12-1992

An Efficient Over-the-Cell Routing Algorithm for High Performance Circuits

Sivakumar Natarajan

AN EFFICIENT OVER-THE-CELL ROUTING ALGORITHM FOR HIGH PERFORMANCE CIRCUITS

by

Sivakumar Natarajan

A Thesis
Submitted to the
Faculty of The Graduate College
in partial fulfillment of the
requirements for the
Degree of Master of Science
Department of Computer Science

Western Michigan University
Kalamazoo, Michigan
December 1992

AN EFFICIENT OVER-THE-CELL ROUTING ALGORITHM FOR HIGH PERFORMANCE CIRCUITS

Sivakumar Natarajan, M.S.

Western Michigan University, 1992

In this thesis, we present a three-layer and a two-layer over-the-cell (OTC) channel routing algorithm (WILMA3 and WILMA2 respectively) for high speed circuits. This router not only minimizes the channel height by using over-the-cell areas but also achieves the net's timing requirements.

We have implemented our routers in C on SUN Sparc 1+ workstation and tested it on MCNC benchmarks Primary I and II. Experimental results show that WILMA3 can achieve results which are 72% better (on the average) than the conventional two layer channel router, 61% better than two layer over-the-cell router and 51% better than three layer greedy channel router (3GCR). Compared with 3-layer OTC routers, WILMA3 improves the delay (i.e., estimated as the length of a longest net) by as much as 8.5%, while increasing the channel height by 17% in the worst case. WILMA2 produces routings which are comparable to existing 2-layer over-the-cell routers.

ACKNOWLEDGMENTS

First, I thank my advisor, Prof. Naveed A. Sherwani. I could not have made so much progress without his guidance and help in my school years. Second, to him and other members of my committee, Prof. Fred Boals and Prof. Ajay Gupta, I extend my sincere appreciation for their supervision and support. I would like to thank all the faculty in the Department of Computer Science and Western Michigan University, Kalamazoo who have ever provided opportunity, encouragement, and support to me.

I extend my special thanks to Nancy Holmes and Prof. Majid Sarrafzadeh for their help and support without which this work would not have come to fruition. I also want to thank my fellow students in the night group for their encouragement and help.

Finally, to my parents, I owe a debt of gratitude. I would not be what I am without their innumerable sacrifices and their unstinted support in everything I undertake. Their love for me and their confidence in me cannot be described in words.

I thank my wife, Shoba, for her patience, love and support throughout this process. I am thankful to her for being very understanding and supportive at a time when I needed it most.

I know it is the happiest thing for her and my parents to see the completion of my thesis and my graduation.

Sivakumar Natarajan

To my parents and my wife, Shoba

Order Number 1351255

An efficient over-the-cell routing algorithm for high performance circuits

Natarajan, Sivakumar, M.S.

Western Michigan University, 1992

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

TABLE OF CONTENTS

Table of Contents – Continued

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Recently, the standard cell design style has increased in popularity as it offers a good compromise between time-to-market and area. In other words, standard cell offers a fast turnaround time without poor real estate utilization as in completely synthesized FPGAs and other gate array technologies. Therefore, standard cell is playing an important role in the market, and this role is likely to expand in the future.

A standard cell design usually takes more area than a full-custom or a hand-crafted design. However, as more and more metal layers become available for routing, the difference in area between the two design styles will gradually reduce. In order to be more competitive with full-custom designs, area-efficient standard cell design technologies must be developed. The total layout area in the standard cell design style is equal to the sum of the total cell area and the total channel area. For a given layout, the total cell area is fixed. Thus, the total area of a layout can only be reduced by decreasing the total channel area. This problem has been extensively studied, and many two layer and three layer channel routers have been developed, several of which can produce "near optimal" routings for most channels [18]. Despite this fact, as much as 15% of the area in most standard cell layouts is still consumed by channel routing.

As several channel routers have been developed that complete channel routing with the number of tracks very close to the channel density, further improvement in the layout area is impossible if routing is done only in channels. As a result, several researchers have investigated the use of area over the cells to obtain further reduction in channel height [5, 6, 11, 14, 16, 19, 24]. This routing technique is referred to as *over-the-cell* routing and is possible due to the fact that most cell libraries do not allow use of the second (M2) and third (M3) metal layers for connections within the cells.

Internal routing of cells is completed using the first metal layer (M1). Therefore, the second metal layer (M2), and if the technology allows the third metal layer (M3), over-the-cell is un-utilized. The area in M2 and M3 can be utilized for routing of nets in order to reduce the channel height. As the number of layers allowed for routing increases, the over-the-cell routing problem becomes important. Since the conventional channel routing problem is known to be NP-hard [26], and as the over-the-cell channel routing problem is a generalization of the conventional channel routing problem, it is easy to see that the over-the-cell channel routing problem is also NP-hard [11].

Several algorithms for over-the-cell routing have been presented, and the technique has proven to be very effective [5, 6, 14, 19]. Several heuristics for over-the-cell channel routing have been presented for the two metal layer process [5, 14, 19]. Two of them consider only nets whose terminals are on a single row, either top or bottom, and produce up to a 20% reduction in channel height for over-the-cell channel routing, as compared to conventional channel routers [5, 19]. The third router allows a much wider variety of nets to be routed over the cell rows and produces up to a 35% reduction in channel height as compared to conventional channel routers [14]. Two heuristics have been presented for over-the-cell routing using the three metal layer process, both of which produce a 59% reduction in channel height as compared to a traditional three layer channel router [14].

Despite the dramatic performance of OTC routers, a major shortcoming of the existing routers is the increase in the total wire length and the length of the longest net. Analysis of existing results shows that the total wire length may be increased by as much as 20% in [6] and 35% in [14]. Although no results on wire length are reported, it is very likely that the net length also increases in case of [19]. However, it is possible that the net length in [19] is less than the corresponding net lengths reported in [6, 14]. This may be due to the fact that the main objective of their router is to minimize the number of routing tracks used in the over-the-cell area, as well as in the channel. However, *no direct attempt at wire length minimization is presented*, and it is possible that critical wires may be rather long in completed over-the-cell routings.

In this thesis, we present a new three layer over-the-cell channel routing algorithm, WILMA3 (WIre Length Minimization Algorithm), for high-speed circuits. This router not only minimizes the channel height by OTC routing but also minimizes the net lengths while satisfying the bounds on the lengths. Our algorithm is novel in two aspects. Firstly, we optimize the track assignment of each net with respect to delay. We compute the track bound for each net which ensures that the net delay is no greater than the given bound (this bound can be obtained in a pre-processing step) – we use the linear RC delay model. Using this track bound, nets are selected for OTC routing. For this purpose, we develop an $O(dn)$ time algorithm for finding an optimal subset of nets, where d is the density and n is the number of nets. Secondly, we use 45^o segments to route the nets over the cells to further reduce the net length.

We have implemented our router in C on SUN Sparc 1+ workstation and tested it on MCNC benchmarks Primary I and II. Experimental results show that our three-layer router, WILMA3, can achieve results which are 72% better (on the average) than the conventional two layer channel router, 61% better than two layer OTC router and 51% better than three layer greedy channel router. In particular, for PRIMARY I, WILMA3 produces a routing with 83 tracks. Compared with 3-layer OTC router [15], WILMA3 improves the delay – estimated as the length of a longest net – by as much as 8.5%, while increasing the channel height by 17% in the worst case. This ensures that all nets achieve their timing requirements which is an important factor in high speed circuits. Experimental results with our two-layer router, WILMA2, indicate that we can produce routings which are comparable to existing 2-layer OTC routers: both WILMA2 and WILMA3 achieve these results, while maintaining the net length bounds.

We first describe the physical constraints for over-the-cell routing.

1.1 Physical Model

In this section, we describe the physical model and their effect on routing techniques.

In the traditional cell layout style, there are two parallel horizontal diffusion rows, one for the P-type transistors and the other for N-type transistors. The first metal layer (M1) is used to complete connections which are internal to the cells. Feedthrough routing is also done using the M1 layer. Power and ground lines are routed in the second metal layer (M2) in the center of the area over-the-cell rows. Terminal rows are available in all layers and are located on the boundaries of the cells [14]. This leaves a rectangular, over-the-cell routing area for each terminal row of the standard cells. The number of tracks available for over-the-cell routing is determined by the height of these rectangular areas and may vary depending on the cell library used. If the cell library has a cell height of 108λ, six routing tracks are allowed each over the cell routing region. Clearly, the net segments routed in the area over the cell rows must be planar since only the M2 layer can be used for over-the-cell routing. (The M1 layer is typically allocated for completing intra-cell connections.) In the channel, the vertical (branch) net segments are routed on the M2 layer, and the horizontal (trunk) net segments are routed on M1. The physical model used for over-the-cell is depicted in Figure 1. If we use the 150λ cell library, thirteen tracks are available for routing in the M3 layer of the over-the-cell regions. This model is used by most existing over-the-cell routers [6, 14, 19].

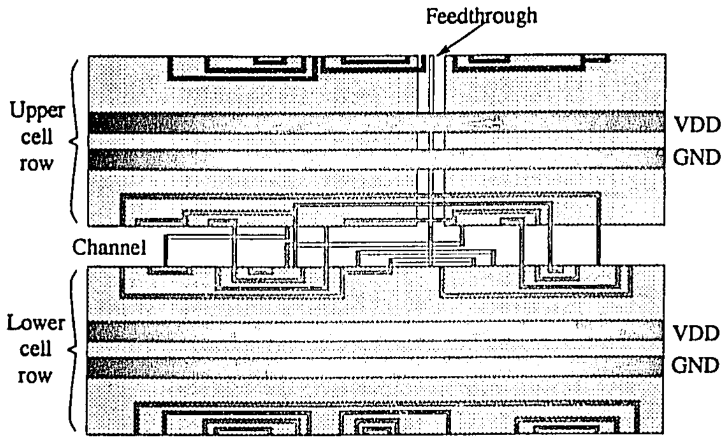


Figure 1. Physical Model.

In three-layer cell model, two metal layers (M2 and M3) are available for routing over each cell row. The assumptions about terminal locations, routing within the cells and and routing of ground and power are the same as two-layer model. It should be noted that the entire over-the-cell region may be used for routing in the third metal (M3) layer, however in order to avoid routing over power and ground lines, we partition the M3 region horizontally into two rectangular regions of equal size. Vias may or may not be allowed in over the cell regions depending on the process used.

1.2 Objectives of Thesis Research

Since the main objective of the over-the-cell routers is to minimize channel height the resulting routing may end up with increasing the length of the longest net. This is unacceptable for high performance circuits in which the longest nets determines the performance of the circuit. What we need is a router which bounds the delay while decreasing the channel height.

The main objective of this thesis research is to develop a performance driven router for high-performance circuits, that not only reduces the channel height but also bounds the length of the nets such that no net exceeds it timing budget.

1.3 Organization of Thesis

The thesis is organized as follows: Chapter II gives a background of the routing problem along with an introduction to the graph terminology and describes two routers in detail. Chapter III gives an overview of WILMA with a description of net classification and net weight assignment. Chapter IV describes the track bound computation for the different net types and gives the algorithm for selecting nets with the track bound constraint. Chapter V gives the detail of the algorithm along with the experimental results and Chapter VI gives the conclusions and future research.

CHAPTER II

BACKGROUND

In this chapter, we discuss the basic definitions in graph theory followed by a definition of the commonly used graphs in routing. Then we describe two two-layer over-the-cell routers followed by a review of few routers.

2.1 Preliminaries

A *graph* is a pair of sets $G = (V, E)$, where V is a set vertices, and E is a set of pairs of distinct vertices called *edges*. If G is a graph, $V(G)$ and $E(G)$ are the vertex and edge sets of G, respectively. A vertex u is adjacent to a vertex v if $\{u, v\}$ is an edge, i.e., $\{u, v\} \in E$. The degree of a vertex u is the number of edges incident with the vertex u. There are several classes of graphs that we will use in the development of properties of routing problems; here we briefly review definitions related to these classes.

A *bipartite graph* is a graph G whose vertex set can be partitioned into two subsets X and Y, so that each edge has one end in X and one end in Y; such a partition (X, Y) is called *bipartition* of the graph. A *complete bipartite graph* is a bipartite graph with bipartition (X, Y) in which each vertex of X is adjacent to each vertex of Y; if $| X |= m$ and $| Y |= n$, such a graph is denoted by $K_{m,n}$. An important characterization of bipartite graphs is in terms of odd cycles. A graph is bipartite if and only if it contains no odd cycle.

2.2 A Graph Model

Let R be a set of evenly spaced terminals. A *net* N is defined to be a subset of nodes in R, i.e., $N \subseteq R, | N |\geq 2$. N is called a *simple* net if $| N |= 2$,

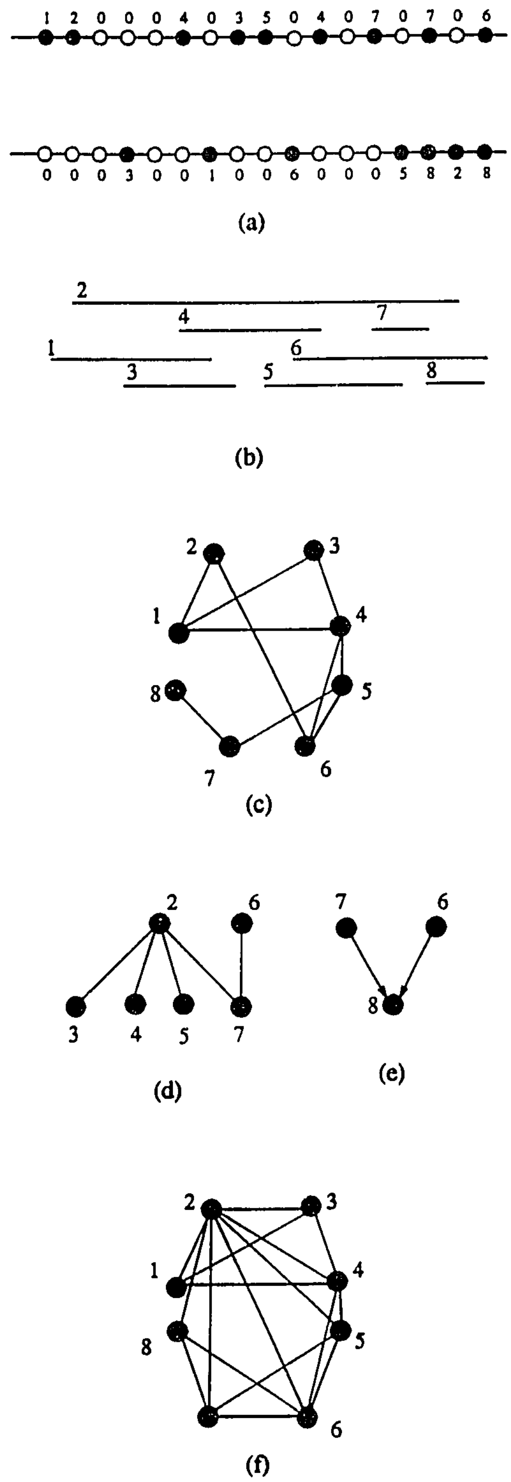


Figure 2. Graph Representation of a Net List.

otherwise it is called a multi-terminal net. Let $L = \{N_1, N_2, \ldots, N_n\}$ be a set of nets defined on R. Each net N_i can be uniquely specified by two distinct terminals l_i and r_i called the left point and the right point, respectively, of N_i. Abstractly, a net can be considered as an interval bounded by left and right points. Thus for a given set of nets, an interval diagram depicting each net as an interval can be easily constructed. Figure 2(b) is the interval diagram for the net list given in Figure 2(a). Given an interval diagram corresponding to a set of nets, two graphs representing the routing problem can be defined as follows.

Define an *overlap graph* $\vec{G}_\mathrm{O} = (V, \vec{E}_\mathrm{O})$,

$$V = \{v_i \mid v_i \text{ represents interval } I_i \text{ corresponding to net } N_i\}$$

$$\vec{E}_\mathrm{O} = \{(v_i, v_j) \mid l_i < l_j < r_i < r_j\}$$

In other words, each vertex in the graph corresponds to an interval representing a net and a directed edge is drawn from v_i to v_j if and only if the interval defined by N_i overlaps with that of N_j but does not completely contains or is completely contained by N_j. Figure 2(c) gives the overlap graph for the net list shown in Figure 2(a).

Similarly, define a *containment graph* $\vec{G}_\mathrm{C} = (V, \vec{E}_\mathrm{C})$, where the vertex set V is the same as defined above and $\vec{E}_\mathrm{C}$ a set of directed edges defined below:

$$\vec{E}_\mathrm{C} = \{(v_i, v_j) \mid l_i < l_j, r_i > r_j\}$$

In other words a directed edge is drawn from v_i to v_j if and only if the interval corresponding to N_i completely contains the interval corresponding to N_j. Figure 2(d) gives the containment graph for the net list shown in Figure 2(a).

We also define an *interval graph* $G_\mathrm{I} = (V, E_\mathrm{I})$ where the vertex set V is the same as above, and two vertices are joined by an edge if and only if their corresponding intervals have a non-empty intersection. It is easy to see that

$E_{\mathrm{I}} = E_{\mathrm{O}} \cup E_{\mathrm{C}}$. Figure 2(f) gives the interval graph for the net list shown in Figure 2(a).

We also define an *vertical constraint graph* (hereafter referred to as VCG) $\vec{G}_{\mathrm{V}} = (V, \vec{E}_{\mathrm{V}})$,

$$V = \{v_i \mid v_i \text{ represents interval } I_i \text{ corresponding to net } N_i\}$$

$$\vec{E}_{\mathrm{V}} = \{(v_i, v_j) \mid l_i = l_j \text{ or } r_i = r_j \text{ or } l_i = r_j \text{ or } r_i = l_j\}$$

where the vertex set V is the same as above, and two vertices are joined by an directed edge only if their corresponding intervals have a vertical constraint(i.e., the nets have at least one end point in the same column). If the nets have both common end points, it will have two directed edges in the VCG, which form a cycle. Figure 2(e) gives the vertical constraint graph for the net list shown in Figure 2(a).

We also define an *horizontal constraint graph* (hereafter referred to as HCG) $G_{\mathrm{H}} = (V, E_{\mathrm{H}})$, where the vertex set V is the same as above, and two vertices are joined by an edge if and only if their corresponding intervals have a non-empty intersection. It is clear that the horizontal constraint graph is the same as interval graph.

It is well known that the class of overlap graphs is equivalent to the class of circle graphs [2]. Similarly, the class of containment graphs is equivalent to the class of permutation graphs [10].

2.3 Over-the-Cell Routers

The two-layer OTC routing problem essentially is a selection of two planar sets of segments. One of them is routed in the upper over-the-cell area and the other is routed in the lower over-the-cell region. For a three-layer OTC routing, four planar sets are selected, two sets for the two layers in the upper over-the-cell area and the other two sets for the two layers in the lower over-the-cell area. The nets that are not selected are routed in the channel area. In the following, we discuss two algorithms for over-the-cell routing [4] and [14].

2.3.1 Basic Over-the-Cell Router

In [4], Cong and Liu presented an algorithm for the over-the-cell channel routing. It divides the problem into three steps, routing over the cells, choosing net segments in the channel, and routing in the channel.

The first step is formulated in a very natural way as the problem of finding a maximum independent set of a circle graph. Since the later problem can be solved in quadratic time optimally, an efficient optimal algorithm is obtained for the first step. Also, the second step is formulated as the problem of finding a minimum density spanning forest of a graph. The minimum density spanning forest problem is shown to be NP-hard, so, an efficient heuristic algorithm is presented which produces very satisfactory results. A greedy channel router [21] is used for the third step.

There are two routing layers in the channel, and there is a single routing layer over-the-cells for intercell connections. Clearly, the over-the-cell routing must be planar.

The first step of the over-the-cell channel routing problem is to connect terminals on each side of the channel using over-the-cell routing area on that side. The same procedure is carried out for each side (upper or lower) of the channel independently. Let t_{ij} denote the terminal of net N_i at column j. In a given planar routing on one side of the channel, a hyperterminal of a net is defined to be a maximal set of terminals which are connected by wires in the over-the-cell routing area on that side. For example, for the terminals in the upper side of the channel in Figure 3, $\{t_{5,4}, t_{5,6}, t_{5,11}\}$ is a hyperterminal of net 5. $\{t_{2,2}\}$ is

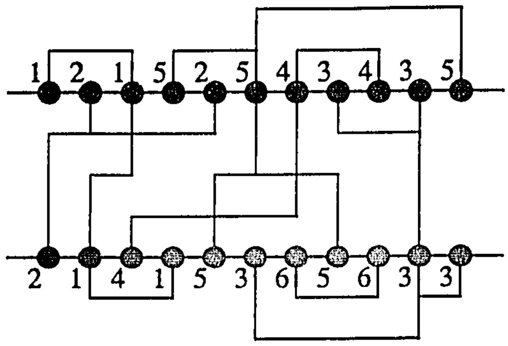


Figure 3. An Instance of Over-the-Cell Routing Solution.

also a hyperterminal. Obviously, when the routing within the channel is to be done, all the hyperterminals of a net need to be connected instead of connecting all the terminals of the net, because the terminals in each hyperterminal have already been connected in the over-the-cell routing area. Intuitively, the fewer hyperterminals are obtained after routing over the cells, the simpler the subsequent channel routing problem. Thus the first step of the problem can be formulated as routing a row of terminals using a single routing layer on one side of the row such that the number of hyperterminals is minimum.

After the completion of the over-the-cell routing step, the second step is to choose net segments to connect the hyperterminals that belong to the same net. A net segment is a set of two terminals of the same net that belong to two different hyperterminals. Thus the second step of the problem is to choose net segments to connect all the hyperterminals of each net such that the resulting channel density is minimum.

After the net segments for all the nets are chosen, the terminals specified by the selected net segments are connected using the routing area in the channel. The problem is now reduced to the conventional two-layer channel routing problem. A greedy channel router [21] is used for this step. Other two-layer channel routers may also be used.

Net Selection for OTC Routing: The first step of the over-the-cell channel routing problem is to route a row of terminals using a single routing layer on one side of the channel such that the resulting number of hyperterminals is minimized. This problem is called the multiterminal single-layer one-sided routing problem (MSOP).

MSOP can be solved by a dynamic programming method in $O(c^3)$ time, where c is the total number of columns in the channel. Given an instance I of MSOP, let $I(i,j)$ denote the instance resulting from restricting I to the interval $[i,j]$. Let $\mathcal{S}(i,j)$ denote the set of all the possible routing solutions for $I(i,j)$. Let:

$$M(i,j) = \max_{S\in\mathcal{S}(i,j)} \{\sum_{k\geq 2}(k-1)d_k(S)\}$$

where $d_k(S)$ is the number of hyperterminals of degree k in S. If there is no terminal at column i, clearly, $M(i,j) = M(i+1,j)$. Otherwise, assume that the terminal at column i belongs to net n. Let $x_{n_1}, x_{n_2}, \ldots, x_{n_s}$ be the column indices of other terminals that belong to net n in interval (i,j). Then, it is easy to verify that

$$M(i,j) = \max(i+1,j), \max_{1\leq l\leq s}\{M(i+1,n_l) + M(n_l,j)\}$$

It is easy to see that this recurrence relation leads to an $O(c^3)$ time dynamic programming solution to MSOP.

Channel Segment Selection: After the over-the-cell routing, a set of hyperterminals is obtained. The terminals in each hyperterminal are connected together by over-the-cell connections. The next problem is to choose a set of net segments to connect all the hyperterminals of each net such that the channel density is minimized. This problem can be transformed to a special spanning forest problem, as discussed below.

For an instance I of the net segment selection problem, the connection graph $G = (V, E)$ is defined to be a weighted multigraph. Each node in V represents a hyperterminal. Let h_1 and h_2 be two hyperterminals that belong to the same net N_i. For every terminal t_{ij} in h_1 and for every terminal t_{ik} in h_2 there is a corresponding edge (h_1, h_2) in E, and the weight of this edge $w((h_1, h_2))$ is the interval $[j, k]$ (assume that $j \leq k$, otherwise, it will be $[k, j]$). Clearly, if h_1 contains p_1 terminals and h_2 contains p_2 terminals, then there are $p_1 \times p_2$ parallel edges connecting h_1 and h_2 in G. Furthermore, corresponding to each net in I there is a connected component in G.

Given an instance I of the net segment selection problem, since all the hyperterminals in the same net are to be connected together for every net in I, it is necessary to find a spanning forest of $CG(I)$. Moreover, since the objective is to minimize the channel density, the density of the set of intervals associated with the edges in the spanning forest must be minimized.

Therefore, the net segment selection problem can be formulated as Minimum Density Spanning Forest Problem (MDSFP). Given a weighted connection graph $G = (V, E)$ and an integer D, determine a subset of edges $E' \subseteq E$ that form a spanning forest of G, and the density of the interval set $\{w(e) | e \in E'\}$ is no more than D. In [4], it was shown that this problem is computationally hard.

Theorem 1 *The minimum density spanning forest problem is NP-complete.*

In view of NP-completeness of the MDSFP, an efficient heuristic algorithm has been developed for solving the net segment selection problem [4]. The heuristic algorithm works as follows. Given an instance I of the net segment selection problem, a connection graph $G = (V, E)$ is constructed. For each edge $e \in E$, the relative density of e, called $RD(e)$, is defined to be $d(e)/d(E)$, where $d(e)$ is the density of the set of intervals which intersect with the interval $w(e)$, and $d(E)$ is the density of the interval set $\{w(e) | e \in E\}$. The relative density of an edge measures the degree of congestion over the interval associated with the edge. The algorithm repeatedly removes edges from E until a spanning forest is obtained.

2.3.2 Over-the-Cell Router Using Vacant Terminals

In [14], Holmes, Sherwani and Sarrafzadeh presented a new algorithm called WISER, for over-the-cell channel routing. There are two key ideas in their approach: use of vacant terminals to increase the number of nets which can be routed over the cells, and near optimal selection of 'most suitable' nets for over the cell routing. Consider the example shown in Figure 4(a). Four tracks are necessary using a conventional channel router or an over-the-cell router. However, using the idea of vacant terminals, a two-track solution can be obtained (see Figure 4(b)). Furthermore, it is clear that the selection of nets which minimize the maximum clique, $h_{\max}$, in horizontal constraint graph is not sufficient to minimize the channel height. For example, channel height for the routing problem shown in Figure 4 is determined strictly by $v_{\max}$, that is, longest path in the VCG (vertical constraint graph). Thus, the nets which cause long paths in VCG should be considered for routing over the cells to obtain a better over the cell routing solution.

The algorithm WISER was developed to take advantage of the physical characteristics indigenous to cell-based designs. One such property is the abundance of vacant terminals. A terminal is said to be *vacant* if it is not required for any net connection. Examination of benchmarks and industrial designs reveals that most standard cell designs have 50% to 80% vacant terminals depending on the given channel. A pair of vacant terminals with the same x-coordinate forms a *vacant abutment* (see Figure 4). In the average case, 30% - 70% of the columns in a given input channel are vacant abutments. The large number of vacant terminals and abutments in standard cell designs is due to the fact that each logical

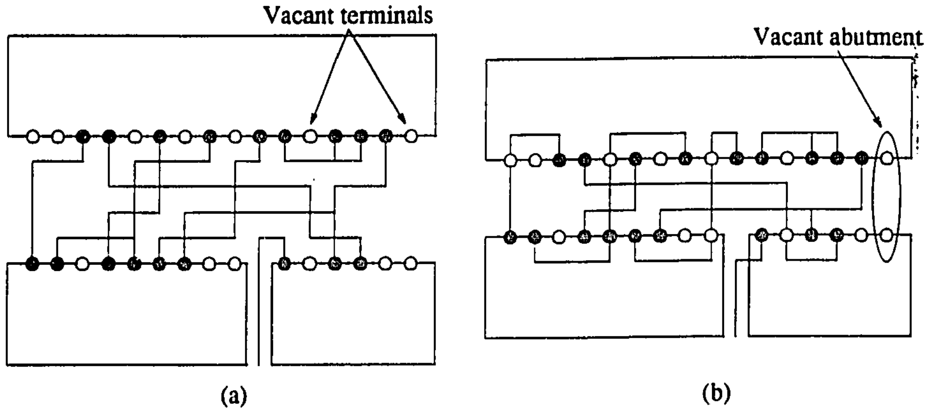


Figure 4. Effect of Using Vacant Terminals in Layout.

terminal (inputs and outputs) is provided on both sides of a standard cell but, in most cases, need only be connected on one side. It should be noted that the actual number of vacant terminals and abutments and their locations cannot be obtained until global routing is completed.

An informal description of each of the six steps of algorithm is given below.

1. **Net Classification:** Each net is classified as one of three types which, intuitively, indicates the difficulty involved in routing this net over the cells.

2. **Vacant Terminal and Abutment Assignment:** Vacant terminals and abutments are assigned to each net depending on its type and weight. The weight of a net intuitively indicates the improvement in channel congestion possible if this net can be routed over the cells.

3. **Net Selection:** Among all the nets which are suitable for routing over the cells, a maximum weighted subset is selected, which can be routed in a single layer.

4. **Over-the-Cell Routing:** The selected nets are assigned exact geometric routes in the area over the cells.

5. **Channel Segment Assignment:** For multi-terminal nets, it is possible that some net segments are not routed over the cells, and therefore, must be

routed in the channel. In this step, 'best' segments are selected for routing in the channel to complete the net connection.

6. **Channel Routing:** The segments selected in the previous step are routed in the channel using a greedy channel router.

The most important steps in algorithm WISER are net classification, vacant terminal and abutment assignment, and net selection. Channel segment assignment is done using an algorithm similar to the one presented in [4]. The channel routing is completed by using a greedy channel router [21].

2.3.3 Review of Other Over-the-Cell Routers

Holmes, Sherwani, and Sarrafzadeh [15] introduced two models for three-layer, over-the-cell channel routing in the standard cell design style. For each model, an effective algorithm is proposed. Both of the algorithms achieve dramatic reduction in channel height. In fact, the remaining channel height is normally negligible. The novelty of this approach lies in use of 'vacant' terminals for over-the-cell routing. For the entire PRIMARY I example, the router reduces the routing height by 76% as compared to a greedy 2-layer channel router. This leads to an overall reduction in chip height of 7%.

Wu, Holmes, Sherwani, and Sarrafzadeh [34] presented a three-layer over-the-cell router for the standard cell design style based on a new cell model (CTM) which assumes that terminals are located in the center of the cells in layer M2. In this approach, nets are first partitioned into two sets. The nets in the first set are called critical nets and are routed in the channel using direct vertical segments on the M2 layer, thereby partitioning the channel into several regions. The remaining nets are assigned terminal positions within their corresponding regions and are routed in a planar fashion on M2. This terminal assignment not only minimizes channel density but also eliminates vertical constraints and completely defines the channel to be routed. In the next step, two planar subsets of nets with maximum total size are found and they are routed on M3 over-the-cell. The rest of the nets are routed in the channel using a HVH router.

Terai, Nakajima, Takahashi and Sato [27] presented a new model for over-the-cell routing with three layers. The model consists of two channels and routing area over a cell row between them. The channel has three layers, whereas the over-the-cell area has two layers available for routing. An over-the-cell routing algorithm has been presented that considers over-the-cell routing problem as a channel routing problem with additional constraints.

Bhingarde, Panyam and Sherwani [1] introduced a new three-layer model for, over-the-cell channel routing in standard cell design style. In this model the terminals are arranged in the middle of the upper and the lower half of the cell row. They develop an over-the-cell router, called MTM router, for this new cell model. This router is very general in nature and it not only works for two and three layer layouts but can also permit/restrict vias over-the-cell.

CHAPTER III

OVERVIEW OF WILMA AND NET CLASSIFICATION

In this chapter, we will give an overview of the WIre Length Minimization Algorithm (WILMA) and describe the classification of nets. Before we give an overview, we will describe performance driven over-the-cell routing.

3.1 Performance Driven Over-the-Cell Routing

Despite the dramatic performance of OTC routers, a major shortcoming of the existing routers is the increase in the total wire length and the length of the longest net. Careful analysis of existing results shows that the total wire length may be increased by as much as 20% in [6] and 35% in [14]. Although no results on wire length are reported, it is very likely that the net length also increases in case of [19]. However, it is possible that the net length in [19] is less than the corresponding net lengths reported in [6, 14]. This may be due to the fact that the main objective of their router is to minimize the number of routing tracks used in the over-the-cell area, as well as in the channel.

In [20] a three-layer and a two-layer over-the-cell channel routing algorithm (WILMA3 and WILMA2 respectively) for high performance circuits is presented. This router not only minimizes the channel height by using over-the-cell areas but also attempts to route all nets within their timing requirements. This algorithm is based on two ideas. First, it optimizes the track assignment of each net with respect to delay. It identifies the track bound for each net which ensures that the wire length is no greater than the length of the net if routed in the channel. Using this track bound, nets are selected for over-the-cell routing. Secondly, 45° segments are used to route the nets over-the-cells to further reduce the net length.

The basic idea of the algorithm is as follows: all the multi-terminal nets are decomposed into two-terminal nets and classified. Then weights are assigned to each net. The weight of a net intuitively indicates the improvement in channel congestion possible if this net can be routed over the cells. A channel router is then used to obtain the channel density (d_c) if routed in the channel. For each net N_i, track in which N_i is routed is recorded. An over-the-cell router is used to obtain the channel density (d_o) for over-the-cell routing. For each net N_i, the track bound k_i is computed, which ensures that if the net is routed over-the-cell at a track less than or equal to k_i, it will have a wire length less or equal to the net length when routed in the channel. This is based on the estimated channel heights d_c and d_o. Among all the nets which are suitable for routing over the cells, four (two) maximum-weighted planar subsets are selected, subject to the track bound constraint for the three-layer (two-layer) model. Once the nets are selected, a set of vacant terminals (vacant abutments) in the case of Type II (Type III) nets are assigned to each net N_i depending on its weight. These vacant terminal/abutment locations will later be used to determine an over-the-cell routing for N_i. Over-the-cell routing is done with 45^o segments and rectilinear segments. In order to avoid design rule violations, any net N_i routed over-the-cell on track t_i must contain a vertical segment of length ρ_i before 45^o segments can be used. The net segments that have not been routed in the area over the cells are routed in the channel. After the channel routing is done the channel density (d_θ) due to over-the-cell routing of nets is obtained. If $d_\theta > d_o$, d_o is set equal to d_θ and the process is repeated. This iteration does not take place more than once or twice in practice. In the worst case, the iteration is bounded by the difference between d_o and d_c.

3.2 Overview of the Algorithm

Algorithm WILMA3 consists of seven basic steps which are described, briefly, as follows. Input to the algorithm is a global routing and a set of bounds on net's length. These bounds either come from timing requirements or may be selected by a designer to reduce long nets in the circuit.

1. **Net Decomposition & Classification:** Multi-terminal nets are decomposed into two-terminal nets and classified.

2. **Net Weight Assignment:** The weight of a net intuitively indicates the improvement in channel congestion possible if this net can be routed over the cells. Selecting the nets with maximum weight is necessary to achieve maximal reduction in channel height.

3. **Estimation:** In this phase of the algorithm, a channel router is used to obtain the channel density (d_c) if routed in the channel. For each net n_i, we record the track in which n_i is routed. An over-the-cell router is used to obtain the channel density (d_o) for over-the-cell routing. This information is used to compute track bounds.

4. **Track Bound Computation:** For each net n_i, we compute the track bound k_i, which ensures that if the net is routed over-the-cell at a track less than or equal to to k_i, it will have a wire length less or equal to a given bound; this bound depends on net's criticality and timing requirements. This is based on the estimated channel heights d_c and d_o.

5. **Net Selection With Track Bound Constraint:** Among all the nets which are suitable for routing over the cells, we select four (two) maximum-weighted, planar subsets, subject to the track bound constraint for the three (two) layer model. Once the nets are selected, a set of vacant terminals (vacant abutments) in the case of Type II (Type III) nets are assigned to each net n_i

depending on its weight. These vacant terminal/abutment locations will later be used to determine an over-the-cell routing for n_i.

6. **Over-The-Cell Routing with 45° Segments:** Perform hybrid over-the-cell routing with 45° segments and rectilinear segments. In order to avoid design rule violations, any net n_i routed over-the-cell on track t_i must contain a vertical segment of length ρ_i before 45° segments can be used (we shall elaborate on ρ_i in subsequent chapters.)

7. **Channel Routing:** The net segments that have not been routed in the area over the cells are routed in the channel. In algorithm WILMA2 and WILMA3, channel routing algorithms from [21] and [7], respectively, are used. After the channel routing is done the channel density (d_θ) due to hybrid over-the-cell routing of nets is obtained. If $d_\theta > d_o$, we set $d_o = d_\theta$ and go to Step 4. We could also use channel routers that employ 45° wires, e.g., the one proposed in [33].

Note that if the routing cannot be finished then we have to increase the channel height by one, recalculate the bounds, and proceed. The iterative process mentioned above takes place very rarely as for most examples our algorithm can complete the routing using no more tracks than d_o. However, in cases when d_θ is indeed greater than d_o we have noticed that the difference is usually one or at most two tracks.

3.3 Net Classification

Given a set of nets $N = \{n_1, n_2, n_3, \ldots, n_k\}$ in a channel C with top row R_t and bottom row R_b, such that $n_i = \{t_{i1}, t_{i2}, \ldots, t_{ij}\}$ where t_{ij} is a terminal belonging to n_i, a terminal t_i is called *vacant* if it not used for interconnection by any net *i.e.*, $t_i \notin \cup_{n_i \in N} n_i$. The notion of vacant terminals was first introduced and exploited in [14]. For any given t_i, the function $OPP(t_i)$ returns the terminal directly across the channel in the same column, the function $ROW(t_i)$ returns the row to which the terminal t_i belongs, (either R_t or R_b), and $COL(t_i)$ returns the column of terminal t_i.

Nets are classified according to the location of net terminals with respect to vacant terminals and vacant abutments in the channel. A *vacant abutment* is a column in which both the terminals are not used for interconnection by any net *i.e.*, t_i and $OPP(t_i) \notin \cup_{n_i \in N} n_i$. In a typical standard cell design, approximately 50% - 80% of the terminals are vacant and 30% - 70% of the columns are vacant abutments.

We decompose each m-terminal net into exactly $m-1$ two-terminal nets at adjacent terminal locations. For example, a six-terminal net $n_i = \{t_1, t_2, t_3, t_4, t_5, t_6\}$, is decomposed into five two-terminal nets: $n_{i1} = \{t_1, t_2\}$, $n_{i2} = \{t_2, t_3\}$, $n_{i3} = \{t_3, t_4\}$, $n_{i4} = \{t_4, t_5\}$, and $n_{i5} = \{t_5, t_6\}$. Clearly, $m-1$ two-terminal nets are sufficient to preserve the connectivity of the original m-terminal net.

After decomposition, we classify all of nets as one of three basic types. Let $n_j = \{t_1, t_2\}$ be a net where t_1 (t_2) is the leftmost (rightmost) terminal of n_j. A net n_j is a Type I(a) net if $ROW(t_1) = ROW(t_2)$. Net n_j is a Type I(b) net if $ROW(t_1) = ROW(t_2)$ and that $OPP(t_1)$ and $OPP(t_2)$ are both vacant. Type I nets may be routed over the cell rows on either the top or the bottom of the channel. Net n_j is a Type II net if $ROW(t_1) \neq ROW(t_2)$ and that either $OPP(t_1)$ or $OPP(t_2)$ are vacant. Type II nets may be routed over the cells on either side (top or bottom) of the channel. Net n_j is a Type III net if $ROW(t_1) \neq ROW(t_2)$ and there exists a terminal t_i such that $COL(t_1) < COL(t_i) < COL(t_2)$ subject to the condition that t_i and $OPP(t_i)$ are vacant and $OPP(t_1)$ and $OPP(t_2)$ are not vacant.

The three basic net types are illustrated in Figure 5. Hereafter, due to the geometrical nature of the discussion, a net n_i will be denoted by its left terminal (l_i) and right terminal (r_i), such that $COL(l_i) \leq COL(r_i)$.

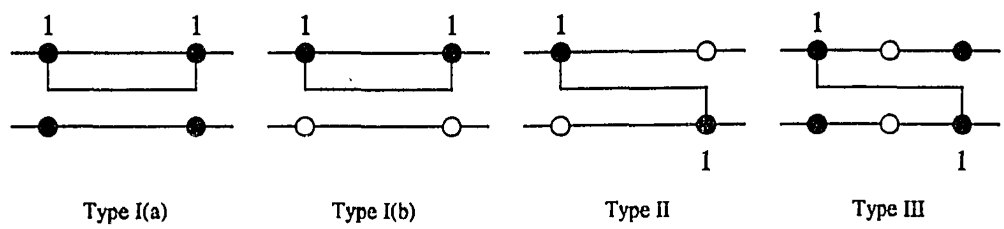


Figure 5. Three Basic Net Types.

This classification operation is rather straight forward and can be done in $O(T+B)$ time, where T is the number of terminals in the top row and B is the number of terminals in the bottom row of the channel.

3.4 Net Weight Assignment

In this section, we discuss the assignment of weights to nets depending on their criticality. This is computed by using the weight of the ancestors and descendents of each net.

Each two-terminal net $n_j = (l_j, r_j)$, where l_j and r_j denote the column locations of the left and right terminals of n_j respectively, is assigned a *Net Weight.* The Weight $W(n_j)$ of net n_j indicates the reduction in channel height possible if n_j is routed over the cell and can be computed based on the relative density of the channel in the interval $[l_j, r_j)$ and the relative path position of n_j in the vertical constraint graph (VCG.) Relative density for net n_j is given by $r_d(n_j) = \frac{l_d(n_j)}{h_{max}}$ where $l_d(n_j)$ is the maximum of the local densities at each column location x such that $l \leq x \leq r$ and h_{max} is density of the CRP (the size of a maximum clique in HCG.) The relative path position $r_p(n_j)$ of net n_j in VCG is given by:

$$r_p(n_j) = \frac{d_{anc}(n_j) + d_{des}(n_j) - abs(d_{anc}(n_j) - d_{des}(n_j))}{v_{max}}$$

where $\quad d_{anc} = \max(\{d(s, n_j) \mid s \text{ is an ancestor of } n_j\})$

$d_{des} = \max(\{d(n_j, s) \mid s \text{ is a descendant of } n_j\})$ and

v_{max} is the length of the longest path in VCG.

In general, the weight of a net n_j is computed as follows:

$$W(n_j) = \frac{h_{max}}{v_{max}} r_d(n_j) + \frac{v_{max}}{h_{max}} r_p(n_j)$$

The net weight for a set N of nets is given by:

$$W(N) = \sum_{n \in N} W(n)$$

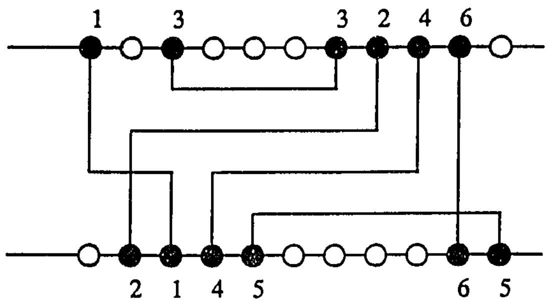
The weight of a net intuitively indicates the improvement in channel congestion possible if this net is routed over the cell. This weight is used to select nets to be routed over the cells to achieve maximum reduction in channel height.

CHAPTER IV

TRACK BOUND COMPUTATION AND NET SELECTION

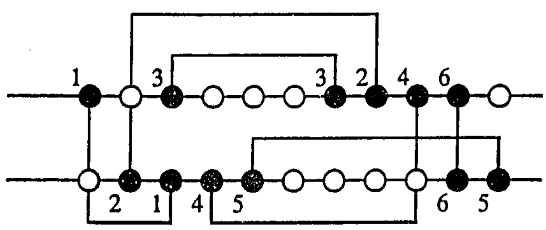
In this chapter, we explain the track bound, and its computation. Track bound specifies the track above which, if a net is routed will have a delay more than what it will when routed in the channel. This bounds the delay of the net thus making the routing applicable to high performance circuits. Figure 6 describes the effect of bounding a net's delay. In Figure 6 (a) we show the instance of a channel routing problem which has a 4 track solution with the length of the longest net being 11. In Figure 6 (b) we give the over-the-cell routing solution to the instance of a channel routing problem in Figure 6 (a). We have a 1-track solution, but with the length of the longest net having been increased to 12. In Figure 6 (c), we have the over-the-cell routing solution with the idea of bounding the delay of the nets. Bounding the delay of each net ensures that no net is routed with a length more than its length in channel routing. This yields a 3 track solution with the length of the longest net no greater than the one in channel routing. Note that although we have a channel density more than that in over-the-cell routing, the delay of net is bounded.

We also use a angular routing to further increase the track bound which gives us more flexibility in routing the net. When we perform angular routing, we need to ensure that design rules are not violated. In order not to violate the design rules, we first route a vertical segment before routing the angular segment, as shown in Figure 7.



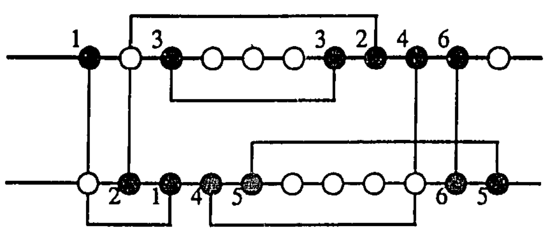
Tracks = 4
Longest Net = 11

(a)



Tracks = 1
Longest Net = 12

(b)



Tracks = 2
Longest Net = 11

(c)

Figure 6. Track Bound.

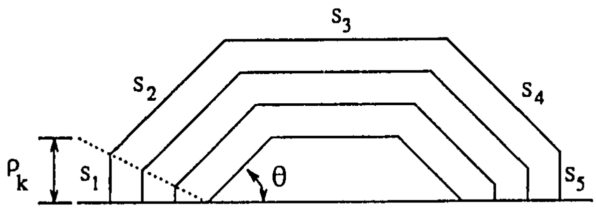


Figure 7. OTC Routing Using 45° Segments.

4.1 Track Bound Computation

Hybrid routing for a net n_i with terminals l_i and r_i at track k_i will have (at most) five segments $s_1, s_2, \ldots, s_5$ as shown in Figure 7. The segments s_1 and s_5 are the *vertical segments*, s_2 and s_4 are the *angular segments*, and s_3 is the horizontal segment routed at track k_i. The vertical segment ρ^i_k for any track k_i is $\delta(k_i - 1)(\frac{1-\sin\theta}{\cos\theta})$, where δ is the minimum feature separation for the technology (which is 3λ for CMOS) and θ is the angle at which the angular segment is routed with respect to the horizontal. If $len(s_j)$ represents the length of a segment s_j, then the length of the net routed over-the-cell using hybrid routing is given by

$$length\ of\ net\ n_i = \sum_{j=1}^{5} len(s^i_j)$$

where the lengths of the individual segments are given by:

$$len(s^i_1) = len(s^i_5) = \rho_{k_i} = \delta(k_i - 1)\frac{(1-\sin\theta)}{\cos\theta}$$

$$len(s^i_2) = len(s^i_4) = \frac{\delta k_i - s^i_1}{\sin\theta} = \frac{\delta(k\cos\theta + k\sin\theta - k + 1 - \sin\theta)}{\sin\theta\cos\theta}$$

$$len(s^i_3) = (r_i - l_i) - \frac{2(\delta k_i - s^i_1)}{\tan\theta} = (r_i - l_i) - \frac{2\delta(k\cos\theta + k\sin\theta - k + 1 - \sin\theta)}{\sin\theta}$$

Let d_o and d_c be the channel densities as estimated in Phase 3. Let θ be the orientation of the angular segments used in routing with respect to the horizontal. Any given net n_i may be routed in a track in the channel (t^c_i) or a track in the upper over-the-cell area (t^u_i) or a track in the lower over-the-cell area (t^l_i). Thus, it may be assigned to a track in one of the three possible ways. It is possible that the horizontal segment h_i of the net is split into two segments, and the two segments are one routed on the upper cell at track k_i^u and the other routed in the lower cell at track k_i^l as in Type III nets. In order to compute the track bound, we need three lengths, the total length of the net when routed in the channel denoted by L_i^c, the total length when routed over-the-cell denoted by L_i^o and the total length when routed over-the-cell using angular segments denoted by L_i^θ.

The length L_i^c of a net n_i is equal to the sum of the length of horizontal segment and length of vertical segment. The length of the horizontal segment is equal to $(r_i - l_i)$. The length of the vertical segment depends on whether both the terminals lie on the top or bottom of the channel, and is given by:

$$\text{Length of the vertical segment of net} = \begin{cases} 2t_i^c & if\ (l_i, r_i)\ on\ R_t \\ 2(d_c - t_i^c) & if\ (l_i, r_i)\ on\ R_b \end{cases}$$

Without loss of generality, we will assume that both terminals of n_i lie on R_t. Hence, the length of a net n_i routed in the channel is given by $L_i^c = (r_i - l_i) + 2t_i^c$. It should be noted that L_i^c is same for all net types, whereas L_i^o and L_i^θ vary depending on the type of the net. Since the length increase due to OTC routing varies with the type of net, we identify for each class of net the track bound in terms of the above mentioned parameters.

4.1.1 Type I Nets

Recall that Type I nets are those nets which, due to the location of vacant terminals with respect to net terminals, may be routed over the cell rows on exactly one side of the channel. Type I(a) nets are those Type I nets with both terminals on a single row (top or bottom) of the channel; whereas, Type I(b) nets have terminals on opposite channel boundaries. We consider, each of these sub-types separately.

4.1.1.1 Type I(a) Nets

A net of Type I(a), with both its terminals on the same side of the cell is routed over the cell on the side in which the terminals originate. The routings in the channel, over-the-cell, using hybrid routing for this net type are illustrated in Figure 8.

Lemma 1 *Given a net n_i of Type I(a), $L_i^\theta \leq L_i^c$ if*

$$k_i \leq \frac{t_i^c \sin\theta\cos\theta - \sin^2\theta + \cos\theta + 2\sin\theta - \sin\theta\cos\theta - 1}{2\sin\theta + 2\cos\theta - \sin\theta\cos\theta - 2}$$

Proof: Let $n_i = (l_i, r_i)$ be a type I(a) net. Let t_i^c be the track on which net n_i is routed in an initial channel routing. The wire length L_i^c due to channel routing when net n_i is routed in track t_i^c in the channel is equal to the sum of the length of its vertical segments and horizontal segments. Hence, the total length of the net when channel routing is employed is given by:

$$L_i^c = r_i - l_i + 2\delta t_i^c$$

The total length of the net if routed over-the-cell with angular segments given as:

$$L_i^\theta = 2 \times len(s_1) + 2 \times len(s_2) + len(s_3)$$

Hence, L_i^θ is given by:

$$L_i^\theta = 2\delta(k_i - 1)\frac{(1-\sin\theta)}{\cos\theta} + \frac{2(k_i\cos\theta + k_i\sin\theta - k_i + 1 - \sin\theta)(1-\cos\theta)}{\sin\theta\cos\theta} + (r_i - l_i)$$

Thus, the difference in routing in the channel and routing over the cell using angular segments is given by:

$$\Delta L = L_i^\theta - L_i^c$$

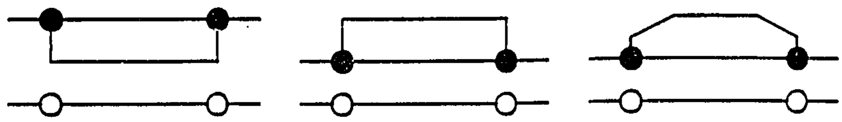


Figure 8. Type I(a) Net Wire Length Difference.

In order to ensure that the net length does not increase if net n_i is routed over the cell using angular segments, $\triangle L$ must be zero. Hence, we equate L_i^θ with L_i^c to obtain the track bound as:

$$k_i \leq \frac{t_i^c \sin\theta\cos\theta - \sin^2\theta + \cos\theta + 2\sin\theta - \sin\theta\cos\theta - 1}{2\sin\theta + 2\cos\theta - \sin\theta\cos\theta - 2} \quad \blacksquare$$

4.1.1.2 Type I(b) Nets

A net of Type I(b), with terminals on the same side of the cell is routed on the other side of the cell. This type of OTC routing certainly uses more wire length than Type I(a) net. However, in the case of availability of space over-the-cell on the opposite cell and no space on the side of the net, a potential reduction of one track in the channel is possible, which in turn may reduce the net length of some of the other nets. The three routing strategies for the net of Type I(b) are illustrated in Figure 9.

Lemma 2 *Given a net n_i of Type I(b), $L_i^\theta \leq L_i^c$ if*

$$k_i \leq \frac{(t_i^c - d_o)\sin\theta\cos\theta - \sin^2\theta + \cos\theta + 2\sin\theta - \sin\theta\cos\theta - 1}{2\sin\theta + 2\cos\theta - \sin\theta\cos\theta - 2}$$

Proof: For a net n_i of type I(b), the routing that is done in addition to a Type I(a) routing is the dual vertical traversal of the channel. The routing for this net type in the channel remains the same as for Type I(a) net. The total length of

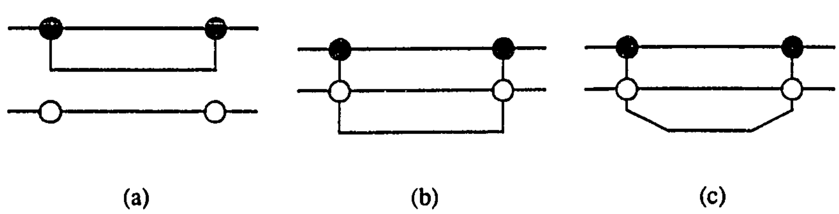


Figure 9. Type I(b) Net Wire Length Difference.

the net employing angular routing L_i^θ is given as:

$$L_i^\theta = 2d_o + 2 \times len(s_1) + 2 \times len(s_2) + len(s_3)$$

Hence, L_i^θ is given by:

$$L_i^\theta = 2d_o + 2\delta(k_i - 1)\frac{(1 - \sin\theta)}{\cos\theta}$$

$$+\frac{2(k_i \cos\theta + k_i \sin\theta - k_i + 1 - \sin\theta)(1 - \cos\theta)}{\sin\theta\cos\theta} + (r_i - l_i)$$

As in Net Type I(a), the difference between routing in the channel and routing over the cell using angular segments is given by:

$$\triangle L = L_i^\theta - L_i^c$$

We ensure that there is no increase in net length if routed over the cell using angular segments by equating L_i^θ with L_i^c giving the track bound as:

$$k_i \leq \frac{(t_i^c - d_o)\sin\theta\cos\theta - \sin^2\theta + \cos\theta + 2\sin\theta - \sin\theta\cos\theta - 1}{2\sin\theta + 2\cos\theta - \sin\theta\cos\theta - 2} \quad \blacksquare$$

4.1.2 Type II Nets

For a net of Type II, with terminals on opposite sides of the channel, the net is routed over one of the cells subject to the availability of vacant terminals. Figure 10 illustrates for this net type, the routing for channel, over-the-cell, and hybrid routing strategies.

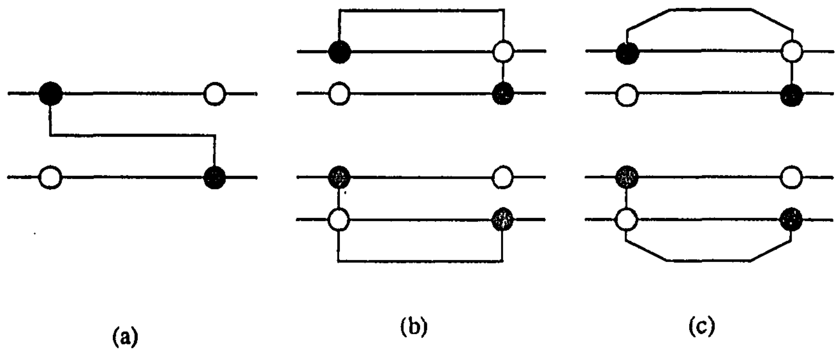


Figure 10. Type II Net Wire Length Difference.

Lemma 3 *Given a net n_i of Type II, $L_i^\theta \leq L_i^c$ if*

$$k_i \leq \frac{\frac{(d_c-d_o)}{2}\sin\theta\cos\theta - \sin^2\theta + \cos\theta + 2\sin\theta - \sin\theta\cos\theta - 1}{2\sin\theta + 2\cos\theta - \sin\theta\cos\theta - 2}$$

Proof: For a net n_i of Type II, the total length of the net in channel routing, L_i^c is given as $(r_i - l_i) + \delta d_c$. If routing is done over the cell, the total length of the net employing angular routing L_i^θ is given as:

$$L_i^\theta = d_o + 2 \times len(s_1) + 2 \times len(s_2) + len(s_3)$$

Hence, L_i^θ is given by:

$$L_i^\theta = d_o + 2\delta(k_i - 1)\frac{(1-\sin\theta)}{\cos\theta}$$

$$+\frac{2(k_i\cos\theta + k_i\sin\theta - k_i + 1 - \sin\theta)(1-\cos\theta)}{\sin\theta\cos\theta} + (r_i - l_i)$$

In order to ensure that net length does not increase if routed over-the-cell, we set the difference between routing in the channel and routing over the cell using angular segments to be zero as in Net Type I and solve the equations giving the track bound as:

$$k_i \leq \frac{\frac{(d_c-d_o)}{2}\sin\theta\cos\theta - \sin^2\theta + \cos\theta + 2\sin\theta - \sin\theta\cos\theta - 1}{2\sin\theta + 2\cos\theta - \sin\theta\cos\theta - 2} \quad \blacksquare$$

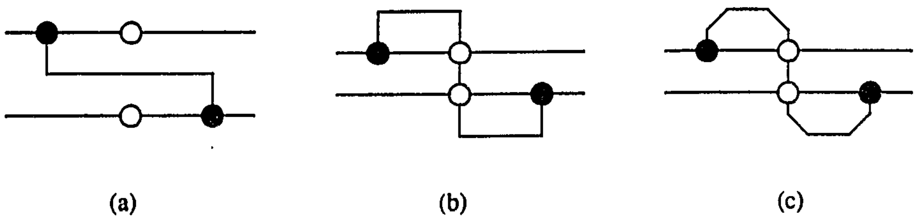


Figure 11. Type III Net Wire Length Difference.

4.1.3 Type III Nets

A Type III net, with terminals on opposite sides of the channel, with occupied opposite terminals can be routed through a vacant abutment, such that routed net spans the over-the-cell areas both on the top and bottom cell. Figure 11 gives the net length difference due to different routing strategies employed, for this net type.

Lemma 4 *Given a net n_i of Type III, $L_i^\theta \leq L_i^c$ if*

$$k_i^u + k_i^l \leq \frac{\frac{(d_c - d_o)}{2}\sin\theta\cos\theta - 2\sin^2\theta + 2\cos\theta + 4\sin\theta - 2\sin\theta\cos\theta - 2}{2\sin\theta + 2\cos\theta - \sin\theta\cos\theta - 2}$$

Proof: A Type III net if routed in the channel would have its length L_i^c as the same as for a Type II net. If routed over-the-cell using angular segments, the length of the net L_i^θ would be the sum of the lengths in top and bottom cell, plus the channel density. Let k_i^u and k_i^l be the track assignments on the top and bottom cell. Then the length of the net for over-the-cell using angular segments, L_i^θ, is given as:

$$L_i^\theta = d_o + 2 \times len(s_1^u) + 2 \times len(s_2^u) + len(s_3^u) + 2 \times len(s_1^l) + 2 \times len(s_2^l) + len(s_3^l)$$

where s_j^u and s_j^l represent the segments in angular routing for upper and lower cells respectively.

Table 1

Comparison of Track Bounds for $\theta = 45^o$ and $\theta = 60^o$

Net Type	$\theta = 45^o$	$\theta = 60^o$
I(a)	$k_i = \frac{0.5t_i^c+0.121}{0.328}$	$k_i = \frac{0.433t_i^c+0.049}{0.299}$
I(b)	$k_i = \frac{0.5(t_i^c-d_o)+0.121}{0.328}$	$k_i = \frac{0.433(t_i^c-d_o)+0.049}{0.299}$
II	$k_i = \frac{0.5(\frac{d_c-d_o}{2})+0.121}{0.328}$	$k_i = \frac{0.433(\frac{d_c-d_o}{2})+0.049}{0.299}$
III	$k_i^l + k_i^u = \frac{0.5(\frac{d_c-d_o}{2})+0.242}{0.328}$	$k_i^l + k_i^u = \frac{0.433(\frac{d_c-d_o}{2})+0.098}{0.299}$

Hence, L_i^θ is given by:

$$L_i^\theta = d_o + 2\delta(k_i^u - 1)\frac{(1-\sin\theta)}{\cos\theta}$$

$$+\frac{2(k_i^u\cos\theta + k_i^u\sin\theta - k_i^u + 1 - \sin\theta)(1-\cos\theta)}{\sin\theta\cos\theta} + (r_i - l_i)$$

$$+2\delta(k_i^l - 1)\frac{(1-\sin\theta)}{\cos\theta} + \frac{2(k_i^l\cos\theta + k_i^l\sin\theta - k_i^l + 1 - \sin\theta)(1-\cos\theta)}{\sin\theta\cos\theta} + (r_i - l_i)$$

The difference in length due to routing in the channel and routing over the cell using angular segments is given by:

$$\Delta L = L_i^\theta - L_i^c$$

This difference must be zero, in order to ensure that the net length does not increase if routed over the cell using angular segments. Hence, we equate L_i^θ with L_i^c giving the track bound as:

$$k_i^u + k_i^l \leq \frac{\frac{(d_c-d_o)}{2}\sin\theta\cos\theta - 2\sin^2\theta + 2\cos\theta + 4\sin\theta - 2\sin\theta\cos\theta - 2}{2\sin\theta + 2\cos\theta - \sin\theta\cos\theta - 2} \quad \blacksquare$$

In Table 1, we compute the track bounds for all the above mentioned net types for $\theta = 45^o$ and $\theta = 60^o$. We have seen that the nets lengths are a minimum for angular routing for $\theta = 45^o$. Hence, we use 45^o segments for routing over-the-cell. However, in asymmetric and low density OTC routings, we use other orientations for segments.

4.2 Net Selection With Track Bound Constraint

In this section, we present a new algorithm for finding an independent set in an unweighted graph with track bound constraint. This algorithm runs in $O(dn)$ time, where d is the density of nets and n is the number of nets. For finding the independent set in a weighted graph we can use the $O(kn^2)$ time algorithm presented in [6].

Consider a set $\mathcal{I} = \{I_1, \ldots, I_n\}$ of intervals, where $I_i = (l_i, r_i)$ is specified by its two end-points. The *left-point* l_i and the *right-point* $r_i, l_i < r_i$. We assume all end-points are distinct real numbers (the assumption simplifies our analysis and can be trivially removed). Two intervals $I_i = (l_i, r_i)$ and $I_j = (l_j, r_j)$ are *independent* if $r_i < l_j$ or $r_j < l_i$; otherwise, they are *dependent.* Two dependent intervals are *crossing* if $l_i < l_j < r_i < r_j$ or $l_j < l_i < r_j < r_i$. If $l_i < l_j < r_j < r_i$ then we say I_i *contains* I_j.

Assume track $i+1$ is above track i, $1 \leq i \leq t-1$. Given $\mathcal{I}$, the restricted track assignment problem (RTAP) is the problem of assigning intervals into tracks 1 to t such that:

p1) In each track, intervals are pairwise independent.

p2) If an interval I_i contains another interval I_j then $\tau_i > \tau_j$, where τ_a is the track to which I_a is assigned.

RTAP has been studied by Sarrafzadeh-Lee [22]. They showed it is NP-hard to minimize the number of tracks and proposed a 2-approximation algorithm running in $O(n \log n)$ time, where n is the number of intervals. That is a restricted tracks assignment in less than $2d$ tracks, where d is the problem density. We

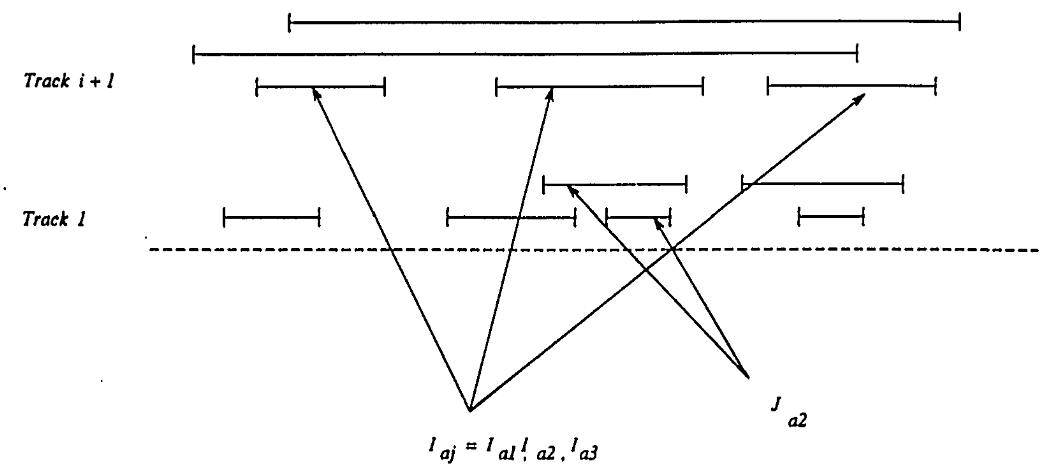


Figure 12. A Restricted Track Assignment of a Set of Intervals.

will see that for our purpose an approximation algorithm is "as good as" an optimal algorithm! (since the approximate bound will be a cost factor in the time complexity)

Consider an overlap representation $\mathcal{I} = \{I_1, \ldots, I_n\}$ of an unweighted overlap graph, where each interval I_i is assigned a bound b_i being the maximum track to which it can be assigned. A track assignment satisfying the bounds is called a bounded tracks assignment. We want to find, among all bounded independent sets with density K, one with maximum size. We will refer to this problem as the BMIS-K problem.

For each interval $I_i = (l_j, r_j)$ in $\mathcal{I}$ we want to calculate a size s_i being an MIS (maximum independent set) of the intervals (in the overlap model) with endpoints in the closed interval $[l_j, r_j]$ (i.e., it includes I_i) among all independent sets with density less than or equal to K. Also, for each maximum independent set obtained, we want to calculate its density k_i, $1 \leq k_i \leq K$. (Note that, $w_i = 1$ for all i). First, $\mathcal{I}$ is assigned into tracks 1 to t in a restricted manner, $d \leq t \leq 2d - 1$

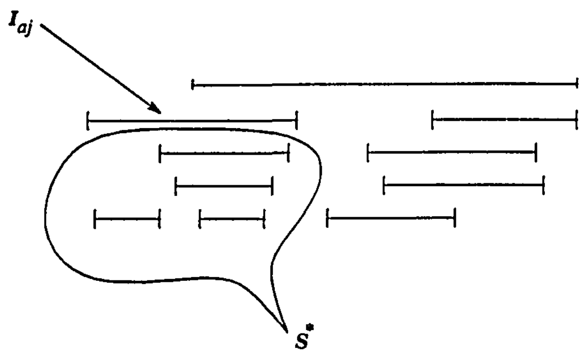


Figure 13. Proof of Case 1.

(that is, a restricted track assignment is obtained). $\mathcal{I}_i$ denotes the set of intervals placed in track i, $1 \leq i \leq t$. $\mathcal{I}_{i+1} = \{I_{a_1,\ldots,}I_{a_m}\}$ denotes the set of intervals assigned to track $i+1$. Certainly, intervals in $\mathcal{I}_{i+1}$ are pairwise independent. Let $\mathcal{J}_{a_j}$ be the set of intervals contained in I_{a_j}, $1 \leq j \leq m$. Clearly, $\mathcal{J}_{a_j} \cap \mathcal{J}_{a_k} = \phi$, for all j and k $(j \neq k)$.

Inductively, assume tracks 1 to i have been processed and for each interval I_j in the set of processed tracks the size s_j and its density k_j is known. We proceed as follows. The basis is trivial – for each interval I_j in track 1, $s_j = 1$ and $k_j = 1$. Track $i+1$ is processed as follows. For each $\mathcal{J}_{a_j}$ (as before, $\mathcal{J}_{a_j}$ is the set of intervals contained in I_{a_j}, $1 \leq j \leq m$ (see Figure 12), a maximum weighted independent set – denoted by $\mathcal{S}_{a_j}$ – in the interval graph representation (not the overlap representation) is found, where the weight of an interval is its size (and is known, by the induction hypothesis). Let s'_{a_j} denote the weight of $\mathcal{S}_{a_j}$ and k'_{a_j} be the maximum of k_f, for all I_f in $\mathcal{S}_{a_j}$. There are two cases to be considered.

Case 1: If $k'_{a_j} = K$ or if $k'_{a_j} > b_{a_j}$ then I_{a_j} will be deleted and will not be considered in subsequent steps. The reason is as follows. Consider an optimal solution that contains I_{a_j} (see Figure13). Let $\mathcal{S}^*$ denote the set of intervals in this optimal solution that is contained in I_{a_j}. The density of $\mathcal{S}^*$ is less than K. Otherwise, the density of $\mathcal{S}^* \cup \{I_{a_j}\}$ would be more than K. Certainly,

$|\mathcal{S}^*| \leq |\mathcal{S}_{a_j}| - 1$; otherwise, $\mathcal{S}^*$ should have been selected in the inductive step, for its density is less than k'_{a_j}. Now $\mathcal{S}^* \cup \{I_{a_j}\}$ in the optimal solution can be replaced with $\mathcal{S}_{a_j}$ to obtain a solution of the same (or more) size.

Case 2: If $k'_{a_j} < K$ and $k'_{a_j} \leq b_{a_j}$, then we set the following two parameters: $k_{a_j} = k'_{a_j} + 1$ and $s_{a_j} = s'_{a_j} + 1$, and continue.

An interval $I_{n+1} = (l_{n+1}, r_{n+1})$ is introduced, where l_{n+1} is the minimum of l_i's minus 1 and r_{n+1} is the maximum of r_i's plus 1 (for $1 \leq i \leq n$). Repeating the just described process one more time results in an optimal solution. As before it takes $O(nt) = O(nd^*)$ time to obtain the size s_i of all intervals.

Lemma 5 *An unweighted instance $\mathcal{I}$ of BMIS-K for overlap graphs can be solved in $O(dn + n\log n)$ time, where d is the density of $\mathcal{I}$ and $n = |\mathcal{I}|$ (note that $d \geq k$).*

Note that the proposed technique cannot be used for the weighted version of the problem, for even one interval in higher tracks may have a large weight and thus replaces (some or all) the previous intervals (the argument in Case 1 does not hold any more). The weighted version can be readily solved employing a straight-forward modification of the algorithm proposed in [6]. We conclude:

Lemma 6 *An arbitrary instance $\mathcal{I}$ of BMIS-K for overlap graphs can be solved in $O(Kn^2)$ time, where $n = |\mathcal{I}|$.*

Using this algorithm, we compute four independent sets for the three layer model to be routed on the top and bottom cell for the second metal layer(M2) and third metal layer(M3). In the two layer model, we compute only two sets as we have only one metal layer (M2) in which to do over-the-cell routing.

CHAPTER V

ALGORITHM WILMA AND EXPERIMENTAL RESULTS

In this chapter, we will describe the tradeoff between channel height and wire length that we are dealing with, followed by an discussion on the iterative phase of WILMA, and finally give the formal algorithm.

As shown in Figure 14 , on one side we have shorter channel heights and longer net lengths, which is a outcome of over-the-cell routing, and on the other side we have bigger channel height with shorter net lengths. What we are looking for is a tradeoff between channel height and length of the net, the only condition being that no net exceed its bound.

We start with d_o, the channel density for over-the-cell routing and try to route the nets considering the bounds on the net. If it is not possible to route, we increase the channel height by one and then route it again. The worst case would be when the routing is the same as that in channel routing.

Figure 15. illustrates the flow of the algorithm WILMA. Note that if the routing cannot be finished then we have to increase the channel height by one, recalculate the bounds, and proceed. The iterative process mentioned above takes place very rarely as for most examples our algorithm can complete the routing using no more tracks than d_o. However, in cases when d_θ is indeed greater than d_o we have noticed that the difference is usually one or at most two tracks. The iterative process is a controlled one and is bounded by $d_c - d_o$.

The formal algorithm is given below:

Algorithm WILMA3()

Input : $N = \{n_1, n_2, \ldots, n_m\}$ **is a set of nets where each net**

$n_j = \{t_i \mid ROW(t_i) \in \{R_t, R_b\}, 1 \leq i \leq \text{num\_of\_columns}\}$

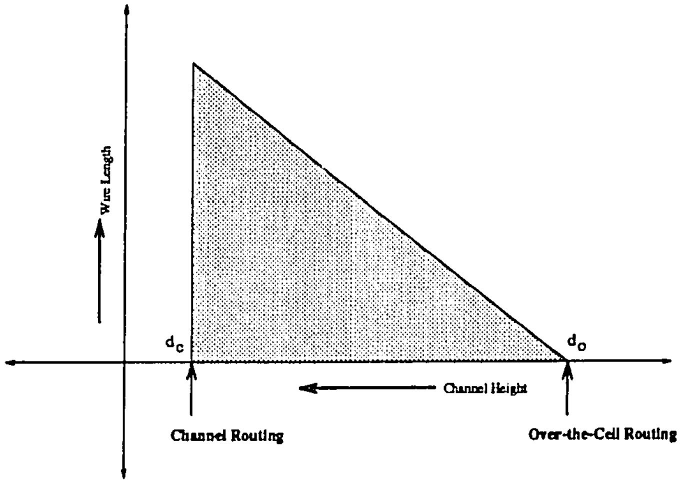


Figure 14. Channel Height Vs Wire Length Tradeoff.

Output: Over-the-cell channel routing.

begin Algorithm

PHASE 1: Net Decomposition and Classification

For each $n_j \in N$ do

/* $n_j = \{t_1, t_2, \ldots, t_m\}$ is a m-terminal net */

$N' = N' \cup \{n_x = (t_i, t_{i+1}) \mid 1 \leq i < m\}$

where i refers to the i^{th} terminal of m-terminal net n_j

$N = N'$ /* Replace the original set N with the set N' */

For each $n_i = (l_i, r_i) \in N$ do {

/* Type I nets */

If $(ROW(l_i) = ROW(r_i))$ {

If ($OPP(l_i)$ and $OPP(r_i)$ are both not vacant))

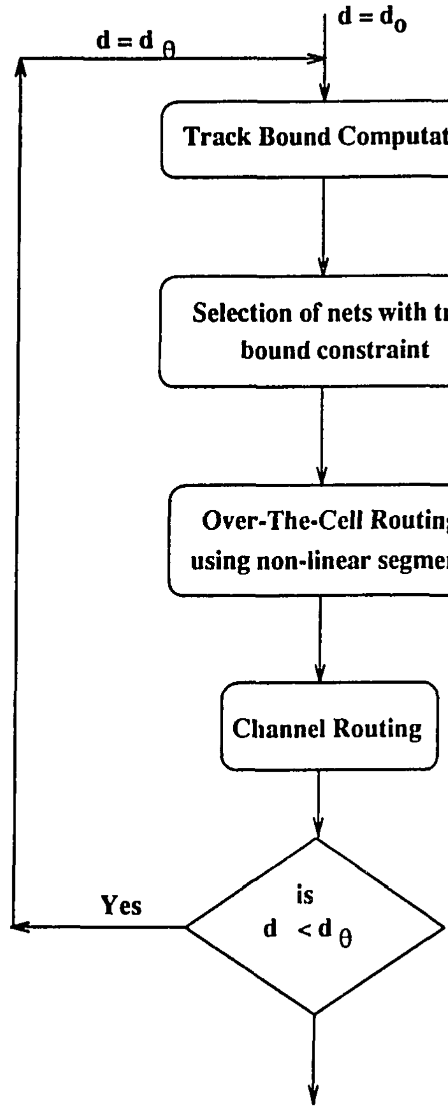


Figure 15. Iterative Process of Algorithm WILMA.

Then $N_1 = N_1 \cup \{n_i\}$

Else If ($OPP(l_i)$ and $OPP(r_i)$ are both vacant)

Then $N_1 = N_1 \cup \{n_i\}$

}

Else /* $ROW(l_i) \neq ROW(r_i)$ */ {

/* Type II nets */

If ($OPP(l_i)$ and $OPP(r_i)$ are both vacant)

Then $N_2 = N_2 \cup \{n_i\}$

/* Type III nets */

If ($OPP(l_i)$ and $OPP(r_i)$ are both not vacant))

Then $N_3 = N_3 \cup \{n_j\}$

}

} /* For all nets */

PHASE 2: Channel Density Estimation for Channel Routing

/* Use a Greedy Channel Router to give estimates */

Input : N

Output: Length L_i^o of each net n_i and the channel density d_c

GREEDY_CHANNEL_ROUTER(N)

PHASE 3: Channel Density Estimation for Over-the-cell Channel Routing

/* Estimate Channel density for OTC routing */

Input : N

Output: Channel density d_o

OTC_CHANNEL_ROUTER(N)

PHASE 4: Track Bound Computation

```
/* Compute the track bound for each net */
Input : N, d_o, d_c
Output: Track bound k_i for each net n_i ∈ N
For each n_i = (l_i, r_i) ∈ N do {
    /* Type I nets */
    If ((ROW(l_i) = ROW(r_i))
        If (OPP(l_i) and OPP(r_i) are both not vacant)
            Then k_i = (1.26 + t_c)/2.49
        Else If (OPP(l_i) or OPP(r_i) are vacant)
            Then k_i = (1.26 + t_c - 2 * d_o)/2.49

    /* Type II nets */
        If (OPP(l_i) is vacant and OPP(r_i) is vacant)
            Then k_i = (1.26 + d_c/2 - d_o/2)/2.49

    /* Type III nets */
        If (OPP(l_i) and OPP(r_i) are both not vacant)
            Then k_i = (2.52 + d_c/2 - d_o/2)/4.98
}
```

PHASE 5: Net Weight Assignment

/* Assign weights to the nets to give estimates */

ASSIGN_WEIGHTS(N)

PHASE 6: Selection of Nets With Track Bound Constraint

/* Select independent sets $I_{t2}, I_{b2}, I_{t3}, I_{b3}$, for the top and bottom of second layer and top and bottom of third layer respectively */

For each ordering $O \in \{\beta_x \mid \beta_x$ an orderings of set {t,t,b,b } } do {

$S_1 = \text{MIS}(N)$

$S_2 = \text{MIS}(N - S_1)$

$S_3 = \text{MIS}(N - S_1 - S_2)$

$S_4 = \text{MIS}(N - S_1 - S_2 - S_3)$

$\text{WEIGHT}[O] = \sum_{i=1}^{4} Wt(S_i)$

}

Select ordering O such that WEIGHT[O] is maximum, and assign $I_{t2}, I_{t3}, I_{b2}, I_{b3}$ according to ordering O.

PHASE 7: Over-The-Cell Routing Using 45° Segments

Input : Four independent sets $I_{t2}, I_{t3}, I_{b2}, I_{b3}$

Output: Over-the-cell routing

OVER_THE_CELL_LAYOUT($I_{t2}, I_{t3}, I_{b2}, I_{b3}$)

PHASE 8: Channel Segment Assignment and Channel Routing

Input : Nets remaining after over-the-cell routing

Output: Channel density d_θ and routing

ASSIGN_CHANNEL_SEGMENTS($I_{t2}, I_{t3}, I_{b2}, I_{b3}, NL$)

CHANNEL_ROUTE(N)

If $d_\theta > d_o$ *then* {

$d_o = d_\theta$

Goto PHASE 4 /* Iterative step */

}

end Algorithm

5.1 Complexity of the Algorithm

Phases 1,4, and 5 take linear time in the number of nets, as they are performed once for every net. Phase 6 is of time complexity $O(dn)$. Phases 2 and 3 have a time complexity of $O(n^2)$, as they involve finding the independent sets. Since the iteration takes place only once or twice, the complexity of WILMA3 is $O(n^2)$.

5.2 Experimental Results

The algorithm WILMA2 differs from WILMA3 in two aspects: net selection and over the cell routing. In net selection, instead of finding four independent sets, we find only two independent sets, one for the top and other for the bottom over-the-cell area of the channel. In over-the-cell the routing is done for just one layer, that is, M2. Because of the reduction in the complexity of algorithm, WILMA2, has better running times than WILMA3.

Both algorithms WILMA3 and WILMA2 have been implemented on a SPARC 1+ workstation in C, and has been tested on several benchmarks including Primary I and Primary II. For an example, we show the performance of WILMA2 and WILMA3 on the Primary I benchmark. The placement of Primary I was obtained from TimberWolfSC Version 5.1 [17], and the global routing is from [3].

We use a RC model for computing the delays in the net. The per unit resistance and capacitance of a net is 30 $m\Omega$ and .20 fF respectively [30]. This yields an effective delay of 6 $\times 10^{-9} ps$. We ignore the effect of vias in the channel while computing the net delay. This is because of the fact that via's are process dependent and the delay varies with the process used. Ignoring the delay in via's gives the advantage to the channel routing - as it gives a conservative delay bound for over-the-cell routing.

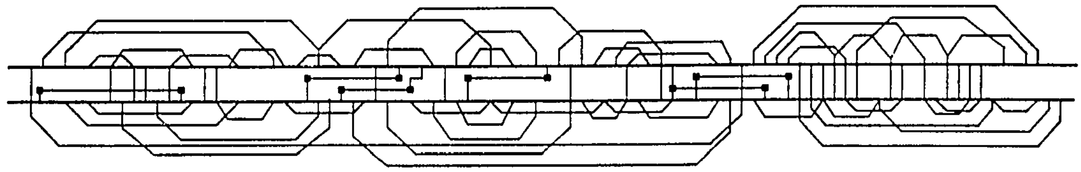


Figure 16. Portion of Channel 14 of PRIMARY I Routed by WILMA3.

Results for the Primary I, indicating the number of horizontal tracks used in the channel are given in Table 2 in comparison with results obtained by a greedy channel router and a conventional channel router. As can be seen in Table 2, WILMA3 reduces the channel height by an average of 52% as compared to best available three-layer greedy channel router. On the average, WILMA3 performs 73% better than a conventional two-layer channel router, 62% better than two-layer OTC routers. WILMA2 performs 35% better than a conventional two-layer channel router. The delay in the longest net in Channel 14, for example, is 229 using conventional OTC router and is 194 using WILMA.

Experimental results indicate that both routers significantly reduce the total wire length and the length of the longest net. Table 3 gives the total wire length and the length of the longest net for the channels of PRIMARY I benchmark for the conventional over-the-cell router and WILMA2. WILMA2 reduces the total wire length by 7.5% and achieves a 8.5% reduction in the length of the longest net on an average. The reduction in total wire length and length of the longest net obtained by WILMA3 are similar to that of WILMA2. A portion of Channel 14 of PRIMARY I routed by WILMA3 is shown in Figure 16.

Table 2

Channel Heights for PRIMARY I by WILMA3

Chan.	No. of Tracks Produced					% Impro. by WILMA3		
	2CRP	2OTC	3CRP	3OTC	WILMA3	2CRP	2OTC	3CRP
1	11	5	6	1	2	82	60	66
2	16	12	9	5	5	69	58	44
3	21	16	11	6	6	71	62	45
4	24	21	15	6	9	62	57	40
5	21	17	11	6	7	67	58	36
6	23	18	12	7	7	69	61	41
7	22	14	14	4	5	77	64	64
8	24	18	16	7	7	71	61	56
9	21	16	13	6	6	71	62	53
10	15	11	8	4	5	67	54	37
11	17	12	12	3	5	70	58	58
12	15	11	8	3	4	73	64	50
13	13	9	8	3	3	77	67	62
14	13	9	7	3	4	69	55	42
15	11	7	6	2	2	81	71	66
16	11	7	6	2	2	81	71	66
17	14	8	7	3	3	78	62	57
18	6	3	3	0	1	83	66	66
Total	298	214	172	71	83	72	61	52

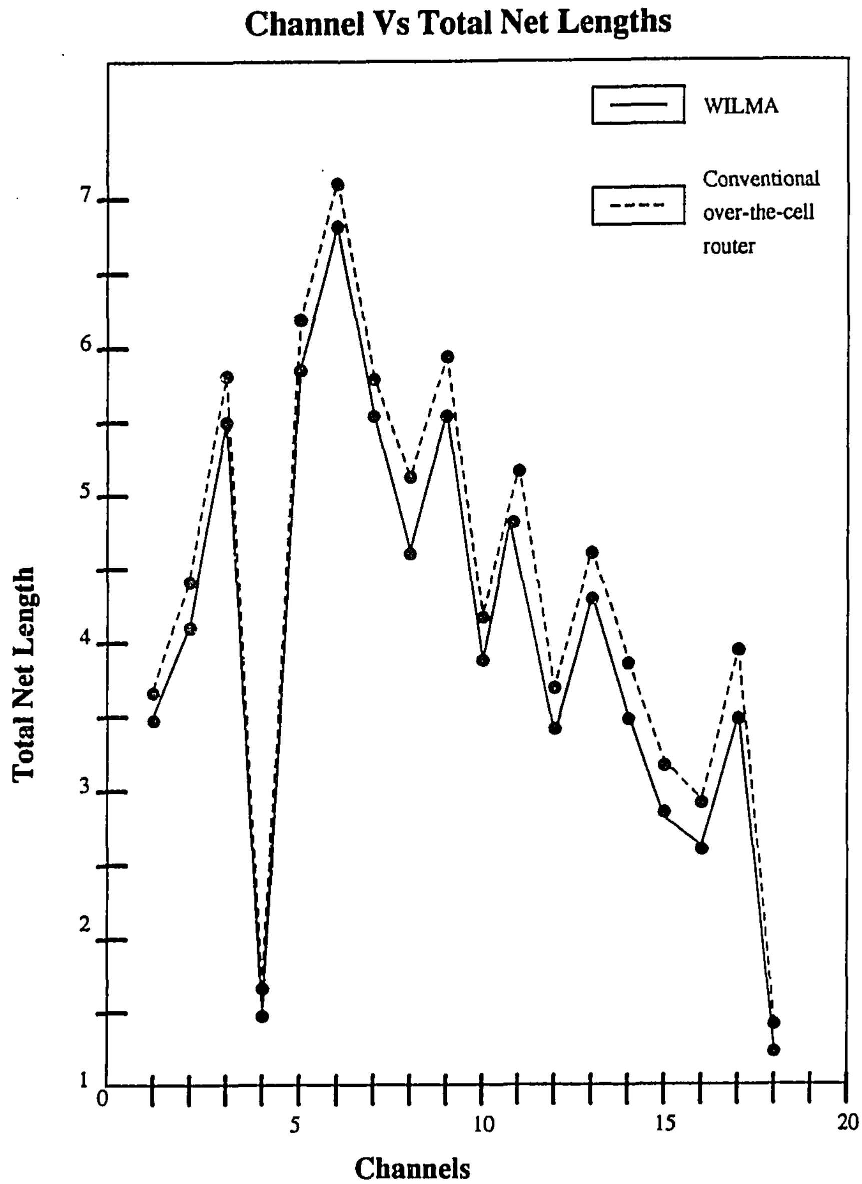


Figure 17. Total Length for OTC Routing and WILMA.

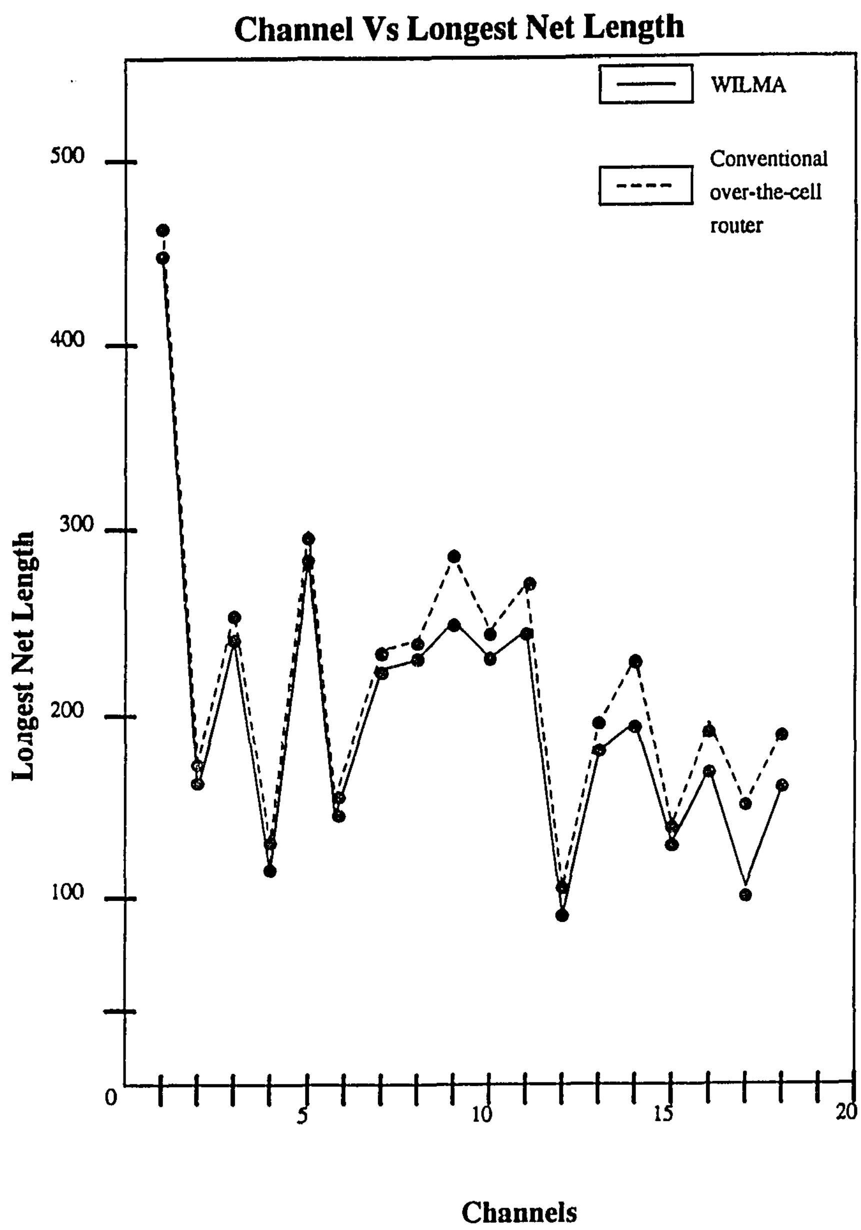


Figure 18. Length of the Longest Net for Channel OTC Routing and WILMA.

Table 3

Wire Lengths for Channels of PRIMARY I for WILMA2

	Total length			Longest net		
Ch	WILMA	2OTC	% imp	WILMA	2OTC	% imp
1	3524	3702	4.8	450	463	2.8
2	4159	4437	6.2	160	178	10.0
3	5523	5861	5.8	220	221	0.5
4	1502	1583	5.1	118	129	8.5
5	5854	6240	6.2	291	295	1.4
6	6805	7165	5.0	143	151	5.3
7	5508	5873	6.2	226	232	2.6
8	4622	5107	9.5	231	238	2.9
9	5563	5959	6.7	251	284	12.0
10	3927	4264	7.9	238	242	1.7
11	4870	5201	6.4	241	268	10.0
12	3398	3747	9.3	91	105	13.0
13	4298	4683	8.2	181	198	8.6
14	3463	3844	9.9	194	229	15.0
15	2838	3246	13.0	125	135	7.4
16	2614	2935	11.0	166	195	15.0
17	3569	3949	9.6	100	145	31.0
18	1236	1344	8.0	160	169	5.3

CHAPTER VI

CONCLUSIONS AND FUTURE RESEARCH

6.1 Conclusions

In this thesis, we studied an important VLSI layout problem, over-the-cell routing, catering to a specific class of circuits, high performance circuits.

We have presented a three layer over-the-cell (OTC) channel routing algorithm (WILMA3) for high speed circuits. This router not only minimizes the channel height by using over-the-cell areas but also achieves the net's timing requirements. In this algorithm, the track assignment of each net is optimized with respect to delay. Each net is given a bound on wire length; based on that, we calculate a track bound for each net. Using this track bound, nets are selected for over-the-cell routing. For this purpose, we presented an $O(dn)$ time algorithm for finding an optimal subset of nets, where d is the density and n is the number of nets. Also, we used 45^o segments to route the nets over-the-cells to further reduce the net length.

We have implemented our router in C on SUN Sparc 1+ workstation and tested it on MCNC benchmarks Primary I and II. Experimental results show that WILMA3 can achieve results which are 72% better (on the average) than the conventional two layer channel router, 61% better than two layer over-the-cell router and 51% better than three layer greedy channel router (3GCR). Compared with 3-layer OTC routers, WILMA3 improves the delay (i.e., estimated as the length of a longest net) by as much as 8.5%, while increasing the channel height by 17% in the worst case.

We have also developed a two-layer version of our routing algorithm called WILMA2. Experimental results with WILMA2 indicate that it can produce routings which are comparable to existing 2-layer over-the-cell routers - both WILMA2 and WILMA3 achieve these results, while maintaining the net length bounds.

6.2 Future Research

A number of problems related to performance driven over-the-cell routing remain open. The parameters that have to be considered to account for delay in high-performance circuits are both electrical and geometrical in nature. The electrical parameters are that of resistance capacitance, and inductance. The geometrical parameters are that of the width and thickness of the wire, the thickness of the oxide layer, and routing geometry. These parameters are responsible for the parasitic effects in high-performance circuits such as crosstalk, parasitic capacitance, inductance, and corner reflections.

In [36], an analysis of the above mentioned parameters with respect to the delay has been presented. The inductance effect on delay is an important factor in the sub-nanoseconds range. The wire capacitance also plays an important role in the delay of circuits. One way of overcoming this effect is to reduce the width of the wire subject to certain limiting conditions. The delay of a wire has a linear relationship to both the resistance of the wire and the length of the wire. The width of the wire also determines the delay. As the width of the wire is reduced, the delay increases drastically after a critical width.

Hence, a router catering to high-performance circuits has to consider all the above mentioned effects in reducing the delay in addition to reducing the length of the longest net. Also, as other cell models are gaining popularity, developing routers for high performance circuits using other cell models is a necessity.

REFERENCES

[1] S. Bhingarde, A. Panyam, and N. Sherwani, "On Optimal Cell Models for Over-the-Cell Routing ," to appear in *Proceedings of 6th International Conference on VLSI Design,* January 1993, IEEE Computer Society Press, Los Alamitos, California.

[2] M. Buckingham, "Circle Graphs," Ph.D. Thesis, Courail Institute of Mathematical Science, Computer Science Department, New York University, October 1980.

[3] C. Chiang, M. Sarrafzadeh, and C. K. Wong, "A Global Router Based on Steiner Min-Max Trees," *IEEE Transactions on Computer-Aided Design,* Vol. 9, No. 12, December 1990, pp. 1318-1325.

[4] J. Cong and C. L. Liu, "On the Over-the-cell Channel Routing Problem," *IEEE Transactions on Computer Aided Design,* April 1990, pp. 408-418.

[5] J. Cong and C. L. Liu, "Over-the-Cell Channel Routing," *Proceedings of International Conference on Computer-Aided Design,* 1988, pp. 80-83, IEEE Computer Society Press, Los Alamitos, California.

[6] J. Cong, B. Preas, and C. L. Liu, "General Models and Algorithms for Over-the-Cell Routing in Standard Cell Design," *Proceedings of 27th ACM/IEEE Design Automation Conference,* June 1990, pp. 709-715, IEEE Computer Society Press, Los Alamitos, California.

[7] J. Cong, D. F. Wong, and C. L. Liu, "A New Approach to the Three Layer Channel Routing Problem," *Proceedings of International Conference on Computer-Aided Design,* 1987, pp. 378-381, IEEE Computer Society Press, Los Alamitos, California.

[8] D. N. Deutsch and P. Glick, "An Over-the-Cell Router," *Proceedings of 17th ACM/IEEE Design Automation Conference,* 1980, pp. 32-39, IEEE Computer Society Press, Los Alamitos, California.

[9] J. S. Deogun, A. Roy, and N. A. Sherwani, "Single Row Routing with Bounded Number of Doglegs Per Net," *Proceedings of 1989 IEEE International Symposium on Circuits and Systems,* May 9-11, 1989, pp. 43-46, IEEE Computer Society Press, Los Alamitos, California.

[10] M. C. Golumbic, "Algorithmic Graph Theory and Perfect Graphs," Academic Press, New York, 1980.

[11] G. Gudmundsson and S. Ntafos, " Channel Routing with Superterminals," *Proceedings of 25th Allerton Conference on Computing, Control and Communication,* 1987, pp. 375-376, Springer-Verlag, New York.

[12] U. I. Gupta, D. T. Lee, and J. Y. T. Leung, "An Optimal Solution for the Channel Assignment Problem," *IEEE Transactions on Computers,* Vol. C-28(11), 1979, pp. 807-810.

[13] F. Harary, "Graph Theory," Addison-Wesley, Reading, MA, 1969.

[14] N. Holmes, N. Sherwani, and M. Sarrafzadeh, "New Algorithm for Over-the-Cell Channel Routing Using Vacant Terminals," *Proceedings of 28th ACM/IEEE Design Automation Conference,* June 1991, pp. 126-131, IEEE Computer Society Press, Los Alamitos, California; also to appear in *IEEE Transactions on Computer-Aided Design.*

[15] N. Holmes, N. Sherwani, and M. Sarrafzadeh, "Algorithms for Over-the-Cell Channel Routing Using the Three Metal Layer Process," *Proceedings of IEEE International Conference on Computer-Aided Design,* November 1991, pp. 428-431, IEEE Computer Society Press, Los Alamitos, California.

[16] H. E. Krohn, "An Over-the-Cell Gate Array Channel Router," *Proceedings of 20th ACM/IEEE Design Automation Conference,* 1983, pp. 665-670, IEEE Computer Society Press, Los Alamitos, California.

[17] K. W. Lee and C. Sechen, "A New Global Router for Row-Based Layout," *Proceedings of IEEE International Conference on Computer-Aided Design,* November 7-10, 1988, pp. 180-183, IEEE Computer Society Press, Los Alamitos, California.

[18] T. Lengauer, "Combinatorial Algorithms for Integrated Circuit Layout," Wiley, New York, 1990.

[19] M. S. Lin, H. W. Perng, C. Y. Hwang, and Y.L. Lin, "Channel Density Reduction by Routing Over the Cells," *Proceedings of 28th ACM/IEEE Design Automation Conference,* June 1991, pp. 120-125, IEEE Computer Society Press, Los Alamitos, California.

[20] S. Natarajan, N. Sherwani, N. Holmes, and M. Sarrafzadeh, " Over-the-Cell Routing for High Performance Circuits," *Proceedings of 29th ACM/IEEE Design Automation Conference,* June 1992, pp. 600-603, IEEE Computer Society Press, Los Alamitos, California.

[21] R. L. Rivest and C. M. Fiduccia, "A 'Greedy' Channel Router," *Proceedings of 19th ACM/IEEE Design Automation Conference,* pp. 418-424, 1982, IEEE Computer Society Press, Los Alamitos, California.

[22] M. Sarrafzadeh and D. T. Lee, "Restricted Track Assignment with Application," *Proceedings of International Symposium on Algorithms and Computation,* Nagoya, Japan, 1992, Springer-Verlag, New York; also to appear in *International Journal of Computational Geometry & Applications.*

[23] N. A. Sherwani, "A Graph Theoretic Approach to Single Row Routing Problem," Ph.D dissertation, University of Nebraska-Lincoln, August 1988.

[24] Y. Shiraishi and Y. Sakemi, "A Permeation Router," *IEEE Transactions on Computer-Aided Design,* Vol. CAD-6, No.3, May 1987, pp. 462-471.

[25] H. C. So, "Some Theoretical Results on the Routing of Multilayer Printed-Wiring Boards," *Proceedings of 1974 IEEE International Symposium on Circuits and Systems,* 1974, pp. 296-303, IEEE Computer Society Press, Los Alamitos, California.

[26] T. G. Szymanski," Dogleg Channel Routing is NP-Complete," *IEEE Transactions on Computer Aided Design,* CAD-4, January 1985, pp. 31-41.

[27] M. Terai, K. Takahashi, K. Nakajima, and K. Sato, " New Model for Over-the-Cell Channel Routing with Three Layers," *Proceedings of IEEE International Conference on Computer Aided Design,* 1991, pp. 432-435, IEEE Computer Society Press, Los Alamitos, California.

[28] B. S. Ting, E. S. Kuh, and I. Shirakawa, "The Multilayer Routing Problem: Algorithms and Necessary and Sufficient Conditions for the Single Row, Single Layer Case," *IEEE Transactions on Circuits and Systems,* Vol. CAS-23, December 1976, pp. 768-778.

[29] T. T. K. Trang, M. Marek-Sadowska, and E. S. Kuh, "An Efficient Single-Row Routing Algorithm," *IEEE Transactions on Computer-Aided Design,* Vol. CAD-3, July 1984, pp. 178-183.

[30] R. S. Tsay, "Exact Zero Skew," *Proceedings of IEEE International Conference on Computer Aided Design,* November 1991, pp. 336-339, IEEE Computer Society Press, Los Alamitos, California.

[31] S. Tsukijama and E. S. Kuh, "Double Row Planar Routing and Permutation Layout," *Networks,* Vol. 12, 1982, pp. 286-316.

[32] P. Widmayer and C. K. Wong, "An Optimal Algorithm for the Maximum Alignment of Terminals," *Information Processing Letters,* Vol. 20, No. 2, February 1985, pp. 76-82.

[33] D. Wang and E.S. Kuh, "New Algorithms for Two and Three layer Channel Routing," *International Journal of Circuit Theory and Application,* Vol. 19, No. 6, December 1991, pp. 525-549.

[34] B. Wu, N. Holmes, N. Sherwani, and M. Sarrafzadeh, "Over-the-Cell Routers for New Cell Model" *Proceedings of 29th ACM/IEEE Design Automation Conference,* June 1992, pp. 604-607, IEEE Computer Society Press, Los Alamitos, California.

[35] T. Yoshimara and E.S. Kuh, "Efficient Algorithms for Channel Routing," *IEEE Transactions on Computer-Aided Design,* Vol. 1, No. 1, 1982, pp. 418-424.

[36] D. Zhou, F. P. Preparata, and S. M. Kang, "Interconnection Delay in Very High-Speed VLSI," *Proceedings of ICCD 1988,* 1988, pp. 52-55, IEEE Computer Society Press, Los Alamitos, California.