NOVEL ROUTING SCHEMES FOR IC LAYOUT

by

Deborah C. Wang

Memorandum No. UCB/ERL M91/75

9 September 1991

NOVEL ROUTING SCHEMES FOR IC LAYOUT

by

Deborah C. Wang

Memorandum No. UCB/ERL M91/75

9 September 1991

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

NOVEL ROUTING SCHEMES FOR IC LAYOUT

by

Deborah C. Wang

Memorandum No. UCB/ERL M91/75

9 September 1991

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Novel Routing Schemes for IC Layout

by

Deborah C. Wang

Abstract

With the advent of multiple metal layers in chip design and multi-chip modules, it is important to explore routing techniques which are not strictly Manhattan with H/V (horizontal and vertical) layers. In this thesis, we present a number of new channel routing algorithms. The methods depart from the traditional channel routing approach by considering non-Manhattan wiring and the overlapping model. At first, only the problem of two-layer routing with two-terminal nets is considered. The routing strategy is based on parallel bubble sorting and river routing techniques. An important concept introduced is the POTENTIAL function, which serves as a measure of the degree of difficulty of a particular channel routing problem. Although the methods are initially derived for two-layer routing with two-terminal nets only, extensions to multi-terminal nets and three-layer routing have been made. Preliminary results indicate that, in many instances, our new algorithm outperforms the traditional approaches. In addition, it requires many fewer vias than traditional methods do.

The last chapter of this thesis presents an optimum scheme for interconnecting the chip core and the I/O pads at the final stage of physical design. The pad placement routine, based on linear assignment, determines the dimension of the pad ring and selects the optimum position for each pad with the objective of minimizing the chip area and the total wire length. The ring router is based on a channel routing algorithm which incorporates additional features to address the special needs of the ring configuration. It attempts to achieve 100% routing completion in a rectangular ring-shaped area with two interconnect layers. The complete package has been implemented as part of the BEAR Layout System for custom chip design.

Prof. Ernest S. Kuh
Committee Chairman

To my family and my teachers . . . . . .

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Layout design of integrated circuit is usually divided into several stages: placement, global routing, detailed routing, compaction and verification. The placement stage decides the optimal locations of logic blocks or modules on the chip. The remaining chip area is used for interconnection. To reduce the complex routing problem into smaller and more manageable subproblems, the routing space is divided into regions. Global routing determines the rough topology of each net in terms of regions, but does not specify its exact pattern. Detailed routing determines the final wiring path for each net. In gate array, standard cell and macro cell design methodologies, routing regions are referred to as channels. Channel routing, one of the most important and extensively studied detailed routing problems is the focus of this thesis.

Many breakthroughs [13, 5, 9, 4, 2, 1, 10, 6, 8, 11] in channel routing theory and algorithms have been reported. The classical and well-studied model for channel routing is the directional Manhattan routing model: one layer is used exclusively for vertical wires , another is used for horizontal wires, and vias are introduced for each layer change. The density of a channel gives a lower bound on the channel height. However, routing in density is frequently difficult to achieve due to the complication introduced by the vertical constraints. It is known that the *restrictive routing problem* (when each net can occupy at most one track) with cyclic vertical constraints does not have a solution [3] and in general, the Manhattan problem in the presence of vertical constraints is NP-complete [7, 12].

In Chapter 2 of this dissertation, we present a number of new channel routing algorithms. The methods depart from the traditional channel routing approach by considering non-Manhattan wiring and the overlapping model. With the advent in multiple metal layers in chip design and multi-chip modules, it is important to explore routing techniques which are not strictly Manhattan with H/V (horizontal and vertical) layers. The routing strategy is based on parallel bubble sorting and river routing techniques. An important concept introduced is the POTENTIAL function, which serves as a measure of the degree of difficulty of a particular channel routing problem. Since there is no need to deal with vertical constraints, we can always achieve 100% routing completion. The routing solution produced by our router has unambiguous layer assignment and is design-rule correct. It has been demonstrated that the method can route channels with channel height less than the channel density. In addition, it requires many fewer vias than traditional methods do.

In Chapter 3, we propose a new 3-layer channel routing scheme by exploiting the additional degrees of freedom offered by the 60° Steiner grid. The routing strategy is based on the river routing technique. Wires on different layers are allowed to overlap. We prove that the channel height, h, produced by our router satisfies $2d/5 \leq h \leq d$, where d is the Manhattan density. Compared to the Manhattan routers, our router has the following advantages: (1) It can always achieve 100% routing completion since it does not need to deal with vertical constraints. vertical constraints, (2) The wire length is expected to be shorter due to the use of 60° wires. (3) A small number of vias is required because the routing solution contains a minimal number of net crossings. In particular, at most one via per net is required for routing two-terminal nets across a channel. (4) Due to the simplicity of the river routing technique, it is very fast.

In Chapter 4, the final stage of detailed routing is considered. The core module containing the ensembled logic is to be connected to the I/O pads. The problem is to find an embedding of the I/O pads in a ring enclosing the core module and to define precise conductor paths necessary to interconnect the terminals as specified by the netlist. A 100% routing completion is required and, in addition, the objective is to minimize the chip area and the total wire length. We developed

a complete package of pad placement and ring routing as part of the BEAR Layout System for custom chip design. The pad placement routine based on linear assignment minimizes the chip area and the total wire length. The ring router performs 100% interconnection in two layers based on the left edge algorithm. Each net is routed in the shortest path. The signal nets can be routed near density to yield a ring of minimal thickness. Shrinkage/expansion of each side of the ring is independent. A practical situation involving multiple power/ground nets and variable widths for Power/Ground nets is considered. The computation complexity of the package is $O(N^2) + O(nlogn)$, where N is the number of pads and n is the number of terminals. With optimized pad placement, we achieved 3% - 18% reduction in chip area and 17% - 31% reduction in total wire length.

In Chapter 5, we discuss important features which can still be enhanced in the existing routers and future research for detailed routing. Multi-terminal nets, L-shaped channel routing and multi-layer channel routing remain interesting as yet unsolved problems.

Chapter 2

Two-Layer Channel Routing

Channel routing is one of the most important phases of physical design of VLSI chips as well as some PC boards. Many channel routing theory and algorithms [51, 19, 43, 12, 7, 3, 45, 26, 40, 47] have been reported. The traditional model for channel routing is the directional Manhattan routing model: net terminals are located on vertical grid lines, two wiring layers are available for interconnection - one layer is used exclusively for vertical wires , another is used for horizontal wires, and vias are introduced for each layer change. Multi-terminal nets are allowed. The density of a channel gives a lower bound on the channel height. However, routing in density is frequently difficult to achieve due to the complication introduced by the vertical constraints. It is known that in general, the Manhattan routing problem in the presence of vertical constraints is NP-complete [31, 49].

In this chapter, we propose new 2-layer channel routing methods which perform well on practical channels and attempt to provide accurate and formal analysis on the quality of the solutions. The approach was developed with an interest in the intrinsic complexity of channel routing problems and their combinatorial structure. Two new routing models, the *mini-swap* model and the *overlap* model are introduced. Non-Manhattan geometry as well as rectilinear wires are used. The routing strategy is based on parallel sorting and river routing techniques and is very different from the Manhattan approach. In particular, vertical constraints no longer exist. Furthermore, the solution produced by either model consists of a minimal set of net crossings which

leads to a small number of vias. To characterize and to evaluate the performance of each model, a function called POTENTIAL is used. Intuitively, the POTENTIAL function measures the degree of difficulty for a given channel routing problem. Based on the performance analysis, we attempt to combine the strength of both models to form a new *hybrid* router. The routing solution produced by our router has unambiguous layer assignment and is design-rule correct. Finally, extensions to handle multi-terminal nets will be discussed.

The remainder of this chapter is organized as follows. Section 2.1 contains the problem definition. In Section 2.2, we present the algorithm and theory associated with the *mini-swap* model for 2-terminal net routing. The *overlap* routing model is introduced and analyzed in Section 2.3. Section 2.4 describes the new hybrid router and its performance. Section 2.5 deals with the multiple-terminal problem and Section 2.6 contains concluding remarks.

2.1 The Problem

A channel is a pair of vectors of nonnegative integers - TOP and BOT - of the same dimension (Figure 2.1).

TOP = $t(1),\ t(2),\ .....t(n)$

BOT = $b(1),\ b(2),\ .....b(n)$

We assume that these numbers are the labels of grid points located along the top and bottom edge of a rectangle. Points having the same positive label have to be interconnected, i.e. they define nets. A 100% routing completion is required and the objective is to minimize the channel area and the number of vias. We first consider a special case of the general channel routing problem.

Definition 1 *A channel is dense iff* $t(i) \neq 0$ *and* $b(i) \neq 0$ *for all i, that is, every grid point on the top and bottom boundaries is occupied by a terminal.*

Definition 2 *A non-trivial 2-terminal net is a net that has exactly two terminals, one on the top and another on the bottom.*

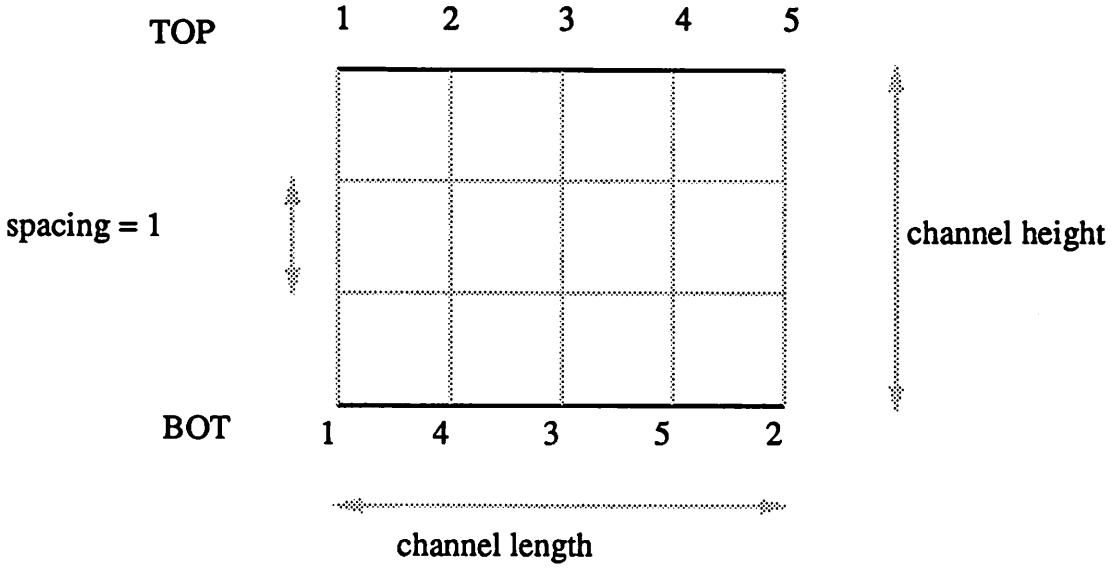


Figure 2.1: The channel routing problem.

Let $\{1, 2, ...n\}$ denote the set of nets. Then in a dense 2-terminal net channel, TOP and BOT are permutations of $\{1, 2, ...n\}$. Without loss of generality, we may assume that nets are arbitrarily ordered on the bottom and are naturally ordered on the top. This is stated as:

Definition 3 *A dense 2-terminal net (D2TN) channel routing problem is specified by:*

TOP = 1, 2, 3, ... n

BOT = a permutation of { 1, 2, ... n }

We denote the top and bottom terminals of net i by (i, q_i).

2.2 The Mini-swap Model

The mini-swap model involves diagonal wires oriented in the +45° and -45° angles as well as vertical wires. The routing strategy is based on parallel sorting techniques [1]. To characterize the channel routing problem under this model, a function called POTENTIAL, originally defined for the parallel bubble sorting [44], is used. Intuitively, the POTENTIAL function measures the degree of difficulty for

a given channel routing problem. We can mechanically evaluate the POTENTIAL function to compute the exact channel height required to route the channel under the *mini-swap* model without actually carrying out the routing steps. This feature makes this model very attractive since knowing the required channel height in the placement stage is valuable.

Furthermore, we prove the solution is optimal under the mini-swap model. The routing solution produced by applying the parallel bubble sorting technique can be unambiguously wired using two interconnect layers. Unlike other Non-Manhattan routers [37] [36] [46] that magnify both column and track spacing by a factor of $\sqrt{2}$ which implies the channel area is doubled, this router ensures that the solution is design rule correct and does not magnify spacing in either direction.

The remainder of this section is organized as follow. In section 2.2.1, we define the new mini-swap model. Section 2.2.2 describes the the routing strategy based on the parallel bubble sorting technique. In section 2.2.3, the POTENTIAL function and its theoretical background are introduced. The optimality theorem based on the POTENTIAL function is established in section 2.2.4. A lower bound for the POTENTIAL function is also given. We compare this routing strategy to the classical Manhattan approach as well as to other non-Manhattan routers in section 2.2.5. Section 2.2.6 contains concluding remarks on the mini-swap model.

2.2.1 Definition of the Mini-Swap Model

The basic idea of the new model is to swap a pair of neighboring nets by two wires, one in the +45° direction, another in the -45° direction (Figure 2.2). We call such a swap a "mini-swap".

Routing of a D2TN channel can be viewed as a vertical stack of steps. A step is a unit-high horizontal strip in which a set of mini-swaps is performed. If a net does not change position in a step, it simply propagates to the next track by a unit vertical wire. An example is shown in Figure 2.3 .

A solution for the D2TN channel routing problem in this model can be constructed in a bottom-to-top step-by-step fashion . The final channel height is

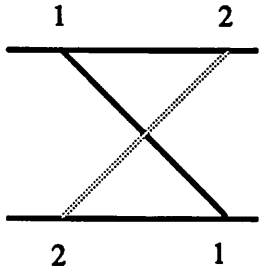


Figure 2.2: A mini-swap.

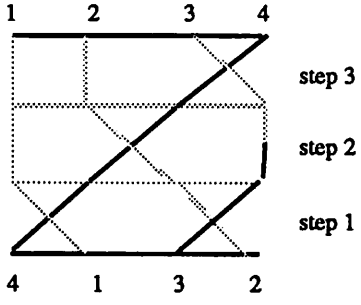


Figure 2.3: In each step, a set of mini-swaps occure.

equal to the number of steps required. Clearly, a D2TN channel routing problem can have many possible solutions under the mini-swap model. Figure 2.4 shows 2 solutions to an instance of a D2TN channel. To measure the performance of a routing strategy, we need to define the optimum solutions under the mini-swap model.

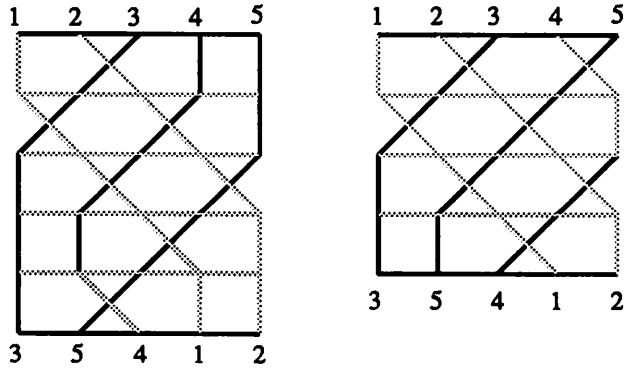


Figure 2.4: Two realizations of the same channel.

Definition 4 *The optimal solution of a two-terminal net channel routing problem under the mini-swap model is one that has*

(a) the minimum channel height and

(b) the shortest total wire length

In search of the optimum solution, the router must determine which pairs of nets to swap in each step so that the final solution is optimum. In the example shown in figure 2.4 , mini-swaps were performed only on pairs of nets that were not ordered. It may be possible that in search of the optimum solution we may want to swap pairs that are already ordered. However, the possibility is ruled out by theorem 1 which states necessary optimality conditions. Before we prove the theorem, we need the following definition.

Definition 5 *A pair of nets is said to be "planar" if it is in the natural order on the bottom. Otherwise, it is said to be "intersecting".*

Theorem 1 *A solution is optimal under the mini-swap model if it has the minimum number of tracks and properties 1 and 2 hold:*

(property 1) planar pairs do not intersect and

(property 2) intersecting pairs intersect only once

Proof: For the sake of contradiction, assume a solution S, has the minimum number of tracks and some planar nets intersect twice. Then there exists a new solution S' (Figure 2.5), which has the same number of tracks and no intersecting planar nets.

Clearly S' has shorter total wire length than S, which implies that S is not optimal.

The proof for property 2 is similar. ■

In other words, a router never needs to swap a pair of nets that are already in the natural order to obtain an optimum solution. This implies that the set of mini-swaps that leads to the solution corresponds to a minimal set of net crossings.

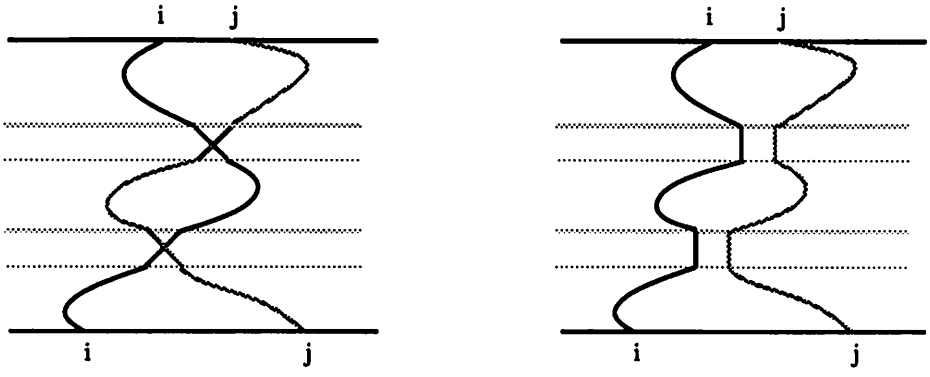


Figure 2.5: For proof of necessary optimality condition.

2.2.2 The Routing Strategy

The basic routing strategy is to construct a wiring path for each net in a bottom-to-top step-by-step manner. In each step, the router selects a set of intersecting pairs to swap. Since decisions made in the earlier steps can affect choices in later steps, choosing an appropriate set of mini-swaps to perform in each step is crucial to the quality of the final solution.

We now describe a routing algorithm based on the parallel bubble sort [1]. In the first step, nets located at odd grid points are compared with the net on their right. If the pair is an planar pair, the pair does not switch positions; otherwise, a mini-swap is performed. The second step is identical to the first one, except this time nets located at even grid points are compared with the net on their right. These two steps are repeatedly performed in this order. The algorithm stops when for two consecutive steps, no pairs of nets switch places. Once the algorithm terminates, the nets are ordered in the natural order. The algorithm must terminate since each pair of nets can switch places only once. Let us demonstrate the algorithm on the example shown in Figure 2.6.

Layer assignment for the wiring path is trivial. Layer 1 is assigned to wires oriented in the +45° angle; layer 2 is assigned to wires oriented in the -45° angle. Vertical wires can be assigned to any layer since they do not cross any other wires.

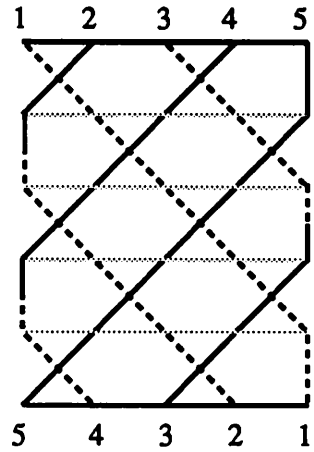
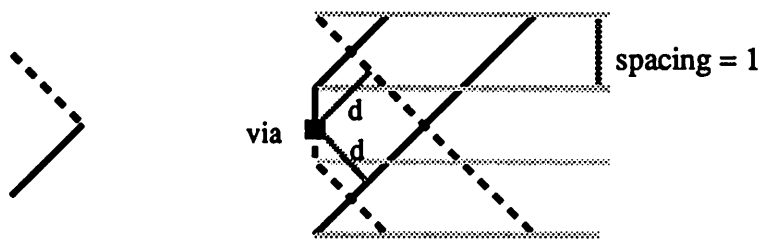


Figure 2.6: Routing by odd-even transposition.



(a) An impossible situation

(b) A via is placed at the center of a vertical segment to endure d > 1.

Figure 2.7: Postions of vias.

Vias are introduced for layer changes between a +45° wire and -45° wire. Due to the odd-even transposition procedure, the situation shown in Figure 2.7(a) could never happen: the +45° wire of a net always connects to a -45° wire through a vertical segment as shown in Figure 2.7(b). We place a via at the midpoint of the vertical segment. This ensures that the wires satisfy the design rule so that there is no need to magnify either the column spacing or the row spacing by $\sqrt{2}$.

Formally, let us denote the permutation of nets on track t by:

$A_t = (a_t(1), a_t(2), ...a_t(n))$

Then, $A_0 = (a_0(1), a_0(2), ...a_0(n))$, where $a_0(i) = b(i)$, for all i. We also assume there is an infinite number of auxiliary nets, represented by ..... a(-1), a(0), and a(n+1), a(n+2),...., where for all $i \leq 0$ and $t > 0, a_t(i) = -\infty$ and for all $i > n$ and $t > 0$, $a_t(i) = +\infty$. These auxiliary nets do not disturb the routing of regular nets at any time. For each track $t \geq 0$ and each integer i, the net, $a_t(i)$, at the ith grid point at track t is given by:

if (i + t) is even , $a_t(i) = min(a_{(t-1)}(i),\ a_{(t-1)}(i+1))$

if (i + t) is odd , $a_t(i) = max(a_{(t-1)}(i-1),\ a_{(t-1)}(i))$

The dynamic behavior of this routing scheme is the same as that of the parallel bubble sorter realized on a two-way infinite linear array. If, for every i, $a_t(i) \leq a_t(i+1)$, then A_t is said to be sorted. The computing time of the parallel bubble system is the smallest t such that A_t is sorted.

2.2.3 The POTENTIAL function

Since the routing scheme is a direct adaptation of the parallel bubble sort, a characterization of the bubble system can also be used to analyze the routing process. In particular, we are interested in the computing time of a parallel bubble system which corresponds to the required height of the D2TN channel routing problem. A function called POTENTIAL, first introduced by [44], proved to be very useful in evaluating the computing time of the bubble system. This section defines the POTENTIAL function and reviews the theorem proved by [44]. Before we introduce the POTENTIAL function, we need to define the following four terms.

Definition 6 *For each (i,j,t), where $1 \leq i, j \leq n$, and $t \geq 0$, and the set of nets, S, we define (Figure 2.8):*

- *ORDER(i,j,t,S) is the number of indices $p \in S$ such that $i < p \leq j$ and $a_t(i) \leq a_t(p)$, or such that $j \leq p < i$ and $a_t(p) \leq a_t(i)$*
- *NOTORDER(i,j,t,S) is the number of indices $p \in S$ such that $i < p \leq j$ and $a_t(p) \leq a_t(i)$, or such that $j \leq p < i$ and $a_t(i) \leq a_t(p)$*

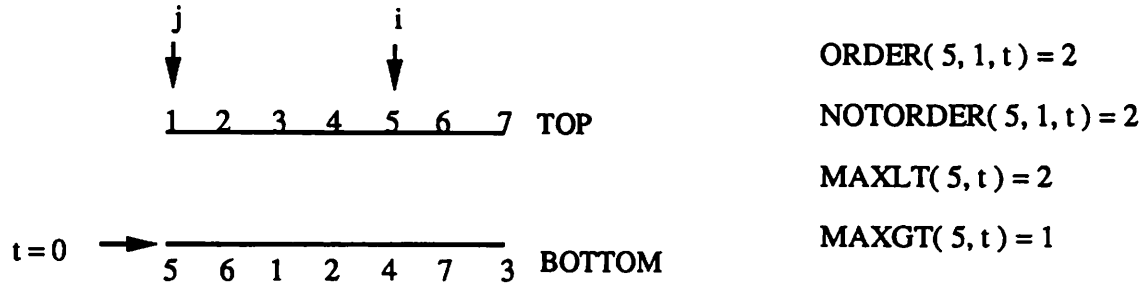


Figure 2.8: Illustration of ORDER, NOTORDER, MAXLT, MAXGT.

- *MAXLT(i,t,S) = max(0* $\cup$ { *ORDER(i,p,t,S) - NOTORDER(i,p,t,S) + 1* | $p \in S, p < i$ *and* $a_t(i) \leq a_t(p)$ })
- *MAXGT(i,t,S) = max(0* $\cup$ { *ORDER(i,p,t,S) - NOTORDER(i,p,t,S) + 1* | $p \in S, i < p$ *and* $a_t(p) \leq a_t(i)$ })

We are now ready to define the POTENTIAL function.

Definition 7 *For any position indexing* $i \leq n$, *and any* $t \geq 0$, *the function POTENTIAL(i,t) is defined as (Figure 2.9):*

- *When NOTORDER(i,1,t,S) = 0,*
  POTENTIAL(i,t) = NOTORDER(i,n,t,S) + MAXGT(i,t,S)
- *When NOTORDER(i,n,t,S) = 0,*
  POTENTIAL(i,t) = NOTORDER(i,1,t,S) + MAXLT(i,t,S)
- *When NOTORDER(i,n,t,S)* $\neq$ *0 and NOTORDER(i,n,t,S)* $\neq$ *0,*
  POTENTIAL(i,t) = NOTORDER(i,1,t,S) + NOTORDER(i,n,t,S) +
  max(1, MAXLT(i,t,S), MAXGT(i,t,S))

The POTENTIAL function for the entire bubble system is defined by:
POTENTIAL(t) = max(POTENTIAL(i,t) | $1 \leq i \leq n$ *)*

Intuitively, $NOTORDER(i, 1, t, S)$ is the number of nets to the left of net i that are to be swapped with i. $NOTORDER(i, N, t, S)$ is the number of nets to the right of i that are to be swapped with i. $MAXLT(i, t, S)$ is the maximum number of

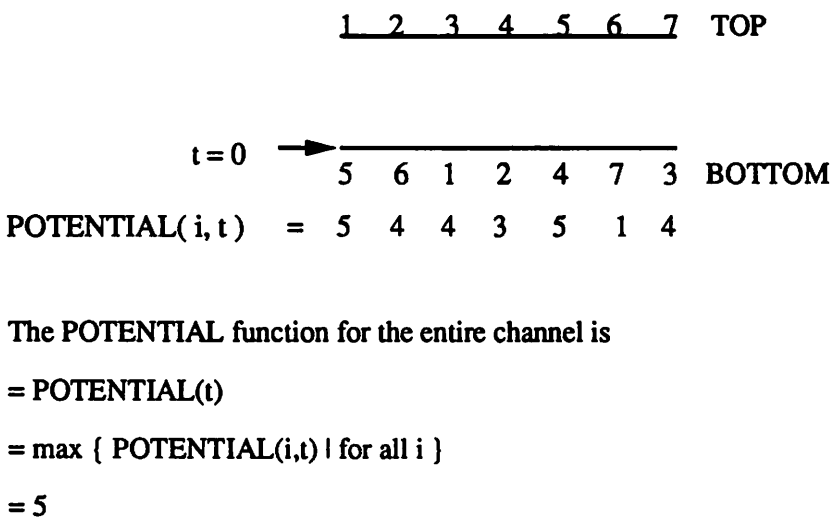


Figure 2.9: Illustration of the POTENTIAL function.

steps it takes for i to swap with a net $j > i$ which is to the left of i and $MAXGT(i,t,S)$ is the maximum number of steps it takes for i to swap with a net $j < i$ which is to the righT of i. $POTENTIAL(i,0)$ is the number of steps required for i to rearch its target postions. Therefore, the overvall POTENTIAL(0) is chosen to be the POTENTIAL value of the worst net. Note that when both NOTORDER$(i,1,t,S)$ = NOTORDER$(i,n,t,S) = 0$, POTENTIAL$(i,t) = 0$ from either (1) or (2). Hence the function is well defined. From the above definition, it is clear that POTENTIAL(i,t) $= 0$ if and only if NOTORDER$(i,1,t,S)$ = NOTORDER$(i,n,t,S) = 0$. An immediate consequence of this fact is:

Fact 1 If POTENTIAL$(t) = 0$, then A_t is sorted.

Fact 2 If POTENTIAL$(i,t) = 0$, then POTENTIAL$(i,r) = 0$ for all $r > t$

We now state the main theorem proved by [44].

Theorem 2 *If $t \geq 1$ and A_t is not sorted, then*
POTENTIAL(t+1) = POTENTIAL(t) - 1

Corollary 2.1 *The computing time of the bubble system is POTENTIAL(0) or POTENTIAL(0)+1.*

The salient feature of the bubble system is that the POTENTIAL value consistently decreases by 1 per step. In other words, when POTENTIAL(0) = k, the required height of the D2TN channel is k or $k+1$. The value of POTENTIAL(0) is defined solely by the initial permutation A_0 , without referring to the intermediate configurations A_t for $t > 0$. The POTENTIAL function can be precisely evaluated to compute the number of tracks require to route the D2TN channel without actually carrying out the routing steps. Another feature is that the decision of whether to swap a pair or not is local and does not depend on the locations of the rest of the nets. This feature makes this algorithm attractive in a parallel mode of operation. Also observe that the wiring path for each net is monotonic in the vertical direction.

2.2.4 The Main Theorem

The parallel bubble sort provides a simple routing strategy to produce a solution in the mini-swap model. Given the initial value of the POTENTIAL function, we saw the bubble scheme consistently decreases the POTENTIAL value by one until the target value zero was reached. Do better algorithms exist for which the POTENTIAL value decreases rapidly , say, by more than 1 per step ? For the example shown in Figure 2.10, the first step of the parallel bubble algorithm would switch the 2 pairs shown in (a). But as many as 6 pairs could have been switched (b). In the following theorem, we prove a sufficient optimality condition for any algorithm under the mini-swap model. See Appendix A and B for detailed proof.

Theorem 3 *Potential value decreases at most by 1 per net per step under the mini-swap model.*

The following theorem gives a lower bound on the POTENTIAL function.

Theorem 4 *For each net i, we define the displacement(i) as the horizontal distance between its two terminals. Then POTENTIAL(0)* $\geq max(\{displacement(i) \mid 1 \leq i \leq n\})$.

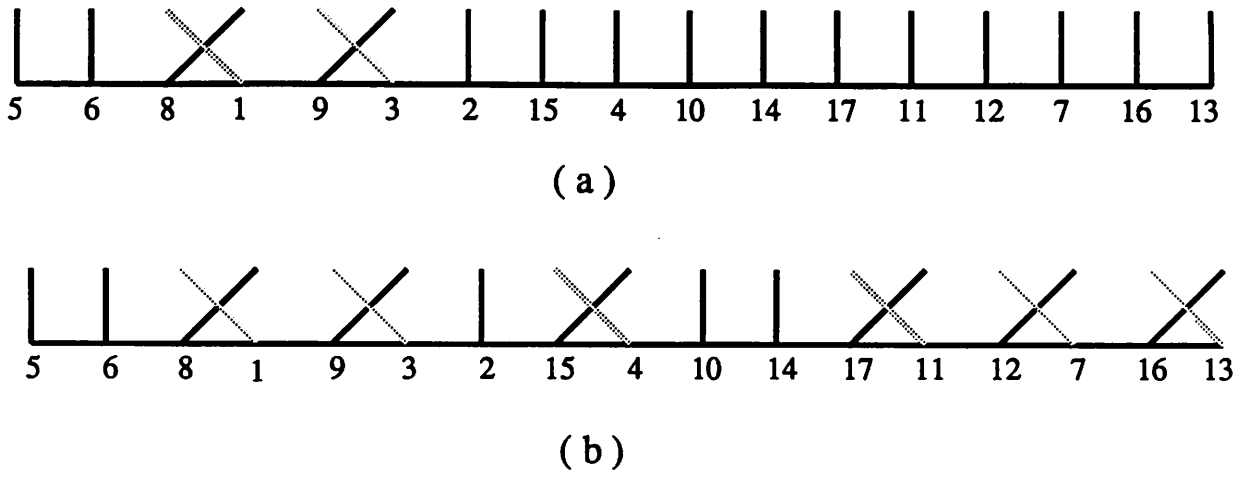
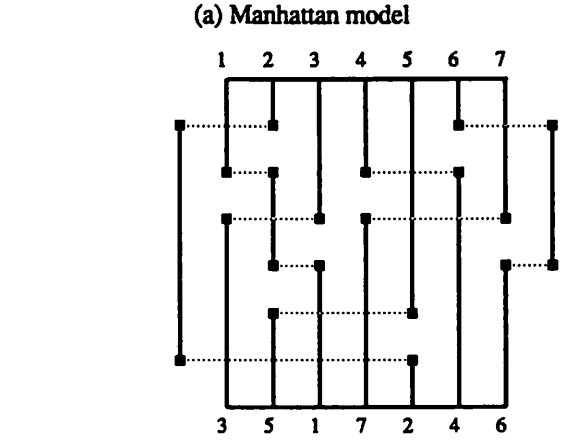


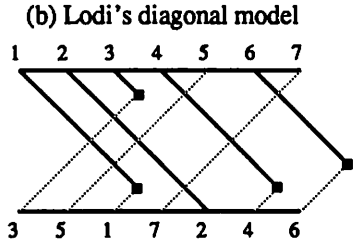
Figure 2.10: Choosing different pairs to swap in a step.

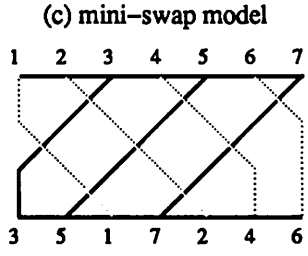
2.2.5 Results and Comparison

Routing results of two D2TN channels are shown in Figure 2.11 and 2.12. Compared to the Manhattan routers, our router has the following advantages:

- There is no need to deal with vertical constraints.
- The required channel height, POTENTIAL(0), can be precisely computed.
- For D2TN channels that have a POTENTIAL(0) value less than the Manhattan density, our router can out-perform any Manhattan routers.
- Extra columns outside the channel's span are never used.
- Wirelength is expected to be shorter due to the use of diagonal wires.
- Being a minimal crossing solution, the routing requires only a small number of vias. We observed that most nets require zero or one via, whereas Manhattan routing typically requires at least 2 vias per net.
- It is inherently suitable for parallel mode of operations.
- In standard cell design, it is very difficult to optimize the assignment of pins to feedthroughs when the objective function is the density. In our environment,







(d) Comparison

	Man	Lodi	Mini Swap
height	7	3	3
area	56	42	18
vias	20	4	0
wire length	71	42	28

Figure 2.11: A channel is routed in three different models. Results are tabulated in (d).

one would want to minimize the maximum displacement of a net. This can be easily done by linear assignment.

In comparison with the diagonal model channel router proposed in [37], we make the following observations. (1) The results produced by our router are design

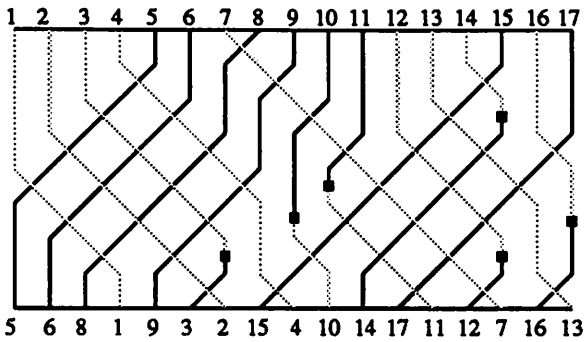


Figure 2.12: A channel with density = 7 is optimally routed by the mini-swaps.

rule correct while the diagonal router [37] violates design rules. In order to rectify their violations, the spacing between terminals for [37] must be magnified by $\sqrt{2}$. This is not a realistic and practical design since pin assignment has been done prior to the channel routing phase. (2) Suppose we do allow the pins to be reassigned in order to make a fair comparison. The diagonal router [37] completes routing of a D2TN channel in $max(\{displacement(i) \mid 1 \leq i \leq n\})$ tracks which is smaller than the height obtained by our router (see theorem 4). But due to the $\sqrt{2}$ factor for both the column spacing and row spacing, the **channel area** produced by the diagonal router [37]is twice the height. On the other hand, our channel area simply equals to the height, POTENTIAL(0). Therefore, our router is better than [37] for channels whose POTENTIAL values are less than twice the maximum displacement.

2.3 Concluding Remarks on the Mini-Swap Model

In section 2.2, we have proposed a new channel routing model, the "mini-swap" model. The routing algorithm is based on the parallel bubble sorting technique. A function named POTENTIAL, originally defined for characterizing the bubble system, can be evaluated to compute the precise number of tracks required to route the channel. The function evaluation of POTENTIAL for a given channel can be obtained without referring to the intermediate routing steps. We have established necessary

and sufficient optimality conditions for routing under the "mini-swap" model. The final routing solution has an unambiguous layer assignment and is design-rule correct. Our results show that a class of dense two-terminal net channels can be routed in a height less than the Manhattan density.

2.4 An Overlap Model

2.4.1 Introduction

The results of the previous section show the advantages of using the mini-swap model for routing channels whose POTENTIAL value is less than or equal to the Manhattan density. Using the mini-swap model to route channels with "long" nets (nets spanning a number of columns which is larger than the Manhattan density), however, leads to unsatisfactory results. How can we overcome this deficiency? Since Theorem 3 confirmed that we could not reduce the required channel height by cleverly selecting pairs to swap, the only alternative that may lead us to a better solution is to relax the constraint of pair-wise transpositions. In other words, we shall allow long nets to swap with more than one net at a time so that they may reach their target positions more rapidly. In this section, we introduce an overlap model. The routing algorithm is based on the river routing technique, in which, like the mini-swap model, vertical constraints do not exist. To analyze the routing performance, the POTENTIAL function again proves to be a useful tool in providing an upper bound of the channel height. We also establish a lower bound of the channel height that incorporates the relative ordering of the nets. For a class of channel routing problems, this demonstrates that density is not a tight lower bound. Unlike other overlap models [29, 42, 6] that used two or more vias per net, the routing solution produced by our router can be wired using two interconnect layers so that at most one via is required per net. Furthermore, all wires are monotonic in both the horizontal and vertical directions.

This section is organized as follow. The routing algorithm based on river routing is described in section 2.3.2. In section 2.3.3, we establish the lower bound

and upper bound of the channel height by considering the permutation of the nets. In particular, we prove the initial POTENTIAL value of the channel is an upper bound. A refinement of the routing procedure is introduced in section 2.3.4. Conclusion remarks on the overlap model are given in section 2.3.5.

2.4.2 Algorithm for the Overlap Model

The basic idea of the new scheme is to allow long nets to swap with more than one net while maintaining the integrity of net crossings. To improve the results of the mini-swap model, the long nets must be given a better chance in swapping with other nets. At the same time, we want to keep the number of net crossings minimal so that the number of vias will not increase.

The routing algorithm is based on the river routing technique. A dense 2-terminal net channel routing problem is specified in Definition 3. A net is said to be a *right* net if $B(i) > i$; otherwise, it is said to be a *left* net (Figure 2.13). The basic idea of the overlap model is to divide the nets into two disjoint subsets: $S1$ = { *right* nets } and $S2$ = { *left* nets } and route each set independently. A top level description of the algorithm is given below:

step 1) Nets are divided into two sets: S1 = { right nets }; S2 = { left nets }

step 2) Sort S1 in decreasing order of their top terminals. Route S1.

step 3) Sort S2 in decreasing order of their bottom terminals. Route S2.

The solution for the D2TN channel routing problem is constructed in a bottom- to-top net-by-net fashion. Let us demonstrate the algorithm on the example shown in Figure 2.14. The right nets are sorted and routed sequentially in a river routing fashion. The procedure begins by constructing a rectilinear wiring path for the first right net. The path begins at the bottom terminal and ends at the top terminal. The wiring path for the second right net simply follows the path of the first right net and ends at its top terminal. This process continues until all of the right

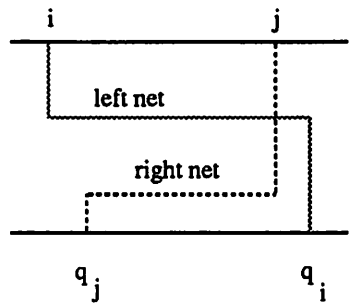
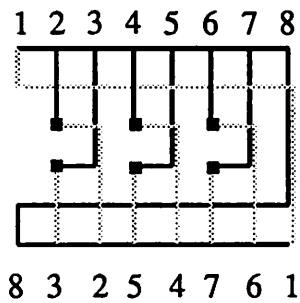


Figure 2.13: Definition of a right net and a left net.



Sorted Right Nets = { 8, 7, 5, 3 }

Sorted Left Nets = { 1, 6, 4, 2 }

Figure 2.14: Routing in the overlap model.

nets are routed. All of the paths are monotonic in both the horizontal and vertical directions. Step 3 is identical to step 2, except that this time the left nets are routed.

Layer assignment for the wiring paths is trivial. For the right (left) nets, the vertical segment attached to their bottom (top) terminals is assigned layer 2 (1) while the remaining wire segments are assigned layer 1 (2). If a right (left) net, i, has $NOTORDER(i,1,0,S) = 0$ ($NOTORDER(i,1,0,S) = 0$), then i can be routed on a single layer, see Figure 2.14 . A via is introduced for each layer change. Clearly, the routing paths for a pair of planar nets do not intersect and intersecting pairs intersect only once. This implies that the routing solution contains a minimal set of net crossings.

2.4.3 Performance Analysis

We establish four results associated with the overlap model: lower- and upper- bounds on channel height, wire length and number of vias. The lower- and upper- bounds on the channel height are derived by considering the permutation of nets. Suppose the D2TN channel routing problem is specified as a pair of vectors: $TOP = \{ 1, 2, ..., n \}$ and $BOT = \{ b(1), b(2), ... b(n) \}$. Let us define the term *NOTORDER* and the function *POTENTIAL* as in Definition 6 and 7.

Theorem 5 *The lower bound on the channel height under the arbitrary overlap model is* $max\{ NOTORDER(1,i,0,S1) + NOTORDER(i,n,0,S2) \mid 1 \leq i \leq n \}$

Proof: At each column i, the number of horizontal tracks required on layer 1 is at least *NOTORDER(i,1,0,S1)*. The number of horizontal tracks required on layer 2 is at least *NOTORDER(i,n,0,S2)*. Since the layer 2 tracks must be stacked on top of the layer 1 tracks, this implies the height of the channel at column i is at least the sum of the two terms. The overall channel height is then the maximum over all columns. ■

The above lower bound is derived by considering the permutation of the nets. It is not only a tighter bound than the Manhattan density, d, but also demonstrates

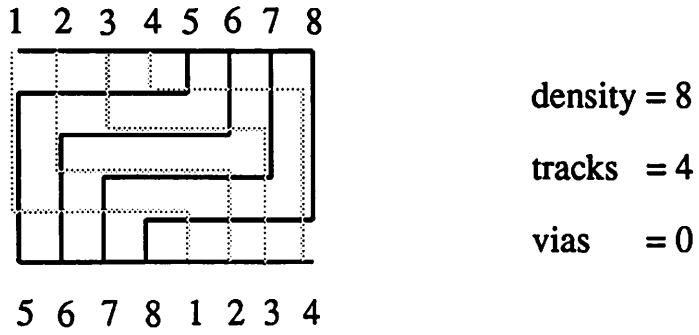


Figure 2.15: A channel is routed in a height equal to the lower bound.

that $d/(L-1)$ [34, 28] is not a universal lower bound under the unrestricted overlap models, where L is the number of layers. The example in Figure 2.15 is routed by the overlap router in a height equal to the lower bound, which is half of the Manhattan density. We prove in the following theorem that, in contrast to the bubble system, the POTENTIAL value will decrease by more than 1 per track (amortizedly) under the overlap model.

Theorem 6 *The upper bound on the channel height under the arbitrary overlap model is*
$POTENTIAL(0) + 2$.

Proof: The proof is established by amortized analysis. The total decrease of POTENTIAL value is average over the total number of tracks. A accounting method guarantees the average performance of each net in the worst case. See Appendix C for detailed proof. ■

The essential implication of Theorem 3 and Theorem 6 is that when long nets are present in a channel routing problem, the POTENTIAL value of the channel is much larger than the density. This means that the overlap model works better than the mini-swap model. On the other hand, if all nets are "short", i.e. the POTENTIAL value of the channel is equal to or less than the density, the mini-swap model is more attractive. In section 2.4 we shall introduce a hybrid router

that combines the strengths of both models to handle all types of channel routing problems.

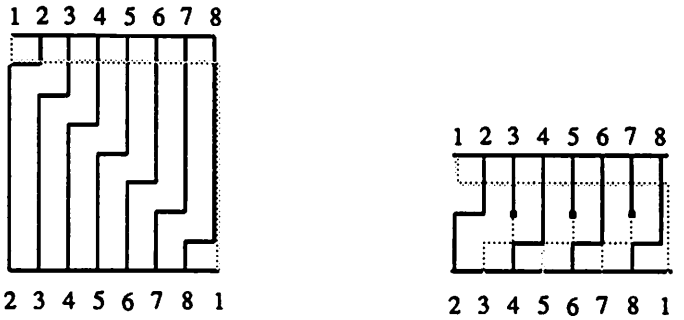
The wiring path for each net is monotonic in both the horizontal and vertical direction. There are no detours in the final solution. This is a direct consequence of the routing algorithm and leads to the following fact.

Fact 3 In the routing solution produced by the rectilinear overlap router, the wire length of net i is equal to $displacement(i)+(finalchannelheight)$, where $displacement(i)$ = horizontal distance between the terminals of net i.

There are a number of reasons that the number of vias should be kept small: (1) In integrated circuit processing, the more vias you have on a chip the lower the yield will be. (2) Every via has an associated resistance which affects the circuit performance. (3) The size of the via is usually larger than the wire width. So more vias means more routing space is needed. The overlap routing procedure has the distinct advantage of generating a small number of vias for D2TN channel routing problems. Given any D2TN channel routing problem, the overlap routing algorithm introduces at most one via per net. Furthermore, the number of vias required can be calculated by the following lemma.

Lemma 1 *For any right (left) net i, if NOTORDER(i,1,0,S1) = 0 (NOTORDER(i,n,0,S2) = 0), then net i has no via in the final solution produced by the overlap router. Otherwise, net i has one via. The total number of vias can be precisely calculated by evaluating*
NOTORDER(i,1,0,S1) or NOTORDER(i,n,0,S2) for all i.

Proof: If a right net i has $NOTORDER(i, 1, 0, S1) = 0$, then by construction, net i does not require a via. Similarly, if a left net j has $NOTORDER(j, N, 0, S2) = 0$, net j needs no via. Hence, the total number of vias can evaluated by adding the number of right and left nets that satisfy the above conditions. ■



(a) Before refinement (b) After refinement

Figure 2.16: Refinement of the overlap routing procedure.

2.4.4 Refinement of Routing Strategy

One weakness of the overlap routing method is its inefficiency in handling a consecutive sequence of planar nets, as shown in Figure 2.16(a). A modification of the routing procedure to improve routing results in such situations is proposed. The basic idea is that instead of routing all right (left) nets in one step, we route the right and left nets alternatingly until the nets are sorted. The remaining nets are planar and can be optimally routed by the "layer per net" method [4, 35]. The "layer per net" method states that, given two interconnect layers and a set of planar nets, $\{ 1, 2, ...M \}$, we assign the odd-numbered nets to layer 1 and the even-numbered nets to layer 2, solve the two one-layer river routing problems that are thus formed to yield the minimum width channel.

Refined Routing Algorithm for Rectilinear Overlap Model

step 1) Divide the nets into two sets: $S1 = \{\text{right nets}\}$ and $S2 = \{\text{left nets}\}$

step 2) Sort right nets in decreasing order of their top terminals: $S1 = \{r1, r2, r3, ...\}$

step 3) Sort $S2$ in decreasing order of their bottom terminals: $S2 = \{l1, l2, l3, ...\}$

step 4) Alternatingly route the right and left nets, ie. route in the order:
$r1$, $l1$, $r2$, $l2$, $r3$, $l3$, ... until the nets are sorted.

step 5) Route remaining planar nets by the "layer per net" model

The example in Figure 2.16(a) is rerouted by the refined procedure to yield a much better solution as shown in Figure 2.16(b).

2.4.5 Concluding Remarks on the Overlap Model

Compared to the Manhattan model, the overlap model has the following advantages:

- There is no need to deal with vertical constraints
- For channels that have POTENTIAL(0) value less than the Manhattan density, our model can out-perform any Manhattan routers
- Extra columns outside the channel's span are never used
- Because it is a minimal crossing solution, we expect only a small number of vias to be required.

The overlap model is better than the mini-swap model in two respects: (1) It yields a small channel area for channel with long nets. (2) Variable wire widths can be incorporated easily. However, overlapping wires may increase crosstalk between signals and unlike the mini-swap algorithm, the overlap algorithm is not suitable for parallel operations.

2.5 The Hybrid Router

We combine the advantages of the overlap and the miniswap models to form a new hybrid router. This router pre-routes long nets using the overlap algorithm so that the amount of overlapping wire is limited. The remaining nets, ie. relatively short nets, are routed by the miniswap algorithm. The user can specify the maximum amount of overlap allowed.

2.5.1 Algorithm

The basic idea is as follows: When the channel is routed solely by the mini-swaps, we know the channel height is POTENTIAL(0). Suppose we pick some long nets which would cause the mini-swap model to disgrace itself, and pre-route them using the overlap strategy. If the resulting channel height is better than before, we can choose more long nets and continue the process provided the maximum amount of overlap is not exceeded. A top level description of the algorithm is given below:

For $i \leftarrow 1$ to $n-1$ do

i) pre-route i longest nets by the overlap model
amount of overlap = i,
calculate current channel height, t(i),
$t(i)$ = (tracks for pre-routes) + (POTENTIAL value of remaining channel)

ii) compare height to last iteration:
if ($t(i) < t(i-1)$) and (max overlap is not exceeded)
$i = i + 1$
continue
else stop

After selecting a set of long nets, it is not essential to pre-route them to their final positions. It suffices to pre-route them to intermediate positions with the objective of minimizing the POTENTIAL value. Our aim is to simultaneously process the long nets and to find an intermediate position for each net. Assignment of a net to a position should be done in such a way that the POTENTIAL value of that net is as small as possible. Using this criterion, many nets may compete for the same position. To find a good assignment of nets to positions, we use a linear bottleneck assignment approach. We build a complete bipartite graph: G(V,U,E), with two sets of nodes. One set, V = P(i), i = 1,...K , represents the nets; the other set, U = 1,2,...K , represents the positions. K is the number of long nets. Edge (i, j) connects

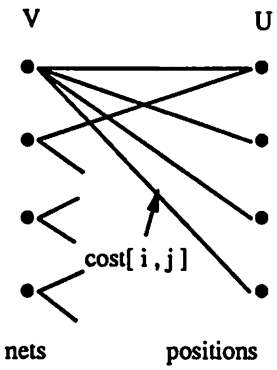


Figure 2.17: Linear Assignment.

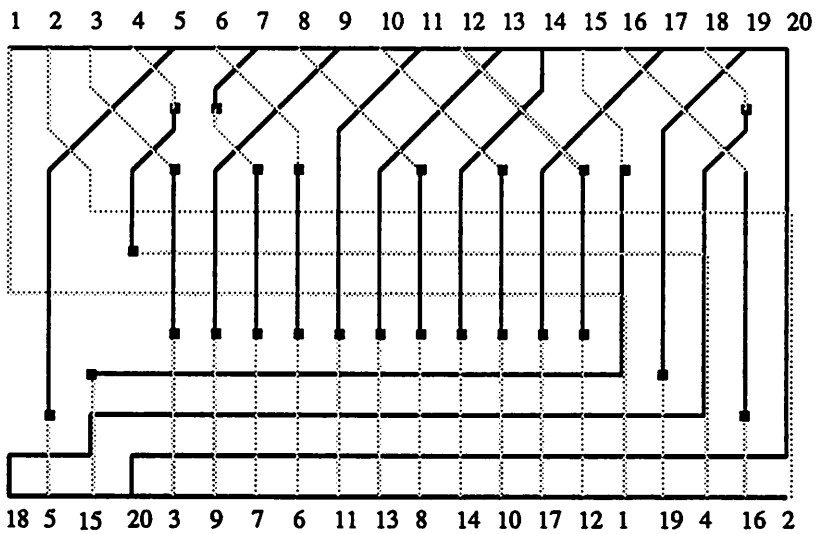


Figure 2.18: A channel is routed in density by the hybrid router.

net node i to position node j. Edge (i, j) is labeled with cost c[i,j] indicating the the POTENTIAL value of net i when it is assigned to position j. (Fig. 2.17). Standard methods may be used to solve this problem. An example routed by the hybrid router is shown in Figure 2.18.

2.5.2 Performance Analysis

Since the hybrid router is a combination of the mini-swap model and the overlap model, it should perform at least as well as either model in the worst case. Given a channel routing problem, if the algorithm terminates in the first $((n-1)\text{th})$ iteration, the solution has purely mini-swaps (overlaps). Therefore, the mini-swap model and the overlap model can be considered special cases of the hybrid router. We summarize the performance bounds of the hybrid router in the following theorem.

Theorem 7 *Lower Bound on the channel height required by the hybrid router*

$$= max\{NOTORDER(i,1,0,S1) + NOTORDER(i,1,n,S2) \mid 1 \leq i \leq n\})$$

Upper Bound of channel height required by the hybrid router

$$= POTENTIAL(0) + 1.$$

Due to the pre-routing strategy, planar nets may intersect twice in the final solution. For example, see net 15 and net 17 in Figure 2.18. Therefore, the number of vias increases in general when the solution does not correspond to a set of minimal net crossings.

2.5.3 Results

The hybrid router is implemented in the C language on a DEC3100 running Ultrix Worksystem V2.1. Figure 2.19 shows the actural routing results of Ex4. Ex4 is a channel with 48 nets. We have attempted to run this example using other routers available [45, 16, 22] but they either failed to complete routing or produced substantially worse results. Table 1 lists the channels tested with 100% routing completion in all cases. Several D2TN channels are routed in a height less than the density.

2.6 Multi-terminal Nets Routing

To extend the routing algorithm to handle multi-terminal nets, we partition each multi-terminal net into 2-terminal subnets (Figure 2.20) and classify each subnet

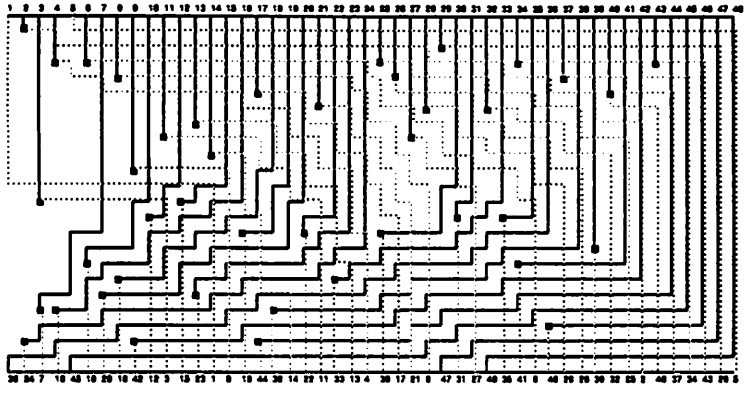


Figure 2.19: Ex4 routed by the hybrid router.

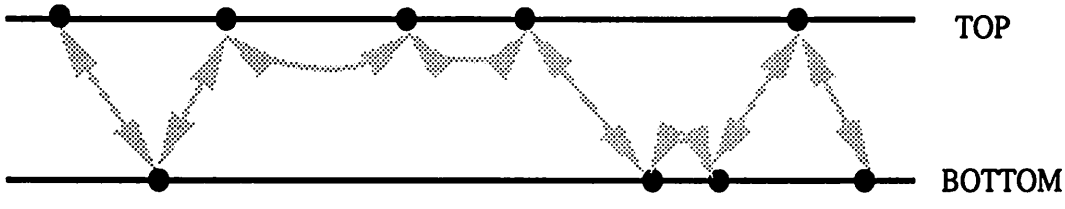


Figure 2.20: Partition of a multi-terminal net into two-terminal subnets.

as a *right* net or a *left* net. A 2-terminal subnet with one terminal on TOP and the other on BOTTOM is said to be a *2-sided* subnet; otherwise, it said to be a *1-sided* subnet. The 2-sided subnets can be categorized as left or right easily. However, the classification of 1-sided subnets is ambiguous and may affect the quality of solution.

Definition 8 *Given I, a two-terminal 1-sided subnet:*

- *I is top-sided iff both terminals of I lie on TOP.*
- *I is bottom-sided iff both terminals of I lie on BOTTOM.*

In the following lemmas, we show that not all 1-sided subnets can be classified either way.

Lemma 2 *Given a 1-sided subnet I,*

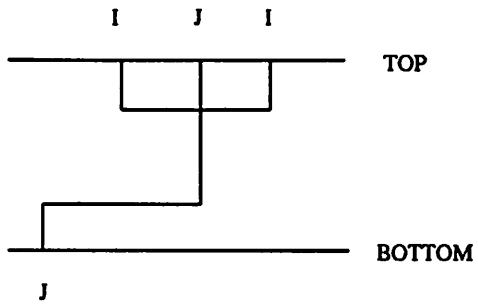


Figure 2.21: For proof of lemma 2(i).

i) I is top-sided and I intersects a right net ⇒ I must be a left net.

ii) I is bottom-sided and I intersects a left net ⇒ I must be a right net.

iii) I is top-sided and I intersects another top-sided subnet J ⇒ I and J can not both be right nets.

iv) I is bottom-sided and I intersects another bottom-sided subnet J ⇒ I and J can not both be left nets.

Proof:

i) Suppose for the sake of contradiction that net I is classified as a right net. Then by the algorithm, net I must be routed before net J on layer 1 (Figure 2.21). This shortes the two wires.

ii) Suppose for the sake of contradiction that net I is classified as a left net. Then by the algorithm, net I must be routed before net J on layer 2 (Figure 2.22). This shortes the two wires.

iii) Suppose for the sake of contradiction that both net I and net J are classified as right nets. Then the algorithm would route both nets on layer 1 (Figure 2.23) and result in conflict.

iii) Suppose for the sake of contradiction that both net I and net J are classified as left nets. Then the algorithm would route both nets on layer 2 (Figure 2.24) and result in conflict.

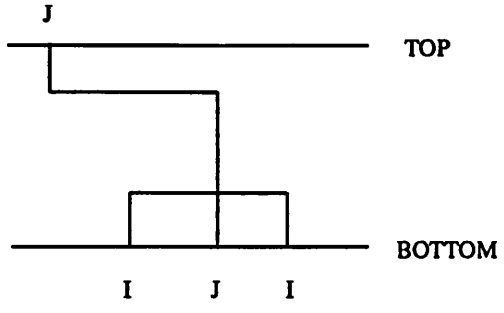


Figure 2.22: For proof of lemma 2(ii).

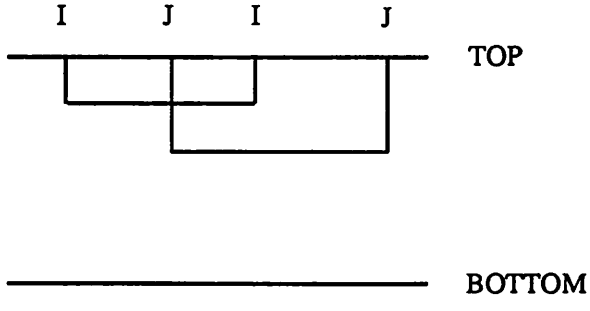


Figure 2.23: For proof of lemma 2(iii).

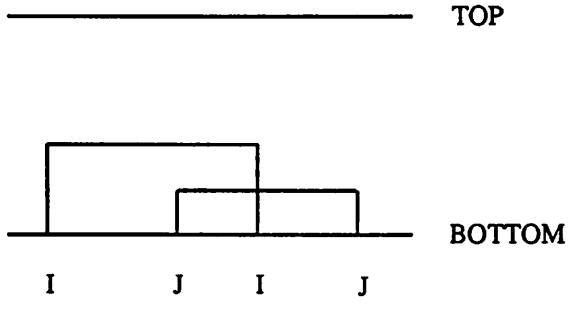


Figure 2.24: For proof of lemma 2(iv).

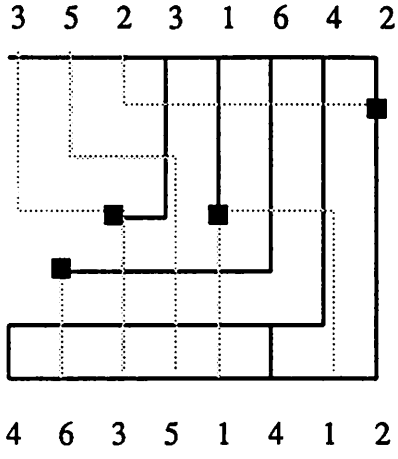


Figure 2.25: Example of multi-terminal net routing.

■

Figure 2.25 shows an example of multi-terminal net routing. Most 1-sided nets in practical examples are classified as above. The classification of the remaining 1-sided subnets is guided by local congestion analysis. The left subnets and right subnets then are sorted independently by their terminal positions and routed in the rivering routing fashion as discussed in the previous section. We observe that this method for routing multi-terminal nets is straight-forward but not necessarily optimal. Future work should investigate better strategy for handling multi-terminal nets.

In a channel routing problem, nets may exit the channel on either its left end or its right end. Routing of these so-called side nets is done by the following procedure (Figure 2.26):

The analysis and theorems introduced for the D2TN channel routing problem in Section 2.3.3 can be generalized for the multi-terminal nets problem by replacing nets with subnets in the equations. Figure 2.27 shows that the Deutsch's difficult channel is routed with 23 tracks and has 25% less vias than [12, 45].

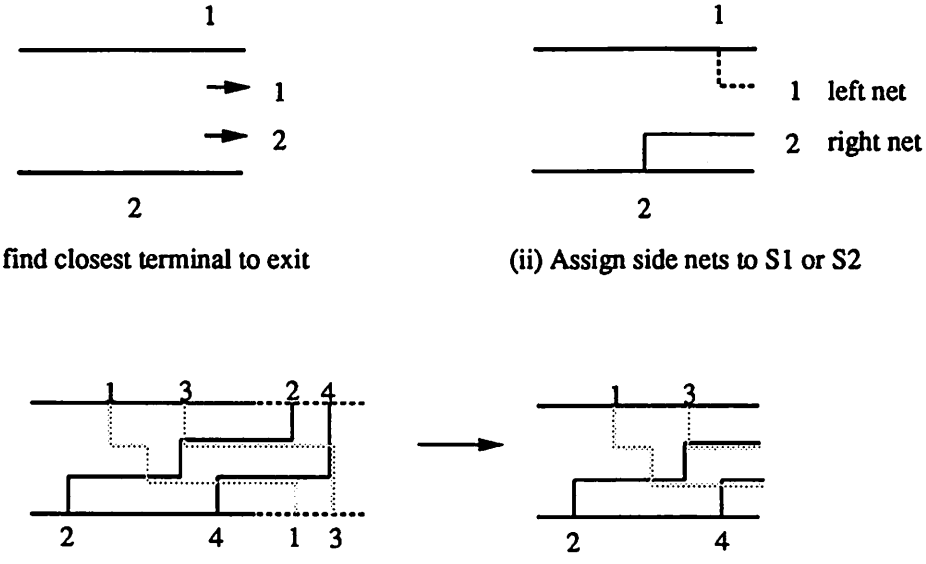


Figure 2.26: Routing side net.

2.7 Conclusion

This chapter departs from the traditional channel routing approach by considering non-Manhattan wiring and the overlapping model. At first, only the problem of two-layer routing with two-terminal nets is considered. Two new routing models, the *mini-swap* model and the *overlap* model are introduced. The mini-swap approach based on bubble sorting is especially attractive for problems with short nets, while the river routing approach is more suitable for problems with long nets. A hybrid method which combines the two is introduced. In the limiting case, the method yields either the mini-swap approach or the river routing approach. The routing techniques are very different from the Manhattan approach. In particular, vertical constraints no longer exist. To characterize and to evaluate the performance of each model, a function called POTENTIAL is used. Intuitively, the POTENTIAL function measures the degree of difficulty for a given channel routing problem. The routing solution produced by our router has unambiguous layer assignment and is design-rule correct.

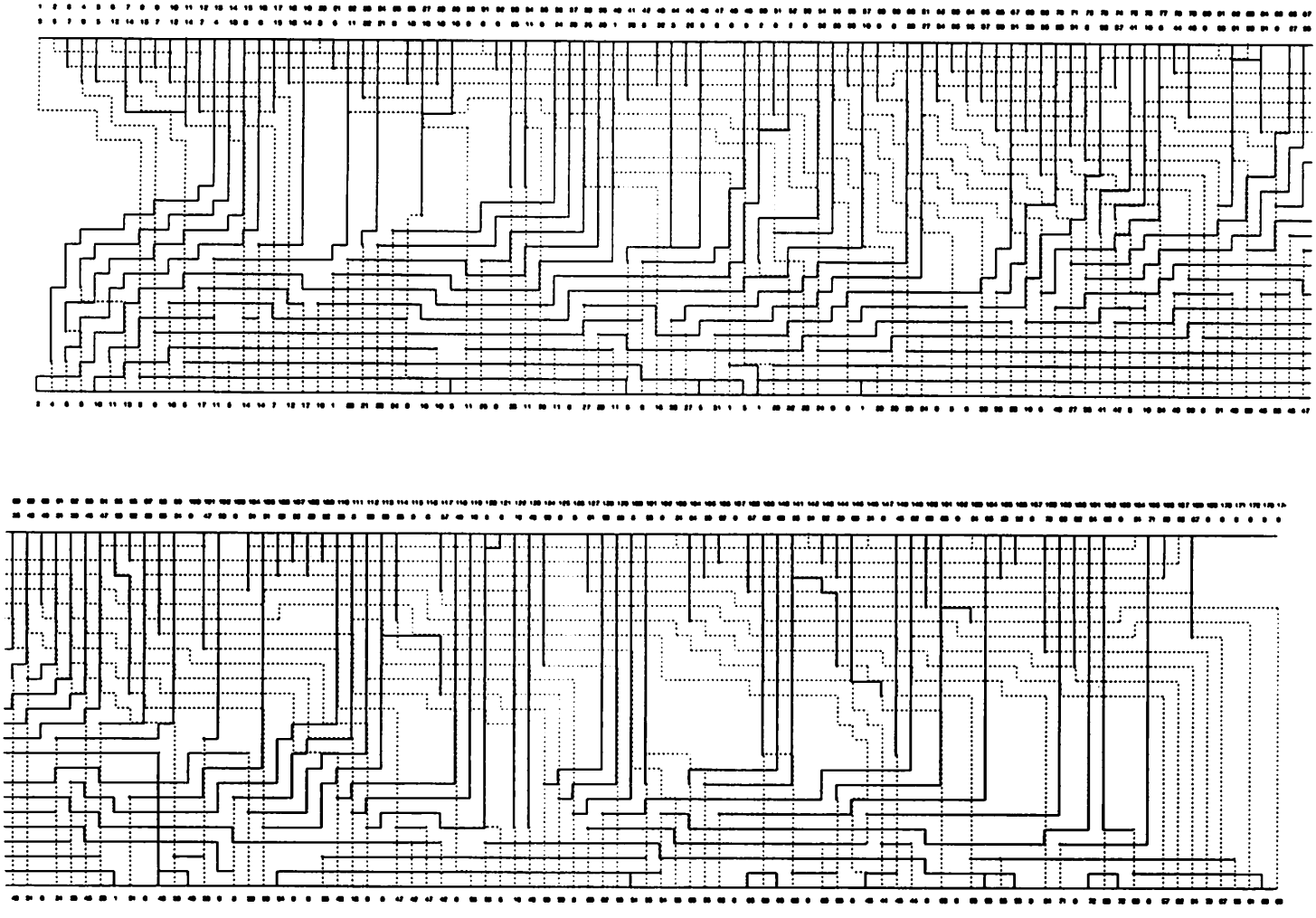


Figure 2.27: Difficult channel.

Finally, a straight-forward extension to handle multi-terminal nets and and side-nets is proposed. It has been demonstrated that the method can route channels with channel height less than the channel density. In addition, it requires many fewer vias than traditional methods do.

Table 1. Experimental results for 2–layer channel routing

channel	density	final height	CPU
Ex1	8	7	0.1
Ex2	7	8	0.1
Ex3	4	3	0.1
Ex4	18	22	0.1
Ex5	4	5	0.1
Ex6	8	5	0.1

Chapter 3

Three-Layer Channel Routing

Recent advance in the manufacturing technology has made multi-layer interconnect a reality. Therefore, it is important to design channel router to handle multi-layer regions. This development is also meaningful for other technologies such as multi-chip modules and hybrid packages. We shall focus on the 3-layer channel routing problem in this chapter.

Most of the 3-layer routing algorithms [22, 16, 8, 17, 21, 24] extend the layer per direction paradigm of Manhattan routing : the top and bottom layers are used for horizontal wires and the middle layer is used for vertical wires (HVH model). The vertical constraints are the major performance- degrading factor under such model. Other algorithms use knock-knee model [41] or overlapping wires [6, 5, 28]. For example, the knock-knee router [41] routes any channel in height equal to the density using 3 alyers. However, these methods may introduce excess vias.

In this chapter, we propose a new 3-layer channel routing algorithm using 60° wires. 60° wires not only provide more degrees of freedom for completing the interconnections but also shorten interconnect length. The algorithm is implemented in a router called *Overture*. *Overture* is designed to route two-terminal nets across a channel, and will also be extended to handle multi-terminal nets. The routing strategy is based on the river routing technique and is very different from the Manhattan method. In particular, vertical constraints no longer exist. Furthermore, the solution produced by *Overture* consists of a minimal set of net crossings which leads to a small

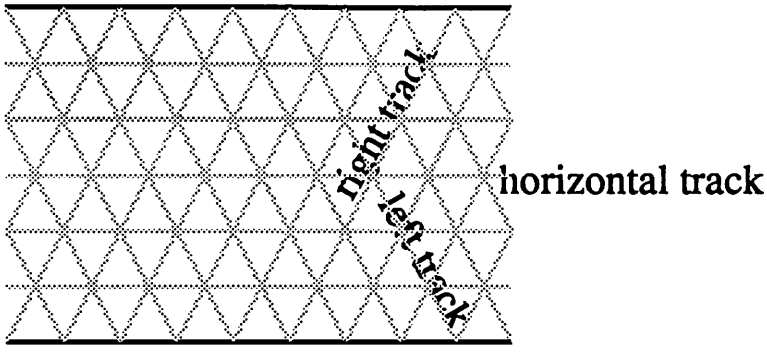


Figure 3.1: The Steiner grid.

number of vias. These characteristics contribute to the effectiveness of *Overture*.

The remainder of this chapter is organized as follows. Section 3.1 describes the Steiner (60°) wiring grid. The routing algorithm is given in Section 3.2. Section 3.3 contains performance analysis. Multi-terminal nets are introduced in Section 3.4. Routing results are presented in Section 3.5. Concluding remarks and future work are given in Section 3.6.

3.1 The Steiner Grid

The *Steiner grid*, first proposed by [13, 14], is composed of three types of grid lines: the *horizontal* tracks, the *right* (+60°) tracks and the *left* (-60°) tracks (Figure 3.1). In principle, three interconnect layers are required: one layer per direction [14, 38]. In contrast to the layer-per-direction Steiner model, we allow wires on different layers to overlap. The only restriction is that wires must lie on the Steiner grid, thereby imposing a uniform unit separation design rule. At each grid point, at most one via is present. Figure 3.2 shows three sample nets routed on the Steiner grid.

Let us now examine the design rules for the Steiner grid. If the terminals on TOP or BOT are separated by the contact-to-contact spacing, D, the distance between parallel tracks is $\sqrt{3}D/2$. If this is greater than or equal to the pitch (line-to-line spacing), the spacing requirement is satisfied. Otherwise, the the spacing between

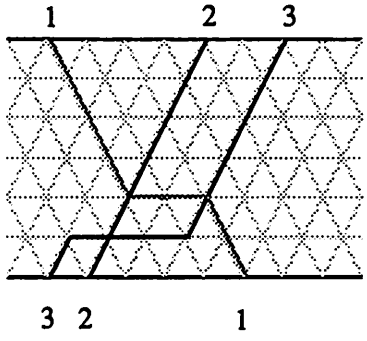


Figure 3.2: Sample nets are routed on the Steiner grid.

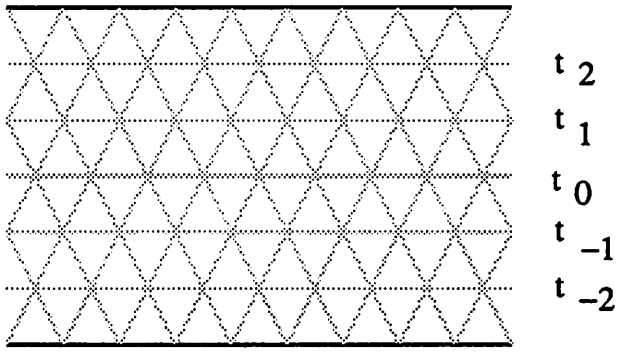


Figure 3.3: Label horizontal tracks.

terminals need to be enlarged. Since the contact-to-contact spacing is typically larger than the pitch, it is highly likely that the spacing rule is satisfied without magnifying the terminal spacing. We also note the spacing between grid points where vias are located, remains design-rule correct.

3.2 Routing Algorithm

In order to align terminals on TOP and BOT on vertical lines, the channel must have odd number of horizontal tracks. Let the horizontal tracks be labeled by $t_w, t_{w-1}, ..., t_0, ..., t_{-w}$ (Figure 3.3). Consider a D2TN channel routing problem specified by Definition 3. Let us define the following terms.

Definition 9 *Given a net i with terminals (i, q_i),*

- *i is a right net iff* $i \geq q_i$.
- *i is a left net iff* $i < q_i$.

We partition the nets into two disjoint subsets: R = { right nets } and L = { left nets } and route each set independently. The routing strategy is based on the river routing techniques. It is an extension of the 2-layer channel router presented in Chapter 3. A top level description of the algorithm is given below:

Top-Level Routing Algorithm

step 1) Nets are divided into two sets: R = { right nets }; L = { left nets }

step 2) Sort R in decreasing order of their top terminals. Route R.

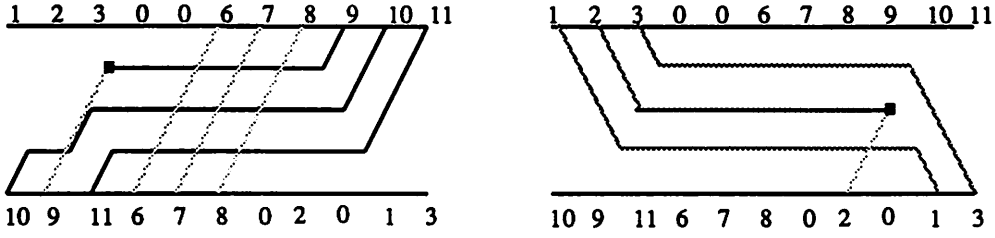
step 3) Sort L in increasing order of their top terminals. Route L.

The solution for the D2TN channel is constructed in a bottom- to-top net-by-net fashion. Let us demonstrate the algorithm on the example shown in Figure 3.4. The right nets are sorted and routed sequentially in a river routing fashion. The procedure begins by constructing a wiring path on the Steiner grid for the first right net. The path begins at its bottom terminal and ends at its top terminal (detail of path finding will be given later). The wiring path for the second right net simply follows the path of the first right net and ends at the top terminal. This process continues until all right nets are routed. Step 3 is identical to step 2, except that this time the left nets are routed.

The layer assignment scheme is as follows. The first interconnect layer is used for horizontal and right tracks; the third layer is used for horizontal and left tracks. If a channel has more right nets than left nets, the middle interconnect layer is used exclusively for the right (+60°) tracks. Otherwise, it is used for the left (-60°) tracks.

To find the wiring path for each individual net on the Steiner grid, we classify the nets according to the following definition.

Definition 10 *Given a channel of height* $h = 2(w+1)$ *and net* i *with terminals* (i, q_i),



(a) Route sorted right nets:

R = { 11, 10, 9, 8, 7, 6 }

(b) Route sorted left nets:

L = { 1, 2, 3 }

Figure 3.4: Example of 3-layer routing on the Steiner grid.

- *i is a right long net iff* $i - q_i > w + 1$
- *i is a right short net iff* $0 \leq i - q_i \leq w + 1$
- *i is a left long net iff* $q_i - i > w + 1$
- *i is a left short net iff* $0 \leq q_i - i \leq w + 1$

For channels with more right nets, the wiring paths for sample nets are shown in Figure 3.5 . A short net with $|i - q_i| = h + 1$ is also called a *direct* net. A direct net is routed with a single segment on the right or left track and requires no via. A short net bends on the horizontal track $i - q_i$. A long net is routed with one horizontal segment and two 60° segments. Direct nets are routed on a single layer and do not require vias. Short and long nets change layer only once and require one via per net. The layer assignment scheme in Figure 3.5 uses the second layer for right tracks. On the other hand, if there are more left nets, ie. the second layer is used for the left tracks, the wiring patterns and layer assignment scheme shown in Figure 3.6 are suitable for this class of channel routing problems.

3.3 Performance Analysis

In this section, we establish four results associated with *Overture*: (1) lower bound on the channel height, (2) upper bound on the channel height, (3) total number

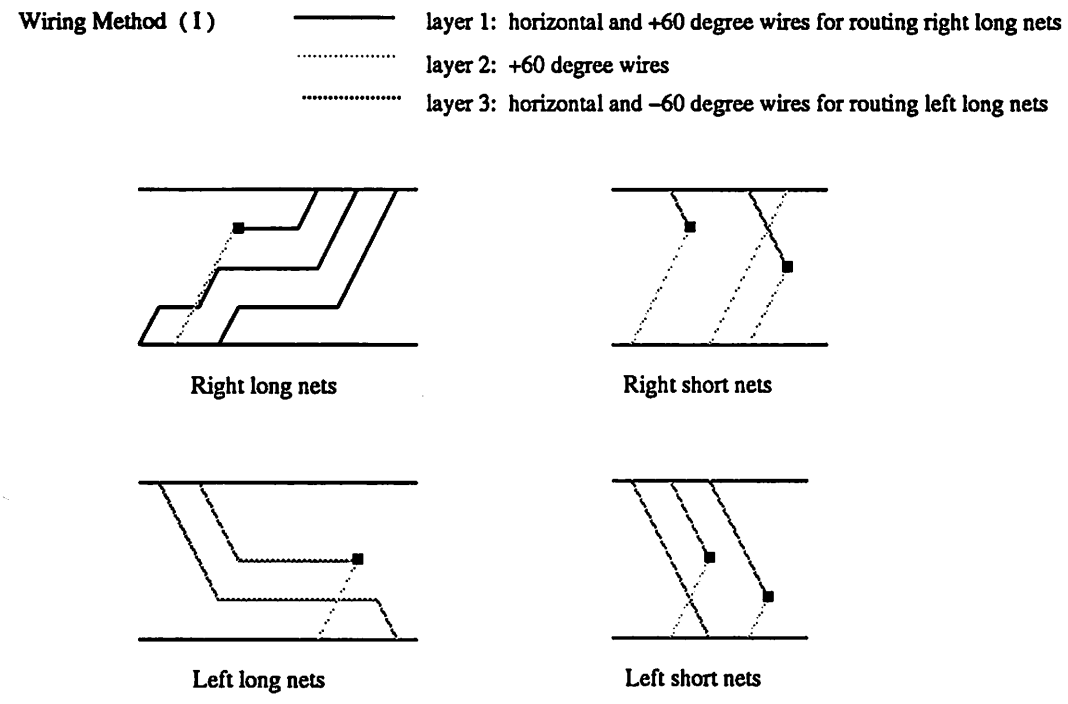


Figure 3.5: Wiring method (I) for the Steiner grid.

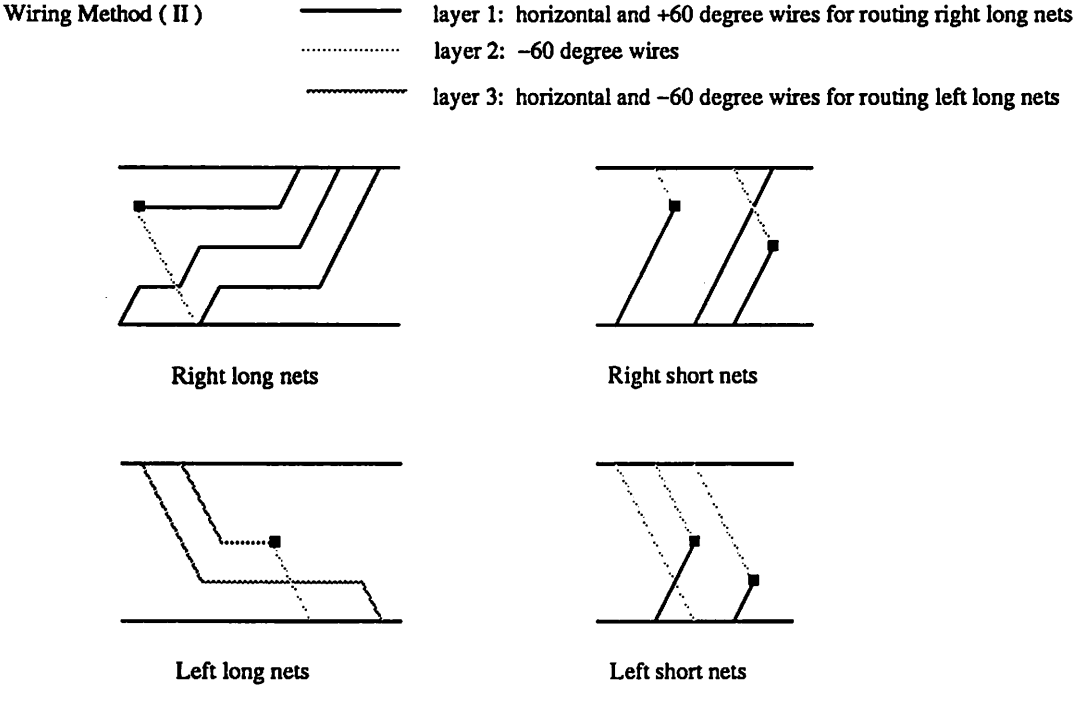


Figure 3.6: Wiring method (II) for the Steiner grid.

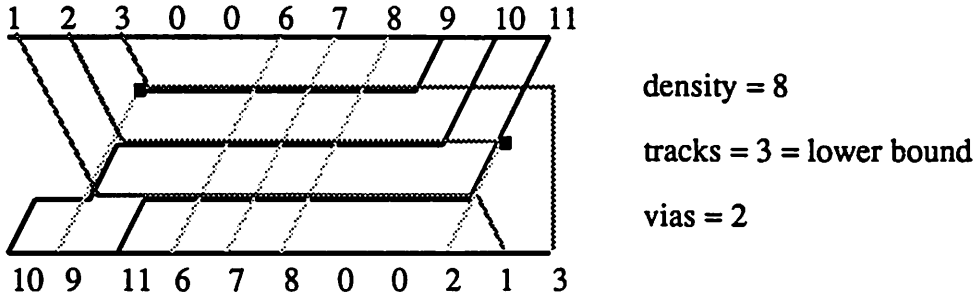


Figure 3.7: A channel is routed in the lower bound on the Steiner grid.

of vias, and (4) wire length. In particular, we prove the final channel height, h, satisfies $2d/5 \leq h \leq d$.

For the 3-layer HVH Manhattan model, $d/2$ is an obvious lower bound on the channel height, since as many as $d/2$ tracks are necessary on each horizontal layer to complete routing. When overlap between wires is allowed, Hambrusch [28] showed that the lower bound is also $d/2$ using 3 layers. We improve this lower bound to $2d/5$ for the 3-layer 60° model, where d is the Manhattan density. In particular, the example shown in Figure 3.7 is routed by *Overture* in a channel of height equal to the lower bound.

Theorem 8 *The lower bound on the channel height produced by Overture is* $2d/5$, *where* d *is the Manhattan density of the channel.*

Proof: Given a channel routing problem and a channel of height $h = 2w + 1$, consider a vertical line L as shown in Figure 3.8 . The horizontal tracks are labeled as $t_w, t_{w-1}, ..., t_0, ..., t_{-w}$. Suppose w is even. Then each of the w grid points located at the intersection of L and the odd numbered tracks allows three distinct wires to pass through. Similarly, each of the $(w + 1)$ grid points located at the intersection of L and the even numbered tracks allows two distinct wires to pass through. Hence the total number of distinct wires cut by L is $3w + 2(w + 1) = 5w + 2 \geq d$. The channel height, $h = 2w + 1$, is then $\geq (2d - 1)/5$. The proof for even w is similar. ■

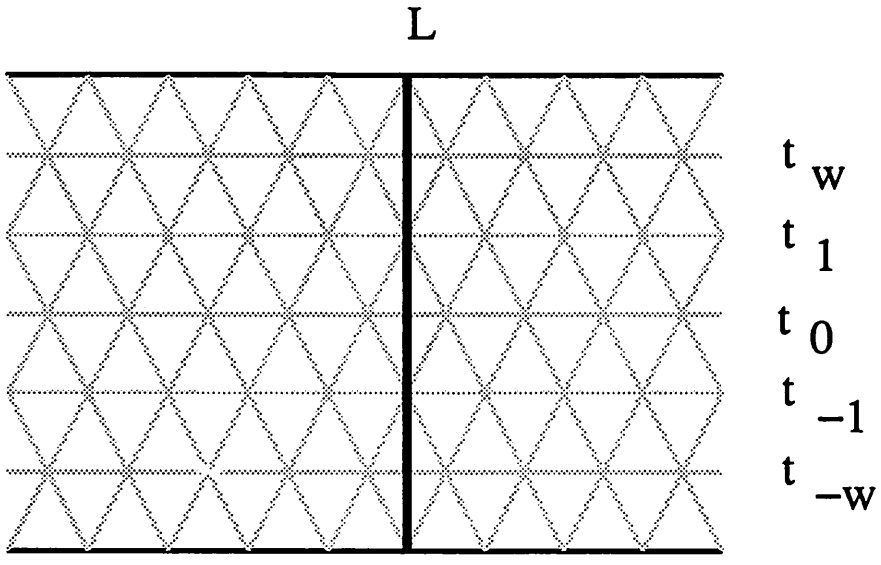


Figure 3.8: For proof of lower bound.

The final channel height, h, is $\max\{\ h(i) \mid 1 \leq i \leq 3\ \}$, where $h(i)$ denote the number of tracks required in the ith interconnect layer. The values of $h(i)$ depend on the density of wires on layer i. Recall the Manhattan density is the maximum of nets that are split by any vertical cut of the channel. We now introduce the notion of the *Steiner density* by considering the number of nets split by any $+60°$ or $-60°$ cut of the channel.

Definition 11 *Given a channel of height h and a set of nets, S,*

$d(S,h,+60°)$ = *the maximum number of nets in S split by any* $+60°$ *cut of the channel*

$d(S,h,-60°)$ = *the maximum number of nets in S split by any* $-60°$ *cut of the channel*

For a given h, let R_l, R_s, L_l, L_s denote the sets of *right long nets, right short nets, left long nets*, and *left short nets* respectively. Since layer 1 contains only right long nets and each such net intersects any $-60°$ cut line once, the number of tracks required on layer 1, $h(1)$, is the Steiner density, $d(R_l, h, -60°)$. The number of tracks required on layer 2 and 3, $h(2)$ and $h(3)$, can be calculated similarly. This is summarized in the following lemma.

Lemma 3 *Given a channel routing problem, assume wiring method I is used, the channel height produced by Overture is* $h = max\{\ h(i) \mid 1 \leq i \leq 3\ \}$, *where*

$h(1) = d(R_l, h, -60°)$

$h(2) = d(R_s \cup L_s, h, -60°)$

$h(3) = d(L_l \cup L_s \cup R_s, h, -60°)$

To find the value of channel height and to construct the final solution, we propose an iterative method. We start with $h = 2d/5$, the lower bound. Given h, we can construct the sets R_l, R_s, L_l, L_s and calculate $h(i)$ by Lemma 1. If $max\{h(i)\} > h$, we repeat the procedure with $h + 1$. Otherwise, the algorithm terminates and the final solution is found. This iteration procedure is summarized below.

Overture Routing Procedure

initialize $h = 2d/5$

repeat

build R_l, R_s, L_l, L_s

calculate $h(i)$ by Lemma 1

while ($h < max\{h(i) \mid 1 \leq i \leq 3\}$)

construct wiring paths (described in Section 4)

To find the upper bound on the channel height, h, and to compare it with the Manhattan density, we analyze the worst case performance of *Overture* in the following theorem.

Theorem 9 *Overture can route any D2TN channel routing problem in at most d tracks, where d is the Manhattan density of the channel.*

Proof: Suppose a D2TN channel routing problem with Manhattan density d is routed in a height h by *Overture* (Figure 3.9). Consider layer 1 and any $-60°$ cut line L. The number of tracks required on layer 1 is $d(R_l, h, -60°)$ (Lemma 1).

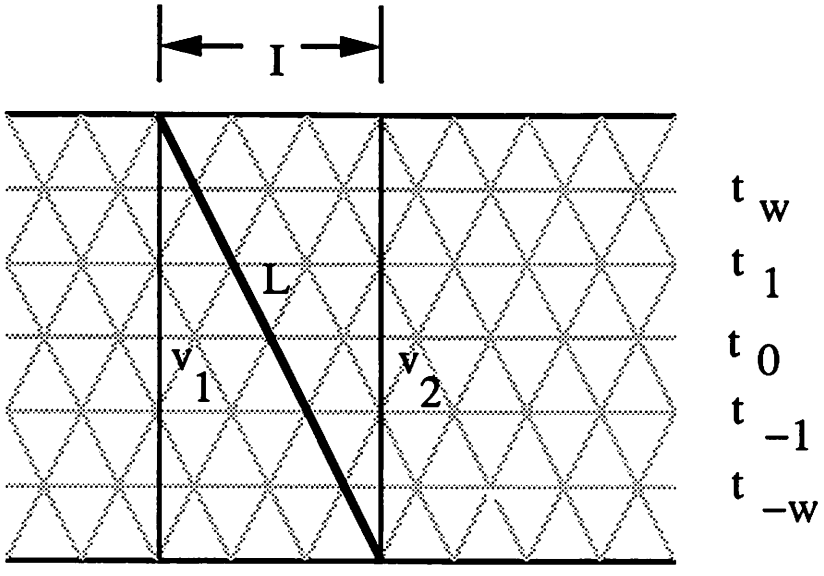


Figure 3.9: For the proof of upper bound.

i) number of grid points on $L = h$,

ii) each right net split by the vertical line v_1 is also split by L. At most $d/2$ right nets are split by v_1.

iii) If all of the $d/2$ right nets split by v_1 are also split by the vertical line v_2, then there can not be additional right nets in the interval I. Hence $h = d(R_l, h, -60°) = d/2$.

iv) If not all of the $d/2$ right nets split by v_1 span the interval I, i.e. say m right nets have top terminals in I, then at most m additional right long nets can be split by L. Since there are $h/2$ grid points on I, $m \leq h/2$. Hence in the worst case $d(R_l, h, -60°) = d/2 + m = d/2 + h/2 =$ number of grid points on $L = h$. Hence $h = d$.

The proof for layer 2 and layer 3 is similar. ■

This upper bound is rather pessimistic because it is the channel height of a pathological case. Experimental results show that *Overture* performs well on practical

channels.

The number of vias required by *Overture* is small because the solution consists of a minimal number of net crossings. In particular, at most one via is required per net for any D2TN channel routing problem. The number of vias can be calculated by the following Lemma.

Lemma 4 *Given a D2TN channel routing problem specified by Definition 3. Let R = right nets ; L = left nets. Given nets i, j and S, a set of nets, we define NOTORDER(i,j,S) as the number of indices* $p \in S$ *such that* $i < p \leq j$ *and* $q_p \leq q_i$ *, or such that* $j \leq p < i$ *and* $q_i \leq q_p$*. Then, for any non-direct right (left) net i, if NOTORDER(i,1,R) = 0 (NOTORDER(i,N,L) = 0), then net i has no via in the final solution produced by Overture. Otherwise, net i has one via. The total number of vias can be precisely calculated by evaluating NOTORDER(i,1,R) or NOTORDER(i,N,L) for all i.*

When wiring method I is used, ie. layer 2 is used for +60° tracks, the wiring paths for the right nets are monotonic in both the horizontal and vertical directions; while the paths for the left nets are monotonic in the vertical direction only. The wire length of each net can be calculated by the following lemma.

Lemma 5 *The exact wire length of net i, L(i), is calculated by:*

$L(i) = span(i) + h/2$ *if net i is a right net*

$L(i) = span(i) + h/2 + NOTORDER(i,N,L)$ *if net i is a left net*

where $span(i) = | i - b(i) |$, h = *final channel height*

3.4 Multi-terminal Nets

To extend the routing algorithm to handle multi-terminal nets, we partition each multi-terminal net into 2-terminal subnets and classify each subnet as a *right* net or a *left* net. Every 2-terminal subnets are classified as a right net or a left net by the same procedure presented in Section 2.5. The right subnets and left subnets are then

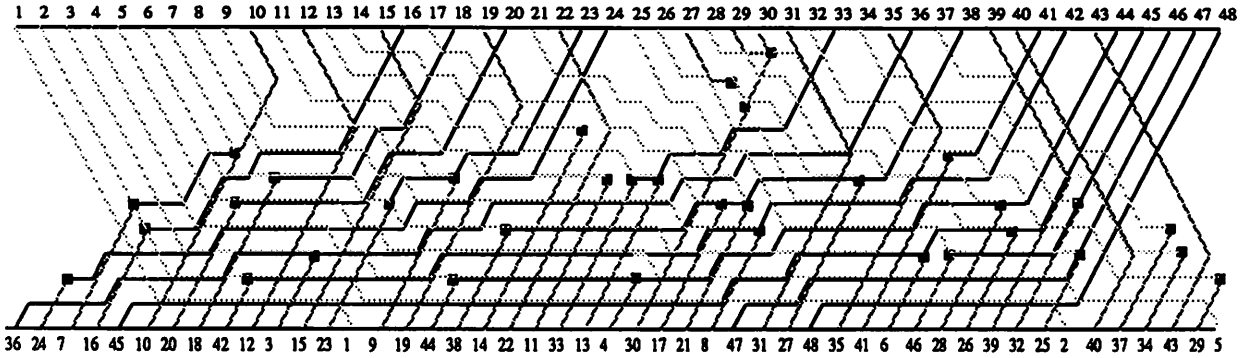


Figure 3.10: 3-Layer routing result of Ex4.

sorted independently by their terminal positions and routed in the rivering routing fashion. We observe that this method for routing multi-terminal nets is straightforward but not necessarily optimal. Future work should investigate better strategy for handling multi-terminal nets.

The analysis and theorems introduced for the 2-terminal nets in Section 3.3 can be generalized for the multi-terminal nets problem by replacing nets with subnets in the equations.

3.5 Results

Overture is implemented in the C language on a DEC3100 running Ultrix Worksystem V2.1. Figure 3.10 shows the routing result of a channel with 48 nets. We have attempted to run this example using other routers available[chameleon,trigger] but they produced substantially worse results. Figure 3.11 shows the difficult channel routed in 13 tracks and 181 vias. Table 1 lists the channels tested with 100% routing completion in all cases. Note that (Ch) in Table 1 denotes the router *Chameleon* developed by [22].

Table 1. Experimental results for 3–layer channel routing

Ex	density	Height		Vias		CPU	
		Ch	Ours	Ch	Ours	Ch	Ours
1	7	8	5	34	5	1.3	0.1
2	8	5	3	20	2	0.1	0.1
3	8	13	5	25	0	2.4	0.1
4	18	26	11	111	39	57.0	0.3
Diff	19	11	13	312	181	2.6	1.1

** Ch = Chameleon

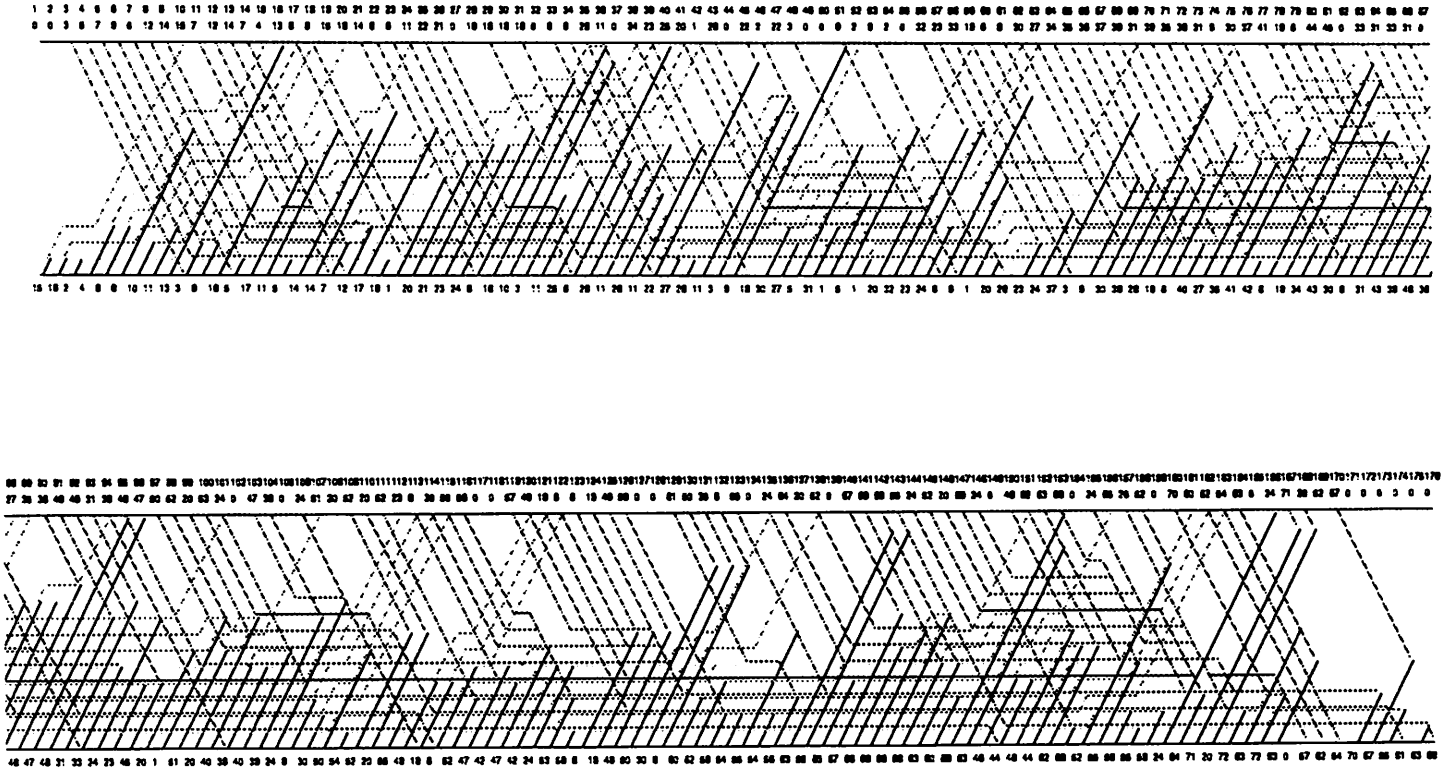


Figure 3.11: 3-Layer routing result of the difficult channel.

3.6 Conclusion

In this chapter, we have proposed a new 3-layer channel routing scheme by exploiting the additional degrees of freedom offered by the 60° Steiner grid. The routing strategy is based on the river routing technique. Wires on different layers are allowed to overlap. The routing solution produced by *Overture* has unambiguous layer assignment and is design-rule correct. We prove that the channel height, h, produced by *Overture* satisfies $2d/5 \leq h \leq d$, where d is the Manhattan density. Compared to the Manhattan routers, *Overture* has the following advantages: (1) It does not need to deal with vertical constraints, (2) The wire length is expected to be shorter due to the use of 60° wires. (3) A small number of vias is required because the routing solution contains a minimal number of net crossings. In particular, at most one via per net is required for routing two-terminal nets across a channel. (4) Due to the simplicity of the river routing technique, it is very fast. The method has been extended to handle multi-terminal nets.

Chapter 4

Pad Placement and Ring Routing

This chapter presents an optimum scheme for interconnecting the chip core and the I/O pads in the final stage of physical design. The pad placement routine, based on linear assignment, determines the dimension of the pad ring and selects the optimum position for each pad with the objective of minimizing the chip area and the total wire length. The router is based on a channel routing algorithm which incorporates additional features to address the special needs of the ring configuration. It attempts to achieve 100% routing completion in a rectangular ring-shaped area with two interconnect layers. The complete package has been implemented as part of the BEAR Layout System for custom chip design.

4.1 Introduction

In the final stage of the layout design, the central core module containing the assembled logic is to be connected to a set of I/O pads. The problem is to find an embedding of the I/O pads in a ring enclosing the core module and to define precise conductor paths necessary to interconnect the terminals as specified by the netlist. A 100% routing completion is required and, in addition, the objective is to minimize the chip area and the total wire length.

In this paper, we assume that I/O pads are to form a rectangular ring, called the *outer core*, encircling the central core module. The interconnection is to take place

in the doughnut-shaped area between the inner core module and the outer core using two interconnect layers. We partition the process into the two steps: pad placement and ring routing.

The two main tasks of the placement step are to determine the dimension of the outer core and to select the best possible position on the outer core for each pad. The question to be answered by placement is : How should the pads be optimally assigned to positions on the outer core. The height and width of the outer core may be flexible. We propose an analytic solution for determining the dimension of the outer core and a pad placement scheme based on linear assignment or network flow optimization.

After placement, the router connects the terminals on the central core module and the pads according to the net specification and design rules. Several routing schemes have been proposed with varying degrees of success [2] [39] [23]. Baker and Pinter [2] required the terminals on the pads and core module to have the same cyclic order so that wires could be river routed in one layer. Smith et al. [23] proposed the concentric and radial wiring model without presenting a complete solution. McGehee [39] used the vertical constraint graph to route the ring. The router presented in this paper is based on an optimal channel routing algorithm which incorporates additional features addressing the special needs of the ring configuration.

Throughout the the paper we will use the following terminology (Figure 4.1):

terminal	a point on the boundary of the core module or a point on one side of the pad where interconnection must be made
net	a set of terminals to be connected
core module	the central block containing the assembled logic of the circuit
inner core	the smallest rectangle enclosing the core module
pad	rectangular I/O blocks
pad ring	a ring of pads
outer core	the rectangle along which the pad ring is formed

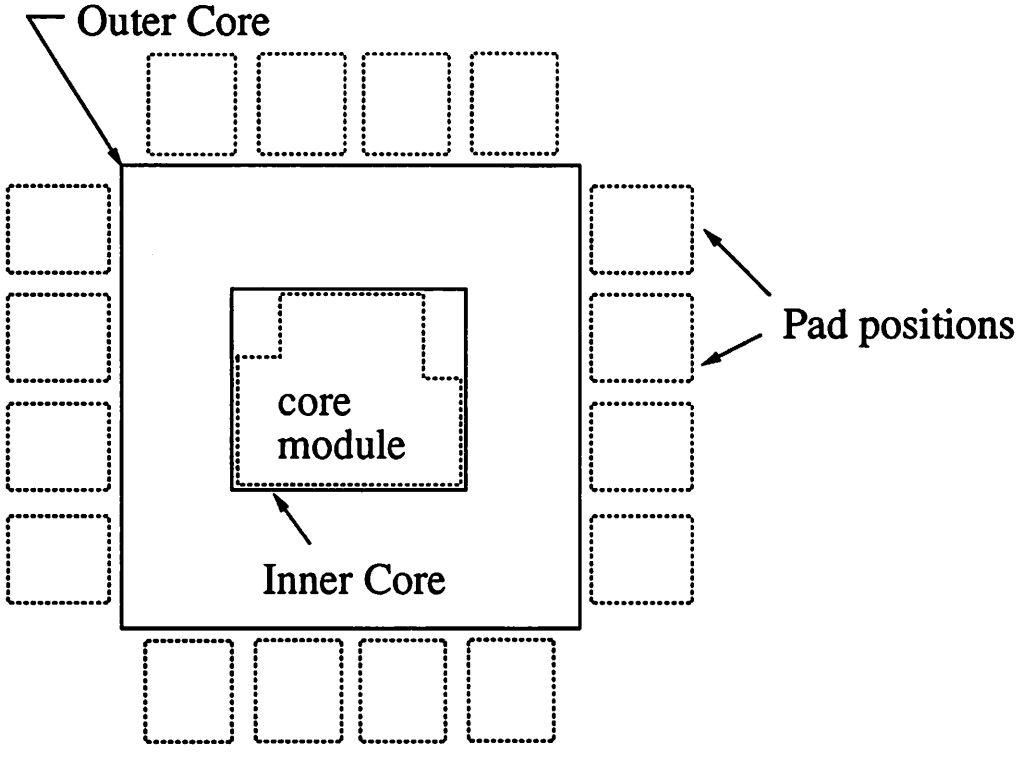


Figure 4.1: Terminology for pad ring.

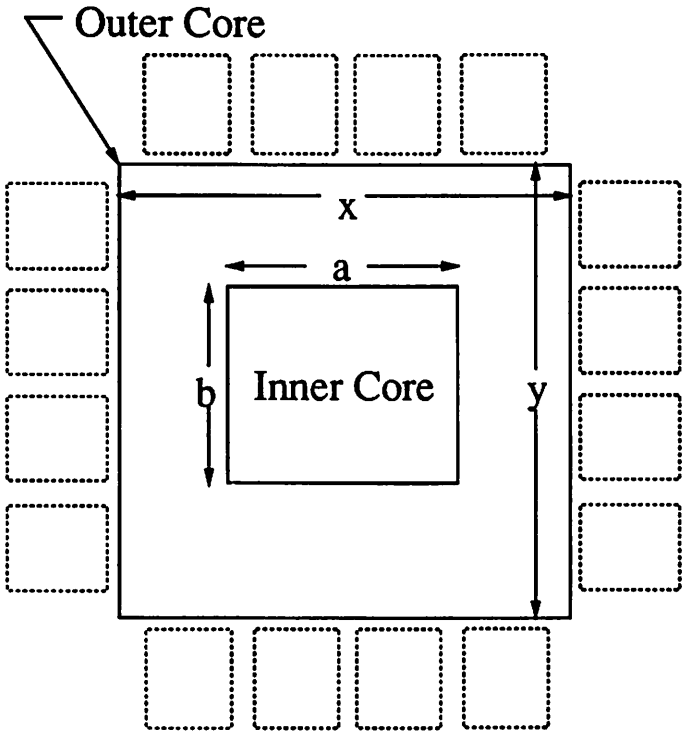


Figure 4.2: Dimensions of outer and inner cores.

4.2 Pad Placement

The goal of the placement step is to assign the best possible position on the outer core to each pad and to determine the dimensions of the outer core. Section (4.2.1) describes an analytic approach to finding the optimum dimension of the outer core. Section (4.2.2) describes the assignment of pads along the outer core.

4.2.1 Optimum Dimension of Outer Core

The dimension of the outer core is represented by two variables x and y, where x (y) represents the number of pad spaces available on one side of the outer core in the horizontal (vertical) direction (Fig. 4.2). The first step in the pad placement routine is to determine optimum values for x and y with the objective of minimizing the chip area.

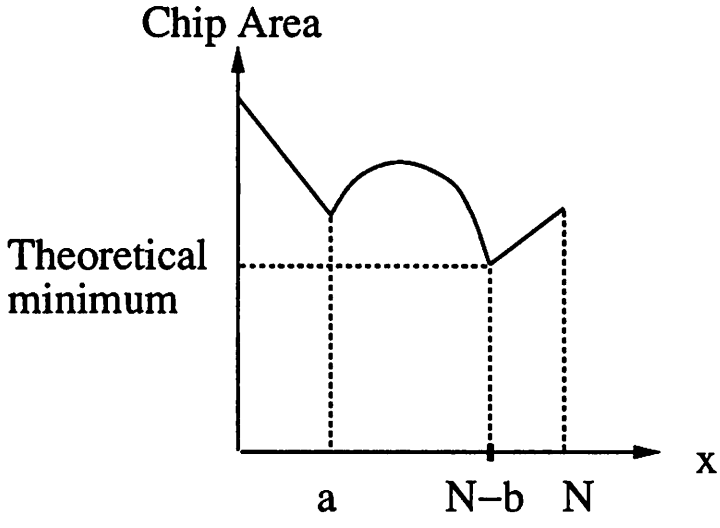


Figure 4.3: Plot of chip area as a function of x.

Suppose we are given $2N$ I/O pads of equal width and an inner core with dimensions a and b, where a (b) represents the number of pad spaces on one side of the inner core in the horizontal (vertical) direction (Figure 4.2). Without loss of generality, assume $a > b$ else rotate the central core by 90 degrees.

A chip is said to be *core bound* if $a + b >= N$, that is, there are more pad spaces than pads. Otherwise, it is said to be *pad bound.* In a core bound chip, the chip area is dominated by the inner core , and minimizing area requires minimizing the routing area in the ring. On the other hand, for a pad bound chip, we have $N = x + y$, and the area of the chip can be represented as

$$Chip\ Area = xy = \begin{cases} x(N-x) & \text{if } a <= x <= N-b \\ a(N-x) & \text{if } x <= a \\ b(N-y) & \text{if } x >= N-b \end{cases}$$

The corresponding plot of the area against x is shown in Figure 4.3.

The theoretical minimum is obtained when $x = (N - b)$ and $y = N - x = b$. However, this corresponds to the configuration shown in Figure 4.4, where there is no routing space either on top or on bottom. This is clearly unrealistic. Hence the next best solution according to the plot above is to increment or to decrement x by one.

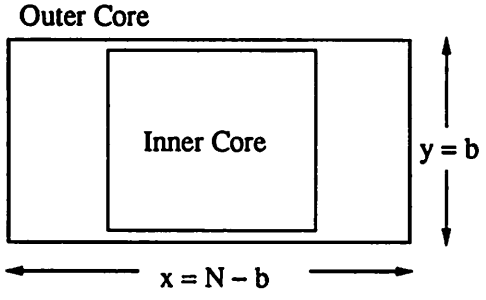


Figure 4.4: Theoretical minimum chip configuration.

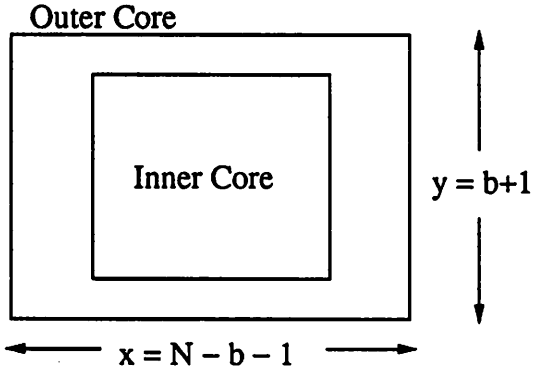


Figure 4.5: Final Chip configuration.

Since to increment x by one does not increase the routing space, the final solution is to decrement x, that is, $x = (N - b - 1)$ pads are placed on top/bottom sides and $y = (b + 1)$ pads are placed on left/right sides (Figure 4.5). We have derived an analytic solution of the outer core dimension problem for a pad bound chip in which the minimum chip area is achieved.

Additional flexibility has also been incorporated into this procedure. For instance, the user may specify the dimension of the outer core to be one of the following three cases:

1. fixed
2. within specified aspect ratios
3. automatically determined by the placement routine

When case 2 or 3 is chosen, the outer core dimension is calculated. If ratio of the resulting x and y exceeds the specified aspect ratio, simply move along the curve in the plot until the desired aspect ratio is obtained.

4.2.2 Assignment of Pads to Positions Along The Outer Core

This section addresses the question of how to assign pads to positions along the outer core with the objective of minimizing the total wire length. Our aim is to simultaneously process the nets and to find an embedding (in terms of pad spaces along the outer core) for each pad.

Assignment of a pad to a position on the outer core should be done in such a way that the increase of wire length of that net is as small as possible. Using this criterion, many pads may compete for the same position. To find a good assignment of pads to positions, we use a linear assignment approach. We build a complete bipartite graph: G(V,U,E), with two sets of nodes. One set, V = P(i), i = 1,...N , represents the pads; the other set, U = 1,2,...N , represents the positions on the outer core. N is the total number of pads to be placed. Edge (i, j) connects pad node

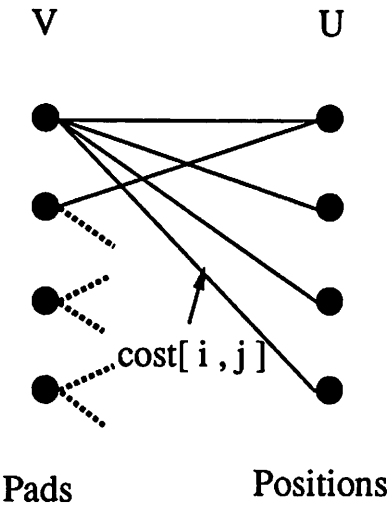


Figure 4.6: Bipartite graph $G(V, U, E)$ for pad placement.

i to position node j. Edge (i, j) is labeled with cost c[i,j] indicating the cost (to be discussed in the next section) of assigning pad i to position j (Figure 4.6). Standard methods [9, 20, 50, 27] may be used to solve this problem.

4.2.3 Cost Function

Choosing an appropriate cost function is crucial to the linear assignment approach. Values of the cost function are stored in an NxN matrix which specifies the cost of assigning pad i (i = *1,...,N*) to position j (j=*1,...,N*). Let P(i) denote the ith pad and Net(i) denote the net that has a terminal on pad P(i). If a net has more than one pads, say P(i), P(j) and P(k), then Net(i) = Net(j) = Net(k). In our implementation, the cost c[i,j] reflects the increase in the wire length of net Net(i) when P(i) is assigned to position j. For each pad P(i) to be placed, we compute and store the shortest path for the net Net(i). This is achieved by drawing a path completely around the inner core and deleting the section with the larger rectilinear distance between two adjacent terminals of Net(i) (Figure 4.7). Now unroll the ring and let the interval (u,v) be the projection of the shorter path on the outer core. To all positions inside this interval, we assign cost zero. To positions outside this range,

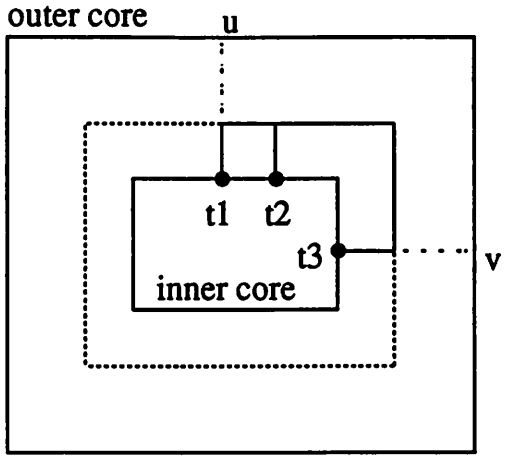


Figure 4.7: Shortest path of a net.

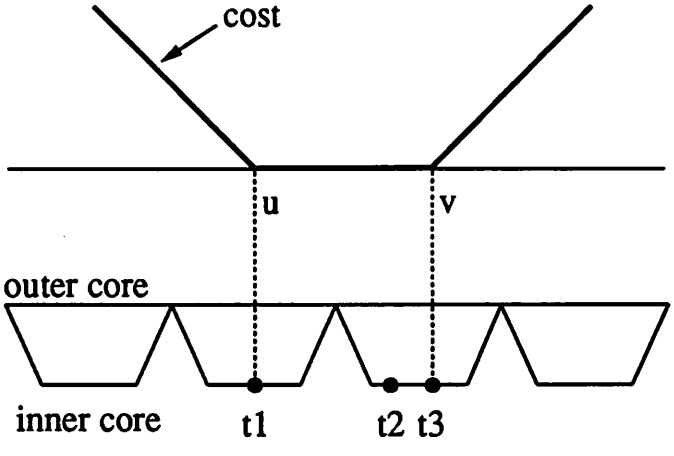


Figure 4.8: Cost funtion.

cost values increase linearly with the distance from the interval (u,v) (Figure 4.8).

The slope of this cost function outside (u,v) may not be the same for all nets. The cost function of more critical nets have higher slopes, giving them an advantage in competing with less critical nets.

For simplicity, we have presented a cost function which allows only a single terminal per pad. It can be extended to handle multi-terminal pads. Suppose a pad has k terminals, *t1, t2, ..., tk.* Each terminal *ti* is a part of net Net(i). The cost

function for this pad then contains k additive terms, with the ith term describing the increase of wire length for net Net(i).

The pad assignment routine is flexible in many ways. For instance, it allows users to preplace some of the pads' positions. When a pad is preplaced, the nodes corresponding to the preplaced pad and the associated incident edges are deleted from the bipartite graph. In another example, the user may specify, for each pad P(i), a subset of positions, *j1, j2,..., jk* as undesirable. A very large cost can be assigned to the edges (i, j1),... (i, jk) to prevent placing this pad at these positions.

4.3 Ring Routing

Upon completion of pad placement, the ring router interconnects terminals on the inner core to the pads according to the netlist. Section (4.3.1) describes the main difficulties associated with the ring routing problem. A solution based on an optimum channel routing scheme is proposed in section (4.3.2). Further refinement of the routing result is outlined in section (4.3.3), followed by a discussion of future directions.

4.3.1 Routing Problem Description

The central core module is a rectilinear polygon defined by 4 boundaries: *top, bottom, left* and *right*. The *top* and *bottom* boundaries are monotonic in the x-direction; the *left* and *right* boundaries are monotonic in the y-direction. The terminals of the module are located on the horizontal segments of the *top* and *bottom* boundaries and on the vertical segments of the *left* and *right* boundaries.

The ring routing problem is analogous to the channel routing problem. The ring routing problem is a channel bent into a ring so that the horizontal tracks become concentric and the vertical columns become radial. One interconnect layer is used for concentric wires. The other layer is used for radial wires [39, 23]. The radial columns are horizontal and vertical lines drawn through the terminals with the exception of the four corners of the ring (Figure 4.9). The thickness of this ring must be determined

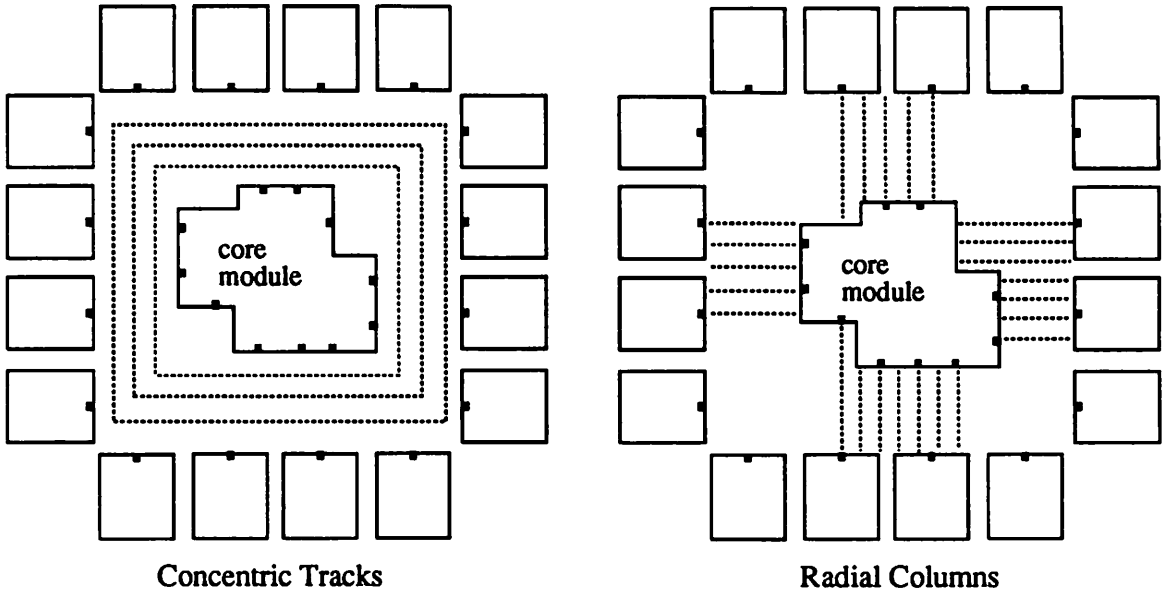


Figure 4.9: Concentric tracks and radial column.

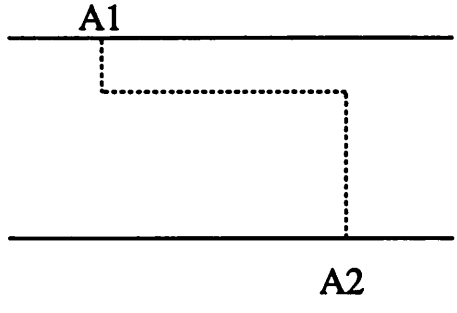


Figure 4.10: There is only one way to route net A in a channel.

by the router: that is we are allowed to add or delete concentric tracks to the ring by moving the four sides of the pad ring. The objective is to minimize the number of tracks in the ring.

One difference between the channel and the ring is that the channel has two open ends whereas the ring is a closed loop. In a straight channel, there is only one direction to route a net (Figure 4.10). But a two-terminal net in a ring can be routed in two ways (Figure 4.11). Under the restrictive routing model proposed by Burstein [11], where the number of concentric tracks for each net is limited to one, an n-terminal net can be routed in n ways.

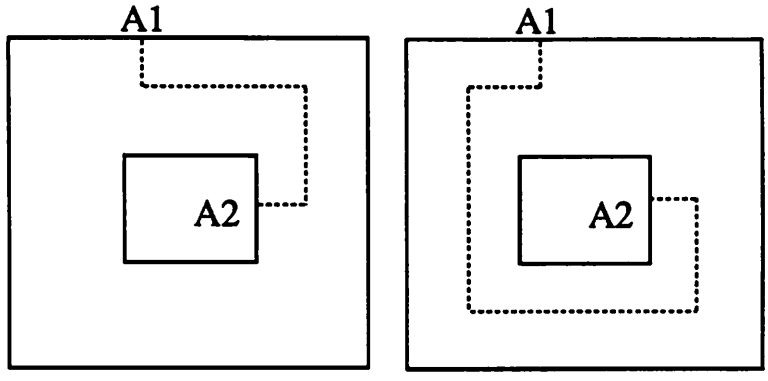


Figure 4.11: There are two ways to route net A in a ring.

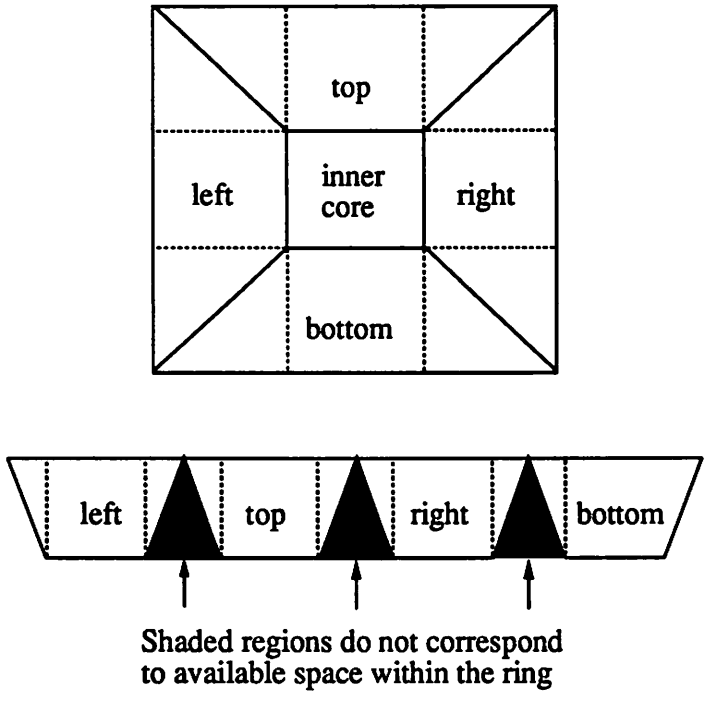


Figure 4.12: Channel produced by unrolling the ring.

Another difference is that the ring has corners. The four corners of the ring impose a key restriction when adapting channel routing algorithms to a ring router. If we cut the ring along a line drawn between the top-left corner of the pad frame and the top-left corner of the module and unroll the ring, the channel we obtain is shown in Figure 4.12 .

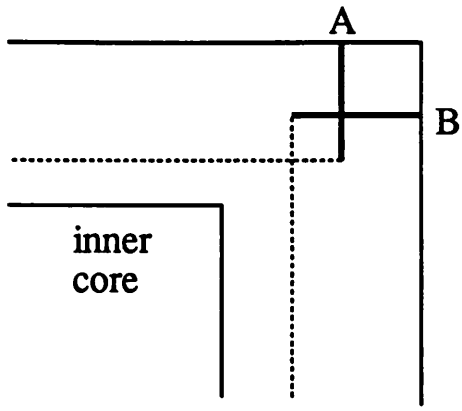


Figure 4.13: Corner net A and corner net B conflict.

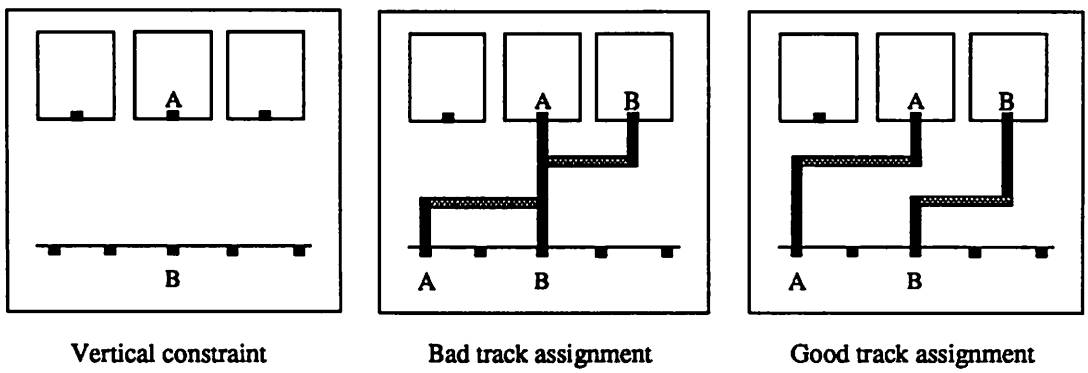


Figure 4.14: Vertical constraints between net A and net B.

The shaded regions represent empty space: they do not correspond to available space within the ring. So if we were to route a corner pad terminal to a track by a vertical column that intersected such non-existing regions, the net could conflict with other nets passing through the corner (Figure 4.13).

The third problem associated with the ring relates to vertical constraints, which are common in channel routing. A vertical constraint arises when a pad terminal and a module terminal have the same abscissa (or the same ordinate) and are to be connected to distinct nets (Figure 4.14). The presence of a vertical constraint indicates that the concentric track of the net 'A' must be positioned above the concentric track of the net 'B'. Vertical constraints lead to significant algorithmic complication and optimality degradation.

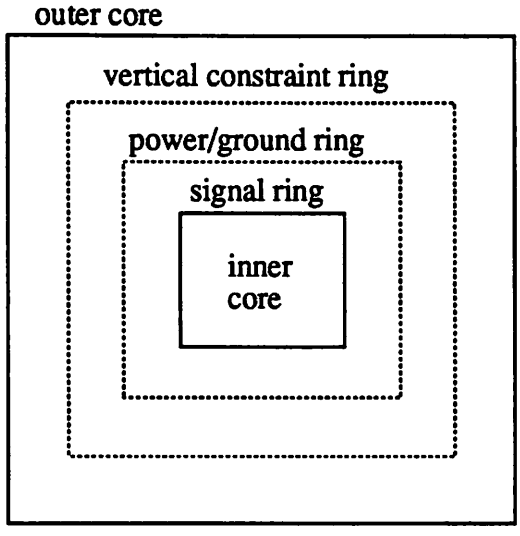


Figure 4.15: Routing area is divided into subrings.

In summary, in order to produce an optimal solution, the ring router must effectively deal with the problems of the routing direction, the corner pads and the vertical constraints.

4.3.2 The Routing Solution

To minimize the interaction between signal nets and P/G nets, the ring is divided into three subrings(Figure 4.15). The inner and outer subrings are used for routing signal nets while the middle subring is used exclusively for P/G interconnections.

The problem of restrictive routing in a ring is to determine a concentric track number for each net. Restrictive routing has the advantage of both being simple and of using a minimal number of vias. When there are no vertical constraints, the restrictive routing problem can be easily and optimally solved by using the left edge algorithm [30]: the number of tracks equals the density. However, when there are vertical constraints, the problem is NP-complete [32] and only a suboptimal solution may be obtained.

One special feature of the ring problem allows us to eliminate all vertical

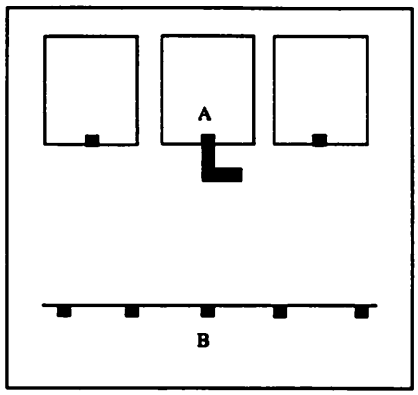


Figure 4.16: Removal of vertical constraints.

constraints so that the left edge algorithm can be applied. This requirement is that the pad terminals must be spaced far enough apart to guarantee the existence of an empty column close to the constraint pad terminal. Usually, at the expense of allocating one track in the outer subring, all of the vertical constraints can be resolved (Figure 4.16). Since the P/G tracks are always outside the signal tracks, this subring is also used for resolving vertical constraints between a signal pad terminal and a P/G core terminal. If one track is not enough to eliminate all vertical constraints, more tracks in the outer subring can be allocated. It is very unlikely that this will be necessary: we have found that one outer track is sufficient for all of the industrial chips we have tested.

Once the vertical constraints are removed, we are ready to apply the the left edge algorithm. First, the shortest rectilinear path for each net is determined. This is achieved by drawing a path completely around the ring and deleting the section with the greatest rectilinear distance between two adjacent terminals of the net (Figure 4.7).

In order to minimize the ring thickness, the routing space between the inner core and the boundary of the core module (Figure 4.17) should be put to good use since it does not contribute to routing tracks. We shall describe a pre-processing procedure which embeds paths in such space prior to the application of left edge algorithm.

Figure 4.18 shows extra routing space in the indentation I(a, b). Such an

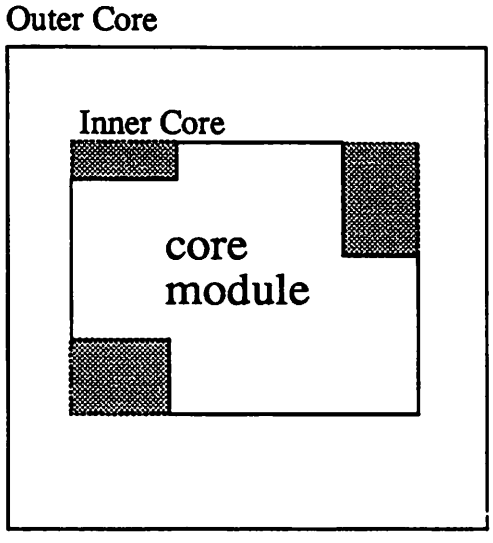


Figure 4.17: Hidden routing space between the inner core and core module boundary.

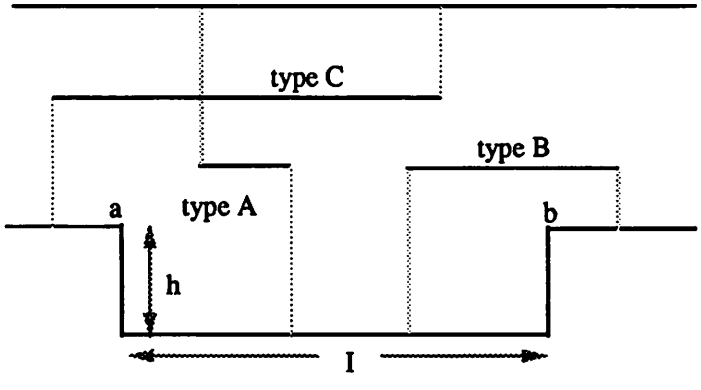


Figure 4.18: Net classification with respect to the indentation.

indentation is critical if it contains the region of maximum density. The height 'h' limits the number of paths that can be embedded in indentation *I*. The goal is to select an appropriate set of paths for embedding in critical indentation.

A path *P* can be classified as type *A*, *B*, or *C* with respect to *I* depending on how it intersect *I*. A path is type *A* if *P is contained in I* (Figure 4.18). A path is type *B* if it has a terminal in *I*. A path is type *C* if *I is contained in P*. Figure 4.19 also shows how each type of path *P* can be embedded in *I*. Since embedding of type *A* (*C*) paths introduces the least (most) number of vias, type *A, B, C* paths are

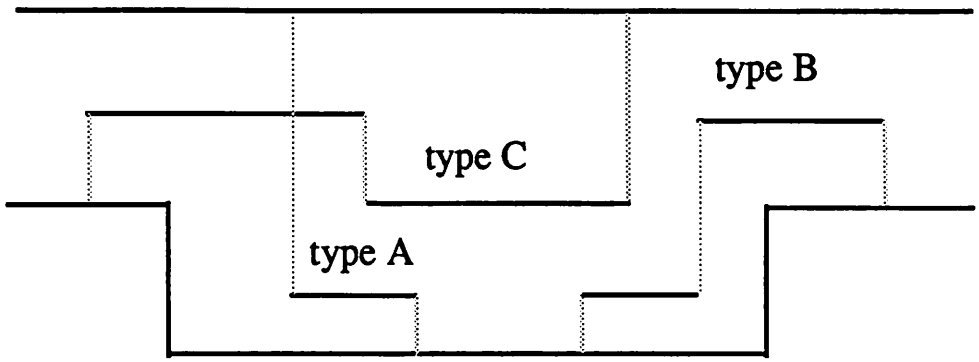


Figure 4.19: Embedding of paths in the indentation.

selected in that order by the preprocessor.

After path selection and pre-processing, we are ready to apply the left edge algorithm. Tracks are assigned from the inside out to keep the wire length as short as possible. Pad terminals are connected to concentric tracks through radial columns with the exception of corner pads.

As described in section 4.3.1, a straightforward radial connection from a corner pad may cause conflicts with other nets. The solution proposed by [39] is to always assign a track outward from the point of intersection for such a corner pad. But in doing so, [39] requires that all tracks be assigned from the outside in, so that concentric wires run along the outer core. For the net shown in Figure 4.20, this strategy results in an unnecessary detour of the wiring compared to that of Figure 4.21.

To overcome this problem, we propose a corner routing scheme resembling river routing. We divide each of the four corners of the ring into two triangular halves along the diagonal line drawn through the pad ring corner and the corresponding module corner (Figure 4.12). Every triangular corner region is routed in the same fashion.

For the example in Figure 4.22 , a wiring path for the first corner pad terminal, A2 (of net A), is constructed by running along the corner boundary until the assigned track is reached. The path for the next corner pad terminal, B2 (of net B), simply 'licks' along the path for A until the track for net B is reached. This

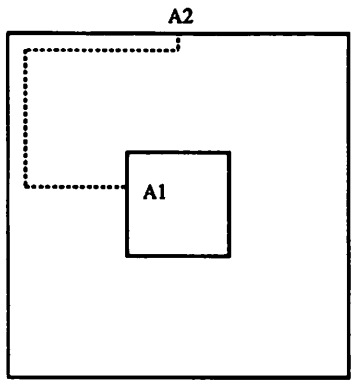


Figure 4.20: Routing along outer core introduces detour.

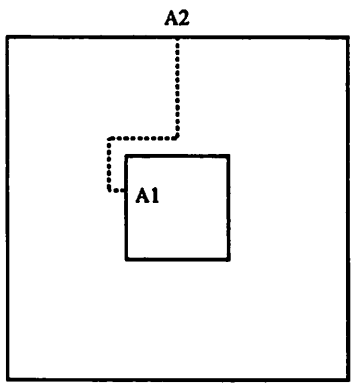


Figure 4.21: Routing along inner core produces shorter path.

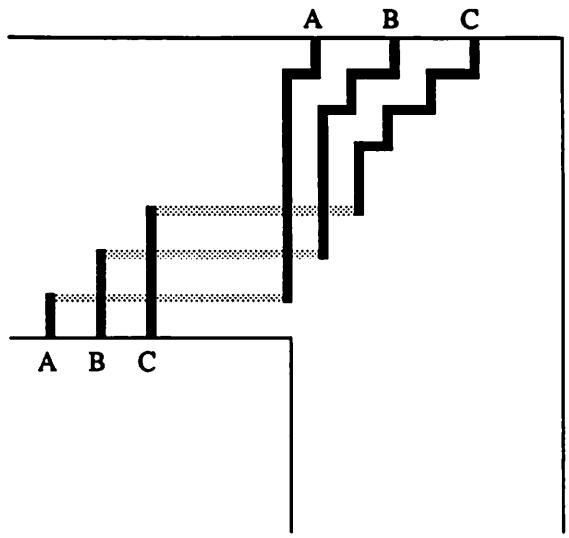


Figure 4.22: Strategy to route corner pads.

process continues until all corner pads in this triangular region are routed. The paths are placed on the same layer as the radial columns so that they do not conflict with the concentric tracks. A similar procedure is repeated for all other corners.

Signal nets can usually be routed within one track of the mathematical lower bound. Compared to the router in [39], the method presented here takes advantage of the ring property and achieves a significant reduction in algorithmic complexity and wire length.

Routing of power and ground nets is done in the same way as signal routing. The ring router is capable of handling variable wire widths for power and ground nets.

4.3.3 Post Processing

The purpose of post processing is to further refine routing results. Its two main tasks are to eliminate unnecessary jogs and to shrink or expand each side of the outer core independently.

Recall that prior to the left edge algorithm, a jog in the outer-most track was introduced whenever a vertical constraint arose between a pad terminal, say A,

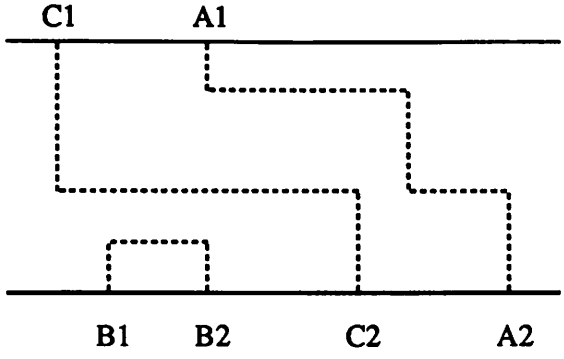


Figure 4.23: Introduction of jog at terminal $A1$ to resolve vertical constraint between terminals $A1$ and $B2$.

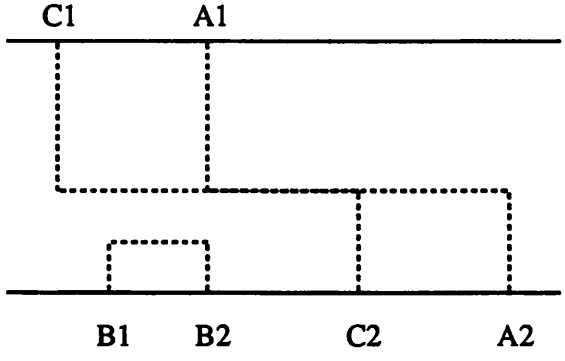


Figure 4.24: Since the deletion of the job causes conflict between nets And C, it should not be deleted.

and a core terminal, say B, to resolve the vertical constraint. After the left edge routine is applied, each net is assigned a track (Figure 4.23). If the track assigned to terminal A is higher than that of terminal B, then it may be possible to delete the jog so as to free the outer track and possibly reduce the ring thickness. Deletion of jogs, however, must be guided so that it does not cause conflicts between nets (Figure 4.24)

Since each side of the ring usually requires different amounts of routing space, we shrink or expand each side independently. For example, maximum density occurs only on one or two sides of the ring so that the ring thickness on the other sides may be further reduced. Likewise, P/G nets or vertical constraints may be present

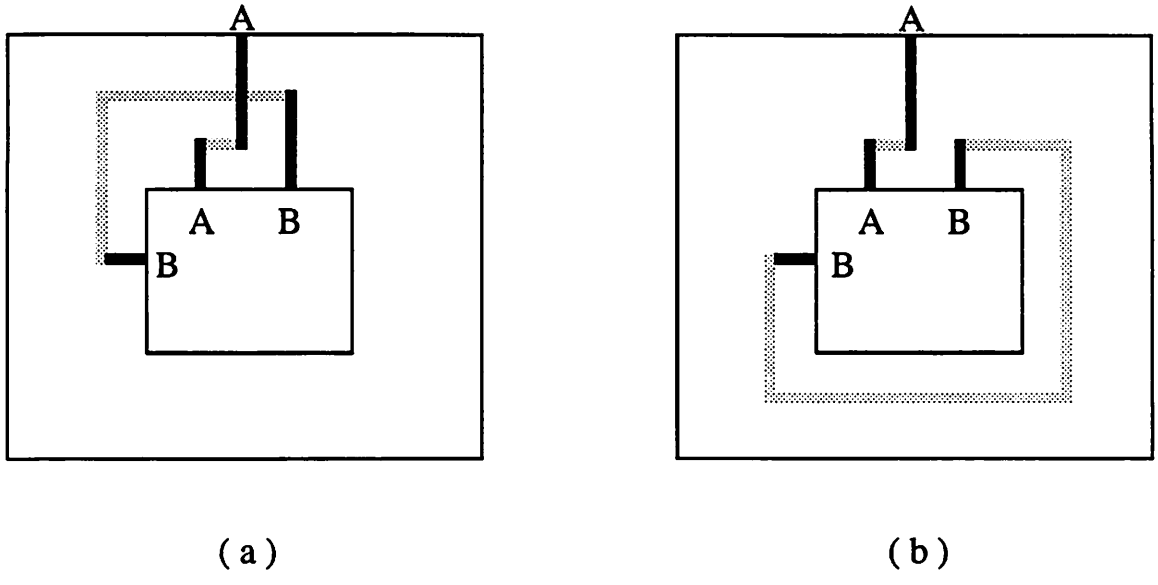


Figure 4.25: Shortest paths vs. preferred paths.

on one or two sides of the ring so that the unused tracks of the other sides may be shrunk to zero.

4.3.4 Shortcomings of the Routing Approach

Exact optimization of the routing space would involve selecting the preferred path, not necessarily the shortest one, to route a net so that the ring thickness on every side is minimized. An example is shown in Figure 4.25 .

Solutions for the two-terminal nets core bound problem (no pad is allowed in the corners) which take into consideration the preferred path are proposed by [33, 48]. For multi-terminal nets or pad bound chips, it remains an open problem.

4.4 Computation Complexity

The placement step based on linear assignment takes $O(N^2)$ time and space where N is the number of pads. The routing step based on the left edge algorithm takes $O(nlogn)$ time and space including sorting on n terminals.

4.5 Results

The ring router is implemented in the C language as part of BEAR [25], a building block layout system. The following tables list the chips tested and the router's performance. The results are obtained on a DEC VAX-11/780 running UNIX BSD 4.3 with 100% routing completion in all cases.

The numbers in parentheses indicate total wire length and chip area improvement with pad placement. These results are also compared with the approach of dividing the ring into 4 channels and using traditional channel router in [25]. In particular, the comparison of the total wire length is shown in Table 3, in which the ring router approach performs favorably.

Table 1. Ring Routing Results Without Pad Placement

Chips	Pads	CPU (sec)	Total Wire Length	Chip Area
Hughes	36	3.6	21336	524342
PrimBBL2	40	7.6	60787	334705
Hp	46	6.2	99167	173232
RTASIC	100	6.4	418228	506000

Table 2. Ring Routing Results With Pad Placement

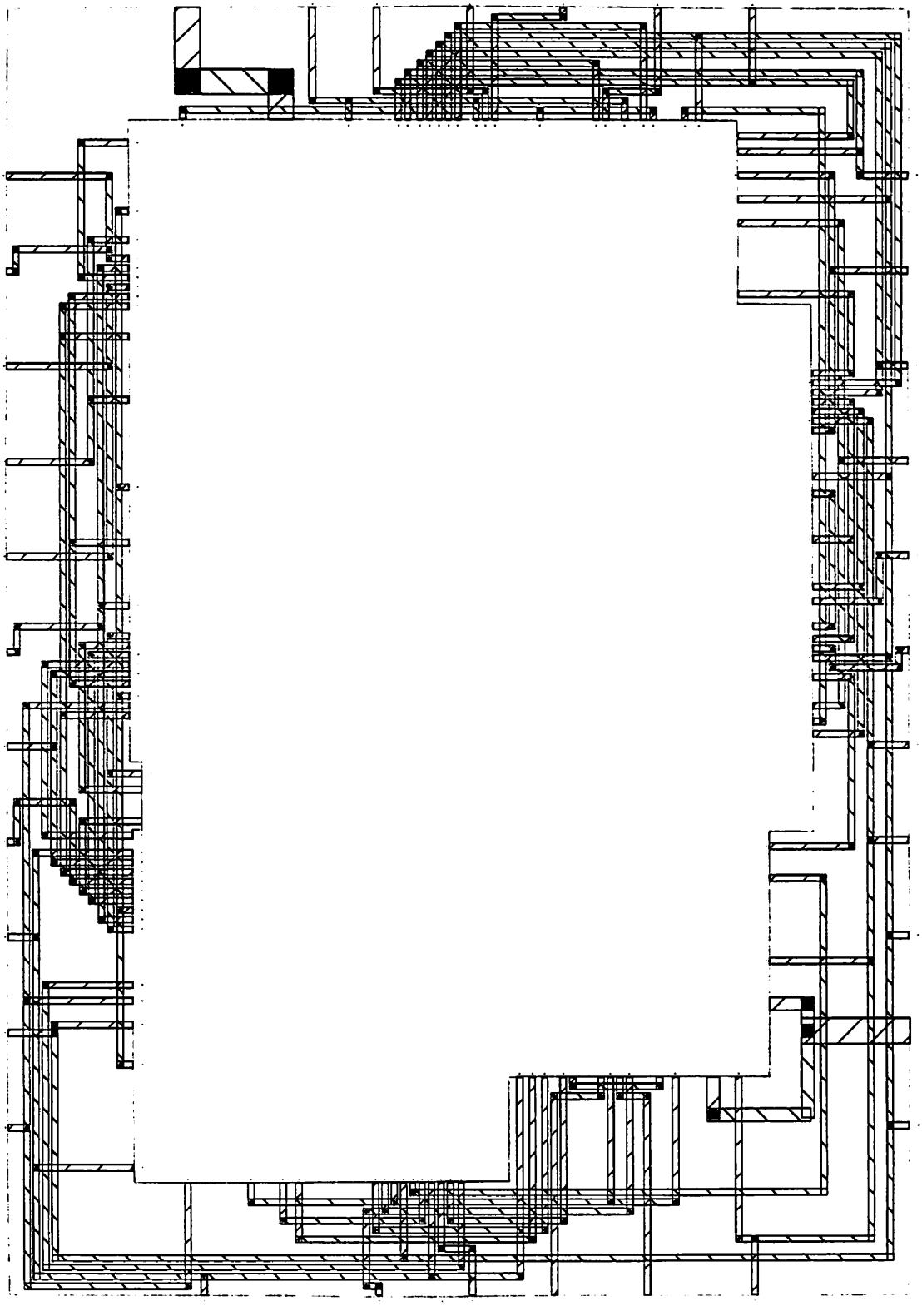
Chips	Pads	CPU (sec)	Total Wire Length	Chip Area	
Hughes	36	8.2	16287 (31%)	467810 (12%)	*
PrimBBL2	40	15.8	48630 (25%)	325981 (3%)	*
Hp	46	13.8	84758 (17%)	158680 (9%)	*
RTASIC	100	65.7	357460 (17%)	429000 (18%)	**

* denotes core bound chip, ** denotes pad bound chip.

Table 3. Total Wire Length Comparison

Chips	Ring Router With Pad Placement	Ring Router Without Pad Placement	Channel Routing
Hughes	16287	21336	22196
PrimBBL2	48630	60787	65562
Hp	84758	97067	98163
RTASIC	357460	418228	471134

The plot on this page shows the actual routing results on a core bound Hughes chip.



4.6 Conclusion

We have developed a complete package of pad placement and ring routing for interconnecting the chip core and the I/O pads. The pad placement routine (1) minimizes the chip area by analytically solving the *outer core dimension* problem and (2) minimizes the total wire length by using the linear assignment formulation. Upon completion of pad placement, the ring router connects terminals according to the netlist in two layers based on the left edge algorithm. We have presented new strategies for eliminating vertical constraints and for routing corner pads. Each net is routed in the shortest path. The signal nets can be routed near density to yield a ring of minimal thickness. Shrinkage/expansion of each side of the ring is independent. A practical situation involving multiple power/ground nets and variable widths for P/G nets is considered. The computation complexity of the package is $O(N^2) + O(nlogn)$, where N is the number of pads and n is the number of terminals. This project has been implemented as part of the BEAR Layout System for custom chip design.

Chapter 5

Future Research for Detailed Routing

5.1 Two-layer Channel Routing

One important feature which can still be enhanced in the existing two-layer channel router is the optimization of the routing for multi-terminal nets. In the current implementation of the two-layer channel router , multi-terminal nets are divided into two-terminal subnets and routed by the generic two-terminal nets algorithm. This approach has the inherent disadvantage of forcing one-sided subnets to be classified as either left or right nets. Therefore how to handle one-sided subnets and how to optimize the routing scheme for multi-terminal nets remains an interesting, and as yet unsolvedproblem.

5.2 L-shaped Channel Routing

The concept of L-shaped channel was first introduced in RRDO [19] to generate a feasible routing order for non-slicing structure placement in building block layout design. Chen [15] proposed an L-shaped channel router by dividing the L region into two subchannels. Depending on the wiring style, vertical or 45° constraint graphs were constructed for both channels. The vertical subchannel and horizontal channels

were then routed and expanded alternatively. Several iterations may be required to complete the routing unless the boundaries were moved in the 45° direction and 45° wires were used. It was not clear whether more iterations were required when the boundaries moved to any angle. Whether or not it is possible to use sorting or river routing techniques to route L channels so that (1) the vertical or 45° constraints can be completely eliminated and (2) routing completion does not require iterations are additional as yet unsolved problems.

5.3 Multi-layer Channel Routing

With the advances in manufacturing technology, multi-layer routing has become a reality. It is important to design detailed routers to handle multi-layer regions. This development is also meaningful for other technologies such as multi-chip modules and hybrid packages. In the multi-layer environment, it is important to explore new routing techniques which are not strictly Manahattan with H/V (horizontal and vertical) wires. Diagonal wires and overlappping wires not only provide more degrees of freedom for completing the interconnections but also shorten the interconnection length. The latter has a profound impact on system performance and cost. In this thesis, 45° and 60° wires have been investigated to route channels using two or three layers. The routing algorithms based on sorting and river routing have the distinct advantage of generating a smaller number of vias since they introduce the minimal number of net crossings. To generalize the generic algorithms to handle n layers, would require three steps. The first step would divide the layers into sets, in which each set contains either two or three layers. Then the second step would distribute nets among the sets of layers. And finally the third step would look at each set of layers and nets as an independent routing problem. The key to optimal results lies in how nets are distributed among the sets of layers so that the sets have uniform density.

Appendix A

Proof of Theorem 3

Theorem 3 *Potential value decreases at most by 1 per net per step under the mini-swap model.*

Proof:

For brevity of notation, let
$A = (a(1), a(2), ..., a(N))$ denote the permutation of nets on track t.
$A' = (a'(1), a'(2), ..., a'(N))$ denote the permutation of nets on track $t+1$.
$DELTA(i,j,t) = ORDER(i,j,t) - NOTORDER(i,j,t) + 1$

Case 1: If $NOTORDER(i,N,t) = 0$, then

$$POTENTIAL(i,t) = NOTORDER(i,1,t) + MAXLT(i,t) \qquad \text{(A.1)}$$

where $MAXLT(i,t) = max(\{0\} \cup \{DELTA(i,j,t) | \, j < i, \, and \, a(i) < a(j)\})$

Subcase 1.1: If the net at position i does not switch position at t, then

$$NOTORDER(i,1,t+1) = NOTORDER(i,1,t) \qquad \text{(A.2)}$$

Let $J = \{ \, j \mid 1 \leq j < i, \, a(i) < a(j) \, \}$
Let $j \in J$.

Subcase 1.1A: If the net at position j switches position with its right neighbor at time t, then $a(j+1) < a(j)$ and $a(i) < a(j)$.
If $a(j+1) < a(i)$, then
$ORDER(i,j+1,t+1) = ORDER(i,j,t) - 1$ and
$NOTORDER(i,j+1,t+1) = NOTORDER(i,j,t)$
Hence, $DELTA(i,j+1,t+1) = DELTA(i,j,t) - 1$

Otherwise, i.e. if $a(j+1) > a(i)$, then
$ORDER(i,j+1,t+1) = ORDER(i,j,t)$ and
$NOTORDER(i,j+1,t+1) = NOTORDER(i,j,t)$
Hence, $DELTA(i,j+1,t+1) = DELTA(i,j,t)$

Subcase 1.1B: If the net at position j switches position with its left neighbor at time t, then $a(i) < a(j) < a(j-1)$. Hence,
$ORDER(i,j-1,t+1) = ORDER(i,j,t)$
$NOTORDER(i,j-1,t+1) = NOTORDER(i,j,t)$
$DELTA(i,j-1,t+1) = DELTA(i,j,t)$

Subcase 1.1C: If the net at position j does not switch position, then
$ORDER(i,j,t+1) = ORDER(i,j,t)$
$NOTORDER(i,j,t+1) = NOTORDER(i,j,t)$
$DELTA(i,j,t+1) = DELTA(i,j,t)$

Since in Subcases 1.1A, 1.1B and 1.1C, $DELTA(i,j,t)$ decreases at most by 1, $MAXLT(i,t)$ decreases at most by 1. Substitute $MAXLT(i,t)$ and equation (A.2) into equation (A.1), we can conclude POTENTIAL decreases at most by 1 for Subcase 1.1.

Subcase 1.2: If net at position i switches position at t, then it must switch with the net on its left since $NOTORDER(i,N,t) = 0$. This implies

$$NOTORDER(i-1,1,t+1) = NOTORDER(i,1,t) - 1 \qquad \text{(A.3)}$$

Let $J = \{\, j \mid 1 \le j < i,\ a(i) < a(j) \,\}$
Let $j \in J$.

Subcase 1.2A: If the net at position j does not switch position at time t, then
$ORDER(i-1,j,t+1) = ORDER(i,j,t)$ and
$NOTORDER(i-1,j,t+1) = NOTORDER(i,j,t) - 1$
$DELTA(i-1,j,t+1) = DELTA(I,J,T) + 1$

Subcase 1.2B: Suppose the net at position j switches with its rihgt neighbor at time t.
If $a(j+1) < a(i) < a(j)$, then
$ORDER(i-1,j+1,t+1) = ORDER(i,j,t) - 1$
$NOTORDER(i-1,j+1,t+1) = NOTORDER(i,j+1,t) - 1$
$DELTA(i-1,j+1,t+1) = DELTA(i,j,t)$
Otherwise, i.e. $a(i) < a(j+1) < a(j)$, we have
$ORDER(i-1,j+1,t+1) = ORDER(i,j,t)$
$NOTORDER(i-1,j+1,t+1) = NOTORDER(i,j,t) - 1$
$DELTA(i-1,j+1,t+1) = DELTA(i,j,t) + 1$

Subcase 1.2C: Suppose the net at position j switches with its left neighbor at time t. This implies $a(i) < a(j) < a(j-1)$. Then
$ORDER(i-1, j-1, t+1) = ORDER(i, j, t)$
$NOTORDER(i-1, j-1, t+1) = NOTORDER(i, j, t) - 1$
$DELTA(i-1, j-1, t+1) = DELTA(i, j, t) + 1$

Since in Subcases 1.2A, 1.2B and 1.2C, $DELTA(i, j, t)$ increases by 0 or 1, $MAXLT(i, t)$ can increase by 0 or 1. Substitute $MAXLT(i, t)$ and equation (A.3) into equation (A.1), we can conclude POTENTIAL decreases at most by 1 for Subcase 1.2.

Case 2: If $NOTORDER(i, 1, t) = 0$, then

$$POTENTIAL(i, t) = NOTORDER(i, 1, t) + MAXGT(i, t)$$

The proof of this case is similar to that of Case 1.

Case 3: When $NOTORDER(i, 1, t) \neq 0$ and $NOTORDER(i, N, t) \neq 0$,

$$POTENTIAL(i, t) = NOTORDER(i, 1, t) + NOTORDER(i, 1, t) + max\{\, 1,\ MAXLT(i, t),\ MAXGT(i, t)\,\} \quad \text{(A.4)}$$

Subcase 3.1: If the net at position i does not switch position at time t, then,

$$ORDER(i, 1, t+1) = ORDER(i, 1, t) \quad \text{(A.5)}$$

$$NOTORDER(i, 1, t+1) = NOTORDER(i, 1, t) \quad \text{(A.6)}$$

Similar to the proof of Subcase 1.1, we can show

$$MAXLT(i, t+1) = MAXLT(i, t) \text{ or } MAXLT(i, t) - 1 \quad \text{(A.7)}$$

$$MAXGT(i, t+1) = MAXGT(i, t) \text{ or } MAXGT(i, t) - 1 \quad \text{(A.8)}$$

Substituting equations (A.5), (A.6), (A.7) and (A.8) into equation (A.4), we conclude POTENTIAL decreases at most by 1 for Subcase 3.1.

Subcase 3.2: If the net at position i switches position with its left neighbor, then

$$ORDER(i-1, 1, t+1) = ORDER(i, 1, t) - 1 \quad \text{(A.9)}$$

$$NOTORDER(i-1, 1, t+1) = NOTORDER(i, 1, t) \quad \text{(A.10)}$$

Similar to the proof of Subcase 1.2, we can show

$$MAXLT(i-1,t+1) = MAXLT(i,t) \text{ or } MAXLT(i,t)+1 \quad \text{(A.11)}$$

To see the changes in $MAXGT(i-1,t+1)$, we again do a case analysis.
Let $J' = \{\ j \mid i \leq j < N,\ a(j) < a(i)\ \}$.
Let $j \in J'$.

Subcase 3.2A: If the net at position j does not switch position at time t, then
$ORDER(i-1,j,t+1) = ORDER(i,j,t)+1$
$NOTORDER(i-1,j,t+1) = NOTORDER(i,j,t)$
$DELTA(i-1,j,t+1) = DELTA(i,j,t)+1$

Subcase 3.2B: If the net at position j switches position with its right neighbor at time t, then $a(j+1) < a(j) < a(i)$. This implies
$ORDER(i-1,j+1,t+1) = ORDER(i,j,t)+1$
$NOTORDER(i-1,j+1,t+1) = NOTORDER(i,j,t)+1$
$DELTA(i-1,j+1,t+1) = DELTA(i,j,t)$

Subcase 3.2C: Suppose the net at position j switches position with its left neighbor at time t. This implies $a(j) < a(j-1)$ and $a(j) < a(i)$.
If $a(j) < a(j-1) < a(i)$, then
$ORDER(i-1,j-1,t+1) = ORDER(i,j,t)+1$
$NOTORDER(i-1,j-1,t+1) = NOTORDER(i,j,t)-1$
$DELTA(i-1,j-1,t+1) = DELTA(i,j,t)+2$
Otherwise, i.e. $a(j-1) > a(i)$,
$ORDER(i-1,j-1,t+1) = ORDER(i,j,t)$
$NOTORDER(i-1,j-1,t+1) = NOTORDER(i,j,t)$
$DELTA(i-1,j-1,t+1) = DELTA(i,j,t)$

Since in Subcases 3.2A, 3.2B and 3.2C, $DELTA(i,j,t)$ increases by 0, 1, or 2, $MAXGT(i,t)$ can increase by 0, 1 or 2. Substitute $MAXGT(i,t)$, and equations (A.9), (A.10), and (A.11) into (A.4), we can conclude POTENTIAL decreases at most by 1 for Subcase 3.2.

Subcase 3.3: If the net at position i switches position with its right neighbor, the proof is analogous to Subcase 3.2.

■

Appendix B

Proof of Theorem 4

Theorem 4 *For each net i, we define the displacement(i) as the horizontal distance between its two terminals. Then POTENTIAL(0)* $\geq max(\{displacement(i) \mid 1 \leq i \leq N\})$.

Proof:

Assume a dense 2-terminal net channel routing problem has N nets. Without loss of generality, the nets are in the natural order on the top and are arbitrarily ordered on the bottom. (Defition 3 in Chapter 2). Given any net i, the top and bottom terminal positions of net i are in columns i and j respectively (Figure B.1).

Case 1: $j \geq i$

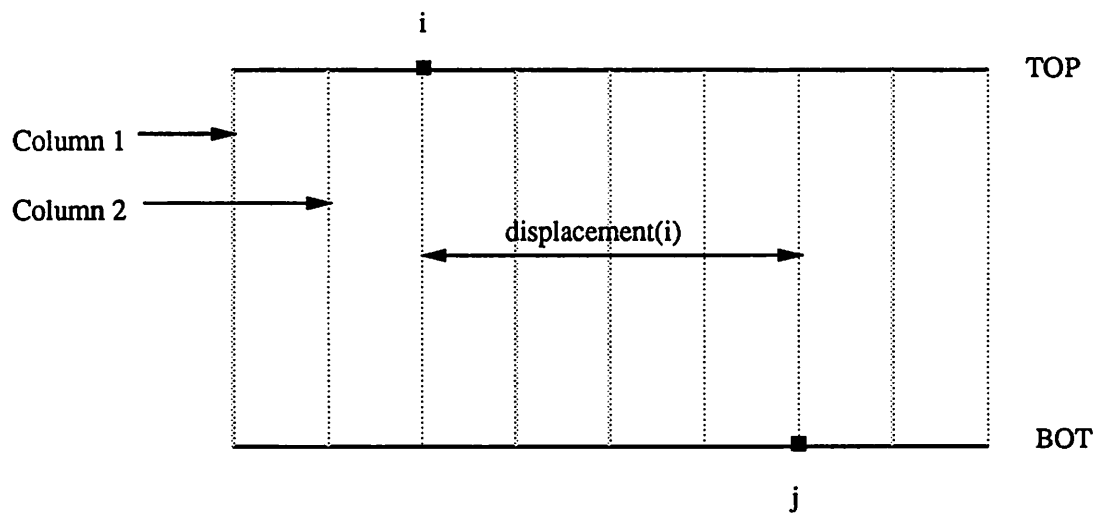


Figure B.1: A net i has top and bottom terminals in columns i and j respectively. The *displacement* of net i is the horizontal distance between the terminals.

Subcase 1.1: When $NOTORDER(i,N) = 0$, then obviously $NOTORDER(i,1) \neq 0$

Let $S = \{ i+1, i+2, ..., N \}$.
Let $S1 \subset S$ be the nets that have bottom terminals in positions $j+1$, $j+2$, ..., N. Clearly,
$|S1| \leq N - j$

Let $S2 = S - S1$ be the nets that have bottom terminals in positions 1, 2, ..., $j-1$. Then

$|S2| \geq (N-i) - (N-j) = j - i = displacement(i)$

Therefore,

$$\begin{aligned} POTENTIAL(i,t) &= NOTORDER(i,1,t) + MAXLT \\ &= |S2| + MAXLT \\ &\geq displacement(i) \end{aligned} \tag{B.1}$$

Subcase 1.2: When $NOTORDER(i,1) \neq 0$ and $NOTORDER(i,N) \neq 0$,

Let $S = \{ i+1, i+2, ..., N \}$
Let $S1 \subset S$ be the nets that have bottom terminals in positions 1, 2, ... $j-1$. Then

$|S1| = NOTORDER(i,1) \geq displacement(i)$

$$\begin{aligned} POTENTIAL(i,t) &= NOTORDER(i,1,t) + NOTORDER(i,N,t) + \\ &\quad max(\,1,\, MAXLT,\, MAXGT\,) \\ &= |S1| + NOTORDER(i,N,t) + \\ &\quad max(\,1,\, MAXLT,\, MAXGT\,) \\ &\geq displacement(i) \end{aligned} \tag{B.2}$$

Case 2: $j \leq i$

This case is simply the mirror image of Case 1. The proof is similar.

■

Appendix C

Proof of Theorem 6

Theorem 6 *The upper bound on the channel height under the arbitrary overlap model is POTENTIAL(0) + 2.*

Proof:

Without loss of generality, the channel can be viewed as having two sections: the bottom section and the top section In the bottom section, the right nets are routed to their target posi- tions; in the top section, the left nets are routed (Figure C.1).

Let us consider the change in the POTENTIAL values of nets in the bottom section. The proof is by Lemma 1, 2 and 3. The main result of these three lemmas is that the POTENTIAL value of every net decreases by at least 1 (amortized) per track in the bottom section. The proof for the top section is similar.

■

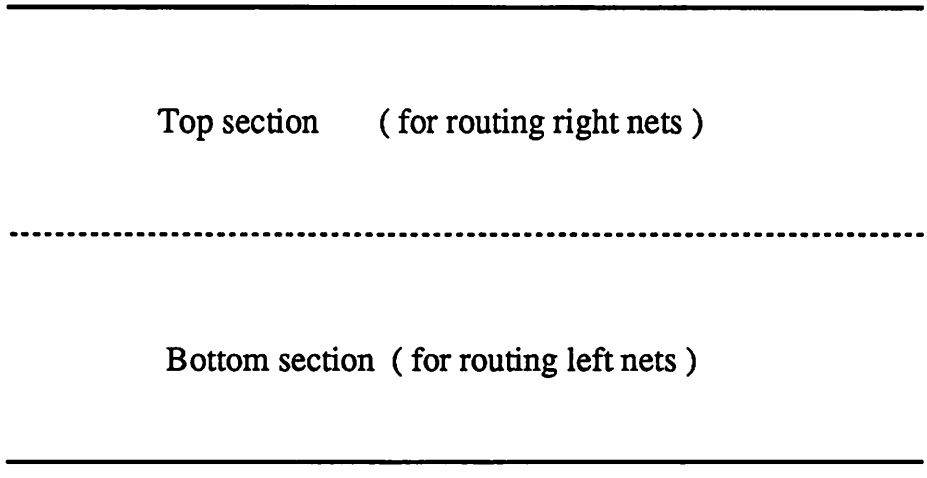


Figure C.1: The channel can be viewed as having two section.

Lemma 6 *In the bottom section, If* $NOTORDER(i,N,t) = 0$, *then*
$POTENTIAL(i,t+1) = POTENTIAL(i,t) - m$
where $m \geq 1$, *unless*
$POTENTIAL(i,t) = 0$ *which implies* $POTENTIAL(i,t+1) = 0$.

Proof:

When $NOTORDER(i,N,t) = 0$,

$$\begin{aligned} POTENTIAL(i,t) = & \quad NOTORDER(i,1,t) + MAXLT(i,t) \\ = & \quad NOTORDER(i,1,t) + max\{DELTA(i,j,t)\} \end{aligned} \tag{C.1}$$

Case 1: Net i is a *right* net:

Claim: Net i has reached its target position, that is
$POTENTIAL(i,t) = 0$

Proof:

Suppose for the sake of contradiction that net i has reached position $i' < i$. Now, there are $N - i$ nets which are greater than i. These nets can not fill up the positions in the interval $[\, i'+1, N \,]$. Hence some net $j < i$ is in this interval. This contradicts the fact $NOTORDER(i,N,t) = 0$. ■

Case 2: Net i is a $left$ net and i crosses a net i':

When i crosses a net,

$$NOTORDER(i,1,t+1) = NOTORDER(i,1,t) - 1 \tag{C.2}$$

Let $J = \{\, j \mid j < i,\ a(i) < a(j) \,\}$
Let $j \in J$.

Subcase 2.1: When j crosses another net:

If j propagates vertically upwards(Figure C.2), then $DELTA(i,j,t+1) = DELTA(i,j,t)$.
Else if j propagates horizontally (Figure C.3) to the right, then there exist a net $j' > j$ such that $j' > i$.

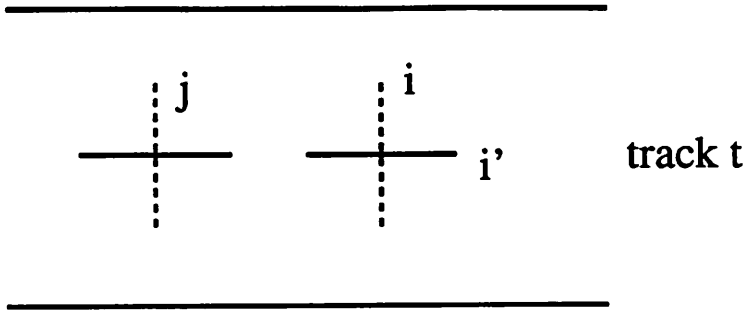


Figure C.2: A right net j propagates vertically upwards in track t.

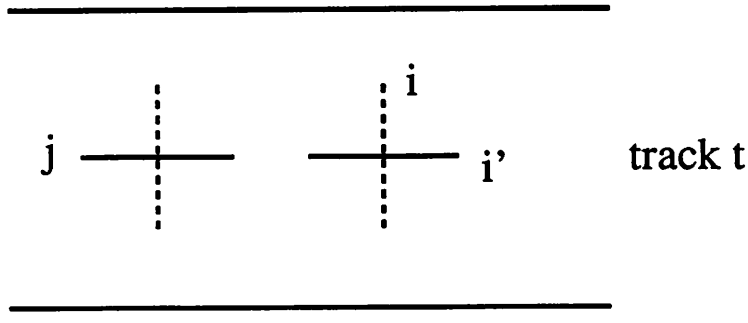


Figure C.3: A right net j propagates horizontally in track t.

Thus,
$ORDER(i,j,t+1) = ORDER(i,j',t)$
$NOTORDER(i,j,t+1) = NOTORDER(i,j',t)$

$$DELTA(i,j,t+1) = DELTA(i,j',t) \tag{C.3}$$

Substituting equations (C.2) and (C.3) into (C.1), we have
$POTENTIAL(i,t+1) = POTENTIAL(i,t) - 1$.

Subcase 2.2 : Suppose j does not cross another net.

If j is a right net(Figure C.4): Then $DELTA(i,j,t+1) = DELTA(i,j',t)$.

Otherwise, j is a left net (Figure C.5):

This implies there exists a right net j' overlapping with j (else j would have been crosses by a right net) such that $j' > j$. Hence,

$$DELTA(i,j,t+1) = DELTA(i,j',t) - 1 \tag{C.4}$$

Substituting equations (C.2) and (C.4) into (C.1), we have
$POTENTIAL(i,t+1) = POTENTIAL(i,t) - 2$.

Case 3: Net i is a left net and i does not cross another net(Figure C.6).

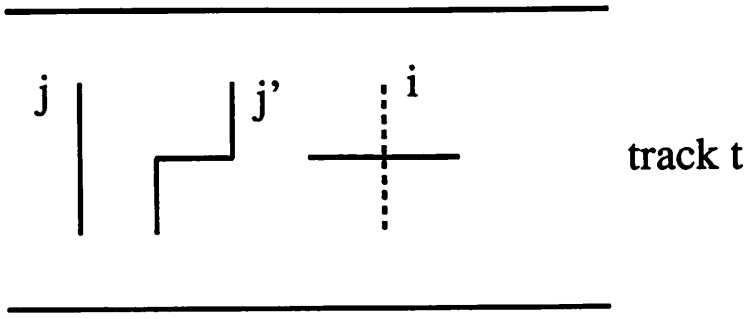


Figure C.4: A right net j does not cross another net in track t.

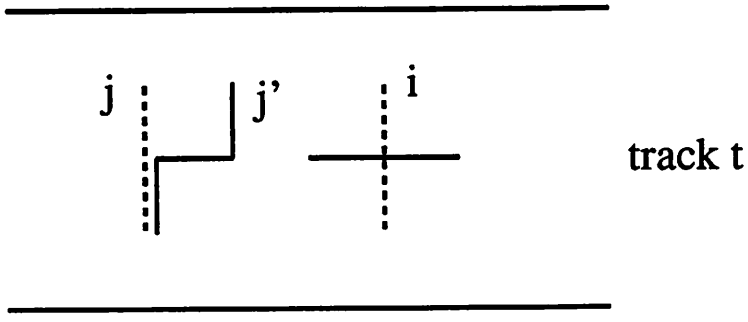


Figure C.5: A left net j does not cross another net in track t.

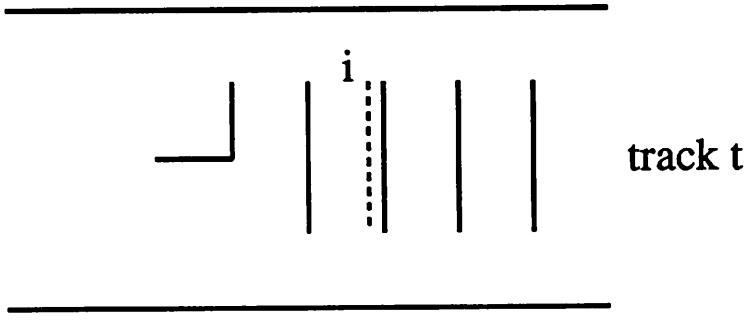


Figure C.6: A left net i does not cross another net in track t.

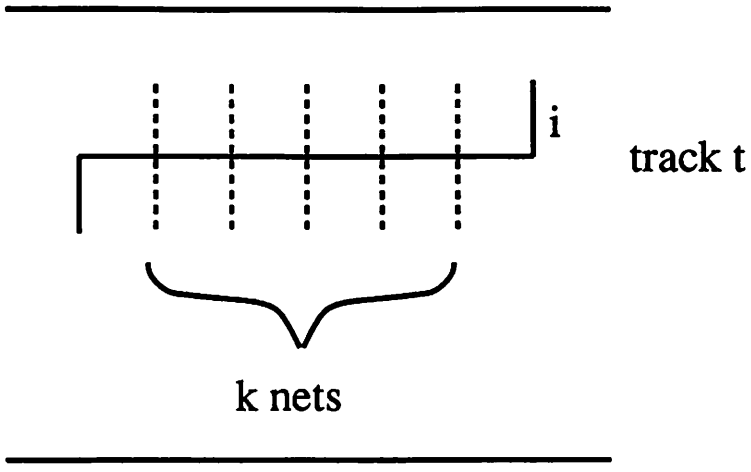


Figure C.7: A right net i crosses k nets in track t.

Since i is a left net and $NOTORDER(i, N, t) = 0$, net i can be routed to its target position at the $t'th$ track in the top section which reduces $POTENTIAL(i, t)$ to 0 (see proof of Lemma 2). This implies the amortized decrease in POTENTIAL value of net i is greater than 1. Note: once $POTENTIAL(i, t+1)$ becomes zero, net i will be dropped from analysis in future steps.

■

Lemma 7 *In the bottom section, If NOTORDER(i, 1, t) = 0, then POTENTIAL(i,t+1) = POTENTIAL(i,t) - m, where* $m \geq 1$.

Proof:

In the bottom section, only a right net could satisfy $NOTORDER(i, 1, t) = 0$. By the routing algorithm, net i would propagte horizontal toward right until another right net i', $i' > i$ is reached. Since $NOTORDER(i, 1, t) = 0$, the $POTENTIAL$ value of net i is defined as

$$\begin{aligned} POTENTIAL(i,t) &= NOTORDER(i,N,t) + MAXGT(i,t) \\ &= NOTORDER(i,N,t) + max\{DELTA(i,j,t)\} \end{aligned} \quad \text{(C.5)}$$

Case 1: Net i crosses k nets as shown in (Figure C.7).

In this case, we have

$$NOTORDER(i, N, t+1) = NOTORDER(i, N, t) - k \quad \text{(C.6)}$$

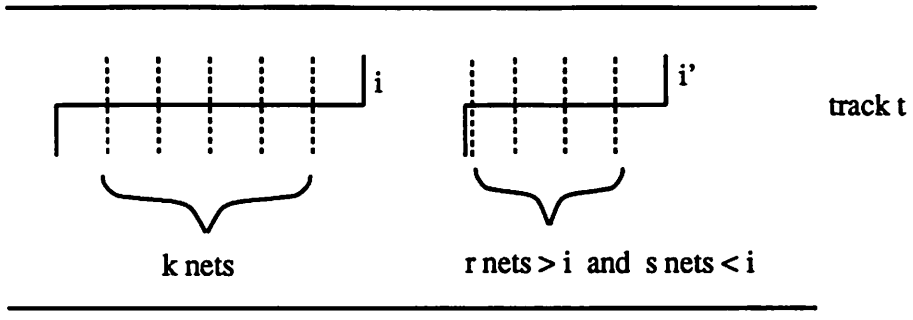


Figure C.8:

Let $J = \{ j \mid j < i,\ a(i) < a(j) \}$
Let $j \in J$.

By examining equations (C.5) and (C.6), we need to show $DELTA(i,j,t)$ does not increase more than k.

If $MAXGT(i,t+1) = 0$, then

$$POTENTIAL(i,t+1) = POTENTIAL(i,t) - k \tag{C.7}$$

Else if $MAXGT(i,t+1) > 0$ (Figure C.8), then

$DELTA(i,j,t+1)$
$= ORDER(i,j,t+1) - NOTORDER(i,j,t+1)$
$= r - s$
$= (r+1) - (k+s) + (k-1)$
$= ORDER(i,j,t) - NOTORDER(i,j,t) + (k-1)$
$= DELTA(i,j,t) + (k-1)$

Case 2: When net i does not cross any net,

$NOTORDER(i,N,t+1) = NOTORDER(i,N,t)$

We need to show that $DELTA(i,j,t)$ decreases.

If $MAXGT(i,t) > 0$ (Figure C.9), then

$DELTA(i,j,t+1)$
$= ORDER(i,j,t+1) - NOTORDER(i,j,t+1)$
$= (ORDER(i,j,t) - 1) - NOTORDER(i,j,t)$
$= DELTA(i,j,t) - 1$

Otherwise, if $MAXGT(i,t) = 0$ (Figure C.10).

This implies $j' < i$ and net i will cross j' and j in track $t+1$. Hence the amortized decrease in the POTENTIAL value of net is 1.

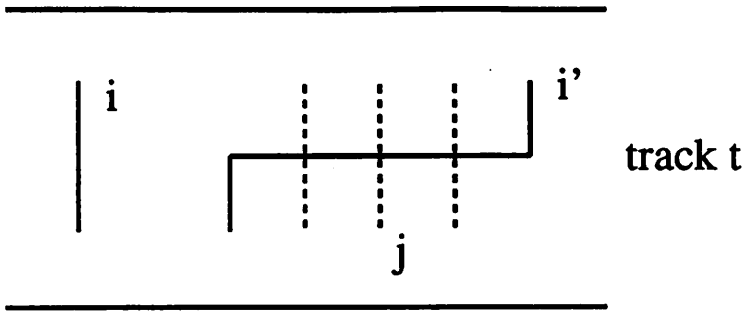


Figure C.9:

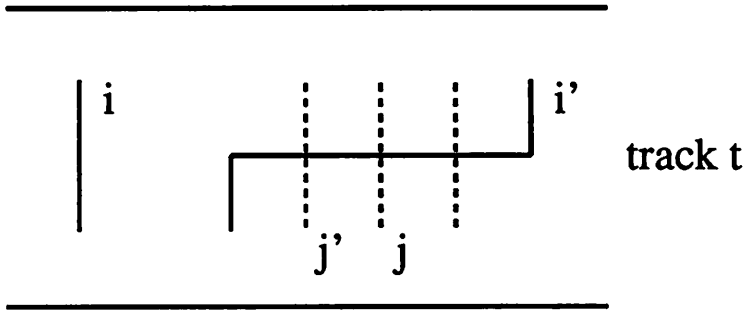


Figure C.10:

■

Lemma 8 *In the bottom section, If NOTORDER(i, 1, t) ≠ 0 and NOTORDER(i, N, t) ≠ 0, then POTENTIAL(i,t+1) = POTENTIAL(i,t) - m where m ≥ 1, unless POTENTIAL(i,t) = 0 which implies POTENTIAL(i,t+1) = 0.*

Proof:

When $NOTORDER(i,1,t) \neq 0$ and $NOTORDER(i,N,t) \neq 0$, net i propagates vertically upwards on layer 2 in the bottom section and

$$POTENTIAL(i,t) = NOTORDER(i,1,t) + NOTORDER(i,N,t) + max1, MAXLT, MAXGT \quad \text{(C.8)}$$

Case 1: If net i crosses another net, then

$$NOTORDER(i,1,t+1) = NOTORDER(i,1,t) - 1 \quad \text{(C.9)}$$

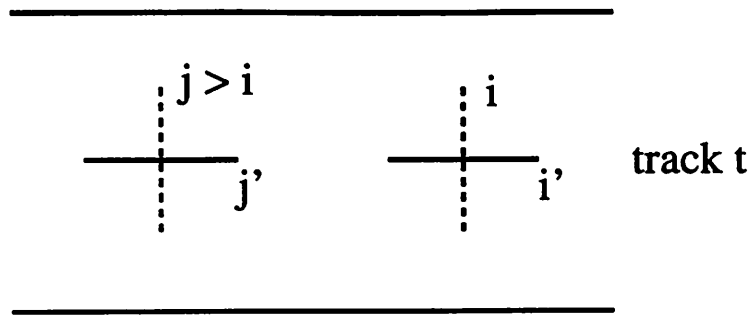


Figure C.11:

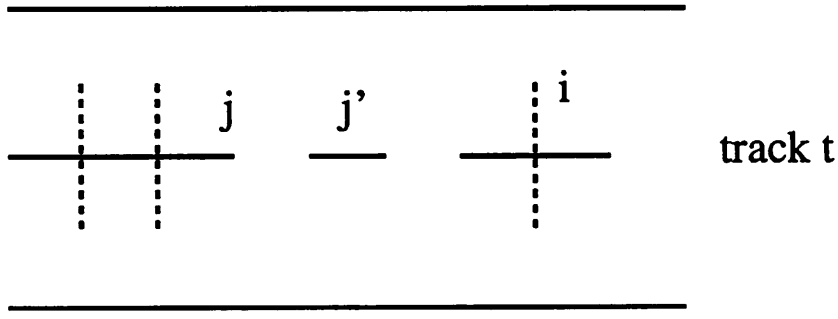


Figure C.12:

$$NOTORDER(i, N, t+1) = NOTORDER(i, N, t) \tag{C.10}$$

By examining equations (C.8), (C.9) and (C.10), we We need to show $MAXLT$ and $MAXGT$ does not increase.

Subcase 1.1: i is a right net

To show the change in $MAXLT$....

Let $J = \{ j \mid j > i,\ a(i) > a(j) \}$.
Let $j \in J$.

Subcase 1.1A: If j crosses a net as shown in (Figure C.11).
Then $j' > j > i$ and $DELTA(i, j, t+1) = DELTA(i, j, t)$.

Subcase 1.1B: if j crosses a net (Figure C.12):
Then $j' > j > i$ and $DELTA(i, j, t+1) = DELTA(i, j', t)$.

Subcase 1.1C: If j does not cross any net (Figure C.13):
then $DELTA(i, j, t+1) = DELTA(i, j', t)$

From Subcases (1.1A), (1.1B) and (1.1C), we conclude that $MAXLT(i, t)$ decreases by ≥ 0 since DELTA(i,t) decreases by ≥ 0.

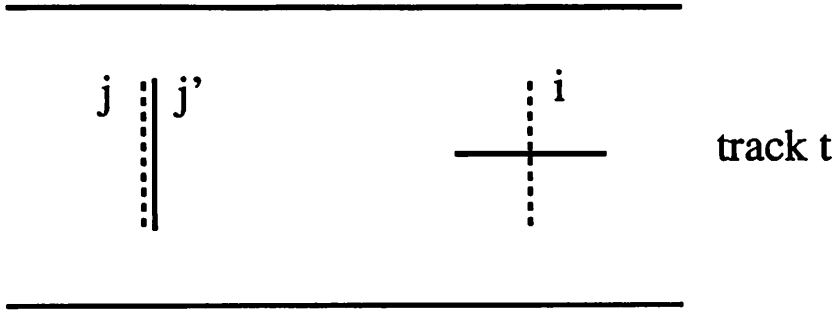


Figure C.13:

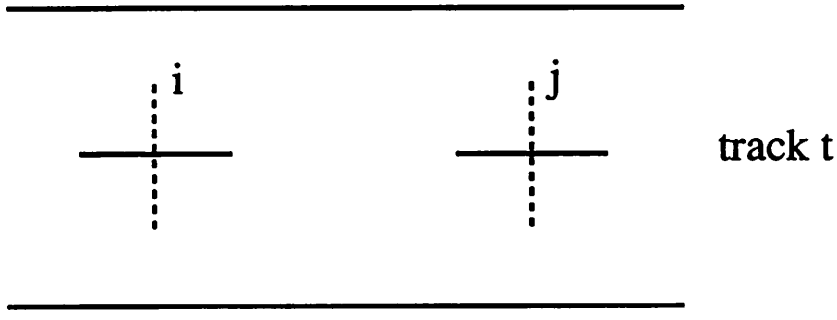


Figure C.14:

To show the change in $MAXGT$....

Let $J' = \{\ j \mid j < i,\ a(i) < a(j)\ \}$
Let $j \in J'$.

Subcase 1.1D: If j crosses another net (Figure C.14).
Then, DELTA(i,j,t+1) = DELTA(i,j,t)

Subcase 1.1E: net j does not cross any net (Figure C.15):
If $DELTA(i,j,t+1) \leq 1$, then done. Else, ie. if $DELTA(i,j,t+1) \geq 2$, then net j will cross 2 or more nets in some track in the top section.

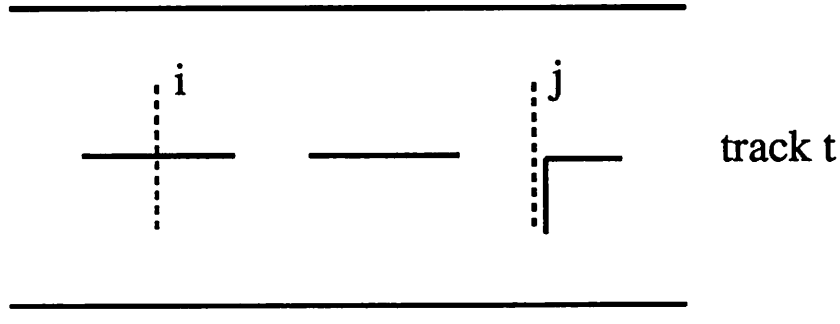


Figure C.15:

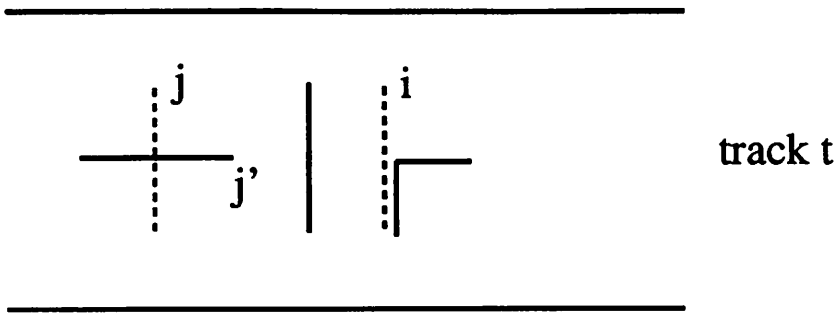


Figure C.16:

This implies the amortized decrease in the POTENTIAL value of net i is greater than or equal to 1.

Subcase 1.2: i is a left net

The proof is analogous to Case 1.1.

Case 2: If net i does not cross any net, then

$$NOTORDER(i,1,t+1) = NOTORDER(i,1,t) \tag{C.11}$$

$$NOTORDER(i,N,t+1) = NOTORDER(i,N,t) \tag{C.12}$$

By examining equations (C.8), (C.11) and (C.12), we We need to show $MAXLT$ and $MAXGT$ decreases.

To show the change in $MAXLT$....,

Let $I = j \mid j > i,\ a(i) > a(j)$.
Let $j \in I$.

Subcase 2A: If j crosses a net (Figure C.16).

Then $j' > j > i$ and
$DELTA(i,j,t+1)$
$= ORDER(i,j,t+1) - NOTORDER(i,j,t+1)$
$= ORDER(i,j,t) - (NOTORDER(i,j,t)+1)$
$= DELTA(i,j,t) - 1$

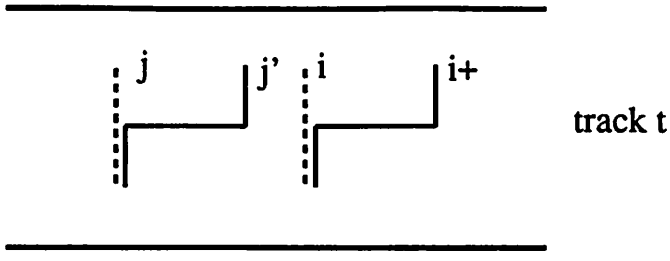


Figure C.17:

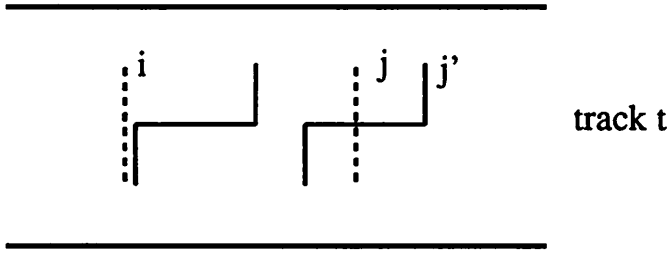


Figure C.18:

Subcase 2B: If j does not cross any net (Figure C.17).

Then,

$DELTA(i,j,t+1)$
$= DELTA(i,j,t)$
$= ORDER(i,j,t) - NOTORDER(i,j,t)$
$= ORDER(i,j',t) - (NOTORDER(i,j',t)+1)$
$= DELTA(i,j',t) - 1$

From Subcases (2A) and (2B), we can conclude that $MAXLT(i,t)$ decreases by 1.

To show the change in $MAXGT$,

Let $I' = \{\ j \mid j < i,\ a(i) < a(j)\ \}$.
Let $j \in I'$.

Subcase 2C: If j crosses another net (Figure C.18):

Then $j' > i$. Hence,

$DELTA(i,j,t+1) = ORDER(i,j,t) - 1 - NOTORDER(i,j,t)$
$= DELTA(i,j,t) - 1$

Subcase 2D: net j does not cross another net

The proof is similar to (1.1E)

From Subcases (2C) and (2D), we conclude that $MAXGT(i,t)$ decreases by 1.

From Case 2 we find that both $MAXGT(i,t)$ and $MAXLT(i,t)$ decrease by 1. Substituting these two terms and equations C.11 and C.12 into equation C.8, we conclude that POTENTIAL decreases by 1 in Case 2.

■

Bibliography

[1] S.G. Akl. Linear arrays. In *Parallel sorting algorithms*, pages 41–59. Academic Press, 1985.

[2] B. Baker and R. Pinter. An algorithm for the optimal placement and routing of a circuit with a ring of pads. In *Proc. 24th Annual Symposium on Foundations of Computer Science*, pages 360–370, 1983.

[3] B.S. Baker, S.N. Bhatt, and F.T. Leighton. An approximation algorithm for manhattan routing. In *Proc. 15th Annual Symp. on Theory of Computing*, 1983.

[4] A.E. Baratz. *Algorithms for integrated circuit signal routing.* PhD thesis, Massachusetts Institute of Technology, August 1981.

[5] M.L. Brady and D.J. Brown. An algorithm for three layer channel routing using restricted overlap. In *Proc. 23rd Annual Allerton Conference on Communication, Control and Computing*, pages 674–675, 1985.

[6] M.L. Brady and D.J. Brown. Optimal multilayer channel routing with overlap. In *Proc. 4th MIT Conference on Advanced Research in VLSI*, pages 281–296, 1986.

[7] D.J. Brown and R.L. Rivest. New lower bounds for channel width. In *Proc. CMU Conf. VLSI Systems and Computations*, pages 178–185, 1981.

[8] P. Bruell and P. Sun. A greedy three layer channel router. In *Proc. of the International Conference on Computer-Aided Design*, pages 298–300, 1985.

[9] R.E. Burkard and U. Derigs. Linear assignment. In *Assignment and matching problems: solution methods with Fortran programs*, pages 1–31. 1980.

[10] M. Burstein. Channel routing. In *Layout Design and Verification*, pages 133–164. 1986.

[11] M. Burstein. Channel routing. In *Layout Design and Verification*, pages 133–164. 1986.

[12] M. Burstein and R. Pelavin. Hierarchical channel router. In *Proc. 20th ACM/IEEE Design Automation Conference*, pages 591–597, 1983.

[13] P. Chaudhuri. An ecological approach to wire routing. In *Proc. of the IEEE International Symposium on Circuits and Systems*, pages 854–857, 1979.

[14] P. Chaudhuri. Routing multilayer boards on steiner metric. In *Proc. of the IEEE International Symposium on Circuits and Systems*, pages 961–964, 1980.

[15] H. H. Chen. Routing l-shaped channels in non-slicing structure placement. In *Proc. 24th ACM/IEEE Design Automation Conference*, pages 152–158, 1987.

[16] H.H. Chen and E.S. Kuh. Glitter: a gridless variable-width channel router. *IEEE Trans. on CAD*, CAD-5:459–465, 1986.

[17] Y.K. Chen and M.L. Liu. Three-layer channel routing. *IEEE Trans. on CAD*, CAD-3:156–163, 1984.

[18] J. Cong, D.F. Wong, and C.L. Liu. A new approach to three- or four-layer routing. *IEEE Trans. on CAD*, CAD-3:1094–1104, 1988.

[19] W.-M. Dai, M. Sato, and E.S. Kuh. Routing region definition and ordering scheme for building-block layout. *IEEE Trans. on CAD*, CAD-4:189–197, 1985.

[20] D.N. Deutsch. A dogleg channel router. In *Proc. 13th ACM/IEEE Design Automation Conference*, pages 425–433, 1976.

[21] J. Edmonds and R. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. In *Journal of the ACM, Vol. 19, No. 2*, pages 248–264, 1972.

[22] R.J. Enbody and H.C. Du. Near-optimal n-layer channel routing. In *Proc. of the 23rd ACM/IEEE Design Automation Conference*, pages 709–714, 1986.

[23] D. Braun et al. Techniques for multilayer channel routing. *IEEE Trans. on CAD*, CAD-7:698–712, 1988.

[24] L. Smith et al. A new area router, the lrs algorithm. In *Proc. IEEE ICCC*, pages 256–259, 1982.

[25] R.I. Greenberg et al. Mulch: A multi-layer channel router using one, two and three layer partitions. In *Proc. of the International Conference on Computer-Aided Design*, pages 88–91, 1988.

[26] W-M. Dai et al. Bear: A new building-block layout system. In *Dig. Tech. Papers, IEEE Int. Conf. Computer-Aided Design*, pages 34–37, 1987.

[27] A. Frank. Disjoint paths in a rectilinear grid. *Combinatorica*, 2(4):361–371, 1982.

[28] H.N. Gabow and R.E. Tarjan. Almost-optimum speed-ups of algorithms for bipartite matching and related problems. In *Proc. 20th Annual Symp. on Th. of Computing*, pages 514–527, 1988.

[29] S.E. Hambrusch. Using overlap and minimizing contact points in channel routing. In *Proc. 21st Annual Allerton Conference on Communication, Control and Computing*, pages 256–257, 1983.

[30] S.E. Hambrusch. Channel routing algorithms for overlap models. *IEEE Trans. on CAD*, CAD-4:23–30, 1985.

[31] A. Hashimoto and J. Stevens. Wire routing by optimizing channel assignment within large apertures. In *Proc. 8th ACM/IEEE Design Automation Workshop*, pages 155–169, 1971.

[32] A. LaPaugh. *Algorithms for Integrated Circuit Layout: An Analytic Approach.* PhD thesis, Massachusettes Institute of Technology, November 1980.

[33] A. LaPaugh. *Algorithms for Integrated Circuit Layout: An Analytic Approach.* PhD thesis, Massachusettes Institute of Technology, November 1980.

[34] A.S. LaPaugh. A polynomial time algorithm for optimal routing around a rectangle. In *Proc. 21st Symp. Foundations of Computer Science*, pages 282–293, 1983.

[35] F.T. Leighton. New lower bounds for channel routing. unpublished manuscript, 1982.

[36] C.E. Leiserson and R.Y. Pinter. Optimal placement for river routing. *SIAM Journal on Computing*, 12(3):447–462, 1983.

[37] E. Lodi, F. Luccio, and L. Pagli. Channel routing for strictly multiterminal nets. *Integration, the VLSI journal*, pages 143–153, 1989.

[38] E. Lodi, F. Luccio, and L. Pagli. A preliminary study of a diagonal channel-routing model. *Algorithmica*, pages 585–597, 1989.

[39] E. Lodi, F. Luccio, and L. Pagli. Routing in the times square mode. Technical report, dipartimento di informatica, Universita Degli Studi Di Pisa, Italy, 1989.

[40] R. McGehee. A practical moat router. In *Proc. 24th ACM/IEEE Design Automation Conference*, pages 216–221, 1987.

[41] K. Mehlhorn, F.P. Preparata, and M. Sarrafzadeh. Channel routing in knock-knee mode: simplified algorithms and proof. *Algorithmica*, 1(2):213–221, 1986.

[42] F.P. Preparata and W. Lipski. Three layers are enough. In *Proc. 23rd Annual Symposium on Foundations of Computer Science*, pages 350–357, 1982.

[43] R.L. Rivest, A.E. Baratz, and G.L. Miller. Provably good channel routing algorithms. In *Proc. of the CMU Conference on VLSI Systems and Computations*, pages 153–159, 1981.

[44] R.L. Rivest and C.M. Fiduccia. A greedy channel router. In *Proc. 19th ACM/IEEE Design Automation Conference*, pages 418–424, 1982.

[45] K. Sado and Y. Igarashi. A function for evaluating the computing time of a bubbling system. *Theoretical Computer Science*, 54:315–324, 1987.

[46] A. Sangiovanni-Vincentelli and M. Santomauro. YACR: Yet another channel router. In *Proc. Custom Integr. Circuits Conf.*, pages 460–466, 1982.

[47] M. Sarrafzadeh. *Hierarchical approaches to VLSI circuit layout.* PhD thesis, University of Illinois, December 1986.

[48] M. Sarrafzadeh. Channel-routing problem in the knock-knee mode is NP-complete. *IEEE Trans. on CAD*, CAD-6(4):503–506, 1987.

[49] M. Sarrafzadeh and F.P. Preparata. A bottom-up layout technique based on two-rectangle routing. In *Integration, the VLSI journal 5*, pages 231–246, 1987.

[50] T.G. Szymanski. Dogleg channel routing is NP-complete. *IEEE Trans. on CAD*, CAD-4(1):31–40, 1985.

[51] N. Tomizawa. On some techniques useful for solution of transportation network problems. In *Networks*, pages 173–194, 1971.

[52] T. Yoshimura and E.S. Kuh. Efficient algorithms for channel routing. *IEEE Trans. on CAD*, CAD-1:25–35, 1982.