

## INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

**The quality of this reproduction is dependent upon the quality of the copy submitted.** Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

# UMI

A Bell & Howell Information Company  
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA  
313/761-4700 800/521-0600

PREVIEW

UNIVERSITY OF CALIFORNIA  
SANTA CRUZ

**RUBBER-BAND BASED TOPOLOGICAL ROUTER**

A dissertation submitted in partial satisfaction of  
the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

by

**Tal Dayan**

June 1997

The dissertation of Tal Dayan is approved:



Professor Wayne Wei-Ming Dai, Chair



Professor Martine Schlag



Professor F. Joel Ferguson



Dean of Graduate Studies

**UMI Number: 9738147**

---

**UMI Microform 9738147**  
**Copyright 1997, by UMI Company. All rights reserved.**

**This microform edition is protected against unauthorized  
copying under Title 17, United States Code.**

---

**UMI**  
**300 North Zeeb Road**  
**Ann Arbor, MI 48103**

Copyright © by  
Tal Dayan  
1997

PREVIEW

## ABSTRACT

A multi-layer, topological detailed-router is described. This is the first router ever reported that uses a rubber-band sketch (RBS) to represent the interconnect. The detailed-router is part of SURF, a routing system for multi-chip modules and VLSI that was designed to handle efficiently large multi-layer problems. The detailed-router supports various routing goals and can generate layouts for rectilinear, octilinear and any-angle wiring rules. It uses a novel approach of unconstrained layer-assignment that makes a better usage of the routing resources by considering a continuous metric of conflict between nets as opposed to the binary go/no-go approach. The layer-assignment is formulated as an optimization problem and various routing goals such as wire and via minimization or constrained-layers can be achieved by simple modifications to the cost function. The layer-assignment partitions the multi-layer problem into a set of single-layer sub-problems that are routed independently by a topological planar-router. The planar-router uses a new net-ordering algorithm that results in shorter wiring than the 'shortest first' approach. The nets are embedded sequentially using an optimal algorithm that finds a shortest planar path in the RBS. The generated RBS is then optimized by a simple re-route algorithm that takes advantage of the 'attachment' relation between branches and terminals in the RBS. A mathematical formulation of the concept of RBS is also presented and is used to prove the correctness of the shortest-path algorithm. This is the first exact analysis of RBS ever published. Empirical results are also shown and they demonstrate the merit of the router.

**Dedicated to Nir**

PREVIEW

## ACKNOWLEDGMENT

I would like to thank the people that enabled and helped this research. First, to my advisor Professor Wayne Dai that his vision, guidance, and leadership was the driving force behind the SURF project. To all the students ('SURF'ers') that worked on various parts of SURF including (in alphabetical order): Joel Darnauer, Mark Fitzpatrick, Duane Foote, Maggie Zhiwei Kang, Kazuhiko Kobayashi, Raymond Kong, Michael Lanzetta, Yizhi Lu, Paul Morton, Paul Pham, David Prather, Darren Senn, David Staepelaere, Joseph Russack, Karen Wang, Jeffrey Su, Marco Man-Fai Yu, William Wu, and Qing Zhu.

Thanks to Professor Joel Ferguson for reviewing this dissertation. Thanks to Professor Martine Schlag for her thorough comments and suggestions regarding contents, organization, style, and spelling. Thanks for Caroline for her sincere notes. Special thanks to David Staepelaere for his enlightening comments on the layer-assignment and the net-ordering chapters, and for his overview of SURF that was the base for SURF overview included in this dissertation. Thanks to Jeffrey Su for his implementation of the 'region' module and for supporting the rubber-band 'engine' used by the local router. Thanks to George 'Joe 19' French for proof-reading the introduction, overview, and the net-embedding chapters. Thank the people at Frame Technology for making available Frame Maker, a great writing tool. And last but not least, I would like to thank Teiko for her support and her help in proof-reading this paper.



## TABLE OF CONTENTS

1	INTRODUCTION .....	1
1.1	General .....	1
1.2	Organization of This Thesis .....	1
1.3	Previous Works .....	1
1.4	Contribution of this work .....	7
2	OVERVIEW .....	9
2.1	Topological Representation and Routing .....	9
2.1.1	Sketches and Topological Classes .....	10
2.1.2	Triangulation Crossing Sketch (TCS) .....	12
2.1.3	Rubber-Band Sketch (RBS) .....	13
2.1.4	Routability Tests .....	14
2.1.5	Topological Routing .....	16
2.2	SURF Routing System .....	17
2.2.1	Introduction .....	17
2.2.2	Layout Refinement Strategy .....	18
2.2.3	Routing Examples .....	22
3	TOPOLOGICAL LAYER-ASSIGNMENT .....	28
3.1	Introduction .....	28
3.2	The Layer-Assignment Problem .....	30
3.2.1	The Input Domain .....	30
3.2.2	The Solution Domain .....	31
3.2.3	The Cost Function .....	33
3.2.4	Properties of the cost function .....	37
3.2.5	User control of .....	42
3.3	The Layer-Assignment Algorithm .....	42
3.3.1	Introduction .....	42
3.3.2	Step I - Breaking the Nets Into 2-Nets .....	44
3.3.3	Step II - Generating The 2-Net Assignment Graphs .....	46
3.3.4	Step III - Solving the Layer-Assignment Problem .....	50
3.3.4.1	Configurations and Assignments .....	50
3.3.4.2	The configuration search algorithm .....	54
3.3.5	Properties of the configuration search algorithm .....	57
3.3.6	The 2-Net Assignment Algorithm (2NAA) .....	59
3.3.7	2NAA complexity Analysis .....	67
3.3.8	LAA Implementation Notes .....	68
3.4	LAA Extensions .....	69
3.4.1	Constrained Assignment .....	69
3.4.2	Supporting Various Metrics in the Layer-Assignment .....	71
3.4.3	Preferring 2-Net assignment to layers .....	72
3.5	Experimental Results .....	73
3.5.1	Benchmark results .....	73
3.5.2	Balancing Between Wiring Length and Via Count .....	75
3.5.3	Estimated vs. Actual Detour .....	78

3.5.4	Comparison of Net Decomposition Methods .....	79
3.5.5	Candidate Via Density Versus Actual Cost .....	81
3.5.6	Using Various Routing Metrics .....	84
3.5.7	LAA Scalability .....	85
3.6	Conclusion and Future Work .....	87
4	TOPOLOGICAL NET ORDERING .....	89
4.1	Introduction .....	89
4.2	Decomposition into 2-Nets .....	89
4.3	2-Net Ordering .....	90
4.4	Planarity Enforcement Operator (PEO) .....	92
4.5	2-Net Ordering Problem (2NOP) Formulation .....	97
4.6	2NOP Complexity .....	101
4.7	Solving the 2NOP .....	102
4.8	Experimental Results .....	104
4.9	Conclusion and Future Work .....	106
5	TOPOLOGICAL PATH SEARCH .....	108
5.1	Least-Cost Topological Path Problem (LCTP) .....	108
5.2	Rubber-Band Sketch Formulation .....	109
5.3	RBS Regions .....	114
5.4	Shortest Planar Path in RBS .....	119
5.5	Reducing the search graph size .....	128
5.6	Conclusion and Future Work .....	139
6	TOPOLOGICAL SKETCH OPTIMIZER (ROAR) .....	141
6.1	General .....	141
6.2	Experimental Results .....	142
7	CONCLUSION AND FUTURE WORK .....	144
8	REFERENCES .....	146

## LIST OF FIGURES

FIGURE 1 - The Puzzle problem.....	9
FIGURE 2 - Homotopic sketches.....	11
FIGURE 3 - Minimal length sketch.....	12
FIGURE 4 - The Triangulation Crossings Sketch (TCS).....	13
FIGURE 5 - Rubber-Band Sketch.....	14
FIGURE 6 - Spokes and ERBS.....	15
FIGURE 7 - Rectilinear ERBS.....	16
FIGURE 8 - Rubber-Band routing example.....	18
FIGURE 9 - SURF routing strategy.....	19
FIGURE 10 - Global routing.....	20
FIGURE 11 - Spokes.....	22
FIGURE 12 - Mixed Analog/Digital.....	23
FIGURE 13 - Rectilinear Routing.....	24
FIGURE 14 - Octilinear Routing.....	25
FIGURE 15 - Euclidean routing.....	26
FIGURE 16 - Larger routing example.....	27
FIGURE 17 - The two level decomposition of a routing problem.....	28
FIGURE 18 - An example of a bin layer-assignment and routing.....	29
FIGURE 19 - An example of non-planar layer-assignment.....	32
FIGURE 20 - Minimizing the via count vs. minimizing wire length.....	33
FIGURE 21 - The basic, actual, and detour lengths of routed components.....	35
FIGURE 22 - Conflict between pairs of components.....	36
FIGURE 23 - Pair-wise conflict vs. the actual detour.....	38
FIGURE 24 - The error of the estimated detour function has no upper bound.....	38
FIGURE 25 - External paths and regions.....	41
FIGURE 26 - Constructing a planar embedding.....	41
FIGURE 27 - The three steps of the assignment algorithm.....	44
FIGURE 28 - Three methods of decomposing a net (a) into 2-Nets.....	46
FIGURE 29 - An example of a simple 2-Net assignment graph.....	48
FIGURE 30 - Two choices for the number of candidate points per 2-Net.....	49
FIGURE 31 - The local-spacing post-processor.....	50
FIGURE 32 - The corresponding solution of a configuration.....	52
FIGURE 33 - Planar embedding of the corresponding solution.....	53
FIGURE 34 - Interdependency between assigned 2-Nets.....	56
FIGURE 35 - Example of assignment improvement.....	56
FIGURE 36 - Local minimum in the configuration search.....	59
FIGURE 37 - Cost of end-point edges.....	61
FIGURE 38 - Cost increase due to branch assignment.....	62
FIGURE 39 - Non-linearity of the branch cost function.....	64
FIGURE 40 - Non-linearity of branch cost in a path.....	65
FIGURE 41 - Adding the short-circuit branch edges.....	66
FIGURE 42 - Splitting the graph nodes.....	67
FIGURE 43 - Constrained layer-assignment.....	70
FIGURE 44 - Candidate via location for Rectilinear metric.....	72
FIGURE 45 - Net ordering and wiring length.....	90

FIGURE 46 - The Triangle problem.....	91
FIGURE 47 - Non-planar 2-Net routing order.....	91
FIGURE 48 - Closed 2-Nets.....	93
FIGURE 49 - Non planar external paths.....	95
FIGURE 50 - Necessary condition for planar embedding.....	96
FIGURE 51 - 2-Net order cost.....	99
FIGURE 52 - Limitations of non-interleaving order.....	102
FIGURE 53 - -sketch.....	110
FIGURE 54 - Terminal's -neighborhood.....	111
FIGURE 55 - Cut's -neighborhood.....	111
FIGURE 56 - Attachments in RBS.....	112
FIGURE 57 - Local nets.....	113
FIGURE 58 - Angle between branch segments.....	113
FIGURE 59 - Implicit attached local nets.....	114
FIGURE 60 - Regions.....	115
FIGURE 61 - Region Visibility.....	116
FIGURE 62 - Neighborhoods intersection.....	118
FIGURE 63 - Link intersections.....	119
FIGURE 64 - Region split by a branch.....	120
FIGURE 65 - Intersection of consecutive visibility links.....	123
FIGURE 66 - Intersection between non-consecutive links.....	124
FIGURE 67 - Link interconnection within regions.....	125
FIGURE 68 - Region on sketch boundary.....	128
FIGURE 69 - RBS triangulation.....	129
FIGURE 70 - Edge intersections.....	130
FIGURE 71 - Corridor.....	131
FIGURE 72 - Corridor internal point.....	131
FIGURE 73 - Link/Edge intersection.....	133
FIGURE 74 - Replacing a link with links.....	134
FIGURE 75 - A branch path of.....	135
FIGURE 76 - Invalid attachment.....	136
FIGURE 77 - Attachment operator.....	137
FIGURE 78 - Invalidating an attachment.....	138
FIGURE 79 - The Triangle problem.....	141
FIGURE 80 - The ROAR operator.....	142

## LIST OF GRAPHS

GRAPH 1 - Number of vias vs. beta .....	77
GRAPH 2 - Wire length vs. beta .....	77
GRAPH 3 - Comparison of net decomposition methods.....	81
GRAPH 4 - Actual cost vs. number of candidate vias. ....	83
GRAPH 5 - Wire length and via count vs. number of candidate vias. ....	84
GRAPH 6 - Wiring length vs. ordering method .....	106

PREVIEW

## LIST OF TABLES

TABLE 1 - Benchmark of SURF vs. SLICE .....	74
TABLE 2 - SURF improvement over SLICE .....	75
TABLE 3 - Number of vias vs. beta.....	76
TABLE 4 - Wire length vs. beta.....	76
TABLE 5 - Detour overestimation vs. beta.....	79
TABLE 6 - Comparison of 2-Net decomposition method. ....	80
TABLE 7 - Wire length vs. number of candidate vias.....	82
TABLE 8 - Via count vs. number of candidate vias. ....	83
TABLE 9 - Effect of the routing metric on the routing results. ....	85
TABLE 10 - LAA scalability.....	86
TABLE 11 - Routing with various orders of the 2-Net.....	105
TABLE 12 - Path actual length / length.....	139
TABLE 13 - ROAR optimization .....	143

PREVIEW

## 1 INTRODUCTION

### 1.1 General

Currently available routers are running out of steam when it comes to meeting the challenges presented by today's VLSI designs and packaging technologies. Multi-Chip Modules (MCM) [1] for example may consist of 60 or more layers and a large number of terminals and nets, and therefore the router must be able to handle large designs efficiently in both time and space. SURF is a routing system developed at University of California Santa Cruz that was designed to answer these challenges. The subject of this research is a rubber-band based topological local router that was developed as part of the SURF project. SURF is the first rubber-band based router ever reported and it has many advantages compared to existing geometrical routers. Its flexible representation of the interconnect enables it to achieve a high completion rate. It handles large designs efficiently and the rubber-band representation of the interconnect provides a powerful environment for manual editing.

### 1.2 Organization of This Thesis

In the rest of this chapter we present an overview of previous works related to our research and a discussion of our original contribution. In Chapter 2 we present an overview of topological representation of interconnect and an overview of SURF. Chapters 3, 4, 5 and 6 cover four aspects of our local router: layer-assignment, net ordering, topological net embedding, and the re-route optimizer respectively. Chapter 5 also includes a formulation of the rubber-band sketch. The formulation is based on mathematical terms from geometry and real analysis. The chapters of this thesis were designed to be independent of each other such that a reader can have a good understanding of each one without reading the previous ones. As a result, some degree of redundancy does exist, especially in the introduction of each chapter where the general terms and context are defined. Finally, Chapter 7 contains a conclusion and a discussion of future research.

### 1.3 Previous Works

A common method of handling large designs is to perform the routing in two phases, *global* and *local* routing. The global routing partitions the routing problem into a set of smaller sub-problems

which are then solved independently by the local router. Each of the sub-problems can be either a *channel* with the terminals on its boundary, or can be *channelless* and have terminals and features inside the routing area as well. Channel routing is more appropriate for Standard or Macro Cells where the routing is done in the spaces between the components, while channelless routing fits better MCM and modern VLSI designs. This dissertation describes a multi-layer, channelless, local router that uses rubber-band topological representation of the interconnect.

The known methods of routing can be classified into the two main categories of *geometric* and *topological* routing. Geometric routers determine the exact geometrical paths of the wires while topological routers use a more abstract representation of the interconnect and determine only the topology of the wires but not their exact paths. Geometric routing is by far the most common routing method used today. Some of the geometric routers mentioned in the literature are maze routers [51] [32] [19] that represent the wires as paths along a two or three dimensional grid, line-probe routers [21] [50] that use a gridless representation, and SLICE [29] which is an efficient plane-sweep router.

Since topological representation is more abstract than geometrical representation, the interconnect is easier to manipulate, and a topological router can avoid making too detailed decisions in early stages of the routing. When the topological routing is completed, it is checked to comply with the spacing requirements and is then transformed to exact geometrical layout.

Various aspects of topological routing have been discussed in the literature including routability check, topological representation, compaction and dynamic updating. The following is a discussion of previous works related to these areas.

The first problem related to topological routing mentioned in the literature is that of *routability test* defined in [54] [56]. This problem is also called *Detailed Routing given an Homotopy* (DRH) and is defined as follows: given a specification of the terminals and modules, and given a homotopy (a rough planar sketch of the wires), is there a geometrical layout that conforms to the homotopy, spacing requirements, and wiring rules? Cole and Siegal [5] proposed a polynomial-time algorithm that finds a complying geometric layout, and showed that such a layout exists if and



only if every *cut* is safe. Leiserson and Maley [35] [44] showed that it is sufficient to check a smaller set of *critical* cuts, and that the test can be done on the *Rubber-Band Equivalent* (RBE) in which wires are as short as possible and zero spacing is allowed (while maintaining the topology of the wires). Kong [31] used the idea of RBE and the concept of *spokes*, introduced by Leiserson and Maley [35], to perform a constructive incremental routability check that results in a complying geometric layout, if one exists. The works of Leiserson and Maley [35][44], and of Kong [31], assume no constraint on the wiring model. Maley, in a later work [46], showed that in the special case of polygonal grid-based wiring rules, the routability test can be performed in  $O(n \log n)$  time.

The topological representations mentioned in the literature fall roughly into two categories: Triangulation Crossing Sketches (TCS) and Rubber-Band Sketches (RBS). TCS use a triangulation of the routing area and capture the crossing of triangulation edges by nets' wires. RBS on the other hand represents the interconnect as flexible rubber-bands that bend and stretch to form the shortest possible connection of a given topology. The RBS representation was originally proposed by Rivest (see acknowledgment in [35]). Leiserson and Maley formalized this concept and called it the *Rubber-Band Equivalent* (RBE) [35] [44] [45]. As far as we are aware of, the only rubber-band based router ever reported is SURF (Santa Cruz ULSI Routing Framework) [9] that is the framework of this research. SURF is based on ideas and results presented by Leiserson and Maley [35] [44] and it covers various aspects of rubber-band based topological routing including: global routing [63], multi-layer topological local routing (this research), representation and manipulation of RBS [8], optimization of an RBS (ROAR optimization, part of this research), testing for routability in an RBS [10] [31], and transformation of an RBS to geometric layout [62]. An overview of SURF is presented later.

The flexible nature of topological representation makes it easier to move features while maintaining the connectivity and the topology of the wires. Murata and Kajitani [52] [53], and Su [65] proposed algorithms for moving objects in TSC and RBS respectively. The flexibility of the topological sketches has also been used to solve the *homotopic compaction* problem. That is, to find the smallest (area wise) layout that conforms to given homotopy, wire rules, and spacing

requirements. Maley [43] proposed a one dimensional compaction algorithm. His work was later extended by [16] [45] [49] [66] [39] [13] and [25].

Several approaches for routing of multi-layer designs were discussed in the literature, including 3-dimensional (3D) search, *maximal-layer* routing, and *layer-assignment*. The 3D search is an extension of single-layer search algorithms such as maze-runner and line-probing that considers layer crossings. In the maximal-layer approach, which is used for example by SLICE [29], the layers are routed sequentially, trying to fit on each layer as much as possible of the missing interconnect. The layer-assignment approach (survey in [26]) considers simultaneously all nets and layers, and assigns nets or partial nets to the available layers.

Known methods for layer-assignment are usually classified as *constrained* or *unconstrained* (also called *topological*). In a constrained layer-assignment, the geometric paths of the wires have already been determined prior to the assignment and the assignment step only determines on what layers they will be routed. In an unconstrained layer-assignment on the other hand, the actual wire paths are not specified to the layer-assignment. In this research we adopted the unconstrained approach because it does not commit in early stages of the routing to specific wire routes, and therefore it is more likely to take advantage of the flexibility of the underlying topological representation.

The concept of unconstrained layer-assignment was introduced by Hsu [24] and since then has received a fair amount of attention in the literature. Hsu [24] and Marek-Sadowska [47] dealt with two-layer problems where all the terminals are on the boundary of the routing area and the via count is to be minimized, while guaranteeing planarity on each layer. This work was extended by Haruyzama, Wong, and Fussell [20] to support multi-terminal nets. Sarrafzadeh and Lee [59] showed that the via minimization problem is equivalent to the *maximum 2-independent set* problem in a circle graph which is NP-complete. Pinter [54] proposed an algorithm that supports terminals inside the routing area. It is based on a partitioning of the routing area into disjoint regions called *clusters* such that the wires in each cluster can be routed on two layers without crossing each other (i.e. each layer is planar). The algorithm then sub-optimally reduces the

problem into finding of a maximal cut in a planar graph, and this problem has a polynomial time solution. The advances in VLSI and MCM which may utilize a significantly larger number of layers resulted in published works that deal with multi-layers. Kiao, Lee, and Sarrafzadeh [36] presented a polynomial algorithm for finding a maximal-weighted planar subset of multi-terminal nets in a switchbox. This algorithm can be used to sequentially ‘peel’ maximal sets of nets for each layer, assuming that all the terminals are on the boundary. If terminals do exist within the routing area, the algorithm has exponential complexity. Other works that restricted terminals to exist on the boundary include [6], [64], and [25]. Cho et al [4] proposed a layer-assignment algorithm that handles terminals within the routing area and considers cross-talk, via count, and wire-length. This algorithm which was developed independently, uses a net pair-wise cost function similar to the one developed in our research (first published in [11]). His algorithm however, assigns complete nets to layers and does not consider insertion of vias as our algorithm does.

Our layer-assignment algorithm (called LAA) is to our best knowledge, the first successful attempt to perform topological layer-assignment with simultaneous minimization of via-count and wire-length. It scales well with an increasing number of layers, it supports multi-terminal nets, and it handles terminals inside the routing area. The LAA models the layer-assignment as an optimization problem with a cost function that considers via-count, and wire length. The cost function can be extended to consider other routing goals such as preferred layers, and cross-talk. The LAA supports rectilinear, octilinear, and any-angle wiring models, and it takes advantage of the flexibility of the underlying RBS.

The LAA decomposes the multi-layer routing problem into a set of independent single-layer sub-problems. Each single-layer problem specifies the terminals (which may include vias introduced by the layer-assignment) and nets or partial nets to be connected on that layer. The nets of each single-layer problem are guaranteed by the LAA to be planar (i.e. they can be topologically routed with no intersections). This kind of routing problem is often referred to as *single-layer* or *planar* routing. Known methods for single-layer routing fall roughly into two categories of *sequential* and *simultaneous* routing. In the sequential approach, nets are routed one-at-a-time, typically on a least-cost path, considering the previously routed nets. The least-cost path can be determined for

example by a maze-runner or a line-probing algorithm. A draw-back of sequential routing is its sensitivity to the order in which the nets are routed. In some cases for example, the optimal solution cannot be achieved by *any* order. Liao and Sarrafzadeh [38] addressed this problem with an algorithm that applies the concept of global routing. It performs the least-cost path search on a rough grid such that it makes decisions with a more global view. In simultaneous routing on the other hand, all nets are routed at the same time and therefore no net gets higher priority than others. This approach is used for example in SLICE [29] where it performs a single-layer routing in a single plan sweep. SLICE's approach is insensitive to the net order but is less likely to find more complex paths and to efficiently handle 'keep-out' areas due to its one directional sweep and short look-ahead.

In this research, we have focused on sequential routing for two reasons. First, the planar routing in SURF is done in a topological sketch and therefore is less sensitive to net ordering. Second, the planar routing is performed on relatively small regions called *bins* defined by the global routing and therefore, any 'bad' decision made while determining the path of a net will have only local and limited affect. Our router uses a net ordering algorithm that estimates the final wire-length for a given order of nets, and a rerouting post-processor that searches for an alternative topology with shorter wiring. The ordering and the planar routing algorithms considers the wire length (Euclidean, rectilinear or octilinear) and are guaranteed to find a planar solution.

Another known concept which is used by our single-layer router is *rip-up and reroute* (RUR). RUR is typically used to improve the completion rate and is performed by deleting some nets from the sketch, routing a residual net, and then rerouting the deleted nets. A variant of this approach, that can be used by routers that are able to represent violations such as cross-over, is to reduce the number of conflicts between already routed nets. The main difficulty in performing RUR is to determine which nets to delete and in what order to reroute them. Several works including [58] [61] [40] [7] [27] and [57] address this issue. The RUR approach is frequently used in sequential routers to compensate for the sensitivity to net ordering. This problem is less significant in topological routing. For example, the Puzzle Problem mentioned in [27] requires a RUR approach to be solved optimally by a geometrical router, but is easily solved optimally using a topological

router because the wires are flexible and are pushed away to create clearance for new wires (Figure 1). Our router uses a RUR optimizer called ROAR that accepts a planar RBS and is guaranteed to result in a planar RBS with the same or lower wire length. To determine which wires to remove and in what order to reroute them, the algorithm uses the *attachment* relation between branches and terminals in the RBS. The ROAR optimizer is described later in this thesis.

#### 1.4 Contribution of this work

Our local router is the first ever published router that is based on rubber-band topological representation. By taking advantage of the flexibility of the RBS, it can efficiently achieve high quality layouts and to handle a wide range of routing problems. Our local router performs the routing in four steps: layer-assignment, net-ordering, net-embedding, and optimization. Our contribution in each of these areas is outlined below.

Our unconstrained layer-assignment algorithm (LAA) uses a new approach in which the output of the layer-assignment is unconstrained as well, that is, it does not specify the actual paths of the wires but only what sub-nets are to be connected on what layers. This approach leaves more freedom to the single-layer router and enables it to make more informed decisions. The layer-assignment is formulated as an optimization problem with a cost function to be minimized. The cost is based on an estimation of the final wire-length and supports user specification of the desired balance between the conflicting goals of minimizing wire-length and via-count. The cost function uses a continuous metric of the conflict between nets as opposed to the go-no-go approach previously used. This enables the LAA to better utilize the routing resources by allowing nets with low conflict to be assigned to the same layer. The cost function can be extended to support other goals such as cross-talk minimization and constrained-layers. The cost function has a finite value if and only if the assignment is planar and this enables the LAA to guarantee that the nets or partial nets assigned to each layer can be routed on that layer without crossing each other. The LAA handles multi-terminal nets and can break a net into multiple layers if this results in a more desirable solution. The LAA handles terminals on the boundary and within the routing area and the terminals are not restricted to be on a grid. If terminals do exist on the boundary, the LAA can consider specification of layer-assignment done previously on the other side of the boundary. This

makes the LAA suitable for routing systems like SURF that perform global routing before the layer-assignment. The LAA is not restricted to a one-layer-one-direction routing style and supports Euclidean, rectilinear and octilinear wiring rules. It uses an algorithm (called 2NAA) that optimally assigns a single two-terminal net, given a sequence of candidate locations for vias and the assignments of previously assigned nets. We have compared our router to SLICE and the experimental results show that it achieves better layouts, even when it is restricted to rectilinear wiring rules.

Similar to the layer-assignment, our net-ordering algorithm is based on a formulation of net ordering as an optimization problem with cost function to be minimized. The cost function estimates the wire-length based on a pair-wise conflict between nets. The planarity of the order is handled by the Planarity Operator that, given a net ordering, finds an order of the same cost that is guaranteed to be routed with no net crossings. Experimental results show that our net ordering algorithm reduces the wire length by 5% and 30% in average compared to the 'shorter-net-first' and 'longer-net-first' approaches respectively.

Our net-embedding algorithm performs a shortest path search in the RBS and is the first rubber-band shortest path algorithm ever reported. The search algorithm is guaranteed to find a shortest planar path in  $O((T^2 + S)\log(T + S))$  time where  $T$  and  $S$  are the number of terminals and number of wire segments respectively in the RBS. The algorithm can be modified to use a smaller search graph and to find a planar path that is likely to be short in  $O((T + S)\log(T + S))$  time. The shortest-path algorithm is also used in our rip-up and reroute algorithm (called ROAR) that is used to compensate for dependency of the generated layout on the net order. The ROAR algorithm uses the 'attachment' relationship between wires and terminals that is unique to the RBS.

We also present a formulation that maps the concept of RBS into known mathematical terms from geometric and real analysis. This clarifies the definition of the RBS as being both planar and of zero spacing. The formulation is used to prove the correctness of the shortest planar path algorithm using exact mathematical tools. To our best knowledge, this is the first time an exact analysis of RBS is published.

## 2 OVERVIEW

In this chapter we present an overview of topological representation of the interconnect and an overview of the SURF routing system. This introduces the terms commonly used in topological routing and explains the context in which the proposed local-router is used.

### 2.1 Topological Representation and Routing

A topological sketch represents the embedding of nets in an abstract form. It contains the general relationship of the nets and terminals but ignores details such as the exact geometrical paths of the nets. This 'partial' representation avoids making too detailed decisions in early stages of the routing (Figure 1) and provides greater flexibility in manipulating the sketch. When the topological routing is completed, it is then transformed into a geometrical layout with exact location of the wires.

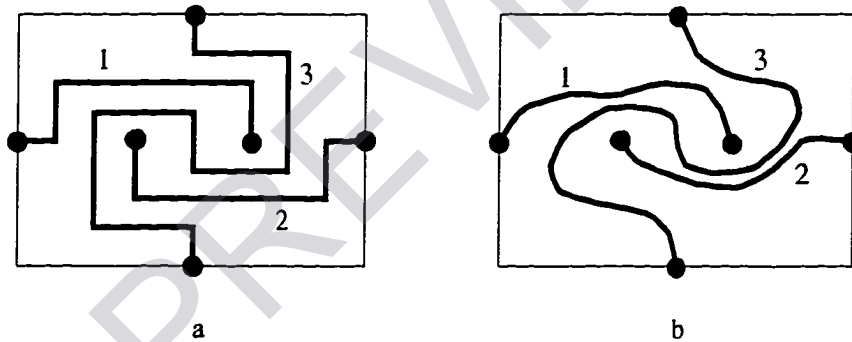


FIGURE 1 - The Puzzle problem. (a) shows an optimal solution to the Puzzle example proposed in [27]. The example has three nets that are routed in the order 1, 2, 3. A typical geometrical router that routes the nets sequentially on a shortest path will fail routing net 3 because there is not enough clearance between the previously routed nets 1, 2 and the two internal terminals. A topological router on the other hand (b) can easily insert net 3 since no commitment was done on the exact geometrical paths of nets 1, 2.

In the rest of this section we present an overview of topological representation of the interconnect. The discussion is limited to single-layer sketches. Multi-layer sketches can be constructed using a set of single-layer sketches and some specification of the layer crossings between terminals of



adjacent layers. For clarity, we deal here with a simplified model of the routing that includes only terminals and nets. Obstacle or 'keep-out' areas can be handled in a similar way.

### 2.1.1 Sketches and Topological Classes

A *geometric sketch* (*sketch* in short) represents the exact paths of the interconnect. It includes a set of *terminals*, each has a unique point inside or on the boundary of the routing area, and *branches*, each a connection between a pair of terminals. The terminals and the branches have zero width and they represent the centers of the pads and the center-lines of the wires respectively. A maximal set of terminals interconnected by branches is called a *net* or *component*. The branches and terminals of a component are assumed to have no cycles (i.e. the components are trees). A branch is represented by a path connecting its two end-terminals. A branch path is a parametric line  $P(t) = (x_t, y_t)$ , where  $t \in [0..1]$ , and  $(x_t, y_t)$  is a point on the plane, such that the two end-points of the path,  $P(0), P(1)$  are the locations of the end-terminals of the branch. The decision which of the two end-points of a branch  $P(0)$  represents is arbitrary. A branch path is planar, it does not cross itself, terminals, or other paths, except for possibly at its end-points  $P(0), P(1)$ , and is strictly contained inside the routing area. The function  $P(t)$  is continuous and piece-wise differentiable.

Let  $A, B$  be two sketches with identical sets of terminals and same branch connectivity (but not necessarily the same paths), and let  $\langle A_i(t) \rangle, \langle B_i(t) \rangle$  be the vectors of branch-paths of  $A, B$  respectively. We say that sketch  $A$  is *homotopic* [35] to sketch  $B$ , or that  $A$  'has the same topology' as  $B$ , if there is a continuous transformation from the paths of  $A$  to the paths of  $B$ ,  $T(r) = \langle P_i(r, t) \rangle, r \in [0..1]$ , such that  $P_i(0, t) = A_i(t), P_i(1, t) = B_i(t), P_i(r, 0) = A_i(0), P_i(r, 1) = A_i(1)$ , and all the intermediate sketches, when  $0 < r < 1$ , are planar. Intuitively this represents a continuous co-deformation of the branches of sketch  $A$ , to achieve those of sketch  $B$ , while maintaining their end-points and their relative locations (Figure 2).