

Routing, Placement, and Partitioning

edited by

George W. Zobrist

TN47
R869

9561654

Routing, Placement, and Partitioning

edited by
George W. Zobrist
University of Missouri-Rolla



E9561654



Ablex Publishing Corporation
Norwood, New Jersey

Copyright © 1994 by Ablex Publishing Corporation

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, microfilming, recording or otherwise, without permission of the publisher.

Printed in the United States of America

Library of Congress Cataloging-in-Publication Data

Routing, placement, and partitioning / edited by George W. Zobrist

p. cm. — (Computer engineering and computer science)

Includes bibliographical references and index.

ISBN 0-89391-784-2

1. Integrated circuits—Very large scale integration—Design and construction—Data processing. 2. Computer-aided design.

I. Zobrist, George W. (George Winston), 1934– . II. Series.

TK7874.R677 1992

621.39'5—dc20

92-15441

CIP

Ablex Publishing Corporation
355 Chestnut Street
Norwood, New Jersey 07648

Routing, Placement, and Partitioning

VLSI DESIGN AUTOMATION SERIES

George W. Zobrist, Series and Volume Editor

Routing, Placement, and Partitioning

VLSI Fault Modeling and Testing Techniques

in preparation

Digital Logic Analysis and Design

VLSI Design Environments and Silicon Compilation

Editor's Preface

This volume of the VLSI Design Automation Series is concerned with state-of-the-art developments in VLSI routing, placement, and partitioning. Contributions from researchers and practitioners are included.

With the rapid advances in VLSI technology, the routing problem has come to assume a position of significance. The routing problem is one of the most widely investigated problems in VLSI design automation. Many VLSI design methodologies are based on the library cell approach, thus routing based on cell-based design approaches is important. Another approach, which is based on slicing topology, uses a gridless router to obtain high-quality results. Other aspects of layout automation are the placement problem and the partition problem; the placement problem places modules on a given board and the partition problem subdivides a very large system into smaller subunits or modules. In the routing problem prior to detailed routing, the routing area is usually divided into smaller routing regions called channels. The channels are physically routed by a channel router. Another objective in VLSI physical design is to obtain a valid layout of a given building block of rectangular shape. The first step towards this goal is to obtain a floorplan, which is a partitioning of the floor rectangle into smaller ones. A survey of existing methods to solve the problem of minimizing the area of a layout when its floorplan and the possible dimensions of the building blocks are given, utilizing both slicing and nonslicing algorithms, is also presented.

The editor wishes to thank the contributing authors for making available the information contained in this book.

George W. Zobrist
University of Missouri—Rolla

Contents

Editor's Preface	vii
1 Routing Techniques <i>Mehdi R. Zargham</i>	1
2 A Gridless Routing System for Macro-Cell Design <i>Patrick Groeneveld and Hong Cai</i>	49
3 Placement <i>Wai-Kai Chen and Kai Wang</i>	96
4 VLSI Placement <i>Chen-Xiong Zhang</i>	121
5 Partitioning <i>Youssef Saab and Vasant Rao</i>	182
6 Channel Definition in VLSI Building-Block Layout <i>Hong Cai and Ralph H.J.M. Otten</i>	215
7 Optimal Determination of Block Dimensions in General Floorplans <i>Shmuel Wimer, Israel Koren, and Israel Cederbaum</i>	237
Author Index	287
Subject Index	292

1

Routing Techniques

Mehdi R. Zargham

Department of Computer Science
Southern Illinois University at Carbondale
Carbondale, IL

1. INTRODUCTION

With the rapid advances in VLSI technology, the routing problem has come to assume a position of significance. High-performance and high density VLSI layouts have become a necessary requirement. The field of design automation has attracted a great deal of attention, especially since it promises to satisfy many of these requirements. By automating the design process to a greater degree, the gap between the conception of an idea and its working prototype has been significantly reduced.

The routing problem is one of the most widely investigated problems in VLSI design automation. Today, many VLSI design methodologies are based on the library cell approach. These cells are arranged on a plane and interconnected by wires. In this type of design methodology, the routing of wires usually is done after the arrangement of cells. The quality of routing is often measured by a number of routing-metrics, namely: the area occupied by the routed wires, the length of interconnection wires, the time taken to complete the routing, and the number of vias. (A via is a feed-through hole or a contact where wire segments in different layers are connected.) By minimizing the length of interconnection wires and the number of vias, we not only increase the reliability and performance of circuits, but also decrease their manufacturing cost.

This chapter represents a survey of routing techniques in cell-based design approach. It provides a review of existing methods, and an in-depth discussion of

specific state-of-the-art algorithms and expert systems for detailed routing problems. However, overall concepts are emphasized rather than implementation details.

This chapter is organized as follows: Section 2 contains some basic definitions and terminology that are used throughout the chapter; Section 3 classifies various types of routing techniques and provides an overview of main steps in automatic routing of today's VLSI circuits; and Section 4 describes a method for two-layer and three-layer channel routing. The main objective of this section is to expose some of the main issues that are relevant in channel routing. Knowledge of these issues facilitates discussing the advantages and drawbacks of some of the well known routing techniques. An in-depth description of these techniques is provided later in Sections 5, 6, and 7. References to the original descriptions and proposed modifications of the techniques are given for readers interested in the details of a particular technique. Finally, concluding remarks and recommendations are presented in Section 8.

2. DEFINITIONS

The terms used throughout this chapter are defined as follows:

- Channel:** A rectangular region with terminals at regular intervals on two opposite sides (the *top* and *bottom*).
- Switchbox:** A rectangular region with terminals at regular intervals on four sides (the *top*, *bottom*, *right*, and *left*).
- Layer:** The wiring surface used for interconnection.
- Net:** A net is specified by a collection of terminals. The goal of a channel router is to connect all the terminals of each net. Usually, vertical wires extending into the channel from the top and bottom terminals are selected from a class of layers, called vertical layers. The horizontal wires which connect these wires are selected from another class of layers, called horizontal layers. The horizontal wires are called **trunks** or **horizontal segments**, and vertical wires connecting the trunks to the top and bottom of the channel are called **branches**.
- Segment or subnet:** A segment is specified by two terminals and a width. Segments implement nets, i.e., a net is constructed of one or more segments.
- Track or row:** The symbolic routing area in the horizontal direction of length equal to the length of the channel. The tracks are used for placing the trunk(s) of each net.
- Column:** The symbolic routing area defined by a top terminal and

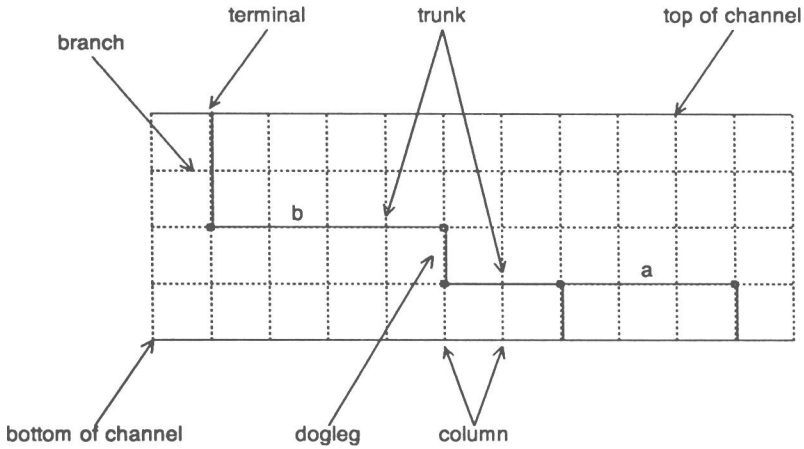


Figure 1.1. Routing Terminology.

its opposite bottom terminal. The columns are used for placing the branches of each net.

Figure 1.1 illustrates the above terminology in a channel with one net which consists of two segments, *a* and *b*.

3. ROUTING TECHNIQUES

To provide a perspective on the various routing algorithms and to aid in organizing the later sections, a routing taxonomy is presented in Figure 1.2. This routing taxonomy is based on the assumption that the entire routing surface is divided into a set of rectangular routing areas called channels and switchboxes as shown in Figure 1.3. Decomposition of a routing problem in this manner makes the automatic routing of today's VLSI circuits possible. At the first level of the hierarchy are the global, detailed, and specialized routers. The global router considers the overall routing region in order to distribute the nets over the channels and/or switchboxes based on their capacities while keeping the length of each net as short as possible. For every channel (switchbox) that a net passes through, the net's id is placed on the sides of the channel (switchbox). Once the terminals of each channel (switchbox) are determined, the detailed router will find a location for the wire segments of each net within the channel (switchbox). The specialized router is designed to solve a specific problem like routing of power and ground wires. These wires require special attention for two reasons: (1) they are usually routed in one layer in order to reduce the parasitic capacitance of contacts, and (2) they are usually wider than other wires (signal and data) since they carry more current.

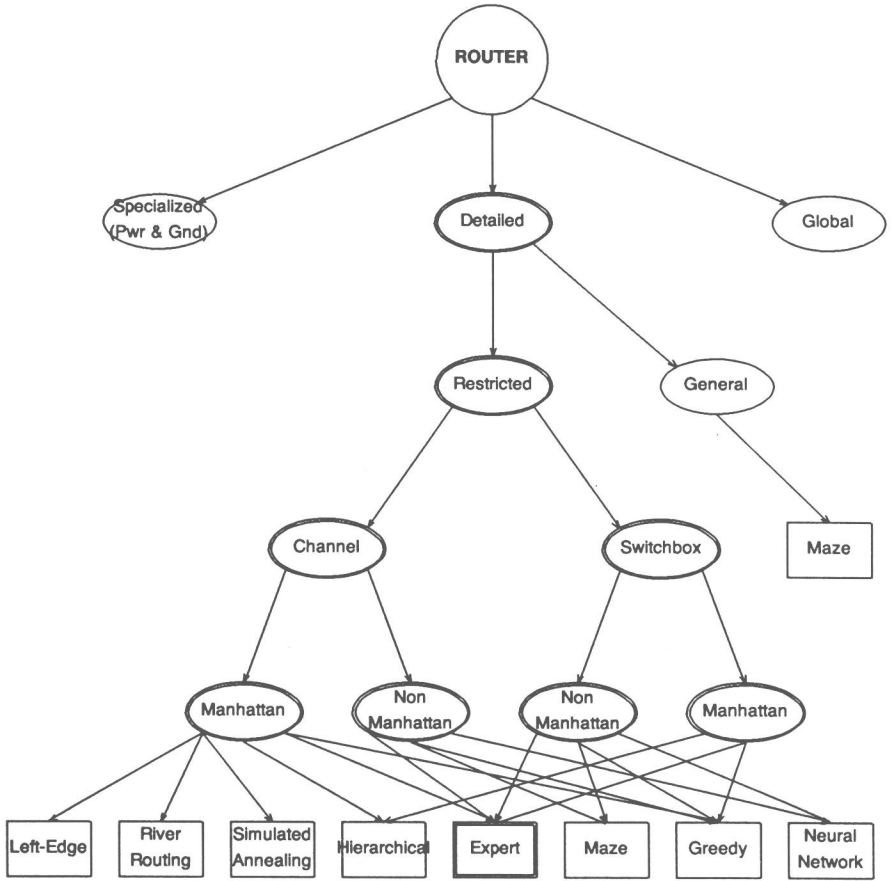


Figure 1.2. Router Taxonomy.

Detailed routers further divide into general-purpose and restricted routers. The general-purpose routers impose very few constraints on the routing problem and operate on a single connection at a time. Since these routers work on the entire design in a serial fashion, the size of the problems they can attempt is limited. On the other hand, the restricted routers require some constraints on the routing problem, such as empty rectangular areas with all of the pins on the periphery. Because of their limited scope, these routers can do a better job of modeling the contention of nets for the routing resources and therefore can be viewed as routing the nets in parallel.

Furthermore, the restricted routers divide into channel and switchbox routers. In order to reduce the complexity of routing, this type of router often uses a rectangular grid on which trunks (horizontal wire segments) and branches (verti-

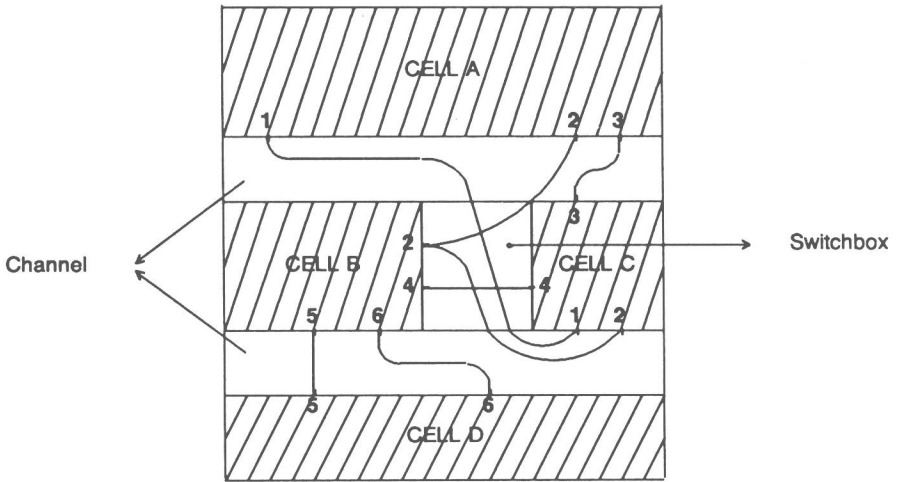


Figure 1.3. Channel and Switchbox.

cal wire segments) are placed on different layers. In other words, the layers supported by the technology are divided into two groups of layers, horizontal and vertical. This is known as Manhattan model. On the other hand, in a non-Manhattan model, the assignment of layer to vertical or horizontal direction is not enforced. Given the freedom of direction, assignment to layers reduces the channel width and vias in many cases; the latter model usually produces a better result than the former.

At the lowest level of the hierarchy in Figure 1.2, different techniques are presented. In general these techniques can be grouped into three different approaches; (1) algorithms, (2) expert systems, and (3) neural networks. Each of these approaches is discussed in later sections.

4. TWO- AND THREE-LAYER CHANNEL ROUTING

There are numerous techniques to solve the channel routing problem. Although these techniques differ from each other in basic ideas, many of them use common terminology. To understand this terminology and the restrictions in channel routing, this section describes a method for two-layer and three-layer routing.

4.1. Two-Layer Channel Routing

Even though recent advances in fabrication technology allow more than two layers for routing, it is still important to know methods for two-layer channel

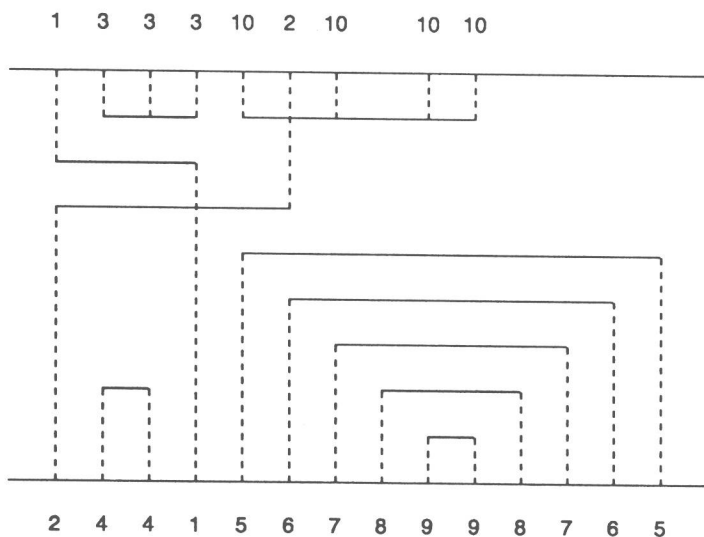


Figure 1.4. Two-Layer Channel Routing.

routing. This is because it is possible to extend some of these methods to multi-layer routing. In addition, these methods introduce some techniques which are basic to channel routing.

Let two layers be available for routing where the horizontal wires are in one layer and vertical wires in another. Each net consists of one or more subnets and each subnet consists of two terminals. Figure 1.4 shows a two-layer channel-routing problem.

Obviously, by the above assumption, the horizontal segments of a set of subnets that intersect a given column cannot overlap. In general, there could be a set of adjacent columns where the subnets that intersect them cannot overlap. To each such set, a sequential number which represents a *zone* is assigned. Figure 1.5 shows the zone representation corresponding to the channel routing problem of Figure 1.4. The zone representation shows that nets belonging to a zone cannot have overlapping horizontal segments and should be assigned to different tracks. For example, the set of subnets intersecting column 2 include every set of subnets in columns 1, 3, and 4. The first zone includes columns 1, 2, 3, and 4. A second zone will include columns 5 and 6, and zone 3 will contain columns 7 through 14.

One of the lower bounds in the number of tracks needed to solve a two-layer channel routing problem is determined by a zone with a maximum number of nets in a given column. The number of nets is called the *maximum density* of the channel, and the corresponding column is the maximum density column [1, 2]. For example, the zone representation in Figure 1.5 specifies that columns 9 and

Net and Zone Representations		
Column	Net	Zone
1	1,2	1
2	1,2,3,4	
3	1,2,3,4	
4	1,2,3	
5	2,5,10	2
6	2,5,6,10	
7	5,6,7,10	3
8	5,6,7,8,10	
9	5,6,7,8,9,10	
10	5,6,7,8,9,10	
11	5,6,7,8	
12	5,6,7	
13	5,6	
14	5	

Figure 1.5. Zone representation.

10 are maximum density columns and the given channel routing problem will have at least six different tracks.

Vertical overlaps should also be avoided because vertical segments are routed in one layer. For example, in Figure 1.4, net 1 should stay above net 2 in order to avoid electrical connections between them. This restriction is known as vertical constraint and is represented by a directed graph [1, 2]. A vertical constraint graph corresponding to the channel routing problem of Figure 1.4 is shown in Figure 1.6. Note that the vertical constraint is transitive. For example, the channel routing problem of Figure 1.4 shows that net 3 should stay above net 1 and net 1 should stay above net 2; therefore, net 3 should also stay above net 2, otherwise a vertical constraint loop will be created. (In the following algorithm a channel routing problem with a vertical constraint loop cannot be handled; therefore, vertical constraint loop creation should be avoided.)

After finding the maximum density column, nets which have intersections with this column will be assigned to different tracks. In this section, one assumes that nets have been sorted into ascending order of their left end-point x-coordinates. At this point, the channel has been divided into two regions; one to the right of the maximum density column, and another region to the left. Nets have to be grouped and merged with the existing tracks in order to reduce the number of tracks. The strategy incorporated into algorithms 1 and 2 of [2] starts grouping and merging of nets to the right of the maximum density column. However, the algorithm suggested by [1] considers both sides of the maximum density column at the same time.

Next, the algorithm enters an iterative loop in which a subnet is considered for

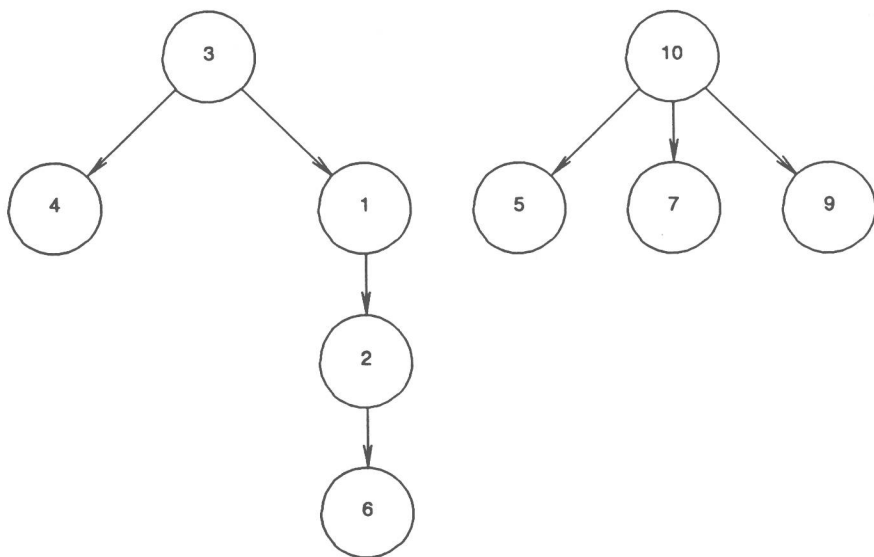


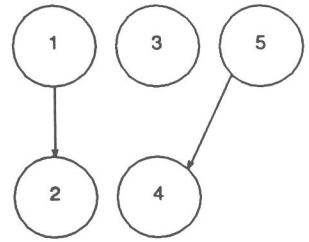
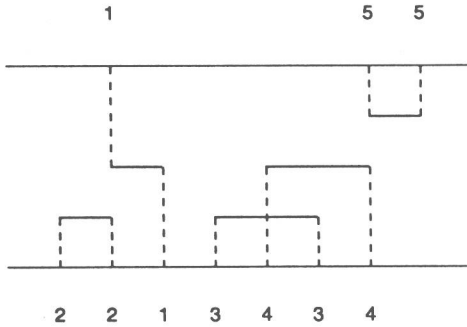
Figure 1.6. Initial Vertical Constraint Graph.

assignment; every other subnet which is considered for assignment in this iteration should have either vertical or horizontal constraint with each subnet that has already been selected for assignment. Otherwise, the subnet will not be considered for assignment at this iteration.

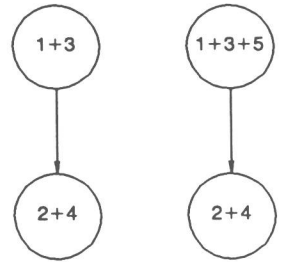
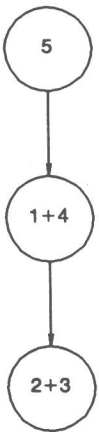
The set of subnets at each iteration obviously cannot be merged together because they have either horizontal or vertical constraint. However, the set of subnets can be merged with the existing tracks if they do not have any constraints. If a subnet which has been selected for assignment at this iteration cannot be merged with any of the existing tracks, it will be assigned to a new track.

Let the critical path length of a node v in the vertical constraint graph, denoted as $CPL(v)$, be defined as the length of the longest path through v . The maximum length of the vertical constraint graph (VCG), denoted by $CPL(VCG)$, is the maximum length of all nodes in the graph [1]. Note that $CPL(VCG)$ is another lower bound for the number of tracks needed in order to realize a given channel routing problem. (This is because each subnet consists of a single trunk.) In other words, the number of tracks needed in a channel is greater than or equal to $\text{Minimum}\{\text{maximum density}, CPL(VCG)\}$.

Merging a subnet with an existing track has its drawbacks since the vertical constraint graph tends to grow, or the maximum length of the vertical constraint graph increases as a result of merging. This situation has been depicted in Figure 1.7. Initially, subnets 1 and 2 are assigned to different tracks. Subnets 3 and 4 are considered for assignment in the first iteration. Subnet 5 cannot be considered in



(a). Two-Layer Channel Routing (b). Initial Vertical Constraint Graph



(c). Final Vertical Constraint Graph (d). Final Vertical Constraint Graph

Figure 1.7. Merging a subnet.

this iteration because even though it has vertical constraint with subnet 4, it does not have any constraint with subnet 3. Therefore, it will be considered in the next iteration. If subnets 1 and 4, 2 and 3, are merged at this iteration it is necessary to assign subnet 5 to a new track in the next iteration because subnet 5 should stay above the track containing subnet 4, according to the initial vertical constraint graph. This situation is shown in Figure 1.7(c). Note the maximum length of the initial vertical constraint graph was 2, and after merging, it has been increased to 3, which means that the router requires three tracks in order to realize the

problem. However, if subnets 1 and 3, and 2 and 4 are merged at the first iteration, it would be possible to merge subnet 5 with the existing track containing subnets 1 and 3. This situation is shown in Figure 1.7(d). Note that the maximum length of the vertical constraint graph has not been increased and it is possible to realize the problem using two tracks.

The algorithm, therefore, should merge subnets and tracks so that the maximum length of the vertical constraint graph is minimized. This problem has been viewed as bipartite matching [1, 2]. Source nodes are the existing tracks to the left of the bipartite matching graph, and destination nodes are selected subnets ready for assignment at each iteration to the right side of the graph. There is an edge between a source node and a destination node, if and only if the selected subnet can be assigned to the track. In [1] a heuristic algorithm has been introduced by means of weights on the edges of the bipartite graph in order to choose a matching between a selected subnet and an existing track that minimizes the CPL of the vertical constraint graph.

The two-layer channel routing algorithm discussed can always provide a solution for the problem if there are no initial vertical constraint loops and the height of the channel is adjustable. The algorithm provides optimum solutions in terms of the number of tracks required to solve a channel routing problem. However, the efficiency of the algorithm can be further improved by introducing another layer into the layout design. The required changes to the two-layer channel routing algorithm and the impact on efficiency of the algorithm are discussed in the next section.

4.2. Three-Layer Channel Routing

Multi-layer channel routing is also possible by using the above two-layer channel-routing algorithm. A three-layer channel-routing algorithm, which is the main focus of this section, has been introduced in [2]. There are three layers available for routing, therefore the router can realize the problem in two ways. The first way is to route the horizontal segment of each net in the first layer. The vertical segments that are connected to the top of the channel are routed in the second layer, while vertical segments that are connected to the bottom of the channel are routed in the third layer. This method is called Vertical Horizontal Vertical (VHV). A second way is to divide the horizontal segments in half. One half will be routed in the first layer, the second half will be routed in the second layer, while the vertical segments will be routed in the third layer. This latter method is called Horizontal Vertical Horizontal (HVH). Each method is described in the following paragraphs.

4.2.1. Vertical Horizontal Vertical (VHV)

Consider the channel-routing problem of Figure 1.8(a) which is routed in two layers and whose initial vertical constraint graph is shown in Figure 1.8(b). The