1986

An automated routing method for VLSI with three interconnection layers

Chong Ho Lee
Iowa State University

Recommended Citation

Lee, Chong Ho, "An automated routing method for VLSI with three interconnection layers " (1986). *Retrospective Theses and Dissertations*. 8092.
https://lib.dr.iastate.edu/rtd/8092

U·M·I Dissertation Information Service

University Microfilms International
A Bell & Howell Information Company
300 N. Zeeb Road, Ann Arbor, Michigan 48106

Lee, Chong Ho

AN AUTOMATED ROUTING METHOD FOR VLSI WITH THREE INTERCONNECTION LAYERS

Iowa State University PH.D. 1986

University
Microfilms
International 300 N. Zeeb Road, Ann Arbor, MI 48106

An automated routing method for VLSI with three interconnection layers

by

Chong Ho Lee

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Department: Electrical Engineering and
Computer Engineering
Major: Computer Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa

1986

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Since there has been continuing demand for scaling up integrated circuits, the layout design of ICs has become more complicated and more important than ever. As the number of gates per chip increases, the number of interconnections between chips decreases and the number of interconnections between gates within the chip increases. Moreover, it is found that the total length of interconnections on a chip actually increases faster than the increase of the number of gates [1]. In polycell LSI layouts, up to 80 % of the chip area is used for interconnections between the modules [2]. Thus, the routing area and the routability condition become dominant factors in integrated circuit(IC) layout design problems.

The increase in complexity and cost of layout design should bring in automated tools as part of the design process; practically, the automatic layout design methods used for ROM s, PLA s or gate arrays are employed already by many manufacturers. Design automation also is attractive in that the same design data bases used for interconnections can be used for other stages of IC design, such as logic simulation, speed analysis, and test pattern generation.

The prime objective of the layout problem is to minimize the total area required to locate the circuit components and the wiring between them while satisfying the layout design rules characterized by physical and electrical properties.

Layout problems are traditionally broken down into two phases: placement and routing. In such methodology, we first place pre-designed building-blocks of circuits on a chip and then route wires to connect necessary terminals.

1.1 Placement and routing

The goal of placement is to arrange the blocks in such a way that the interconnections can be routed and the overall chip area is minimal. Minimum wire length of all connections originally was another goal of placement but it gradually gave way to the criterion of a better routability condition because it caused crowded or unroutable channels. Placement methods usually assume that the blocks are rectangular and orthogonally oriented.

The placement stage of the problem typically has three objectives [2]:

- to avoid cyclic constraints between nets, otherwise the simple router will not work;
- to minimize the channel density;
- to minimize the span over which the maximum density occurs.

In the placement stage, we first arrange a compact placement of the cells ignoring the channel areas between cells. We assume that there is enough space between any two cells to allow as many wires to pass as required in the global routing stage [3]. This assumption requires the routing system to provide feedback so that the placement can be automatically adjusted as needed.

A different scheme in compacting the cell layout is to arrange cells to minimize the chip area while maintaining certain widths for channels between cells based on a prediction of routing requirements. The global routing stage adjusts the placement slightly to reduce the free area or to accommodate overcrowded wires on certain channels.

Placement methods can also be classified into two groups: bottom up and top down placement. Bottom up placement starts with one cell (seed) and other cells are clustered one by one according to a certain standard [4] [5] [6]. This method is sensitive to the selection of the seed and likely to lack in global point of view. Top down placement repeatedly partitions the given set of cells by the min-cut principle [7] [8] [9] [10]. Each partition divides a set of cells into two subsets of approximately equal size in such a way that the number of nets incident to cells in different subsets is minimized. Along with this partition of cells, the chip area accordingly is also divided to accommodate the subsets of cells. Both methods need an additional compaction step by squeezing, rotating or interchanging cells at the end of initial placement.

Although good placement is the key ingredient to a successful layout, there is still much to be done in the area of computerized tools for placement. However, this dissertation concerns only the routing problem assuming the initial placement is done.

The routing stage of layout consists of two parts, global routing and detailed routing. In global routing, the routing region is considered as the sum of rectangular channels or global cells. Each channel or cell is assigned nets which run through it. The global routing distributes the nets all over the chip area evenly so that no channel becomes a bottle neck. Thus, the global routing builds the net data and channel density for every channel in the chip. In the detailed routing stage, the channel router, receiving the net data for each channel from global router, assigns tracks for wire segments such that the width of the channel is minimized.

1.2 Overview of thesis research

1.2.1 Motivation and results

Historically, the IC layout design methods have been developed from the layout methods of the printed circuit board, which has two layers (top and bottom) for wiring; each of the two layers is used for the wires of one of the orthogonal directions. A large scale integrated circuit, which consists of many internal circuit modules, has two layers, a metal layer and poly/diffusion layer for interconnection. This wiring can easily be compared to the printed circuit board wiring. Ever since C.Y. Lee developed the first computational method [11] to solve the routing problem for the directional model, many routing tools have been developed and have generally achieved increased performances. However, as the modern fabrication technologies allow using more than two layers for interconnections, it is worth while to introduce a third layer for the IC routing problems.

Our research focuses on routing with three interconnection layers. In our three-layer model, horizontal wire segments are to be located on the top and bottom layers, whereas the vertical wire segments are to be located on the middle layer. In this way, the capacitance problem caused by parallel wires on adjacent layers can be reduced and, as in the Manhattan model, wires of different direction can always cross one another.

Based on this model, our router solves the channel routing problem using only $\lceil d/2 \rceil$ + 2 tracks where "d" is the density of the channel.

1.2.2 Outline of dissertation

This dissertation presents a layout model which has three interconnection layers and a routing algorithm for that model.

Chapter 2 reviews some of the previous approaches to the routing problems. In Chapter 3, similarities and differences of signal net routing and power/ground routing are discussed. Characteristics of routing the power and ground bus for our three-layer model are observed and an algorithm for bus routing is proposed. The first section of Chapter 4 states the purpose of the global routing and presents an overview of the global routing method. The rest of this chapter discusses the two different global routing models, one for gate arrays and the other for general cells. In Chapter 5, a new algorithm for the three-layer channel routing problem is presented in detail. The lower bounds of various channel routing models are compared. The experimental results of this router are given in Chapter 6. Chapter 7 discusses routability conditions of the channel routing problem in regard to vertical constraints, terminal locations, and channel densities. The upper bound of channel width is observed. Discussions of the advantages and limitations of our model along with some of the proposed extensions to this research are given in Chapter 8.

2. PREVIOUS WORK

The interconnection problem consists of finding paths in a chip so that

1. terminals of each net are made electrically common,
2. all nets are electrically isolated from each other,
3. the area of routing region is minimized.

To achieve this goal, numerous layout design automation tools [12] [13] [14] [15] have been developed and are being used selectively in accordance with the design styles. For polycell and masterslice layouts, automated placement and interconnection techniques have been successfully employed since the early 1970s. In this chapter, some of the better known techniques and algorithms used for automating the interconnection of LSI circuit chips are presented.

2.1 Classical approaches to routing problems

Routing of LSI circuits can be approached in a number of different ways. Nets may be connected using Steiner trees or may be restricted to spanning trees. Connection paths may be allowed to change layers at arbitrary points or at fixed points or may not be allowed at all. Most algorithms used to route wire paths can be categorized by one of the methods presented in this chapter. We are going to describe the common categories of routing algorithms.

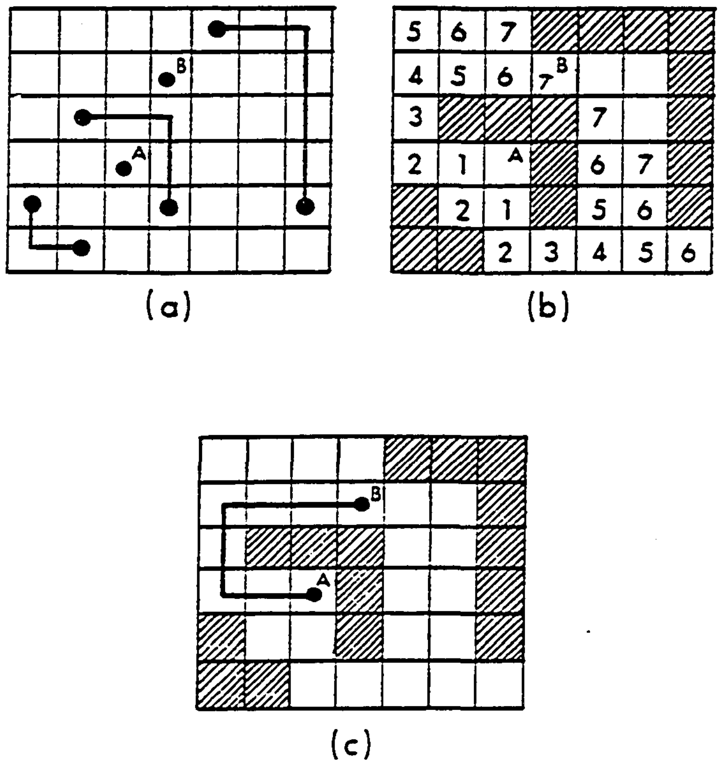
2.1.1 Maze routers

These routers find the shortest path between two points on a grid plane where obstacles may exist. The grid is scaled so that the center to center distance between any two wires on the grid lines meets the constraints imposed by the technology ground rules.

The most well-known procedure for solving this problem is due to C. Y. Lee [11]. Figure 1 shows how this router finds the shortest path connecting points A and B [16]. Lee's algorithm can easily be modified to proceed with two or more layers. The main drawback with the Lee's algorithm is its relatively large demand for computer storage and computation time.

2.1.2 Line routers

Line routers are path construction algorithms which do not imply the grid system. The best known line routing procedure is due to Hightower [17]. When two target points are given to be connected, line segments are extended in horizontal and vertical directions from the targets until



(a)

5	6	7	▨	▨	▨	▨
4	5	6	7 B			▨
3	▨	▨	▨	7		▨
2	1	A	▨	6	7	▨
▨	2	1	▨	5	6	▨
▨	▨	2	3	4	5	6

(b)

(c)

FIGURE 1. Maze router

they hit obstacles. When the escape line hits an obstacle, another escape line starts perpendicularly to its predecessor at the closest escape point where the new escape line can avoid the obstacle. This extension of line continues until two lines - one from each target - hit each other. Thus, a path is constructed between two target points. Figure 2 illustrates the process to find a path between two points A and B [16]. This technique is a very efficient procedure for finding paths on a plane with blockages. However, this router may not find a minimum length path and, in some cases, may not even find a path even though one exists.

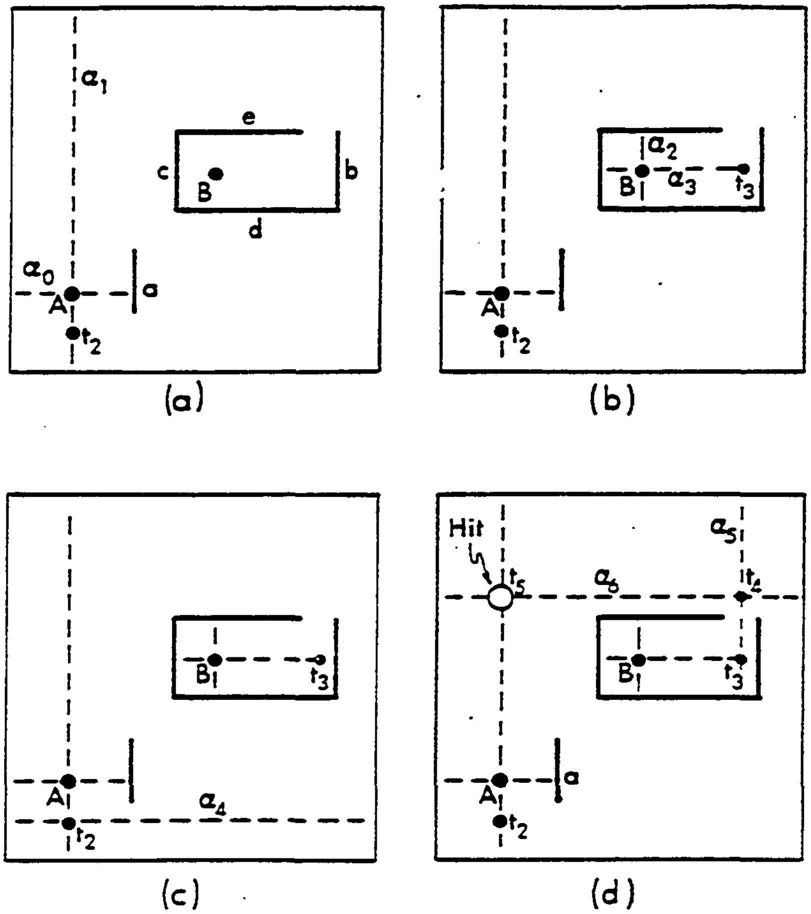


FIGURE 2. Line router

2.1.3 Channel routers

One of the problems of the previous two routing methods is that the nets are routed sequentially, that is, the path currently being routed takes into account the previously routed wires but is independent of those nets which are not yet routed. Ideally, it would be desirable for all the nets to be considered simultaneously for the wire routing of each net. This condition is met to a certain extent in the channel routing method by devising the routing process into two parts, global routing and track assignment.

The channel routing technique was developed initially for routing with horizontal and vertical wires on separate layers [18]. Each path first finds a set of channels which the net traverses without considering conflicts within each channel. After all the nets are globally routed using channel areas, the segments within each channel are assigned tracks so that no two parallel wire segments overlap. Further discussion of channel routing method occurs in Section 2.2 and Chapter 5.

2.1.4 Single-row routers

In this router, a routing plane is considered as a set of rows and columns of terminals, some of which need to be connected as shown in Figure 3 [16].

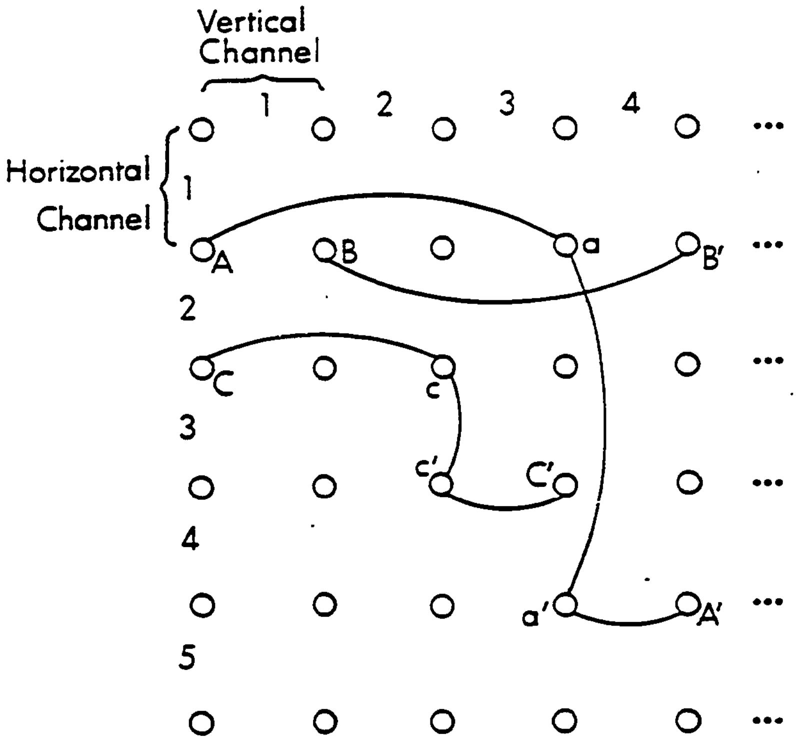


FIGURE 3. Single-row router

The global routing stage of the channel router may be borrowed and modified to assign channels for each path and to indicate pairs of points in rows or columns which are a part of a net. Then the pairs of points along a line will be connected row by row and column by column in the single-row routing (Figure 4).

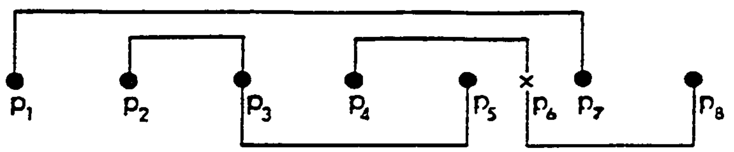


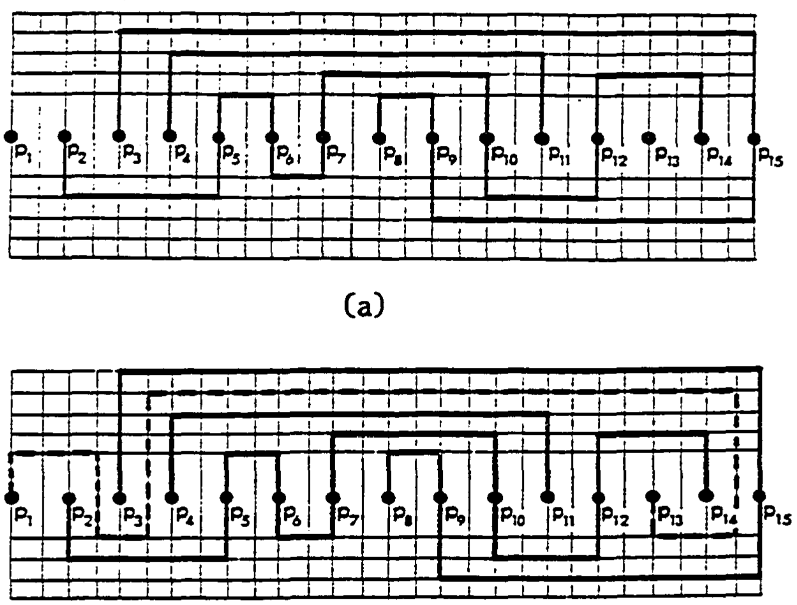
FIGURE 4. Routing row by row in detailed routing stage

This router is easily applied to multi-layer routing problem [19] [20] [21]. The goal of this router is to minimize the line length as well as the number of tracks per each line of terminals. Experimental results show that this router obtains 100 % routing completion for all cases [16].

One noticeable characteristic of this router is the ability to move previously routed paths to provide spaces for the net currently being routed, as shown in Figure 5 [16]. Other classical routing algorithms that lack this characteristic require an interactive run or trying all the sequences in order to find optimal solutions.

2.2 Channel routing algorithms for two interconnection layers

After the global router finds paths for given nets, the whole routing region is partitioned into smaller rectangular channels, each with its own net requirements. There have



(a)

(b)

FIGURE 5. Dynamic path-finding

been two different approaches for the assignment of tracks and columns for wire segments. They are the directional model and the knock-knee model. Both models find paths on grid planes where the wire width is ignored and the separation between two grid lines is assumed to satisfy the technological ground rules.

The directional model uses a separate layer for each direction of the wire segments. In this way, any two crossing wires can pass each other without physical contact.

The left-edge algorithm [18] is a well-known method of this type. Several modifications have been added to this algorithm to improve performances [22] [23] [24] [25].

The knock-knee model [26] [27] [28] is another approach to routing nets on a rectangular channel in which there are no obstacles. In this model, two nets are allowed to share a corner, i.e., they may have common turning points, but no wire segments are allowed to overlap. Whenever a corner is shared by two nets, they are on different layers. Figure 6 compares the solutions of a channel routing problem using these two methods.

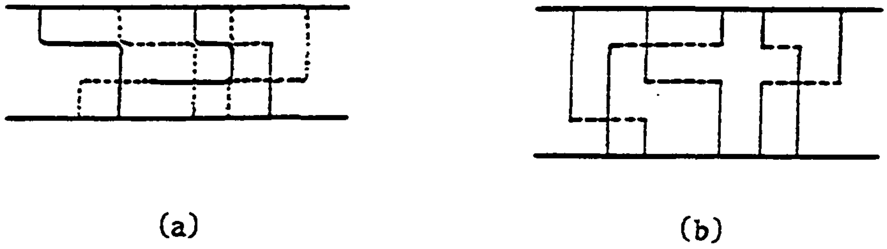


FIGURE 6. Knock-knee model vs. directional model

The knock-knee model can avoid the capacitive coupling problem which may occur in the two-layer overlap model and can reduce the channel width down to the channel density for all cases if three layers are available for interconnection [28]. However, the complexity of the layer assignment of this model may become an overriding issue.

Combined usage of the knock-knee and overlap models is suggested by some authors to reduce the channel width in two-layer channel routing model [29] and in multi-layer overlap models [30] [31] [32].

Analyses have been made to compare the routing space requirement of the three-dimensional layout and the two-dimensional (or two-layer) layout [33] [34].

3. ROUTING FOR POWER AND GROUND

3.1 Overview

The power/ground routing problem is to lay wires that connect every power connection point on the modules to the power pad and every ground connection point to the ground pad. The power/ground routing problem is basically different from the signal routing in the following aspects:

- each wire must be wide enough to accommodate the large amount of current which often flows through it during operation of the chip.
- power/ground wires need to be entirely on the metal layer(s) to maintain certain electrical characteristics.

In common two-layer routing models, the requirement of using only metal layer for power and ground routing imposes the restriction that wires of power net can not cross the wires of ground net. Thus, the power/ground routing problem is the problem of finding two non-crossing Steiner trees on a metal layer. Figure 7-(a) shows an example of this case.

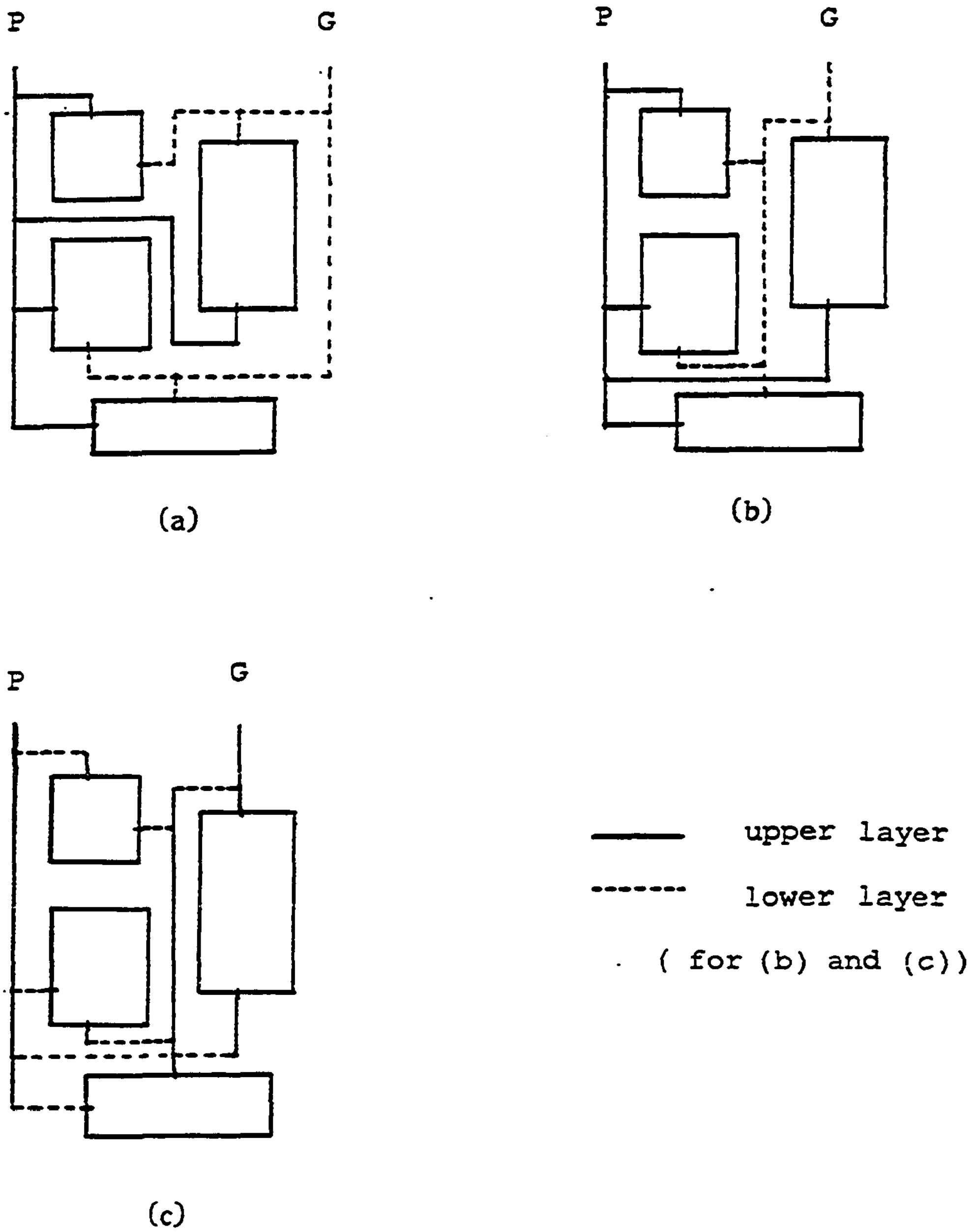


FIGURE 7. Three types of power/ground routings

In this case, if a module in an IC contains more than one power terminal and more than one ground terminal, the power and ground nets may not complete the wiring without crossing each other [35]. Figure 8 shows two cases of a module with two power terminals and two ground terminals.

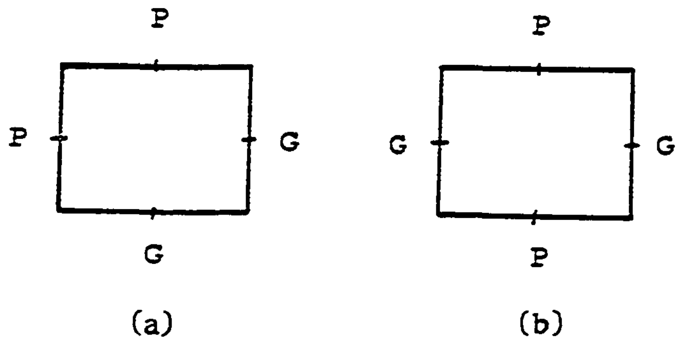


FIGURE 8. Modules with two power/ground terminal pairs

For the case of Figure 8-(b), it is topologically impossible to lay non-crossing wires that connect the two power terminals together and two ground terminals together outside the module and using only one layer. Another drawback of this model is that the power/ground wires become obstacles in the signal net wiring stage because the signal wires are routed on direction-per-layer basis, whereas the power/ground wires are not.

For the models where two metal layers are available besides a poly/diffusion layer, the power/ground routing problem is given the flexibility of constructing two non-touching nets. We can think of two methods to route power/ground nets on two metal layers. The first method is to build two Steiner trees on the separate metal layers, one for the power bus and the other for the ground bus (Figure 7-(b)). We don't need to worry about the crossings of power and ground wires in this case, and obviously, the total bus length will be smaller than it would be when two non-crossing Steiner trees are placed on one metal layer. However, the method which uses Steiner trees for the power/ground buses violates our direction-per-layer scheme and results in higher complexity and more area for the signal net wires to avoid contacting the power/ground wires.

The other method is to route the power/ground wire segments like ordinary signal net routing. That is, one of the metal layers can be used for horizontal segments of power/ground wires and the other for vertical segments of power/ground wires (Figure 7-(c)). Thus, the direction-per-layer strategy is kept for power/ground routing as well, and the signal net routing does not face non-directional bus obstacles. This approach, however, has technological problems such that the layer transfers of power/ground wires may degrade the regulation of power source.

The major differences between the power/ground routing of this case and the signal routing is the number of layers used and the width of the wire segments. Since only the two metal layers are used for the power/ground wires, there will be no parallel overlaps. Thus the problem of capacitive coupling between the power and ground lines is diminished.

As the power/ground wires carry larger currents, these nets need to be wider than the signal nets. In the grid model where each wire segment is considered as a line segment, we regard all nets as having a uniform wire width. However, the power/ground wires must be much wider than any signal wires and they can not be laid on the area assigned for single grid line. A good solution to this problem is to use two grid lines for a power/ground wire. Figure 9 shows how the power/ground wires are laid on the grid plane and occupy two grid lines per power/ground wire.

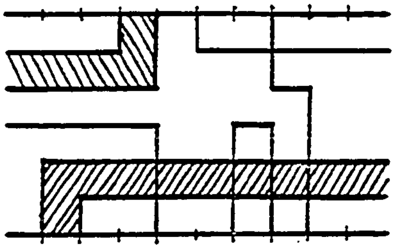


FIGURE 9. Allocating two grid lines for power/ground wires on grid plane

Using this model, power/ground wires can be routed by the same router that we use for signal net routing.

In the following sections, we will describe the power/ground router for the model where two layers are available for power/ground wires and three layers are utilized for signal net routing.

3.2 Power/ground routing and signal routing

Power/ground routing is similar to signal routing in that the router connects sets of terminals in a routing region. For both types of routing, it is desirable to keep the routing area small. However, in our routing model, the signal nets use three layers for interconnections in a horizontal-vertical-horizontal fashion, whereas the power/ground nets use two layers in a horizontal-vertical fashion. The use of two layers for horizontal wire segments of the signal nets reduces the channel width up to half of that of the two interconnection layer model by allowing overlaps of the wire segments. The layer-per-direction usage for the power/ground routing prevents a potential capacitance problem between signal and power lines although there may be some electrical degradation at each via point of the power/ground wires.

Another advantage of the directional routing strategy for the power/ground routing is that we can use the same algorithm for the signal net routing and the power/ground routing. One thing we have to consider when dealing with the power/ground routing is that a power/ground wire segment occupies two grid lines. In this way, a power/ground wire is more than three times wider than a signal net wire. Currents in the signal wires are typically small enough to permit the wire width to be the minimum allowed by fabrication technology, whereas the larger current flowing through a power/ground wire requires more width. According to the Mead-Conway design rules [36], two grid lines must be at least 8λ apart to maintain the minimal separation of 4λ between wires, even when two contact cuts face each other. Therefore, we can allow a width up to 11λ for power/ground wires on metal when we bundle two grid lines together. However we can allow only 3λ for signal net wires on the metal layer and 2λ for signal net wires in polysilicon or diffusion layers. Figure 10 explains this situation.

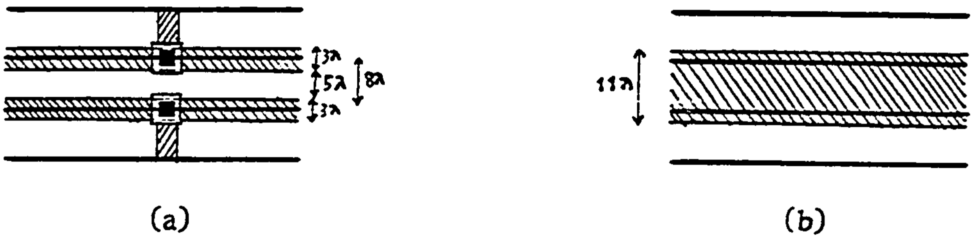


FIGURE 10. Width comparison of signal net wire and power/ground wire

Because we assume that a signal net wire is represented by a widthless grid line in our router, we model a power/ground wire as two grid lines bundled together regardless of its current requirement. More elaborate models calculate the local current values and adjust the bus width accordingly [37] [38].

When two grid lines are tied together to locate the power/ground wires, special treatments are needed when applying the Dogleg or Merge algorithms. An easy way to do this is to locate power/ground wires before we assign the tracks for the signal nets and place the power lines using the two highest possible tracks in a channel and place the ground lines using the two lowest possible tracks in a channel subject to vertical constraints. In order to do this, the vertical constraint trees, which contain power/ground nets, need to choose dogleg nets adjacent to the power/ground nets so that the power/ground nets are neither leaves nor roots of the tree. In this way, the power and ground nets are not chosen for doglegging and consequently would not cause a serious capacitance problem because they are separated from each other toward opposite sides of the channel.

3.3 Restrictions imposed on the power/ground problem

Our model allows two metal layers for placing the power/ground wires. These wires can change their directions by transferring to the other layers. This scheme provides freedom in routing non-crossing wires - either power/ground wires or signal net wires - at the price of a degradation of the electrical properties at every turning point of power/ground wire.

As we discussed in previous sections, the power/ground wires use two grid lines bundled together. That is, power/ground wires have a uniform width any place on the chip. This scheme is chosen for simplicity as we consider the signal net wires as line segments on a grid plane. We do not calculate the minimum width of each power/ground wire that can provide the necessary current and a stable voltage for the worst case power consumption case. We do however, leave the possibility of adding a variable width consideration to our router in order to be more area-efficient.

Typically, the critical current density for aluminum wires in an IC is 1 to 2 mA per square μm of cross-section [39]. For CMOS ICs, the current demand of a 2000-gate array can be met by minimum sized metal track. For n-MOS VLSI, we have no general estimation of the peak current through the power/ground lines that connect all the modules inside the

IC. Our scheme that uses two grid lines for a power/ground wire could be extended to use more than two grid lines for larger power demand.

3.4 Algorithm description

There have been several algorithms developed for power/ground routing [35] [37] [40]. Each of them made an attempt to build the shortest non-crossing trees for the power and ground nets on a plane with certain restrictions. As our model uses two metal layers, however, the routing scheme is completely different. In our model, power and ground wires can cross as many times as needed to make the shortest nets. This scheme eliminates the restriction posed by locations of power and ground terminals on modules as discussed in Section 3.1 and allows as many power/ground terminals as needed at any places on each module.

Another advantage is that the power/ground nets also follow the direction-per-layer scheme like signal nets. In this way, signal nets need not change layers to cross under the power or ground wires and thus there is no exception for direction-per-layer rule.

The first step of the power/ground routing is to determine the locations of power and ground pads along the perimeter of the chip [38] [41]. As every module in an IC has at least one power terminal and one ground terminal, the locations of power and ground pads are important factors to determine the bus length.

The next step is to find sequences of channels where the ground net and the power net traverse such that the total bus length is minimized. This is done by global router. With each channel, the router locates the horizontal segments of the power wires to the two highest possible tracks and locates the horizontal segments of ground wires to the two lowest possible tracks as the vertical constraints allows. The vertical segments of the power/ground nets are assigned on the second metal layer. In this assignment process, there is a possibility of conflict between vertical segments of the power and ground nets, although the chance is very small. As we do not want to use a dogleg net for the power or ground net, this conflict should be avoided in the placement stage of layout design by shifting or reflecting either the top side or bottom side module. The channel router considers the bus wires as a pre-existing wire set and avoids these obstacles by keeping the locations of these in the data sets of the channel router.

4. GLOBAL ROUTING

4.1 Purpose of the global routing

From the placement data of an IC chip, every net needs to be arranged on a routing plane such that the total wire length is minimized and the wires are distributed to avoid local congestion.

The global routing is a preliminary stage of wire routing which provides an approximate routing of all connections. A problem in routing a large number of nets on a plane can be caused by the sequential nature of routing assignments. The space required for routing many nets on a plane is usually order-dependent. By routing one net at a time and making the best choices in placing the wire segments of the net, a router may prevent later nets from being wired or wired efficiently due to lack of look-ahead capability.

The importance of global routing is in its global nature. It takes all the related interconnection requirements into account simultaneously. Global routing, therefore, tries to emulate the independent net routing process. All signal nets compete for space to complete their wiring. Congested wiring areas are then analyzed and rerouted as needed.

This preliminary stage of routing was of limited use when ICs had a small number of modules. As the number of modules of an IC grew, this pre-arrangement of wirings became an important part of the routing problem.

The global routing scheme of the three-layer routing model is not much different from that of the two-layer routing model unless over-the-cell routing is allowed for the three-layer model. Two global routing methods, among others, are noticeable. The first method divides the entire chip area into rectangular cells and determines which edges of the cells are crossed by each net. This method is suitable for the standard cell or gate array layout. The other method, which is for general cell layout, divides the routing region into rectangular channels and constructs the channel graph where each channel is represented by an edge. Using the channel graph, this router finds a set of channels for each net to traverse. The common technique employed in global routing methods is to subdivide the whole chip area into sections. An efficient way to do this is by the hierarchical method where the plane is divided into a certain number of subregions and each subregion is considered as a unit cell. The internal wiring of this cell is considered later in a detailed routing stage. This two stage routing reduces the complexity in finding globally efficient wirings.

4.2 Global routing for gate arrays

Gate array ICs have been widely used in today's IC manufacturing. The regular structure of gate arrays is appropriate for layout automation. The total plane of a gate array consists of two kinds of portions, active areas and channels between them. As we mentioned, the purpose of the global routing is to avoid local congestion in order to yield a high probability of success for subsequent detailed routing while keeping the total wire length minimum. In order to do this, the total plane is divided by grid lines parallel or perpendicular to the gate rows [42] [43] [44] as shown in Figure 11.

The concept of this two stage routing scheme is sometimes called 'hierarchical VLSI routing' [45]. Each rectangular area bounded by grid lines is called a global cell and its sides are called edges. Each edge has capacity which represents the number of line segments allowed to cross the edge. The global routing stage, then, is to determine which edges each net crosses over to connect the necessary terminals. The location of each terminal of a net is represented by the cell number that contains the terminal.

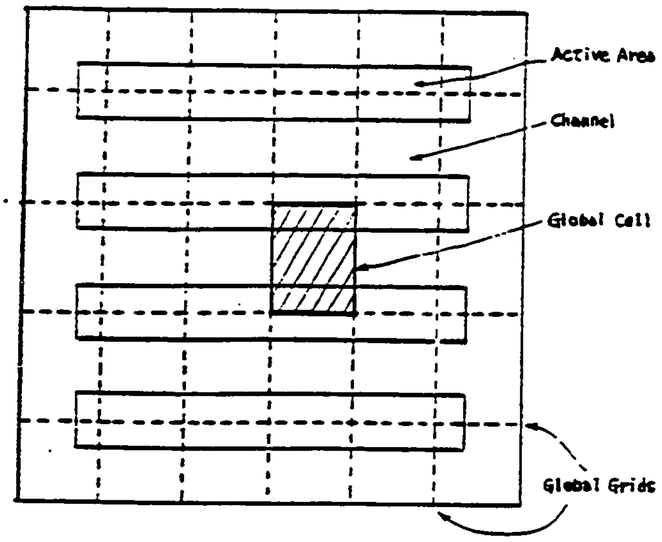


FIGURE 11. Partitioning the chip area into global cells

A set of terminals for each net is connected using a rectilinear Steiner tree construction to provide a minimum Manhattan distance. Initially, every net is wired via a Steiner tree without being affected by other nets. This independence provides equal chances for all nets to find their shortest interconnection patterns and eliminates the bias introduced by different sequences of routing nets. After the global routing tentatively locates all the nets through the global cells, the capacities of all the edges of the cells are examined to check the routability condition. For the over-congested edges, we need to reroute some connections to prevent the bottle necks [12] [42] [46].

Each of the nets passing a congested edge is weighted according to its number of crossings of other congested edges and the crowdedness of the congested edges which the net passes. The net with highest weight is chosen first to be rerouted and other nets with positive weights are rerouted successively in iterative fashion. Depending upon how the weighting function and the iterative nature of global routing are determined, there may exist a factor of dependency between net routings. Chen et al. [12] state that this iterative rerouting phase almost always completes the global routing without over-congested cell edges. If this is not the case, the placement of active areas needs to be adjusted to give extra room for over-crowded channels. Figure 12 shows the global routing steps [47].

4.3 Global routing for general cell layout

A more general type of layout automation tool has been developed for two-layer general cell layout problems. Among the noticeable tools are the BBL system [12] and PI system [13] [48]. The global routing scheme for general cell IC layout is different from that for a gate array although the purposes are same. In general cell layout problems, the routing region is divided into rectangular channels and routing within each channel is treated separately by the channel routing. The global routing assigns a set of

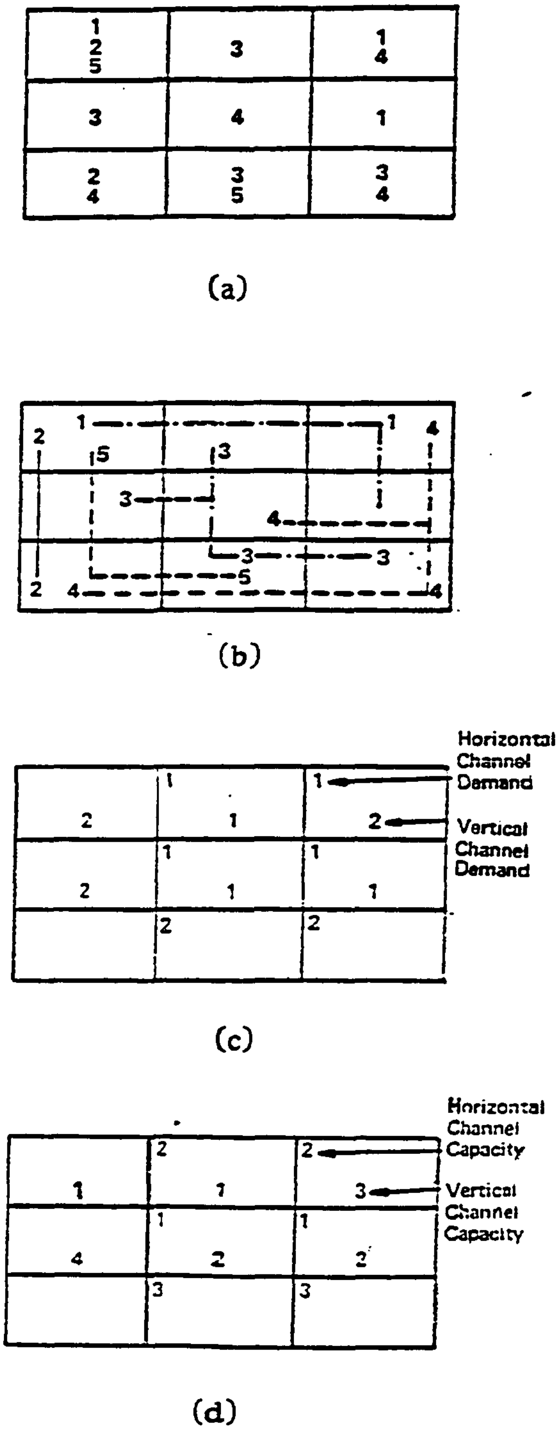
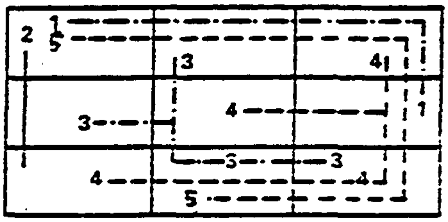


FIGURE 12. Global routing steps



(e)

FIGURE 12. Global routing steps (continued)

channels for each net to traverse.

The global routing provides each channel with its internal net-lists, which will be used in the channel routing stage. Consider a layout of a general cell IC where cells are placed as shown in Figure 13-(a) [12]. The routing region consists of rectangular channels, each of which has its capacity (Figure 13-(b)). The channel graph, which is sometimes called global routing graph [12] or subchannel intersection graph [46], represents each channel by an edge along with its channel capacity and shows geometrical connectivities between channels (Figure 13-(c)). As was done with gate arrays, the global router finds minimum length paths which run through the edges of the channel graph while keeping the density of each channel smaller than the channel capacity.

The difference between the gate array method and the general cell method, other than the regularity of global

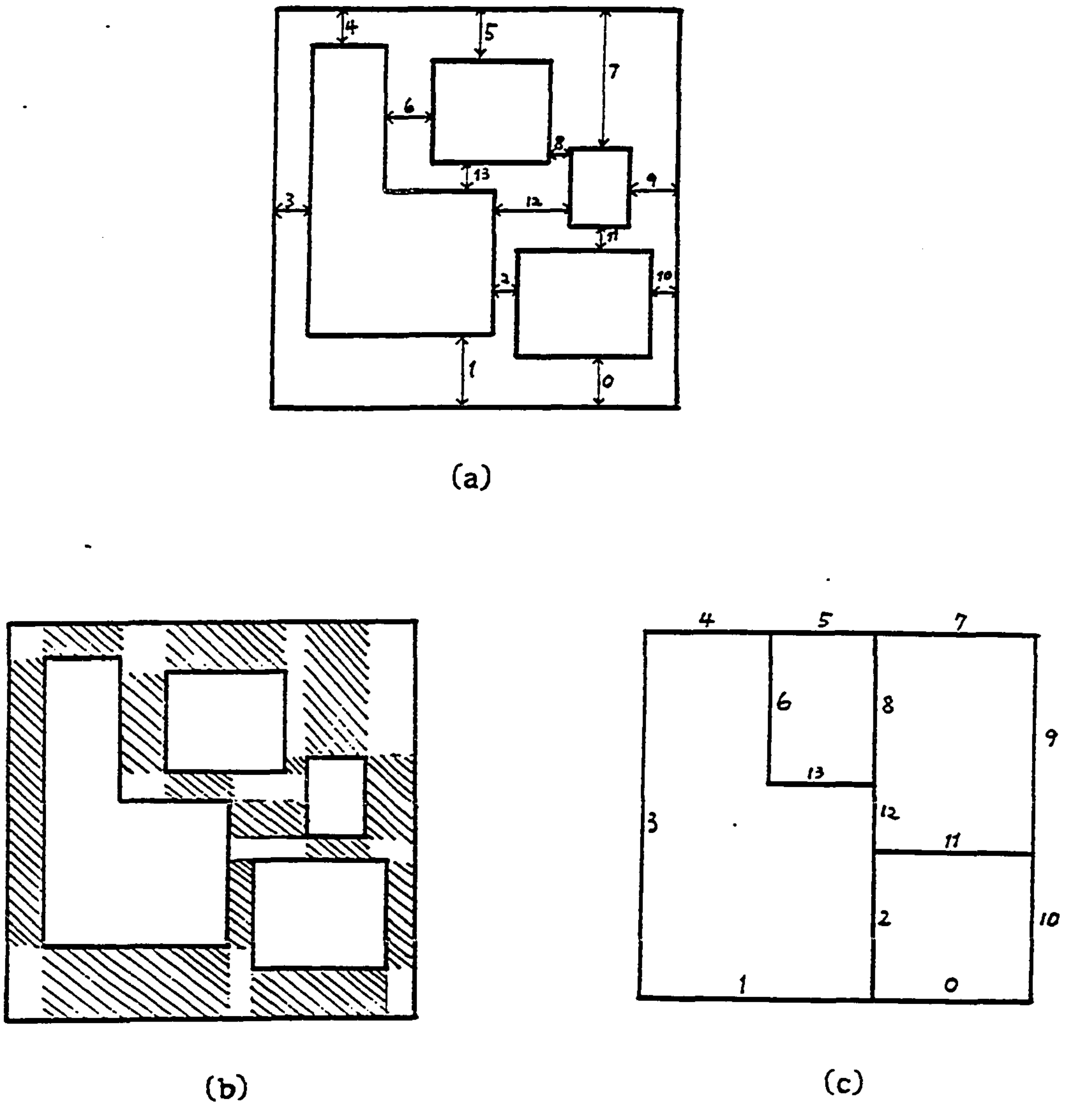


FIGURE 13. General cell layout and the channel graph

cells in gate arrays, is that the edge capacities are considered for all four sides of each cell in gate arrays, whereas in general cell structure, an edge which corresponds to a channel has only one directional capacity.

A different approach to controlling the channel capacity is described bellow. Each edge (channel) in a global routing graph is given a weight which reflects the congestion and the length instead of the capacity of the associated channel [12]. The edge weight is updated each time a signal is routed. At the beginning, the length factor is dominant. As the global router proceeds, the congestion factor becomes more important. In this way, each net finds its global path according to the edge weights. If any net is not completed until all the possible edge weights are large enough (over-congested), the placement of circuit blocks needs to be adjusted. A distance function as a performance index is considered when finding global paths. The distance function may be defined to be the Manhattan distance between endpoints plus penalty values for turning corners or passing an edge that is nearly full.

5. THREE-LAYER CHANNEL ROUTING

There have been several approaches which use three interconnection layers to solve channel routing problems. Among those are two three-layer directional models which were proposed by Chen and Liu [49]. They used HVH (horizontal-vertical-horizontal) type and VHV type channel routing algorithms, which allow overlaps of not only wire segments which cross other wires, but also wire segments in the same direction. Using one of their models, the width of a channel could be reduced to up to one half of the best two-layer channel router in many cases. However, as these algorithms tried to reduce the channel width only by overlapping horizontal or vertical wire segments, they failed to find optimal solutions when the length of the vertical constraints graph was longer than the density of the channel.

In this chapter, a new HVH type channel routing algorithm, which uses an extensive Dogleg algorithm and Merge algorithm, will be presented and the performance will be discussed. This algorithm achieves the lower bound of one half of the channel density and finds near optimal solutions in all cases.

5.1 Channel routing problems

5.1.1 Decomposition of routing region into channels

A channel is a rectangular area bounded by circuit modules and by adjacent channels. The entire routing region needs to be decomposed into channels in order to be treated separately by channel routing algorithms.

In the building block or general cell approach, it is probably a good strategy to decompose with a minimum number of channels. That is, making each channel as large as possible for the given placement of modules; this provides the channel router with greater flexibility in routing and produces a smaller number of floating terminals. Figure 14 shows an example of a decomposed routing region of a chip [50].

The decomposition of a routing region for gate array structure is straightforward. Cells of the same height are located in rows and any two rows of cells make a rectangular routing region in between; this type of structure is appropriate for gate arrays and is convenient for the channel router to use.

5.1.2 The new model

Our three-layer channel router is basically of the HVH type, which uses two interconnection layers for horizontal wire segments and one layer for vertical wire segments. The

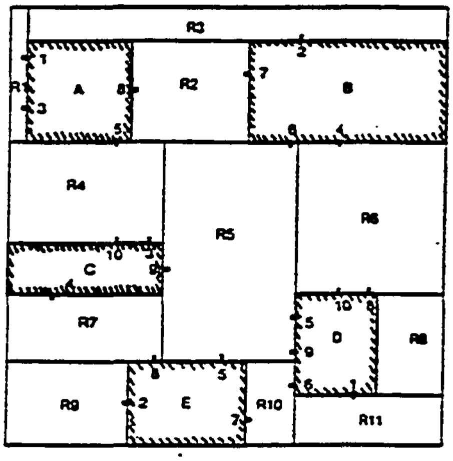


FIGURE 14. Decomposition of routing region

characteristics of our three-layer channel router are summarized as follows.

1. Horizontal wire segments are located on the top and bottom layers and allowed to be overlapped.
2. Vertical wire segments are located on the middle layer.
3. Dogleg nets are used to remove not only cyclic vertical constraints but the effect of long vertical constraints to the width of channel.
4. A dogleg net uses the longest available column for its vertical dogleg segment to reduce the height of the vertical constraints graph as much as possible.
5. The concept of the left edge algorithm is applied

for the three-layer case to compact the horizontal segments and to find the minimum channel width.

In our model, as the layer changes occur from top (horizontal) layer to middle (vertical) layer or from middle (vertical) layer to bottom (horizontal) layer, vias connecting two adjacent layers will not be obstacles on the other layer. The HVH arrangement is also advantageous with regard to the capacitive coupling, which can be a problem for those models allowing overlaps of the wires on adjacent layers. The middle layer of this model consists only of vertical wires and separates the two horizontal layers so that the capacitance problem is reduced. Unlike the knock-knee model, this model can handle multi-terminal nets and is expandable to more than a three layer model.

5.1.3 Some thoughts about the number of layers of multi-layer directional models

Two-layer channel routers have been used largely in IC layout design. The Manhattan model prevails among others. The Manhattan model prevents short circuits at the intersections of horizontal wires and vertical wires by assigning different layers for wire segments of each direction. As the chips get bigger, the horizontal wire segments tend to be longer and the horizontal tracks become more crowded while the vertical constraints become less critical. Thus, there arises a need to introduce an extra layer for sharing the horizontal wire segments.

In the HVH model, signal paths need to transfer to the adjacent layers at every turning point: from top or bottom to middle layer or from middle to top or bottom layer. Each via connects a horizontal wire segment on one layer to a vertical wire segment on the adjacent layer and does not block a grid point on the other layer. If we consider more than three layers, vias may have to connect two layers which are not adjacent and this causes obstacles on the layers in between. We can easily see that the HVHV type four-layer model confronts via obstacle problem when a signal path transfers from the first layer to the fourth layer. Also, this model is no better than HVH model in that the long vertical constraints problem can be solved using our extensive dogleg method. Similarly, the HVVH model does not have much advantage over the HVH model, either.

If we continue our consideration with five and six layer models, the HVHVH model is expected to have some gain in channel width, but, as we discussed, this model causes more via obstacles and the complexity of the routing algorithm will be considerably higher than that of the HVH model. We can not conclude whether it is worth while to adopt this model to achieve the small increase, possibly, in performance at the price of the increased complexity. From the observation herein, the next efficient model is thought to be a six-layer model. As our HVH type model is the most area efficient among three or four layer models, the HVHHVH model will not only be area efficient, but also easily expandable from the HVH model. Other possible models for six layers are HVHVHV type and HHVVHH type. The HVHVHV model is no better than the HVHVH model as the HVHV model is no better than the HVH model. The HHVVHH model will confront more via problems than the HVHHVH model if there are no restrictions on the layers to be transferred.

5.1.4 Lower bounds of channel width

As the Manhattan type models use different layers for horizontal wire segments and vertical wire segments, we need to consider only the interferences between wires of same direction when we lay out the wires. The two constraints relevant to the width of the channel are horizontal constraints, which concern the overlap of horizontal wire segments, and vertical constraints, which prevent the overlap of vertical wire segments. These two constraints bring in two parameters, density d and height h, to the channel routing problems. As we discussed in Chapter 2, the density of channel is a factor in determining the lower bound of the number of tracks required; the height of the longest vertical constraints graph is another factor. The lower bound of channel width of the two-layer Manhattan model is therefore the worst of the two parameters, that is, max(d,h) [49].

If we add the third layer to HV type two-layer model and place horizontal wires on it, the trivial lower bound relevant to the density becomes max($\lceil d/2 \rceil$, h) [49]. If we assign the third layer for vertical wires to make a VHV model, the vertical constraints are no longer effective because the vertical wire segments can overlap. The lower bound in this case is the density of the channel.

Any model with more than three layers will confront other factors like via obstacles and locations of vertical dogleg segments in determining the lower bound, and we exclude them from our discussion.

As our model uses two layers for horizontal wires and applies the extensive dogleg method to remove the lower bound factor of h, the trivial lower bound of the channel width is $\lceil d/2 \rceil$. A more detailed discussion will be presented in Section 5.3 and Chapter 7.

From these observations and the discussion in Chapter 2, the lower bounds of various models are summarized in Table 1.

TABLE 1. Lower bound comparison

Model	Number of interconnection layers	Number of overlap layers	Lower bound of channel width
Manhattan type	2	no overlap	max(d,h)
HVH [49]	3	2	max($\lceil d/2 \rceil$,h)
VHV [49]	3	2	d
knock-knee	2	no overlap	d
knock-knee [28]	3	2	d [1]
overlap model [30]	L	K	$\lceil d/K \rceil+1$ [2]
1-2-3 model [51]	3	2	$\lceil d/2 \rceil$
our model	3	2	$\lceil d/2 \rceil$

[1] This is also the upper bound.

[2] For $K \leq \lceil L/2 \rceil -2$.

5.2 Input and output specifications of the three-layer channel router

The purpose of channel routing is to find the minimum width interconnection layout for given net data. Each net consists of terminals on the top or bottom side of the channel and these terminals are to be connected using horizontal and vertical wire segments so that the resulting layout realizes the minimum channel width. The input to the channel routing problem is given in the form of Net-lists that give net numbers of the terminals on every column of the channel. Figure 15 shows an example of Net-lists.
A set of Net-lists is given as the input which represents

columns	1	2	3	4	5	6	7	8	9
top	0	0	1	0	2	0	3	0	4
bottom	1	2	0	0	3	0	4	0	0

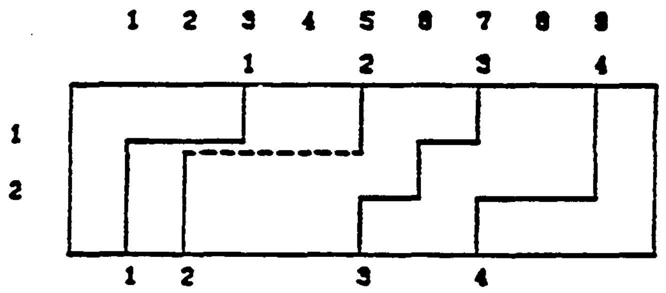
FIGURE 15. An example of Net-lists

the connection information among the terminals on both sides of the channel. It shows the locations of terminal sites for every net. The output of the channel router is a set of Nets which, along with the Net-lists, describes the layout of the wires in the channel. Figure 16 is a set of Nets and its layout which is the solution of the channel routing example of Figure 15.

5.3 Vertical constraints and the extensive dogleg method

As mentioned in Section 5.1.4, the vertical constraints prevent the overlap of vertical segments by constraining the sequence of locations of horizontal segments. When the vertical constraints graph (VCG) has cycles, it is impossible to route the channel with one vertical layer. The dogleg method is used to remove cycles in the vertical constraints graph [23]. Figure 17 explains this situation.

If a set of Net-lists makes a cycle in VCG, the left-edge algorithm [18] can not find proper connections (Figure



(a)

nets	l	d	r	lt	ll	rt	rl
1	1	0	3	1	1	0	0
2	2	0	5	1	3	0	0
3	5	6	7	2	1	1	1
4	7	0	9	2	1	0	0

(b)

FIGURE 16. Nets layout and its data structure

17-(a)). To release the cycle, a net in the cycle has to be deformed into a dogleg net by breaking the horizontal segment of the net and introducing a vertical dogleg segment. Thus, the new VCG becomes directed acyclic graph (Figure 17-(d)). This problem can be solved by many conventional two-layer directional models or the HVH model [49] producing the channel width of four (Figure 17-(c)).

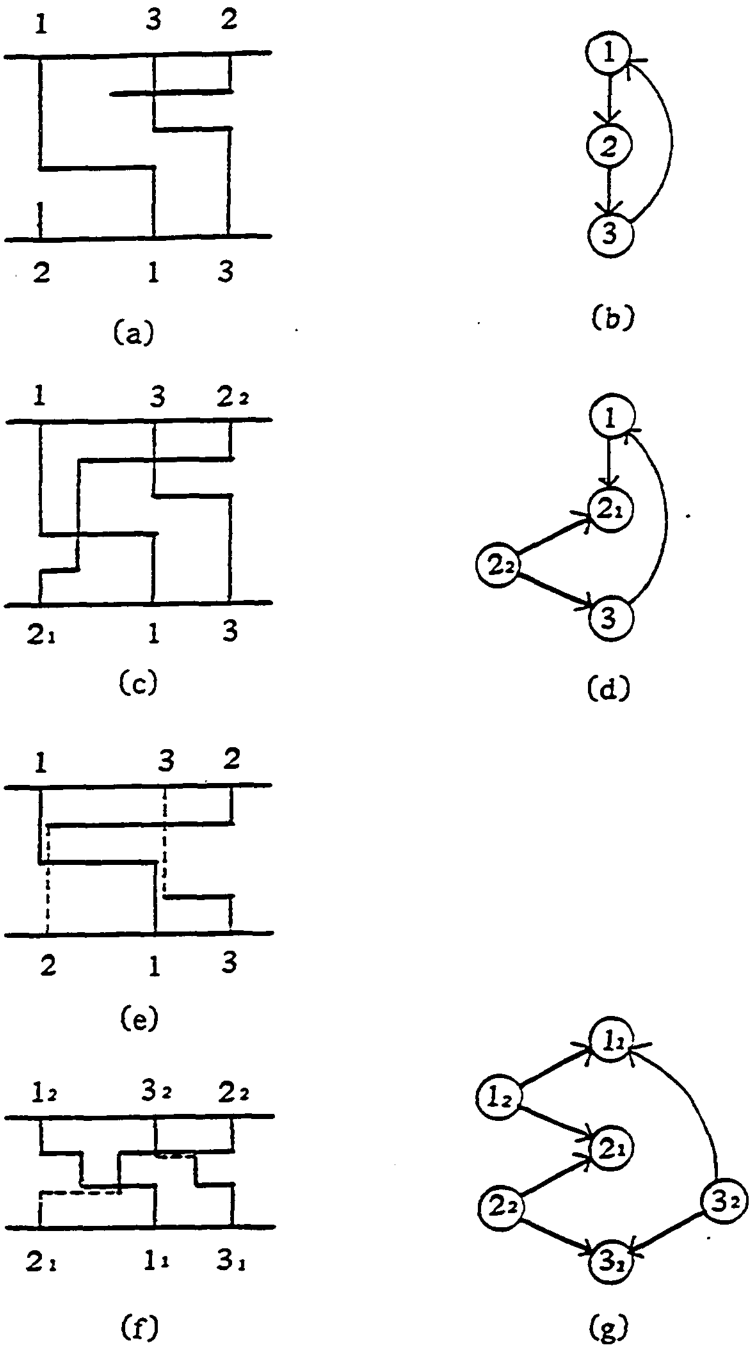


FIGURE 17. Vertical constraints graph and the dogleg method

As the height of VCG is four, this is the minimum width realization for these models. However, if we apply the dogleg method further to those nets in the middle of the longest VCG, we can reduce the height of VCG, which determines the lower bound of channel width in this case. Figure 17-(f) is the VCG of height two after these additional doglegs. In this way we can eliminate the height factor from the lower bound of the channel width and we get the trivial lower bound of $\lceil d/2 \rceil$.

Figure 17-(e) shows the routing result which achieves this lower bound of $\lceil d/2 \rceil$. In the VHV model, the vertical constraints have no meaning because overlaps of vertical wire segments are allowed. The routing of this model, being restricted by horizontal constraints only, gives the width of three, which is equal to the density (Figure 17-(g)).

5.4 The algorithm for three-layer dogleg channel router

5.4.1 Initial routing

In the initial routing stage, the router builds a VCG from given Net-lists and assigns a track for every horizontal wire segment such that the initial track assignment satisfies the vertical constraints. When the VCG contains cycles, a net from each cycle is chosen and sent to the dogleg part of the algorithm, where the net is doglegged to break the cycle. Once an edge in the cycle is broken, the rest of the cycle can be assigned tracks according to the sequence of the VCG. This stage is for initializing the VCG, the array of Nets, and the TLO (track-layer-occupancy) rather than for minimizing the channel width; this is considered in the merge part of the algorithm. The following are the major steps of the initial routing stage of the algorithm.

1. Read the array of Net-lists.
2. Build the VCG.
3. Detect cycles, if any, in the VCG and choose a proposed dogleg net from every cycle.
4. Send each dogleg net to the Dogleg part of this algorithm.
5. For every tree in the VCG, assign the highest available track in the channel for the uppermost node. The track availability is provided by checking the TLO which is initially empty and filled one by one as horizontal segments of nets are assigned tracks and placed in the channel.
6. Build the array of Nets as the track assignment proceeds.
7. Build the array of CV (column-vacancy) from Nets and Net-lists.

The data sets formed by the initial routing are:

- VCG (vertical-constraints graph)
- TLO (track-layer occupancy)
- Nets
- CV (column vacancy)

These four data sets and the Net-lists are sent to the dogleg part of the algorithm and kept throughout the execution of the router and updated as needed.

5.4.2 Three-layer dogleg algorithm

In the two-layer Manhattan model, dogleg nets are used in two cases: resolving cycles in VCG and breaking the long vertical constraints graphs. When a VCG contains cycles, the channel is not routable unless a net in the cycle is doglegged to break the cycle. After all the cycles in the VCG have been removed, further doglegs are still needed when the height of the longest VCG is larger than the density of the channel. In this case, the height becomes the lower bound of the channel width and the reduction of the height enables the channel to become narrower. As described in Section 5.3, the dogleg nets break the vertical constraints graph and reduces the height until the height of the longest VCG becomes smaller than the density. This situation is important in three-layer HVH model, where the lower bound of non-dogleg HVH model is the max($\lceil d/2 \rceil$, h). There is a much greater chance of $\lceil d/2 \rceil \le h$ than $d \le h$ in channel routing problems and we need to use doglegs to lessen the height until it becomes smaller than or equal to $\lceil d/2 \rceil$. When the height becomes less than $\lceil d/2 \rceil$, the lower bound of the channel width will be independent of the height of the VCG and it will be $\lceil d/2 \rceil$. Practically, the dogleg nets, by which we achieve $h \le \lceil d/2 \rceil$, are not enough to provide the minimum channel width because there are other factors which restrict the Merge algorithm. We will discuss this in Section 5.4.3 and in Chapter 7.

It is experimentally observed that introducing dogleg nets until h becomes less than $\lceil d/4 \rceil$ results in the minimum channel width in most cases, although there may be unnecessary dogleg nets and vias in some cases. Most of the unnecessary dogleg nets and thus, unnecessary vias, which provide the Merge with more flexibility to compact the channel, return to normal nets by the procedure called 'release-dogleg'.

The three-layer Dogleg algorithm consists of the following major steps:

1. Find the density of the channel from the given set of Nets.
2. Figure out the height of the longest VCG.
3. Choose a net for a proposed dogleg net around the center of the longest VCG.

4. With each of the proposed dogleg nets chosen in the initial routing stage and in step 3, find a column which has the longest unoccupied portion (that is, CV) between the two terminals of the net.
5. Assign this column for the vertical dogleg segment and assign the uppermost track and the lowermost track of this column for the two horizontal segments of the dogleg net.
6. Update VCG, TLO, CV, Net-lists, and Nets accordingly to the changes as the dogleg nets are created.
7. Repeat step 3 to step 6 until $h \le \lceil d/4 \rceil$.

The above procedures convert the chosen nets into dogleg nets, which remove the cycles in VCG and break the long VCG. Two restrictions in making dogleg nets are:

1. A net can not be doglegged twice.
2. A column in a channel is not allowed to locate more than one vertical dogleg segment.

Up to now, the algorithm provides all the necessary conditions for the Merge, which actually stacks up the horizontal segments and compacts the channel.

5.4.3 The merge algorithm for two horizontal layers

In the preceding section, we suggested a method that dealt with vertical constraints and removed those vertical constraints which restricted the width of the channel. The Merge algorithm of this section will actually reduce the channel width by stacking up the horizontal wire segments while satisfying the vertical constraints as well as horizontal constraints.

In the HVH model, where two layers are allowed for locating horizontal segments, we are given much more flexibility in compacting the channel than in the models with one horizontal layer. When we compact the channel, two constraints, vertical and horizontal, need to be satisfied. The horizontal constraints, which are represented by the TLO, restrict any two horizontal segments which overlap each other on the same track, to different layers. The vertical constraints, which are represented by VCG, prevent the overlap of vertical segments of nets. The Merge algorithm basically performs the shifting up of the horizontal segments on lower tracks of the channel to higher tracks. As two layers are assigned for the horizontal segments, each horizontal segment is checked for the availability of a higher track on either layer and is shifted up to the highest available track. The following are the major steps of the Merge algorithm.

```
Merge(fromlayer, tolayer);

begin
  for totrack=1 to width-2
    for fromtrack=totrack+1 to width-1
      for every horizontal segment on fromtrack
             of fromlayer
        if the horizontal segment doesn't violate
             the horizontal constraints on totrack,
          if this proposed shift of the horizontal
                 segment doesn't violate the
                 vertical constraints,
             begin
               shift this horizontal segment to
                  totrack
               update TLO, CV, and Nets accordingly
               if any dogleg net was straightened
                     out (return to normal net) by
                     this shift,
                  make proper changes to Nets,
                     Net-lists, and CV for released
                     dogleg net,
               if any track was evacuated on both
                     layers by this shift-up,
                  delete the track and make
```

```
                         proper changes for Nets, TLO,
                         and the width
                end
     end
```

Although there are four possible combinations of layer parameters of the Merge such as (1,3), (3,3), (3,1), and (1,1), execution of these four Merge's is not complete because the shift-up of one horizontal segment may create space for another horizontal segment that failed to find higher tracks, to move up. In order to accommodate this situation, the set of four merges needs to be executed repeatedly until there has been no single shift in the previous execution. The Merge algorithm tries to reduce the width of channel by shifting up every possible horizontal segment. However, there is a kind of horizontal segment which is better in the lower tracks of a channel. Consider a net whose two terminals are connected to the bottom side of the channel. Unless the up-shift of the horizontal segment gives way to other horizontal segments to shift up, higher location of horizontal segments of this net makes the vertical segments longer. Therefore, a down-shift of this kind of horizontal segment need to be added at the end of Merge algorithm.

5.4.4 Partitioning of multi-terminal nets

This algorithm basically is developed for the channel routing problem which has two-terminal nets. The multi-terminal nets can be routed as well by adding a few more steps to the algorithm. Multi-terminal nets are divided into two-terminal subnets which connect every two adjacent terminals of the multi-terminal nets. After partitioning the nets, each subnet is treated separately as if it is an individual two-terminal net. In this way, long horizontal segments of multi-terminal nets are divided into shorter segments of two-terminal nets. Each horizontal section has its corresponding node in the VCG, and thus it can be treated separately. This partitioning also makes it possible to dogleg a multi-terminal net as many times, if needed, as the number of its two-terminal subnets.

Unlike the dogleg methods used in [23] [24] [49], our algorithm makes dogleg nets from two-terminal nets, and this enables the optimum placement of vertical dogleg segments. However, we have to consider additional constraints at the cost of flexibility and performance. When the Dogleg algorithm is searching for the proper place for a vertical dogleg segment or the Merge algorithm is trying to shift up a horizontal segment, the two adjacent columns, which used to be a column in the channel of multi-terminal nets, need to be considered together. Decisions are governed by the worst of the condition existing in one of the two columns. At the end of this algorithm, the two-terminal subnets are combined to form the original nets.

Figure 18 shows how the multi-terminal nets are partitioned into two-terminal subnets and treated separately. Figure 18-(a) is the channel routing result of a two-layer non-dogleg model where the height of VCG is four, which is the lower bound of channel width for this type. Figure 18-(c) shows the result of the two-layer dogleg model [23]. This model uses only terminal sites of the dogleg net for the vertical dogleg segments and produces the width of three which equals the density as well as the height in this case. The three-layer HVH model by Chen and Liu [49] would give the same result as Figure 18-(c) because it lacks in the ability to handle the long vertical constraints graph. When the multi-terminal nets of Figure 18-(a) are partitioned, the height of VCG is reduced to three because of the partitioned horizontal segments of multi-terminal nets. In Figure 18-(g), two dogleg nets are created to reduce the height to two, which is the lower bound of our algorithm. Thus, our algorithm finds any proper places in the channel - not restricted to the terminal columns of the net - for the vertical dogleg segments to reduce the height as much as needed. Figure

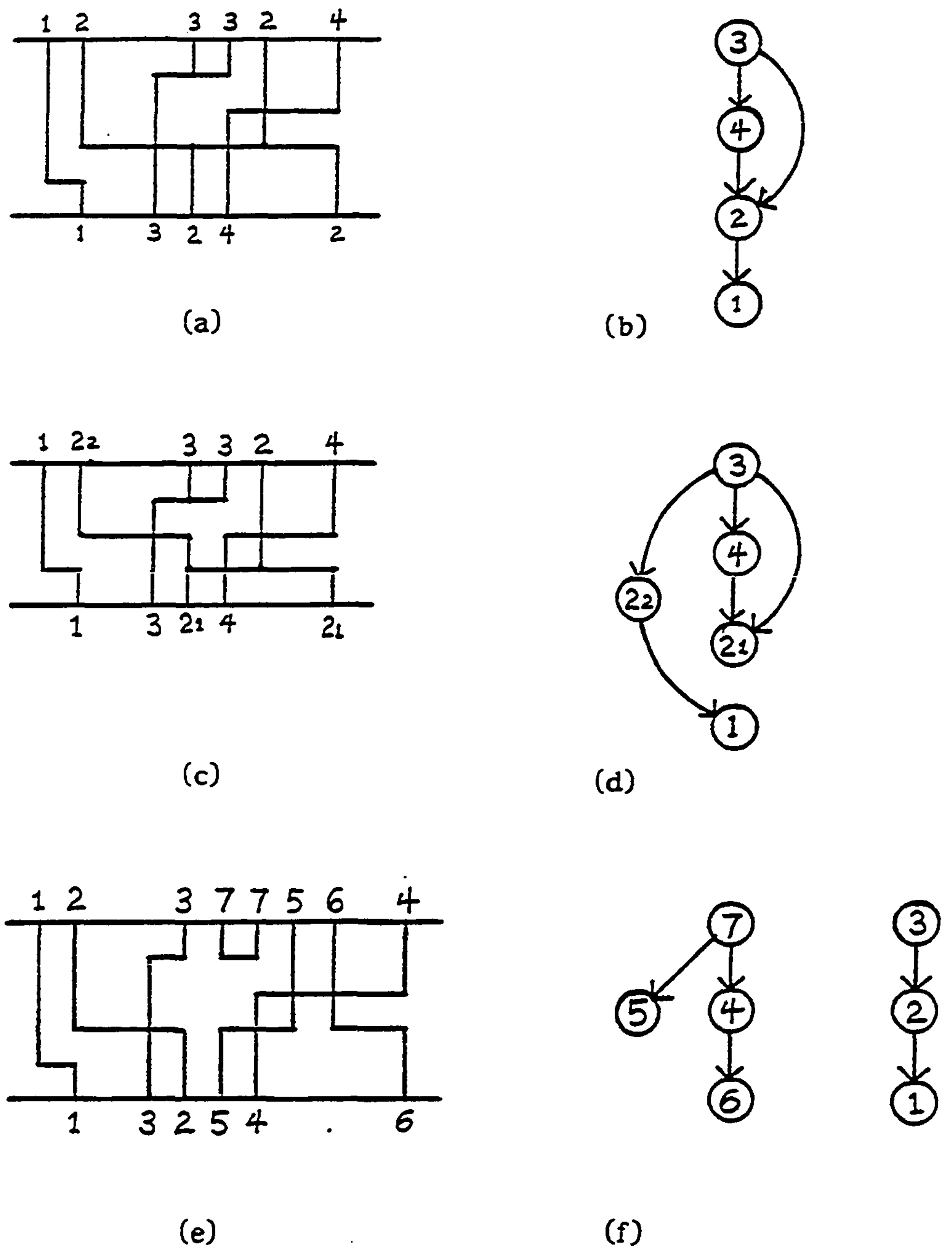
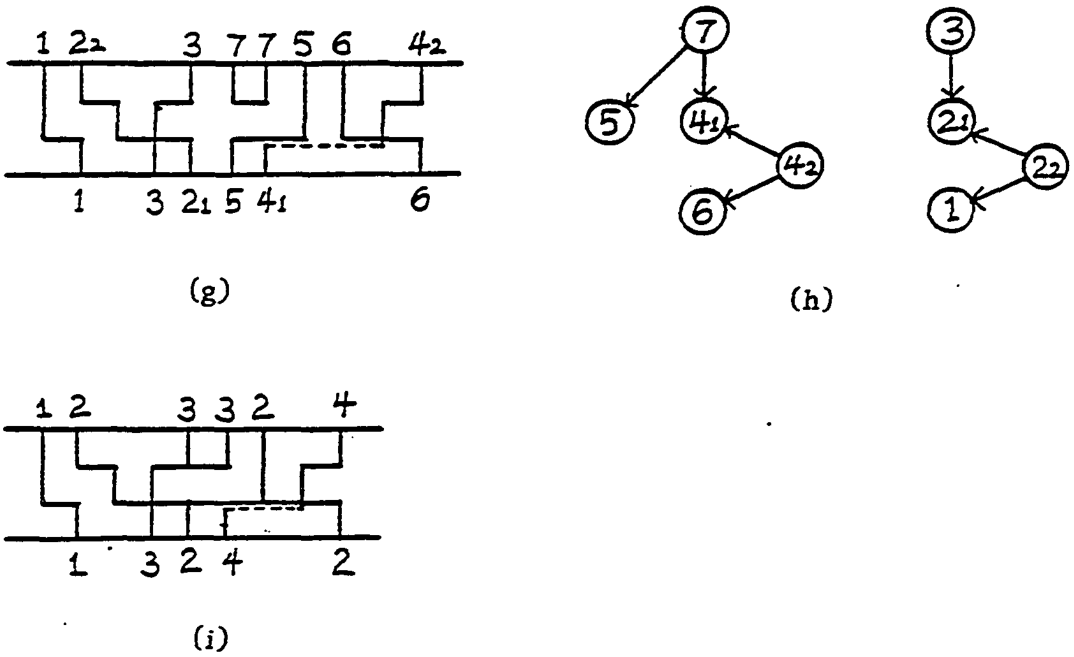


FIGURE 18. Partitioning the multiterminal nets.



(g)

(h)

(i)

FIGURE 18. Partitioning the multi-terminal nets (continued)

18-(i) is the final layout after the multi-terminal nets are restored.

6. EXPERIMENTAL RESULTS

Our router achieves a channel width which is as small as the lower bound in many cases. However, the routability conditions and selection of the sequence of executing the Merge algorithm may sometimes prevent minimum solutions. More discussion of these conditions will be presented in Chapter 7.

In most cases, reducing the height until $\lceil d/4 \rceil$ gives the minimum channel width, although more dogleg nets and vias than are necessary may be created. To prevent this situation and still maintain the minimum channel width for all cases, the height reduction may be determined interactively between $\lceil d/2 \rceil$ and $\lceil d/6 \rceil$. Figure 19 through Figure 24 are the routing results of our router and the performances are summarized in Table 2.

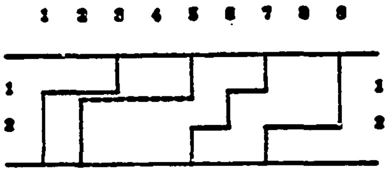


FIGURE 19. Example 1

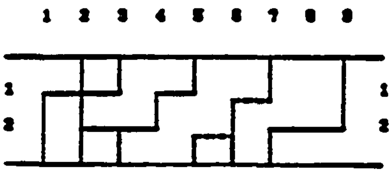


FIGURE 20. Example 2

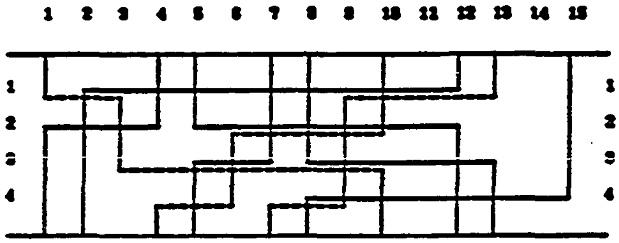


FIGURE 21. Example 3

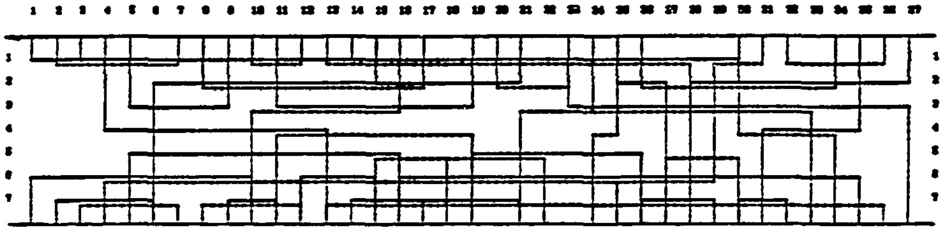


FIGURE 22. Example 4

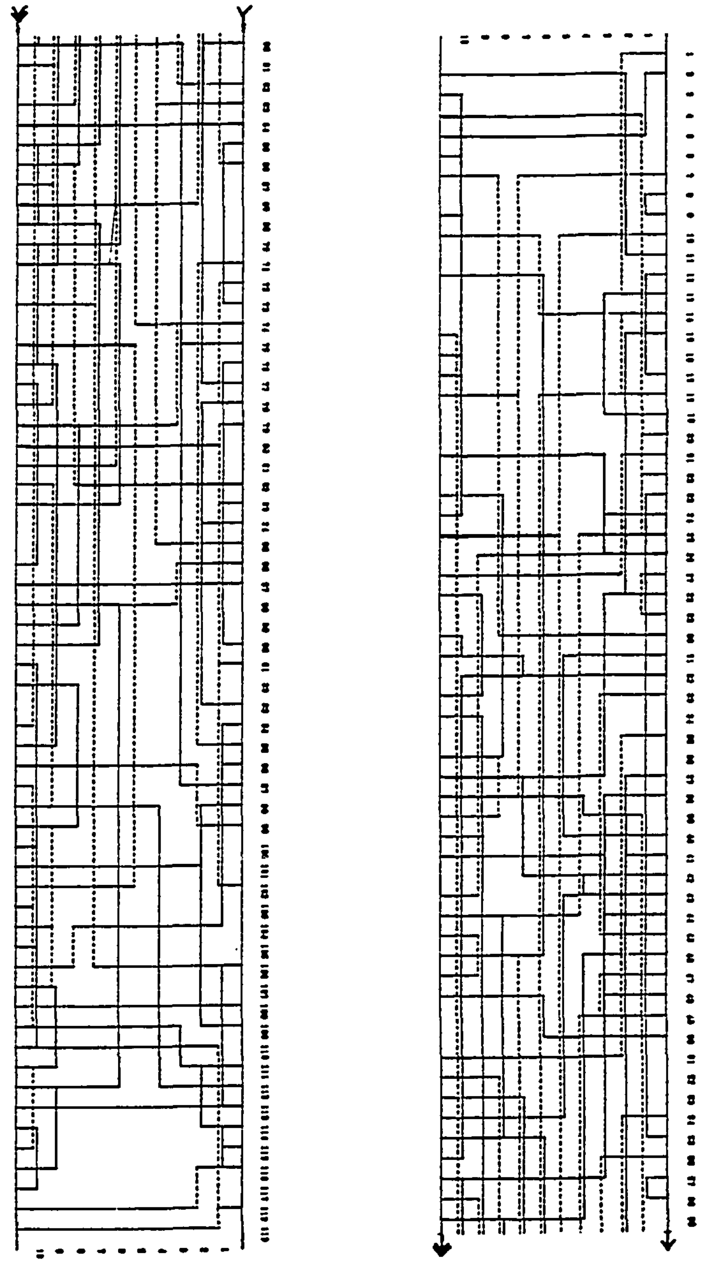


FIGURE 23. Example 5

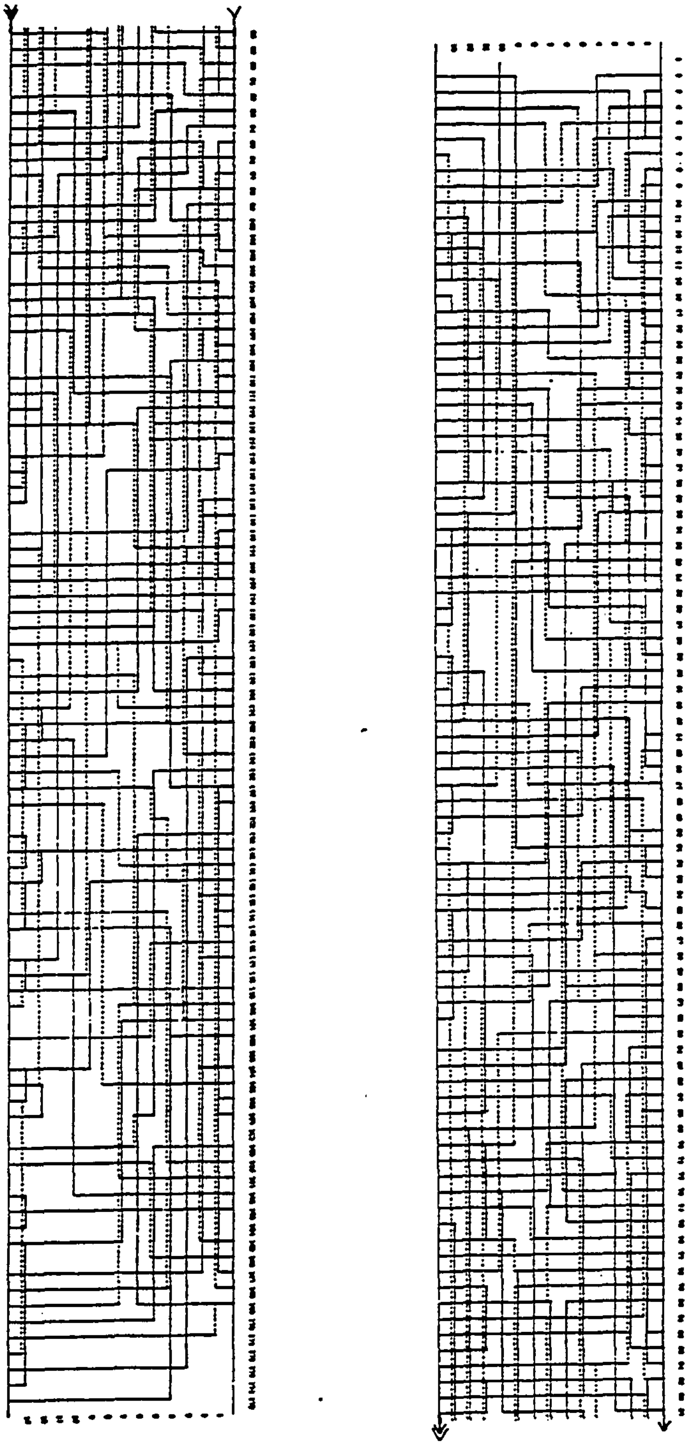


FIGURE 24. Example 6

TABLE 2. Summary of the routing results

Examples	Number of terminals	Density	Width of channel	Remark	Number of dogleg nets	Height after dogleg
example 1	8	2	2	$\lceil d/2 \rceil+1$	1	2
example 2	11	2	2	$\lceil d/2 \rceil+1$	2	2
example 3	18	7	4	$\lceil d/2 \rceil$	3	3
example 4	65	12	7	$\lceil d/2 \rceil+1$	2	3
example 5	186	17	10	$\lceil d/2 \rceil+1$	3	5
example 6	645	19	13	$\lceil d/2 \rceil+3$	10	5

Figure 23 is the example from Chen and Liu [49] where they obtained 13 for the channel width, which was optimum for a non-extensive dogleg HVH model. Applying our router, we got 10 as the channel width. Figure 24 is our solution to the so called 'difficult example' proposed by Deutsch [23]. This example has been tried using various channel routers. The results are compared in Table 3.

The time complexity of our algorithm is $O(d^2 \cdot n)$ where d is the channel density and n is the number of nets. The running time of our router is relatively fast. It took a few seconds of CPU time for each of the examples 1, 2, and 3 using a D.E.C. VAX 11/780. Examples 4, 5, and 6 took approximately 10 seconds, 30 seconds, and 90 seconds respectively using a D.E.C. VAX 11/780.

TABLE 3. Comparison of the results of the 'difficult example'

Router	Number of layers	Number of tracks
Deutsch [23]	2	21
Chan [52]	2	21
Yoshimura and Kuh [24]	2	20
Burstein [53]	2	19
Chen and Liu [49]	3	14
Ours	3	13

7. ROUTABILITY FOR CHANNEL ROUTING

The routability of a CRP (Channel Routing Problem) is determined by two conditions: whether the cycles of the vertical constraints graph are completely removed, and whether the given width of channel is enough for the worst case wiring condition. In this chapter, we are going to discuss the upper bounds and lower bounds of channel widths with respect to to routability.

As we discussed in Chapter 2, there have been two approaches to solve the channel routing problems: the directional model, which is sometimes called the Manhattan model, and the knock-knee model, where wires on different layers can cross each other or share a corner. We have seen in Chapter 5 that the width of channel in any two-layer model is lower bounded by channel density d. Leighton [54] shows that there exist CRPs with density d which can not be wired on two layers using fewer than 2d-1 tracks. The worst case channel width of 2d-1 is obtained by some authors [29] [30]. Preparata and Lipski [28] use three conducting layers for their knock-knee model. They show that any CRP of density d can be wired on a channel of width d, which is optimal under the condition that no overlap is allowed. A generalized knock-knee model was proposed by Hambrusch [30]. She analyzes the L-layer channel routing problem which allows up to K overlaps and provides $\lceil d/(K-1) \rceil$ as the lower bound of the channel width and $\lceil 7d/4 \rceil + c$ as the upper bound.

7.1 Vertical constraints in the directional model

The routability in directional type of channel routing depends on vertical constraints, terminal sites, density, horizontal constraints, and the channel width. Vertical constraints are of importance only for those models which use one layer for vertical wire segments, because vertical constraints are supposed to prevent the overlap of vertical wire segments. That is, a model like VHV is not restricted by vertical constraints and will be excluded from our discussion.

Cycles in the vertical constraints graph are critical with respect to routability and should be resolved before completion of the routing. One way to resolve the cyclic vertical constraints is to rearrange cells [55] such that there are no conflicts. Figure 25-(b) is the resolution of the cycle in Figure 25-(a) by reflection of the upper left cell.

Another way to resolve the cycles is to use a dogleg net as in Figure 25-(c). When the terminal sites are predetermined, making dogleg nets is the only way to resolve

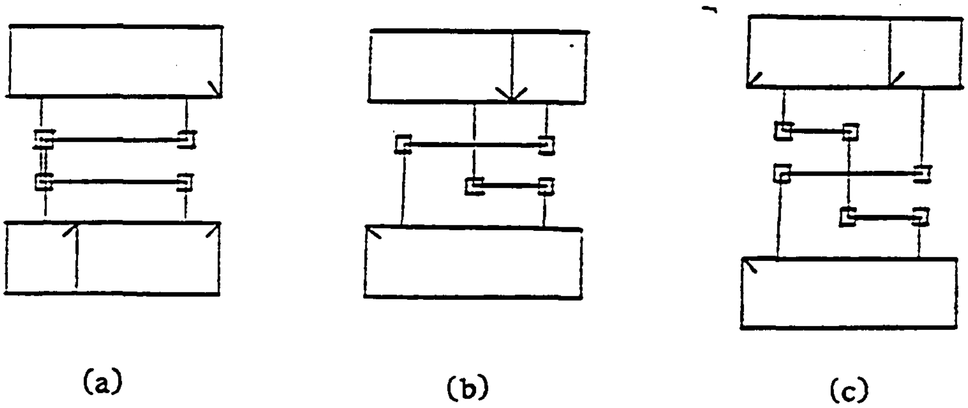


FIGURE 25. Resolving the cyclic vertical constraints

the cycles in VCG.

After cycles in the VCG are removed, the height of the VCG is another factor of routability condition. The width of channel shouldn't be smaller than the length of the longest path in the VCG in order not to violate the vertical constraints. If the above condition is not satisfied, a dogleg of a net in a longest path must be made to shorten the VCG. Thus, the unroutable condition regarding vertical constraints can be removed by successive application of the dogleg method.

7.2 Restrictions concerning terminal locations

In the previous section, we assumed that the dogleg nets can be generated wherever needed. However, before that assumption can be made, it is necessary to be certain that proper space is available in the channel for the extra horizontal segment and vertical segment of the dogleg net. The concept of 'tight set of nets' is defined as follow by Pinter [27].

Definition:

> A cycle in the vertical constraints graph is tight if the nets involved in it occupy a contiguous block of columns.

If a cycle is tight as Figure 26-(a), there is no place for a vertical dogleg segment between the two terminals of a net and the dogleg net can not be routed monotonically.

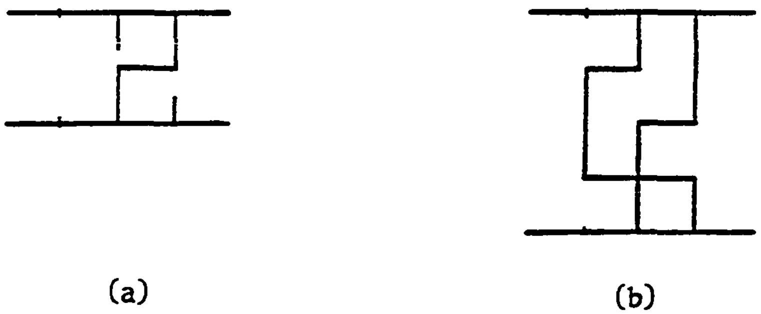


FIGURE 26. Tight set of nets

The dogleg net of Figure 26-(b) uses a detour path to locate a vertical dogleg segment outside the tight interval. It is obvious that the knock-knee model can route this problem using two horizontal tracks with a detour path. A detour net is not desirable in general because it not only uses longer horizontal wires but it increases the local density by one.

Another investigation regarding the terminal locations and routability was done by Frank et al. [56].

Theorem:

> If there are no two consecutive terminals on the bottom side and on the top side of a channel, d+1 horizontal tracks are sufficient for the two-layer channel routing problems.

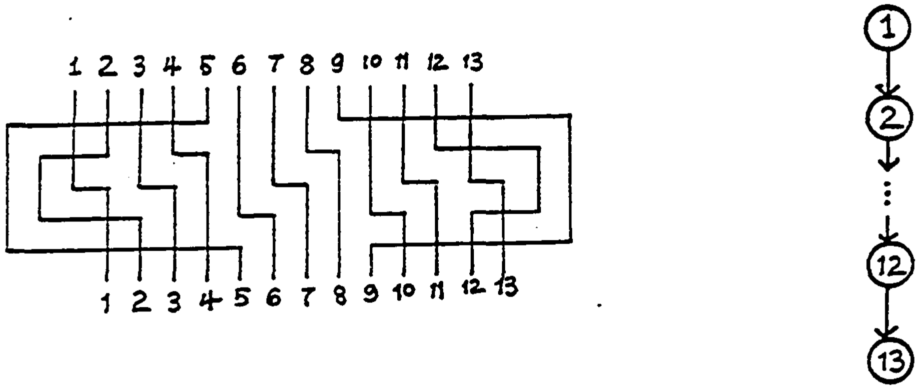
If more dense terminals are allowed, the upper bound increases.

Theorem:

> If there are no three consecutive terminals and every path is a left path, the interconnection problems can be solved on d+3 tracks.

7.3 Lower bound as the function of the number of nets

We have discussed two parameters, d and h, which determine the channel width in channel routing problems. As the third factor in the lower bound of channel width, the number of nets is suggested by Brown and Rivest [57]. They claim that there exist such cases where at least $\sqrt{2n}$ tracks are necessary for the solution of certain channel routing problems which have n nontrivial two-terminal nets. A good example of this case is a 'shift-right-one' problem, as in Figure 27, where the optimal solution uses five tracks although the density is two.



(a) (b)

FIGURE 27. Shift-right-one example

7.4 Upper bound of channel width

In our channel routing algorithm, we use the intensive three-layer dogleg method which not only removes the cycles in VCG in order to make the problem routable but also reduces the height of VCG to produce the trivial lower bound of $\lceil d/2 \rceil$. The above argument, however, is not applicable if the cycle is tight or the proposed dogleg net can not find a proper place for the vertical dogleg segment. We are going to restrict the problem with an assumption before we discuss the upper bound. The assumption we have made with our model is that there is space for a vertical dogleg segment in at least a column between the left terminal and right terminal of the proposed dogleg net. This assumption is reasonable practically because the number of dogleg nets needed is much less than the number of all the nets, and the router has freedom to choose a dogleg net among a group of nets, one of which needs to be doglegged according to locational priority.

In our model, dogleg nets are created until the height of the longest VCG becomes less than $\lceil d/2 \rceil$. Thus, the influence of the longest path in VCG upon the channel width is overridden by the density parameter, which then becomes the only factor that determines the channel width.

Under this condition, the upper bound of the channel width of the three-layer model will be discussed.

If we limit our concern to only the layout of horizontal segments, we need not distinguish multi-terminal nets from two-terminal nets, and we can derive the upper bound of channel width by applying the following theorem of Gallai [56].

Theorem:

Given a family F of closed intervals on a line, F can be partitioned into k parts F_i, $(i=1,2,\cdots k)$, such that each F_i consists of disjoint intervals if and only if no point of the reference line belongs to more than k intervals of F.

This theorem can be interpreted for channel routing problems through the following corollaries. The Corollary 1 corresponds to the above theorem if we interpret 'horizontal wire segments' as 'closed intervals', 'without overlapping' as 'disjoint', and 'd' as 'k'.

Corollary 1:

A set of horizontal wire segments of nets with density d can be placed on a single layer channel using d tracks without overlapping one another.

Corollary 2:

A set of horizontal wire segments with density d can be placed on a two-layer channel using $\lceil d/2 \rceil$ tracks without touching one another.

The above corollaries are effective for the nets with vertical as well as horizontal wire segments if the vertical constraints can be removed. Although the cyclic VCG and the long VCG are resolved in our extensive dogleg method, the following cases remain which require two additional tracks to the trivial lower bound. Figure 28-(c) is a combined case of Figure 28-(a) and Figure 28-(b), each of which requires one additional track over the case of Corollary 2.

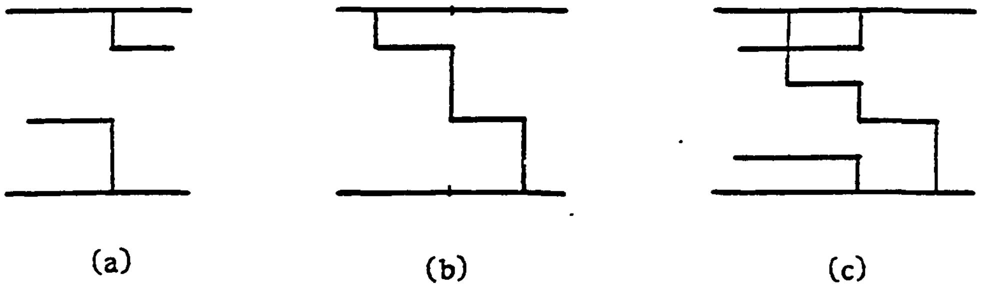


FIGURE 28. The case of two additional tracks

From the above observation and the fact that a column in a channel is not allowed to locate more than one vertical dogleg segment, $\lceil d/2 \rceil + 2$ is the upper bound of channel width for the optimal router of HVH type, if it exists. However, the channel routing problem is believed to be NP-complete and there may not be a generally optimal algorithm. Our algorithm is another better heuristic. We believe that ou:

router is not far from optimal. Almost all the experiments we have made provides the channel widths between $\lceil d/2 \rceil$ and $\lceil d/2 \rceil + 2$.

8. DISCUSSION AND FUTURE WORK

In this thesis research, we establish a new three-interconnection-layer model for the IC routing problem and develop an efficient routing tool for that model. Our router can be applied to the more general class of routing problems; it eliminates some of the restrictions which have been made by other existing routers.

By adding one more layer for interconnection, we could reduce the lower bound of channel width to half of the conventional two-layer model. The modified dogleg method and the merge algorithm for three-layer model are used to achieve the lowest lower bound among existing routers. Experimental results show that the performance of our router is not far from optimal.

However, our model assumes that the 'tight sets of nets' [27] have been avoided in the placement stage of the layout - although there is little chance of existence of such nets practically. If the 'tight sets of nets' have not been completely removed, our router fails to complete the wire routing because our router lacks the capability of 'detour' wiring.

Two restrictions apply to our router; a two-terminal net may not be doglegged twice, and a column in a channel may not locate more than one vertical dogleg segment. These restrictions are made for reducing the complexity of the algorithm. These restrictions, however, do not seem to affect the performance in most cases.

In an attempt to provide more flexibility to the Merge algorithm, our dogleg algorithm reduces the height of the vertical constraints tree further down to $\lceil d/4 \rceil$ (we only need the height of $\lceil d/2 \rceil$ to get a lower bound of $\lceil d/2 \rceil$). This obviously increases the number of dogleg nets (and vias). Although the procedure 'Release-Dogleg' is used to straighten out unnecessary dogleg nets, some of them may be left.

In this thesis, we also present some methods for global routing and power/ground routing for the three-layer model. The interface of the channel router and the global router is another problem. One important problem in combining rectangular channels is the switch box (two-dimensional channel) routing problem. This problem is more complex and often require more width than the adjacent channels. We are planning to extend our research to the three-layer switch box routing problem.

9. APPENDIX: GLOSSARY

(1) routing region: A part of a chip area which is not occupied by circuit building blocks and is used for wiring of nets. This space consists of rectangular subregions, which are called channels.

(2) grid model: A model where the chip area is represented by a grid plane. In this model, wires are considered as line segments along the grid lines and circuit blocks are rectilinear shapes which are bounded by grid lines.

(3) global cell: A unit area in the global routing stage whose internal wiring is not considered in this routing stage.

- edge capacity: The maximum number of wires allowed to cross an edge of a global cell.

(4) Manhattan distance: Rectilinear distance between two points on a grid plane.

(5) track assignment = detailed routing = channel routing

(6) channel: A rectangular region for interconnections between terminals.

- tracks: Horizontal grid lines of a channel which can be used for locating horizontal wire segments.
- columns: Vertical grid lines of a channel which

can be used for locating vertical wire segments.

- side: Top or bottom boundary of a channel.
- width: Number of tracks in a channel.
- local density: Number of horizontal wire segments crossing or touching a column of a channel.
- (channel) density: Maximum of local densities for columns within a channel.
- span: A section of a channel where the local density is equal to the density.

(7) CRP (Channel routing problem): A problem of finding a layout of wires in a channel from given interconnection data.

- directional model: A model to solve CRP's where each interconnection layer allows only one directional - horizontal or vertical - wire segments. Two-layer directional model is sometimes called Manhattan model.
- knock-knee model: There are no directional restrictions for each layer and no overlap of parallel wires are allowed. Instead, a grid point can be shared by two different nets.
- K-fold overlap model: An extension of knock-knee model which has L interconnection layers and allows K overlaps of wire segments.

(8) over-the-cell routing: A routing model where signal nets can pass over the circuit building blocks.

(9) terminal: A point on either side of a channel where an electrical connection to the outside of the channel is to be made.

(10) pad: A point on the perimeter of a chip where electrical connection to the outside of the chap is to be made.

(11) net: A continuous path which connects terminals of equal electrical potential.

- Nets: A data set which represents a given set of nets, each of which is characterized by its terminal sites and locations of horizontal and vertical segments in the channel.
- Net-lists: A set of triplets (top, middle, bottom) of nets which occupy a column.

(12) via: An inter-layer electrical connection which connects wire segments on different layers.

(13) dogleg net: A net whose horizontal segment is partitioned into two sections and located on different tracks in a channel.

- vertical dogleg segment: One of the vertical segments of a dogleg net which is not connected directly to the terminals.

(14) vertical constraints: A set of precedence relations between nets in a channel, e.g., if two or three vertical segments of nets are placed on a column, these nets are ordered in a sequence such that the net of higher-located vertical segment precedes the net of lower-located vertical segment.

- VCG (vertical constraints graph): A directed graph that represents the vertical constraints of all the nets in a CRP.
- height: The length of the longest path in a VCG. Height is defined only when the VCG is acyclic.

(15) horizontal constraints: A set of closed intervals, each of which shows the left-end and right-end of every horizontal segment in a channel.

(16) CV (column-vacancy): A set of closed intervals, each of which represents the empty space of a column in a channel.

(17) TLO (track-layer-occupancy): A data set which represents the location of every horizontal segment on every track by its layer, interval, and net number.

(18) Dogleg: An algorithm which is used to resolve cycles in VCG or to cut down long paths in VCG.

(19) Merge: An algorithm which shifts every possible segments upward as an attempt to compact the channel.

- fromtrack(fromlayer): The track(layer) location of horizontal segment which is going to be shifted.
- totrack(tolayer): The track(layer) location of horizontal segment after execution of the Merge.

10. BIBLIOGRAPHY

1. Heller, W. R.; Mikhail, W. F.; and Donath, W. E. "Prediction of wiring Space Requirements for LSI." *Journal of Design Automation and Fault-tolerant Computing* (May 1978): 117-144.

2. Soukup, Jiri. "Circuit Layout." *Proceedings of the IEEE* 69, No. 10. (Oct. 1981): 1281-1304.

3. Lauther, U. "Cell Based VLSI Design System." In *Hardware and Software Concepts in VLSI* 480-494. Edited by Guy Rabbat. New York: Van Nostrand Reinhold Company, 1983.

4. Schweikert, D. G. "A 2-dimensional Placement Algorithm for the Layout of Electrical Circuits." *Proceedings of Design Automation Conference* San Francisco, CA., (1976): 408-416.

5. Hanan, M. and Kuh, E.S. 'A Study of Placement Technique." *Journal of Design Automation and Fault-tolerant Computing* 1, No. 1 (1976): 28-61.

6. Preas, B. T. and Gwyn, C. W. "Methods for Hierarchical Automatic Layout of Custom LSI Circuit Mask." *Proceedings of the 15th Design Automation Conference* Las Vegas, Nev., June 1978: 206-212.

7. Schweikert, D. G. and Kernighan, B. W. "Partitioning Circuit Layout." *Proceedings of the 9th Annual Design Automation Workshop* Dallas, TX. (June 1972).

8. Breuer, M. A. "Min-Cut Placement." *Journal of Design Automation and Fault-tolerant Computing* 1, No. 4 (Oct. 1977): 343-362.

9. Breuer, M. A. "A Class of Min-Cut Placement Algorithm." *Proceedings of the 14th Design Automation Conference* New Orelans (1977): 284-290.

10. Lauther, U. "A min-Cut Placement Algorithm for General Cell Assemblies Based on a Graph Representation." *Proceedings of the 16th Design Automation Conference* San Diego (1979): 1-10.

11. Lee, C. Y. "An Algorithm for Path Connections and Its Applications." IRE Transactions on Electronic Computers VEC-10 (Sep. 1961): 346-365.

12. Chen, N. P.; Hsu, C. P.; and Kuh E. S. "The Berkeley Building-Block (BBL) Layout System for VLSI Design." In VLSI '83, 37-44. Edited by F. Anceau and E. J. Aas. North-Holland: Elsevier Science Publishers B. V., Aug. 1983.

13. Rivest, Ronald L. "The "PI" (Placement and Interconnect) System." Unpublished Manuscript. Lab. for Computer Science, MIT, Feb. 1981.

14. Antognetti, P.; Pederson, D. O.; and Man, H. De, ed. Computer Design Aids for VLSI Circuits Rockville, Maryland: Sijthoff & Noordhoff, 1981.

15. Ullman, Jefferey D. Computational Aspects of VLSI. Rockville, Maryland: Computer Science Press, 1984.

16. Breuer, M. A. and Carter, H. W. "VLSI Routing." In Hardware and Software Concepts in VLSI 368-405. Edited by Guy Rabbat. New York: Van Nostrand Reinhold Company, 1983.

17. Hightower, D. W. "A Solution to Line Routing Problems on the Continuous Plane." Proceedings of the 6th Design Automation Workshop (June 1969): 1-24.

18. Hashimoto, A. and Stevens, J. "Wire Routing by Optimizing Channel Assignment within Large Apertures." Proceedings of the 8th Design Automation Workshop (1971): 155-169.

19. Tsui, Raymond Y. and Smith, Robert J. "A High-Density Multilayer PCB Router Based on Necessary and Sufficient Conditions for Single Row Routing." Proceedings of the 18th Design Automation Conference (1981): 372-381.

20. Tsukiyama, Shuji; Kuh, Ernest S.; and Shirakawa, Isao. "On the Layering Problem of Multilayer PWB wiring." IEEE Transactions on Computer-Aided Design CAD-2, No. 1 (Jan. 1983): 30-38.

21. Ting, Benjamin S.; Kuh, Ernest S.; and ShirakaWa, Isao. "The Multylayer Routing Problem: Algorithma and Necessary and Sufficient Conditions for the Single-Row Single-Layer Case." *IEEE Transactions on Circuits and Systems* CAD-23, No. 12 (Dec. 1976): 768-777.

22. Kernichan, B.; Schweikert, D.; and Persky, G. "An Optimal Channel-Routing Algorithm for Polycell Layouts of Integrated Circuits." *Proceedings of the 10th Design Automation Workshop* (1973): 50-59.

23. Deutsch, D. N. "A Dogleg Channel Router." *Proceedings of the 13th Design Automation Conference* (1976): 425-433.

24. Yoshimura, T. and Kuh, E. S. "Efficient Algorithms for Channel Routing." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* Cad-1, No. 1 (Jan. 1982): 25-35.

25. Yoshimura, T. "An Efficient Channel Router." *Proceedings of the 21th Design Automation Conference* (1984): 38-44.

26. Thompson, C. D. "A Complexity Theory for VLSI." *Technical Report CMU-CS-80-140* Ph.D. dissertation, Carnegie-Mellon University, Aug. 1980.

27. Pinter, Ron Yair. "The Impact of Layer Assignment Methods on Layout Algorithms for Integrated Circuits." *Ph.D. dissertation*, MIT, Dec. 1982.

28. Preparata, Franco P. and Lipski, Witold. "Optimal Three-Layer Channel Routing." *IEEE Transactions on Computers* C-33, No. 5 (May 1984): 427-437.

29. Rivest, Ronald L.; Baratz, Alan E.; and Miller, Gary. "Provably Good Channel Routing Algorithms." *Proceedings of Carnegie-Mellon Conference on VLSI* (Oct. 1981): 153-159.

30. Hambrusch, S. E. "Channel Routing Algorithms for Overlap Models." *IEEE Transactions on Computer-Aided Design* CAD-4, No.1 (Jan. 1985): 23-30.

31. Hambrusch, S. E. "Minimizing Contact Points and Using Overlap on Two Layers." *Technical Report CSD-TR-455* Purdue University, Aug. 1983.

32. Brady, Martin L. and Brown, Donna J. "Arbitrary Planar Routing with Four Layers." Proceedings of the 1984 Conference on Advanced Research in VLSI MIT, (Jan. 1984): 194-201.

33. Raghavan, Prabhakar and Thompson, Clark D. "Randomized Routing in Gate-Arrays." Report No. UCB/CSD 84/202, Computer Science Division, UC Berkeley, Sep. 1984.

34. Leighton, Frank Thomson and Rosenberg, Arnold L. "Automatic Generation of Three-Dimensional Circuit Layouts." Proceedings of the 1983 IEEE International Conference on Computer Design. (1983).

35. Syed, Zahir A. and Gamal, Abbas E. "Single Layer Routing of Power and Ground Networks in Integrated Circuits." Journal of Digital Systems 6, No. 1 (Spring, 1982): 53-63.

36. Mead, C. and Conway, L. Introduction to VLSI Reading, MA.: Addison Wesley, 1980.

37. Rothermel, H-J. and Mlynski, D. A. "Computation of Power Supply Nets in VLSI Layout." Proceedings of the 18th Design Automation Conference (1981): 37-42.

38. Lie, Margaret and Horng, Chi-Song. "A Bus Router for IC Layout." Proceedings of the 19th Design Automation Conference (1982): 129-132.

39. Read, John W., ed. Gate Arrays Design Techniques and Applications New York: McGraw-Hill, 1985.

40. Moulton, Andrew. "Laying the Power and Ground Wires on a VLSI Chip," Proceedings of the 20th Design Automation Conference (1983): 745-755.

41. Moulton, Andrew S. "Routing the Power and Ground Wires on a VLSI Chip." M.S. thesis, MIT, May, 1984.

42. Ting, Benjamin S. and Tien, Bou Nin. "Routing Techniques for Gate Arrays." IEEE Transactions on Computer-Aided Design CAD-2, No. 4 (Oct. 1983): 301-312.

43. Li, J-T. and Marek-Sadowska, M. "Global Routing for Gate Array." Memorandom No. UCB/ERL M83/60 Electronics Research Laboratory, University of California, Berkeley, Sept. 1983.

44. Karp, R. M.; Leiton, F. T.; Rivest, R. L.; Thompson, C. D.; Vazirani, U.; and Vazirani, V. "Global Wire Routing in Two-Dimensional Arrays." Proceedings of the 24th Annual Symposium on Foundations of Computer Science Tucson, AZ. (Oct. 1983): 453-459.

45. Patel, Ash M.; Soong, Norman L.; and Korn, Robert K. "Hierarchical VLSI Routing - An Approximate Routing Procedure." IEEE Transactions on Computer-Aided Design CAD-4, No. 2 (April 1985): 121-126.

46. Hasset, James E. "Automated Layout in ASHLAR: An Approach to the Problems of 'General Cell' Layout for VLSI." Proceedings of the 19th Design Automation Conference (1982): 777-784.

47. Chen, K. A.; Feuer, M.; Khokhani, K. H.; Nan, N.; and Schmidt, S. "The Chip Layout Problem: An Automatic wiring Procedure." Proceedings of the 14th Design Automation Conference (1977): 298-302.

48. Baratz, Alan. "Algorithm for Integrated Circuit Signal Routing." Ph.D. dissertation, MIT, Aug. 1981.

49. Chen, Yun Kang and Liu, Mei Lun. "Three-Layer Channel Routing." IEEE Transactions on Computer-Aided Design CAD-3, No. 2 (April 1984): 156-163.

50. Ciesielski, M. J. and Kinnen, E. "An Analytical Method for Compacting Routing Area in Integrated Circuits." Proceedings of the 19th Design Automation Conferance (1982): 30-37.

51. Heyns, W. "The 1-2-3 Routing Algorithm or The Single Channel 2-step Router on 3 Interconnection Layers." Proceedings of the 19th Design Automation Conference (1982): 113-120.

52. Chan, Wan S. "A New Channel Routing Algorithm." Third Caltech Conference on Very Large Scale Integration (1983): 117-139.

53. Burstein, Michael. "Hierarchical Channel Router." Proceedings of the 20th Design Automation Conference (1983): 591-597.

54. Leighton, F. T. "New Lower Bounds For Channel Routing." Unpublished manuscript Lab. for Computer Science, MIT, Jan. 1982.

55. Persky, G. "PRO - An Automatic String Placement Program for Polycell Layout." Proceedings of the 13th Design Automation Conference (1976): 417-424.

56. Frank, A.; Levai, P.; Mozes, J.; Scsaurszki, P.; and Tardos, E. "Sufficient Conditions for Solvability of Channel Routing Problems." Proceedings of International Symposium on Circuits and Systems (1984): 1459-1461.

57. Brown, Donna J. and Rivest, Ronald L. "New Lower Bounds for Channel Width." Proceedings of the 1981 Carnegie-Mellon Conference on VLSI Systems and Computers (Oct. 1981): 153-159.

11. ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to my major professor, Dr. Arthur V. Pohm, for his encouragements and guidance during this research. His judgement and technical advice were invaluable for my work.

Special thanks are extended to Dr. Harry Hale, Dr. Robert Stewart, Dr. Charles Wright and Dr. A. E. Oldehoeft for giving suggestions and guidance as my graduate committee.

I am grateful for the financial support I received from the Department of Electrical and Computer Engineering at Iowa State University.

I wish to thank my parents for their love and encouragements. My father inspired me to pursue excellence and taught me the value of education and freedom. His confidence in me in all circumstances has brought me thus far.

Finally, I am most grateful to my family; my wife Eun for her continued faith and selfless help throughout the writing of this thesis, little David for being a constant source of happiness and liveliness.