INFORMATION TO USERS

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

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

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

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

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
YOR: A yield optimizing routing algorithm by minimizing critical areas and vias

Chang, Ting-Mao, M.S.
The University of Arizona, 1991
YOR: A Yield Optimizing Routing Algorithm by Minimizing Critical Areas and Vias

by

Ting-Mao Chang

A Thesis Submitted to the Faculty of the
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
In Partial Fulfillment of the Requirements
For the Degree of
MASTER OF SCIENCE
WITH A MAJOR IN ELECTRICAL ENGINEERING
In the Graduate College
THE UNIVERSITY OF ARIZONA
1991
STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: [Signature]

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

[Signature] Sy-Yen Kuo
Assistant Professor of
Electrical and Computer Engineering

Date 12/6/1990
To my parents and my family
Acknowledgements

During my studies at the University of Arizona, I have met and learn about many people. Most of these have help me in a variety of ways. Since it is not possible to mention by name all who advanced my life, I thank all of you and express my sincere gratitude.

I am deeply indebted to Professor Sy-Yen Kuo, my thesis advisor, for his invaluable help and guidance in development of this work. I am also greatly thankful to Dr. Fredrick J. Hill and Dr. Jerzy W. Rozenblit for their comments and for taking time as my commitee members.

I also indebted to those who help my family in these two years when I was not there. Finally, I will thank my parents for their endless love and support.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>6</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>7</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>8</td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>9</td>
</tr>
<tr>
<td>1.1. General Background</td>
<td>9</td>
</tr>
<tr>
<td>1.2. The routing problem</td>
<td>11</td>
</tr>
<tr>
<td>2. CRITICAL AREAS</td>
<td>14</td>
</tr>
<tr>
<td>2.1. Spot Defects and Critical Areas</td>
<td>14</td>
</tr>
<tr>
<td>2.2. Critical Area Computation</td>
<td>18</td>
</tr>
<tr>
<td>2.3. Via Weights</td>
<td>19</td>
</tr>
<tr>
<td>2.4. Lower Bound on the Size of Critical Areas</td>
<td>23</td>
</tr>
<tr>
<td>3. DEFINITIONS</td>
<td>27</td>
</tr>
<tr>
<td>3.1. Vertical Constraint Graph (VCG)</td>
<td>28</td>
</tr>
<tr>
<td>3.2. Zone representation of horizontal segments</td>
<td>29</td>
</tr>
<tr>
<td>3.3. Weighted Interval Graph (WIG)</td>
<td>32</td>
</tr>
<tr>
<td>3.4. Net Burying</td>
<td>35</td>
</tr>
<tr>
<td>3.5. Net Floating</td>
<td>36</td>
</tr>
<tr>
<td>4. ALGORITHM DESCRIPTIONS</td>
<td>37</td>
</tr>
<tr>
<td>4.1. Merging</td>
<td>38</td>
</tr>
<tr>
<td>4.2. Track Assignment</td>
<td>43</td>
</tr>
<tr>
<td>4.3. Via Shifting and Net Bumping</td>
<td>45</td>
</tr>
<tr>
<td>4.4. Minimize Critical Areas with Redundant Tracks</td>
<td>49</td>
</tr>
<tr>
<td>5. EXPERIMENTAL RESULTS</td>
<td>53</td>
</tr>
<tr>
<td>6. CONCLUSIONS</td>
<td>60</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>62</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

2.1. The routing geometry is a set of parallel conducting paths. 15
2.2. The critical area between parallel wires. 17
2.3. Three types of critical areas. 17
2.4. Critical area calculation. 20
2.5. Example design rules. 22
2.6. The lower bound of a circuit. 26

3.1. Vertical constraint graph. 29
3.2. Cyclic in vertical constraint graph. 29
3.3. Use dog-legging to minimize the number of tracks. 30
3.4. A zone representation. 31
3.5. Interval graph of a net list. 32
3.6. The edge weight $W_e$ between $N_1$ and $N_2$. 33
3.7. Weighted interval graph. 34
3.8. Net burying. 35
3.9. Net floating: (a) before floating (b) after floating. 36
3.10. Partial net floating: (a) before floating (b) after floating. 36

4.1. The yield optimizing routing (YOR) algorithm. 38
4.2. Zone representation and VCG before net merging. 39
4.3. Zone representation and VCG after net merging. 42
4.4. Track assignment procedure. 44
4.5. Example of $TOP(V_v)$, $SON(n)$, and $SON(n)$. 46
4.6. Via shifting procedure. 48
4.7. Example via shifting to minimize critical areas. 49
4.8. Use of net bumping to eliminate critical areas. 50
4.9. Net bumping procedure. 51
4.10. Track assignment with redundant tracks procedure. 52
LIST OF TABLES

5.1. Reduction on the number of vias. ............................. 54
5.2. Reduction on the number of critical area units. ............... 55
5.3. Relationship between channel density and critical area reduction. .... 55
5.4. Chip yields with $D_f = 2.5/cm^2$. ............................ 58
5.5. Chip yields with $D_f = 5.0/cm^2$. ............................. 58
5.6. Percentage yield improvement. ................................. 59
ABSTRACT

Traditionally, the goal of channel routing algorithms is to route the nets with as few tracks as possible to minimize the chip area and achieve 100 percent connection. However, the manufacturing yield may not reach a satisfactory level if care is not taken to reduce the critical areas which are susceptible to defects. These critical areas are caused by the highly compacted adjacent wires and vias in the routing region. A new channel routing algorithm is presented in this paper to deal with this problem. Our approach is to systematically eliminate critical areas by floating, burying, and bumping net segments as well as shifting vias. The yield optimizing routing (YOR) algorithm also minimizes the number of vias since vias in a chip increase the manufacturing complexity which again degrades the yield. The algorithm has been implemented and applied to benchmark routing layouts in the literature. The experimental results show that large reduction in the number of critical areas and significant improvement in yield are achieved, particularly for practical size channels such as the Deutsch's difficult problem.
CHAPTER 1

INTRODUCTION

1.1 General Background

The objectives of integrated circuit design and manufacturing have always been to improve performance and to reduce cost. These goals have been met through increased circuit complexity, smaller geometries, and larger chips. However, as the size and complexity of an integrated circuit continue to increase, there will be a persistent need to limit defect levels and thereby maintain the yield/cost advantages of very large scale integration (VLSI) chips.

Routing is an important problem in VLSI layout design. The primary requirement of routing is to make 100 percent connection that minimizes the chip area [1]. There are often additional objectives such as minimizing the total wire length, minimizing the number of vias, handling prerouted nets or critical nets, for instance. However, a compacted chip with smaller area [2,3] means highly dense routing regions in the chip and will bring wires
and components much closer to each other. Therefore, adjacent wires and devices become more vulnerable to manufacturing defects which reduces yield and increases testing costs. A defect such as extra material or missing material will cause a short or an open circuit between two wires or devices which should be disjoint or connected, respectively. Faults caused by the defects will reduce the yield of the wafer. The yield losses of VLSI manufacturing caused by physical defects were investigated in several papers [4,5]. Some failures can be avoided by appropriately designing in the layout phase. Based on the previous research in yield modeling, some layout geometries were found to generate lower yield. From the results in [6], two types of faults, line stuck-at faults and bridging faults, are accounted for 58% of total faults in VLSI. It was also shown that layout topology has profound impact on the likelihood of certain faults and fault types. Hence, to prevent layout geometries which can cause yield losses during the channel routing process will be of great help in improving the yield.

But very little has been done in the area of critical area minimizing routing algorithms. One notable exception is the DTR algorithm [7] which is a simple routing algorithm to reduce critical areas. It considered only on reducing critical areas in the horizontal layer of the two routing layers and didn’t elaborate on other potential ways. For example, minimizing the number of vias can also reduce critical areas and improve yield. Several papers have been published [8,9] in this area. However, most of them put force on minimizing the number of vias and neglect the effect on critical areas. Since reducing the number of vias may introduce new critical areas, these results cannot be applied directly here.
In our approach, we will minimize the critical areas on both routing layers and minimize the critical areas caused by vias in the routing region to improve the yield. Chapter 2 gives the definition of critical areas and the lower bound of their sizes. Fundamental concepts on minimizing critical areas which will be used in the development of the algorithm are shown in Chapter 3. Chapter 4 outlines the details of the yield optimizing channel routing algorithm (YOR). Experimental results are presented in Chapter 5 followed by the conclusions in Chapter 6.

1.2 The routing problem

The contemporary design methodology is using cell-based design styles wherein cells, selected from a library, are arranged on a plan and interconnected by wiring. Following the placement, components are arranged on a plane and the task remains to connect the electrical equipotential terminal among the components to make them work. For a printed circuit board, the components are IC packages, and electric connections are made with a metal etching process. Metal is uniformly deposited on a carrier surface and unnecessary metal etched away, leaving wire lines. Connections between layers are made by drilling holes through the carrier and plating them with metal. In an integrated circuits, one or two layers of metal separated by insulating layers of oxide are deposited and etched above the silicon to form wire lines. Holes are left in the oxide to make connections between layers. The situation in both cases are similar. The router determine the wire geometry on the layout plan and position of the vias, or holes, to interconnect the layers. The following terminology will be used through this thesis:
1. **Net** - A set of points to be electrically connected.

2. **Layer** - The wiring surface used for interconnection.

3. **Segment** - A straight piece of wire on a single layer, which comprises a portion of the wiring path used to connect a net. The segment in horizontal direction is a horizontal segment, and in vertical direction is a vertical segment.

4. **Via** - A mechanism (hole) for connecting segments on different layers.

5. **Branch** - The segment of a net which connected to a via.

6. **Track** - A line along which the horizontal segments are placed.

In the computer aided design system, the routing problem is modeled by a set of boxes with bounding pins on their periphery and a set of signal nets which specify which pins have to be electrically connected by wires. The routing process employs a *divide and conquer* approach which makes automatic routing possible. A *global router* is used to decompose a large routing surface into small and manageable routing regions and assign each net to a set of routing regions. The routing region is called *channel* if it consists of a rectangular space between two parallel rows of pins or terminal. The location of these terminals are fixed along two opposite sides of the channel, while the location of any pins on the remaining two sides is determined during routing. If fixed pins are located on all four sides of a region, the region is called *switching box*. Following global routing, a *detail routing* is applied. The detail routing for a channel is called *channel routing* and for a switching box is *switching box routing*.

Since most of the routing problems are *NP complete*, acceptable solution are based on good heuristics that yield reasonable time and storage complexities but guarantee
completion. The routing area is limited on a chip so that a small routing area will be necessary. The small routing also minimize the net length which related to the signal delay. However, in some applications, for example gate arrays, the space for routing is fixed, or there are some limitation on reducing the routing area. It means that the density of nets on a routing plan will not be uniform that some area is congested and others are sparse. It is obvious that congested area is more vulnerable to defects than sparse areas. And to uniformly use the whole area instead of squeezing nets in only part of the routing area is important for avoiding the defects infection and improving the manufacturing yield. In this thesis, we propose a heuristic algorithm to optimize the use of the routing area.
CHAPTER 2

CRITICAL AREAS

2.1 Spot Defects and Critical Areas

The manufacturing yield discussed in this thesis is assumed to be the probability of the event that defects do not cause a fault in the fabricated circuit. Defects are local disturbances on the surface of a wafer, with an assumed random character, for example, dust particles. Faults which are caused by defects generate in the circuits. We assume that the defects are spot defects, which is used in most papers in the field of yield modeling. A majority of spot defects is introduced into the IC layer by the lithographic process. Usually they are caused by opaque particles gathered on the surface of the lithography mask or by transparent spots in the mask opaque regions. Transparent particles or opaque may caused extra material or missing material. An extra spot of conducting material or a missing spot of insulating material will cause bridging faults, for example, short between metal paths. Spot defects we discussed in this thesis will cause faults by shorting two
nonequipotential routed nets. On a routing plan, an area in which spot defects will cause a fault is called **critical area**. A spot defect with center in the critical area of routing plan will short two nonequipotential paths. A critical area on a routing plan is determined by the size of defects and the routing geometry. A radius of the spot defect determines the size of the spot defect. The routing geometry can be characterized by parallel conducting paths, as shown in Figure 2.1.

In Figure 2.2, we calculate the critical area between a pair of parallel conducting paths. The critical area between \( l_1 \) and \( l_2 \) is \( A_c(R_d, W) \), where \( R_d \) is the defect radius and \( W \) is the width of spacing between \( l_1 \) and \( l_2 \). The critical width \( W_c \) is the minimal acceptable spacing between two nonequipotential paths. The critical area \( A_c(R_d, W) \) is equal to zero when \( R_d < W/2 - W_c \). Then the critical area increase with slope \( 2L \), where \( L \) is a total length of a conducting path. Thus a larger size of spot defect will result in a larger critical area in this range. When \( R_d \geq W - W_c \), the critical area will saturate with a value \( LW \). For example, if the radius of spot defects is \( 3W/4 \) and \( W_c \) is zero, we will have

![Diagram](image-url)
a critical area $LW/2$, and the width of critical area belt is $W/2$. The size of the critical area between parallel wires with $W_c = 0$ is as follows:

$$A_c(R_d, W) = \begin{cases} 
0 & R_d + W_c < \frac{W}{2} \\
2L(R_d + W_c - \frac{W}{2}) & \frac{W}{2} \leq R_d + W_c \leq W \\
LW & R_d + W_c > W 
\end{cases}$$

In the channel routing process of layout design, we shall concentrate on the bridging faults caused by spot defects in the critical areas between parallel conducting wires and vias, since these are the faults that occur frequently in a channel can be avoided by rearranging the wires and vias. There are several models proposed in [4,5] can be used to calculate the size of a critical area. In a routing channel, we have three types of critical areas: (1) area between two wires, (2) area between a wire and a via, and (3) area between two vias, which are areas A, B, and C, respectively in Figure 2.3. The critical areas between conducting wires were considered in DTR [7], but the other two kinds of areas need also be considered.

As an example, in Figure 2.3, the areas, A, B, and C, are critical areas for the spot defects (a spot defect is assumed to have the shape of a disk) with diameters $3a/2, 3b/2,$ and $3c/2$, respectively $(a > b > c)$. From the previous formula, the widths of critical areas A, B, and C are $a/2, b/2,$ and $c/2$, respectively. If the center of a defect with the diameter $3a/2$ falls on the area A, a bridging fault will exist between the two wires. However, a defect with a diameter greater than $3b/2$ will cause a short if the center of the defect sites in B or C, but not necessarily cause a short if it is in A. So the areas B and C are higher sensitive areas to cause faults than A is.
Figure 2.2: The critical area between parallel wires.

Figure 2.3: Three types of critical areas.
2.2 Critical Area Computation

Generally, calculation of the critical area on a routing plan is more complex than the above considerations. The difficulty is that there are so many different spacings between conducting regions. A via introducing small spacing between via and adjacent conducting region, such as another via or a conducting path. The power line may need larger spacing with signal line to prevent the noise interference. There are three different methods to calculate the critical area on the routing artwork:

1. Monte Carlo method
2. Geometrical method
3. Virtual artwork method

In the Monte Carlo method, the critical area is estimated from the simulation experiment in which a defect disk is placed at random locations of the analyzed artwork. For each Monte Carlo trail, different sizes of disk will be used and the neighborhood of the disk is checked to see whether a short exists. In the geometrical method, entities in the mask are expended by a belt whose width is directly related to the radius of defects. Then the critical area, as an intersection of the expanded entities, is computed. The idea of virtual artwork method is to rearrange the artwork into several parallel wires. Because the computation of critical area between parallel wires is easy, the modified artwork composed of parallel wires will be convenient for computation of critical area and also reserves the statistical characterization of the original artwork.
In this thesis, the bridging fault of short is investigated and we assume the size of defects is not too large to short three parallel conducting paths. Therefore, the critical area is on the insulating strips only and we can find a much simpler approach to calculate the critical area. We assume the the spacing between conducting wires is uniform, except the spacing beside a via. And it will be possible to find a method to replace a via by a pair of parallel wires which cause same critical area such that the artwork will become a uniform spacing artwork which is composed of only parallel wires and easy to calculate the critical area. This method is presented in the next chapter.

2.3 Via Weights

Since areas $B$ and $C$ in Figure 2.3 are vulnerable to smaller defects than area $A$ is, in calculating the size of critical areas for a defect size, we can simplify the calculations by replacing a via by a wire in the following way. In Figure 2.4(a), there exists a critical area $A_1$ between the via $V$ and the wire $W$ with length $l_v$. Since $A_1$ is a type (2) critical area, if $V$ is replaced by a wire $W_2$ as shown in Figure 2.4(a), $W$ has to be replaced by $W_1$ which is longer by the amount $dl$ to make the critical areas $A_1$ and $A_2$ have the same effect on yield. Therefore, $W_1 = W_2 = l_v + dl$ and $dl$ is called the via weight. In Figure 2.4(b), for example, a pair of parallel wires with length $l$ and a via on one of the wires can be replaced by two parallel wires with length $l + dl$ each and no via.

The via weights depend on the manufacturing technology. The MOSIS design rules are considered in this paper. Some relevant design rules are shown below:

- conduction wire width: $3\lambda$
Figure 2.4: Critical area calculation.
space between wires: $3\lambda$
size of via: $2\lambda \times 2\lambda$
space between via and edge of wire: $1\lambda$
horizontal track space: $4\lambda$
column space: $4\lambda$

The technology is a 3 micrometer($\mu$m) process and $\lambda$ is 1.5 $\mu$m. In Figure 2.5, the width of a via is $2\lambda$. Since the space between two wires must be $3\lambda$, the space between horizontal tracks or vertical columns is $4\lambda$ to accommodate vias.

The probability density function of the size of a spot defect follows the Rayleigh distribution function\[4\], but for the defects considered here, the approximate density function $1/x^3$ ($x$ is the radius of the defect) is sufficient to determine the effect of a via on critical areas \[11\]. From the probability density function of the spot defects, we can obtain the weight factor of a via which is adjacent to a wire. Using the probability density function $1/x^3$, we get the probability of a defect with radius larger than $x$ is $1/2x^2$. We can see that the probability of a defect with $x \geq 2\lambda$ is higher than that with $x \geq 3\lambda$. Therefore, the space between adjacent vias is more sensitive to defects than that between adjacent parallel wires. Since the yield is a function of the product $D_d(r)A_c(r)$ \[4\] where $D_d(r)$ is the defect density, $A_c(r)$ is the size of the critical area, and $r$ is the radius of the defect, if two areas have the same yield, they will have the same $D_d(r)A_c(r)$. For example, in Figure 2.4, $A_1$ is the critical area between a via and a wire. We can find an equivalent critical area $A_2$ between two parallel wires. Use the above density function, we have

$$D_dA_c = \int_{d/2}^d 2L\left(r - \frac{d}{2}\right)\frac{1}{r^3}dr + \int_d^{\infty} Ld\frac{1}{r^3}dr = L \ast F(d) = \frac{L}{d}$$
where \( d \) is the space between two parallel wires, \( r \) is the defect radius, and \( L \) is the overlapping length. The first integration term is for defects with radius from \( d/2 \) to \( d \), and the second integration term is for defects with radius more than \( d \). The defects with radius smaller than \( d/2 \) will not cause faults. Assume that \( L_1 \) and \( L_2 \) are the lengths and \( D_1 \) and \( D_2 \) are defect densities of \( A_1 \) and \( A_2 \), respectively. If they have the same yield, we have the following equation:

\[
\frac{D_1 A_1}{D_2 A_2} = \frac{L_1 * F(3)}{L_2 * F(4)} = 1,
\]

and therefore,

\[
L_2 = \frac{4}{3} \times L_1.
\]
Since $L_1 = 4\lambda$, we have $L_2 = 5.33\lambda$ and the via weight of a single via is $1.33\lambda$. It means that we need to increase the length of the wire by $1.33\lambda$ to compensate the deletion of a via. A critical area unit for defects with radius $3d/4$ is defined as the size of a rectangle with the width $d/2 = 2\lambda$ and the length $d + w = 7\lambda$ where $w$ is the width of a wire. Because the size of a critical area is proportional to the length, it also means that we need to add $1.33/7$ critical area units to the total critical area size for each via. Therefore, when we are calculating the size of the critical area between two parallel wires, we will add $1.33/7$ critical area units for each via if there are vias on the wires. For example, if the overlapping length of two adjacent parallel wires is $8(d + w)$ and 2 vias are on the overlapping part, the total size of the critical area is $8 + 2 \times (1.33/7)$ units.

2.4 Lower Bound on the Size of Critical Areas

It is not possible to predicate the total critical area size in a channel before the wires are routed. However, we can get a lower bound on the size of the critical areas on a routing layer given a known number of tracks. Assume that there is no vertical segment on the horizontal layer and no horizontal segment on the vertical layer. The diameter of a defect is assumed to be $3d/2$, and no two or more defects cause a fault together. Given the total length of all horizontal segments, the length of a track, and the number of tracks, we can determine the lower bound of the total critical area size on the horizontal layer. This is done by ignoring the vertical and horizontal constraints and piling up all the horizontal segments in as few tracks as possible. Then we reduce the critical areas by minimizing the overlaps between adjacent parallel lines. This can be done by doubling
the space between nets. The lower bound of the size of critical areas is derived in the following theorem.

**Theorem 1:** Assume that a set of \( N \) nets are routed in a \( K \)-track channel, with length \( C \) for each track. The total length of all horizontal net segments is \( S \). The width of space between adjacent tracks is \( d \). Then the lower bound of the size of the critical areas on the horizontal layer is

\[
\text{lower bound} = \begin{cases} 
Cd(S/C - [K/2]) & \text{if } K \text{ is odd and } S/C > [K/2] \\
Cd(S/C - K/2)/2 & \text{if } K \text{ is even and } K/2 < S/C < K/2 + 1 \\
Cd(S/C - K/2 - 1/2) & \text{if } K \text{ is even and } S/C > K/2 + 1 \\
0 & \text{otherwise}
\end{cases}
\]

**proof:** If all horizontal segments are collected in \( [S/C] \) tracks, we will have \( [S/C] \) full occupied tracks, \( K - [S/C] \) empty tracks, and one partially occupied track. If \( S/C \leq [K/2] \), it is possible to have an empty track between every two occupied tracks and this will not generate any critical area. If \( K \) is odd and \( S/C > [K/2] \), we can arrange \( [K/2] \) full occupied tracks with empty tracks between them again and insert the rest \( S/C - [K/2] \) occupied tracks between those previously assigned tracks. Whenever an empty track which is adjacent to two occupied tracks is assigned net segments, it will generate critical areas on both sides of it. Since we assume that the diameter of a defect is \( 3d/2 \), a critical area between two parallel lines will have a height half the space between two parallel lines. Hence, the size of the critical areas caused by \( S/C - [K/2] \) occupied tracks are

\[
\frac{2Cd(S/C - [K/2])}{2} = Cd(S/C - [K/2])
\]
In the case that \( K \) is even and \( S/C > K/2 \), \( K/2 \) full occupied tracks could be arranged again as above and the others will be assigned to minimize the critical area.

Because of even tracks, there is one empty track which is adjacent to only one occupied track. This empty track result in the critical area only on one side instead of both sides of it such that it has the highest priority to be used for the rest of tracks. If only one more track left, the highest priority track will be used for the last unassigned track. Since it beside only one occupied track, it causes \( d(S - CK/2)/2 \) critical area.

If more than one track left, we will first assign one full occupied track to the highest priority track which will cause \( dC/2 \) critical area. The other unassigned tracks will use the empty tracks between occupied track such that it will cause \( d(S - CK/2) \) critical area. The total critical area is \( d(S - C(K/2 - 1/2)) \)

For example, Figure 2.6, the total length of the wire is 54, the length of channel is 12, and the number of tracks is 6. The spacing between tracks is 1 and radius of defects is 4/3. The critical area is 12.

Although the above formula is not tight enough for low density routing, it is useful for the compact routing case. Surely we can have a tighter lower bound which observes the vertical constraint, but the lower bound will be a too complex equation to be of practical use. The results in Table 1 of [7] should be consistent with our lower bound calculation, but we found that the size of critical areas (241) shown in Example 3a of Table 1 in [7] is less than the calculated lower bound (267) based on Theorem 1. The size of critical areas of example 3c in [12] on horizontal layer is calculated by us to be
398.5 and the lower bound is 366. However, the size of critical areas in example 3c of Table 1 in [7] is claimed to be only 287 which again is inconsistent with our calculations. Due to these inconsistencies, we are not able to compare our results with their results in the rest of the paper. From Theorem 1, we know that there is not much space left for us to minimize the critical areas in a high density routing channel. Hence, we have to find other ways to minimize critical areas instead of only moving horizontal segments within the horizontal layer. Moving horizontal segments to the vertical layer or vertical segments to the horizontal layer such that critical areas can be eliminated will be one feasible way. In the following chapters, we will present methods to perform the task of minimizing critical areas.
In this chapter, we will further define our problem and the terminology we will use in following chapters. Consider a rectangular channel with four rows of terminals along it four sides. The rows of terminals on bottom and top sides are fixed on certain position and the rows of terminals on the other two sides are determined during routing. Each terminal will be assigned to a net which electrically connects it to other terminals. In this thesis, two layers are available for routing. We assume horizontal segment of nets is routed on the top layer and vertical segment of nets is routed on the bottom layer. Two segments of a net on different layer will be connected by a via. Each segment of net on horizontal layer is routed on horizontal tracks and each segment of net on vertical layer is routed on vertical columns. The spacing between tracks and it between columns are the same.
3.1 Vertical Constraint Graph (VCG)

Consider the Figure 3.1, the net 2 has a terminal on the top of channel on column 3 and the net 1 has a terminal on bottom of channel on column 3. To prevent the confliction between net 1 and 2, we have to rout the horizontal segment of net 2 above the horizontal segment of net 1. If we assume that there is only one horizontal segment per net, then it is clear that the horizontal segment of a net connected to the upper terminal at a given column must be placed above the horizontal segment of another net connected to the lower terminal at that column. The relation can be represented by a directed graph, where each node corresponds to a net and a directed edge from node $i$ to node $j$ means net $i$ must be placed above net $j$. The formal description is as following:

**Definition 1.** A digraph $G_v(V, A)$ is a vertical constraint graph, where a node $v_i \in V$ represents net $n_i$, an arc $(v_i, v_j) \in A$ iff net $n_i$ must be placed above $n_j$, and the weight of $v_i$ is the length of $n_i$.

If there is a cycle in the graph, the routing requirement cannot realized without dividing some nets. For example, in Figure 3.2, a cycle $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$ means that net 1 must be placed above itself. The routing is not possible to realize when a net allow only one horizontal segment. However, this kind of conflicting situation can often be avoided by rearranging the placement. To overcome this situation, we can divide the horizontal segment of nets which is called *dog-legging*, see figure 3.2. Using *dog-legging* can avoid the cyclic vertical constraint and minimize the number of horizontal tracks but increases
vias, Figure 3.3. The dog-legging is allowed to horizontal split a net at the column where a terminal is, which implies that no additional vertical track is permitted.

In this thesis, we assume that no cyclic conflict among routed nets and no dogleg is permitted. The cyclic can be solved by dog-legging, and dog-legging can still be applied in our algorithm after some modification. Therefore, this algorithm will not lose any generality by these assumptions.

3.2 Zone representation of horizontal segments

Since we assume the horizontal segment of a net is allowed on one horizontal track only such that the horizontal segment of a net is determined by its leftmost and rightmost
Dog-legging

Figure 3.3: Use dog-legging to minimize the number of tracks.

terminals. Let $S(i)$ be the set of nets whose horizontal segments intersect column $i$. Because horizontal segments of distinct nets must not overlap, the horizontal segments of any two nets in $S(i)$ must not be placed on the same horizontal track. Each column must follow this rule. However, it is easy to see that we only have to consider those $S(i)$ which are not subsets of another set. Therefore, those $S(i)$ which are the subset of a $S(j)$ are assigned to the zone where $S(j)$ is. Each zone in the channel is given a sequential number, for example Figure 3.4, where zone 1, zone 2, etc. are assigned. The number of nets in $S(i)$ is called local density, and the maximum among them is called maximum density.

More clearly description for zone representation could use an interval graph defined by the horizontal segments of nets.

**Definition 2:** A graph $G(V, E)$ is an interval graph corresponding to a set of nets, where a node $v_i \in V$ represents net $n_i$ and an edge $(v_i, v_j) \in E$ iff $n_i \cap n_j \neq \emptyset$ (i.e., net $n_i$ and $n_j$ have horizontal overlap).

The interval graph of the net list in Figure 3.4 is shown in Figure 3.5. A zone is defined by a maximum clique in the interval graph and the clique number is the density of the
Figure 3.4: A zone representation.
Figure 3.5: Interval graph of a net list.

zone. The net in zone $i$ can use the same horizontal track as the net which is in other zones and not in zone $i$ without horizontal confliction. In terms of interval graph, net $i$ and net $j$ could on the same horizontal track if net $i$ and net $j$ do not belong to same clique.

Definition 3: A zone is a set of nodes $\overline{K} \subset \overline{V}$, where $\forall v_i$ and $v_j \in \overline{K}$ and $i \neq j$, $\exists \text{edge}(v_i, v_j) \in E$.

3.3 Weighted Interval Graph (WIG)

The size of a critical area is a function of the space between two adjacent parallel wires. We use the weighted interval graph to represent the critical area between any two nets if they can be next to each other. The weight interval graph will be defined as following:

Definition 4: A graph $G_w(V, E)$ is a weighted interval graph where a node $v_i \in V$ represents a net $n_i$ and an edge $(v_i, v_j)$ is in $E$ iff the horizontal segment of net $n_i$ can be parallely adjacent to the horizontal segment of net
Figure 3.6: The edge weight $W_e$ between $N_1$ and $N_2$.

$n_j$. The weight of edge $(v_i, v_j)$ is $W_e(i, j)$ which represents the size of the critical area between $n_i$ and $n_j$.

The weight for each edge in the WIG is the sum of the overlapping distance of the two nets, and the effect of via on the critical area which is the product of the via weight and the number of vias being adjacent to a wire. For example, in Figure 3.6, vias $a$ and $b$, on net $N_1$ are adjacent to net $N_2$, via $c$ on net $N_2$ is adjacent to $N_1$, but the via $d$ on net $N_2$ is not adjacent to $N_1$. The weight $W_e$ of the edge between $N_1$ and $N_2$ is therefore, $L + 3C$ where $C$ is the via weight. Actually, type (2) and (3) critical areas will have different via weight but only type (2) via weight is happening in the design rule we used. Again, the weight $W_n$ for a node $v_i$ in VCG is the length of $n_i$. If nodes $v_i$ and $v_j$ are two nodes in VCG and $v_i$ is a descendent of $v_j$ but not a direct son of $v_j$, net $n_i$ and net $n_j$ are not possibly adjacent to each other and therefore, node $v_i$ and node $v_j$ will have no edge between them. Figure 3.7 shows an example WIG where node $v_2$ and node $v_5$ have no edge between them in WIG because net $n_2$ is impossible to be adjacent to net $n_5$. 
Figure 3.7: Weighted interval graph.
3.4 Net Burying

In the two layer channel routing, we usually use one layer for the horizontal segments of a net and the other layer for the vertical segments of a net. The major layer for the vertical segments is called the *vertical layer* and the major layer for the horizontal segments is called the *horizontal layer*. We assume that the horizontal layer is on top of the vertical layer. The process of moving a wire segment on the horizontal layer down to the vertical layer is called *net burying*. If there is no wire under a wire \( N \) on the horizontal layer, wire \( N \) can be buried. For example, *net 2* in Figure 3.8 can be buried since there is no wire under *net 2*. After moving to the vertical layer, we have eliminated the critical areas adjacent to *net 2* and the vias on *net 2*. 
3.5 Net Floating

Contrary to net burying, the process of *net floating* is to move the vertical segment of a net to the horizontal layer. A vertical segment of a net at a given column can be moved to the horizontal layer if the path of the vertical segment on the horizontal layer does not cross the horizontal segment of another net on the horizontal layer. For example, in Figure 3.9, the critical areas between the vertical segment of net 4 and those of net 2 and net 5 are eliminated after floating the two vertical segments of net 4. In Figure 3.10, the vertical segment of net 4 is blocked by net 6 such that only partial vertical segment of net 4 can be floated. Similarly, we can have partial net burying.
CHAPTER 4

ALGORITHM DESCRIPTIONS

The algorithm presented in Figure 4.1 is based on the channel routing algorithm proposed by Yoshimura and Kuh[12]. However, we use the WIG instead of an interval graph. The algorithm YOR also uses the zone representation of horizontal segments for net merging and track assignment. The nets in the adjacent zones is hopefully put on the same tracks to minimize the number of tracks for realizing the connections. Zone $Z_s$ represents the first zone which is operated and zone $Z_t$ represents the last one. In this thesis, we start from the leftmost zone and stop at the rightmost zone. However, we could start from the zone which is with highest density in the channel, i.e., the zone is the biggest clique in the interval graph of the set of nets.
Algorithm YOR\((Z_s, Z_t)\);
\{
1. \(L = \{\}\);  
2. for \((Z = Z_s \text{ to } Z_t)\) \{  
3. \(L = L + \{\text{nets which terminate at zone } Z\}\);  
4. \(R = \{\text{net which start at zone } Z + 1\}\);  
5. Merge \(L\) and \(R\) so as to minimize the increase of the longest path in the vertical constraint graph and to minimize the critical area;  
6. Update the vertical constraint graph and weighted interval graph;  
7. \(L = L - \{n_1, n_2, \ldots\}\) where \(n_j\) is a net merged at step 5;  
\}
8. Track assignment that minimize the critical area for every node in the vertical constraint graph;  
9. Float and bury nets;  
\}

Figure 4.1: The yield optimizing routing (YOR) algorithm.

4.1 Merging

The objective of merging is obvious that we try to put nets which have no horizontal overlap on a track and each track put as many nets as possible. In this way, we can use minimal number of tracks and save the chip area. However, we will find some inadequate merging sequences will increase the number of tracks for realizing routing nets because of vertical constraints between nets. In Figure 4.2, if node 10 and node 1 are merged, the length of the most longest path in the vertical constraint graph will increase. Since the length of the most longest path in the vertical constraint graph is equal to the minimal number of tracks for realizing routing nets. Therefore, to minimize the increasing the length of the most longest path, we can merge node 10 and node 9.
Figure 4.2: Zone representation and VCG before net merging
To achieve this goal, we use a heuristic approach which bases on the algorithm proposed by Yoshimura and Kuh [12]. Before describing the algorithm, we will define following operations.

If net \( i \) and net \( j \)

- have no horizontal overlap in the zone representation and
- have no direct path between node \( i \) and node \( j \) in the vertical constraint graph

then, we can merge net \( i \) and net \( j \):

1. merge node \( i \) with node \( j \) in the vertical constraint graph and delete the redundant arc.
2. modify the zone representation by replacing the net \( i \) and net \( j \) by net \( i,j \) which occupied the consecutive zones including those of net \( i \) and net \( j \).

In Figure 4.3, node 10 and node 9 are merged and the zone representation is modified.

Now the problem is how to find a pair of node \( i \) and node \( j \) which will minimize increase of the length of the most longest path in the vertical constraint graph and will not be a obstacle for the following merging sequences for minimizing increase of the length. The following definition will help us describe the algorithm:

1. Given \( P = n_1, n_2, \ldots, n_p \) and \( Q = m_1, m_2, \ldots, m_q \), where \( p \geq q \), are two sets of nets to be merged.
2. node \( s \) is a start node with arcs to every node which has no parent in VCG before adding \( s \).
3. Node $t$ is an end node with arcs from every node which has no child in VCG before adding $s$.

4. $u(n)$, where $n \in P \cup Q$, is the length of the longest path from $s$ to $n$.

5. $d(n)$, where $n \in P \cup Q$, is the length of the longest path from $n$ to $t$.

For condition 1, because two sets of nets $P$ and $Q$ belong to different zones, nets on different sets will never on the same path of the vertical constraint graph. It means that different sets of nets on different paths of vertical constraint graph can be merged and make no confliction. Since the depth of the longest path in the vertical constraint graph is the minimal number of tracks to realize the interconnections, we had to merge the nets so that the depth of longest path in VCG after merged is minimized as much as possible. However, to find an exact minimum merge is too time consuming, hence a heuristic merging algorithm is given. The intuitive ideas is to choose the node $m \in Q$ which is on the longest path before merging; furthermore, it is farthest away from either $s$ or $t$. Then, find a node $n \in P$ of which the increase of the longest path after merging is minimum. If there are two or more choice which will result in a minimum increase, we choose node $n$ such that $u(n) + d(n)$ is nearly maximum and that the condition $u(m)d(m) = u(n)d(n)$ is nearly satisfied. If there are still two or more nodes, we will choose the node $n$ which is the longest net such that we can let long net become longer and it will be useful to prevent critical area in the track assignment process. We will talk about this characteristic in next section. The following algorithm is designed to implement the rule:

Algorithm Merge;
Figure 4.3: Zone representation and VCG after net merging
Given $P, Q$

while ($Q \neq \emptyset$) {

find $\hat{m} \in Q$ which maximum $f(m)$

find $\hat{n} \in P$ which minimum $g(n, \hat{m})$

and which is neither ancestor nor descendent of $\hat{m}$

if ( more than one $\hat{n}$)

chose $\hat{n}$ which is the longest net

merge $\hat{m}$ and $\hat{n} \in VCG$ and $WIG$

delete $\hat{m}$ and $\hat{n}$ from $P$ and $Q$

}

where $m \in Q$ and $n \in Q$ should satisfy following rules

(1) $f(m) = C \ast \{u(m) + d(m)\} + \max\{u(m), d(m)\}$

(2) $g(n,m) = C \ast h(n,m) - \{\sqrt{u(m) \ast u(n)} + \sqrt{d(m) \ast d(n)}\}$

(3) $h(n,m) = \max\{u(n), u(m)\} + \max\{d(n), d(m)\}$

$-\max\{u(n) + d(n), u(m) + d(m)\}$

(4) $C$ is a large value, in this thesis we choose 100

4.2 Track Assignment

After every eligible merging has been made, we get a new VCG graph. Then we assign tracks to the nodes in the VCG graph. The problem of track assignment is how to assign
Procedure Track Assignment;
flag = MIN;
for(i = 1; i ≤ |V_v|; i++) {
  if (flag == MAX) {
    Find node v_i, where v_i ∈ TOP(V_v) and W_n(v_i) + AVG(v_i) is maximum;
    T_i = v_i;
    Delete v_i from G_v(V_v, A);
  }
  else {
    Find node v_i, where v_i ∈ TOP(V_v) and W_n(v_i) + AVG(v_i) is minimum;
    T_i = v_i;
    Delete v_i from G_v(V_v, A);
  }
  if (flag == MIN)
    flag = MAX
  else
    flag = MIN;
}
Figure 4.4: Track assignment procedure.

every node to a track so that the minimum critical area can be obtained. The tracks are
assigned to nodes in an alternating way, i.e., we assign a track to a node with the shortest
net first and then to a node with the longest net, and so on. This procedure is performed
from the top track to the bottom track. The algorithm for track assignment is shown in
Figure 4.4 and described in the following.

Let G_v(V_v, A) be the VCG and G_w(V_w, E) be the WIG. |V_v| is the number of nodes
in V_v. W_n(n) is the weight of a node n ∈ V_v, and W_e(n, m) is the weight of an edge
(n, m) ∈ E in WIG. TOP(V_v) is the set of nodes which have no parent in V_v. Track
T_1, T_2, ..., T_k is in the order from the top to the bottom in the channel.

To scatter the nets on the horizontal layer, we take the following approaches:
1. Interlace the track with long net and that with short net

2. Interlace the track which has many horizontal overlaps with other tracks and the track which has little overlaps with other tracks.

The first approach is dependent on the weights of nodes in VCG and the second approach is dependent on a value, AVG. Interlacing tracks is based on the sum of these two values. The AVG value is the average of the weights of the nodes in WIG which can be assigned to the next track. For example, if node \( v \) is assigned a track, a potential node for the next track will be a node in current \( TOP(V_v) \) or a son of \( v \) which has only one father \( v \). More formally, 

\[
AVG(v_i) = \text{average of all } (W_e(v_j, v_i)) \text{ where } v_j \in (TOP(V_v)+SON(v_i)-\{v_i\}-\overline{SON}(v_i)).
\]

\( SON \) and \( \overline{SON} \) are defined as follows: a node \( v \in SON(v_i) \) if \( v \in V_v \) and \( v \) is a son of \( v_i \) in VCG, and a node \( v \in \overline{SON}(v_i) \) if \( v \in V_v \) and \( v \) has two or more fathers in VCG with one being \( v_i \). For example, in Figure 4.5, \( TOP(V_v) \) is \( \{1, 2, 3, 4\} \), \( SON(1) \) is \( \{5, 6, 7, 8\} \), and \( \overline{SON}(1) \) is \( \{7, 8\} \). If the current track is 1, the next possible track will be one of \( \{2, 3, 4, 5, 6\} \).

The weight of an edge in the WIG includes the via weights and therefore, we minimize the critical areas between parallel wires as well as between a via and a wire or another via simultaneously.

4.3 Via Shifting and Net Bumping

The basic idea in the last section is to scatter the nets on the horizontal layer and to prevent crowded areas. However, from Theorem 1, we know that the ability of interlacing
assignment to reduce the critical areas is still limited. Our objective here is to find an approach which elaborates to minimize the critical areas on both layers.

Up to now we have not considered the critical areas caused by the vertical segments of nets because the vertical segments are much more constrained by the positions of net terminals. The permutation method used for horizontal layer is not suitable for the vertical layer. We propose a simple but efficient method to reduce the critical areas on both layers.

Before describing the algorithm, we first define the term via shifting. In via shifting, a wire $w$ connected to a via $v$ is switched from one layer to another layer and $v$ will be eliminated or shifted along $w$ to another place. Via shifting will be blocked if the wire
crosses other nets. The via shifting process can be implemented by floating or burying nets. From our analysis, three types of vias can be shifted:

1. A via has only two wire branches connected to it and each on a different layer. This via can be eliminated or shifted by floating or burying one of the branches.
2. A via connects to another via by a wire and the wire can be buried or floated.
3. A via has one branch \( b \) on one layer and all other branches on the other layer. This via could be shifted along the branch \( b \) to minimize the critical area and also eliminate vias.

The via shifting procedure is shown in Figure 4.6 and examples on via shifting are given in Figure 4.7.

Net bumping is to bend part of a net segment on a track (column) to another track (column) but keep it on the same layer to remove the critical area caused by it. For example, in Figure 4.8, the two critical areas highlighted are eliminated by bumping the two net segments next to it. Although bumping a net segment will increase the net length, we can prevent excessive net length increase by setting some constraints. In this thesis, we bump a net only if it is possible to reduce more than 2 critical area units. We search the channel to find an empty space with the width of at least \( 2(d + w) \) and the length of more than \( 2(d + w) \) for bumping a net segment. If an empty area was found, one net will be bumped. Because there are two nets next to the empty area which can be bumped, we calculate the size of the critical area caused by each of them and choose the net which generates larger critical area for bumping. The algorithm for net bumping is in Figure 4.9.
Procedure Via_Shifting;
for (each via v) {
    if (v has two branches $b_h$ and $b_v$ on different layers) {
        \[ C_h = \text{size of the critical area adjacent to } b_h; \]
        \[ C_v = \text{size of the critical area adjacent to } b_v; \]
        if ($C_h > C_v$)
            Shift via v along $d_h$;
        else
            Shift via v along $d_v$;
    }
    else if (branch $b_{v,w}$ can be placed on another layer) {
        /* $b_{v,w}$ is the branch from via v to via w */
        Move branch $b_{v,w}$ to another layer if the critical area can be eliminated;
        if (all branches of v are on a single layer)
            delete via v;
        if (all branches of w are on a single layer)
            delete via w;
    }
    else if (# of branches of v on the horizontal layer == 1)
        Shift via v along the branch on the horizontal layer until it is blocked;
    else if (# of branches of v on the vertical layer == 1)
        Shift via v along the branch on the vertical layer until it is blocked;
}

Figure 4.6: Via shifting procedure.
where area\([i, i+m, j, j+n]\) is the area between track \(i\), track \(i+m\), column \(j\), and column \(j+n\).

### 4.4 Minimize Critical Areas with Redundant Tracks

The algorithm described previously tries to minimize the critical areas on a routing plan which uses the minimum number of tracks. However, for example gate array circuits, the number of tracks for routing in a channel is fixed such that the given tracks may be more than necessary. It means that the redundant tracks will happen. For the building block routing [13], the shape and size of unused area for routing is variety, which is different from cell-base design methodology with predefined areas for cells and interconnections. The tracks for each channel in the building block routing is possible more than minimum number of tracks. Therefore, it is obvious that we can use these redundant tracks for
Critical area eliminated by net bumping

Wire on horizontal layer

Wire on vertical layer

Net bumping

Figure 4.8: Use of net bumping to eliminate critical areas.
Procedure Net_Bumping;
for ( i = 1; i < max_track; i++)
  for( j = 1; j < max_column; j++)
    if ( area[i, i + 1, j, j + n] is empty and n > 2, where j + n < max_column) {
      Bump horizontal segments on track $T_{i-1}$ or $T_{i+2}$
      which can minimize the critical areas most;
      if (j + n > max_column)
          break;
      else
          j = j + n;
    }
for (j = 1; j < max_column; j++)
for( i = 1; i < max_track; i++)
  if ( area[i, i + n, j, j + 1] is empty and n > 2, where i + n < max_track) {
    Bump vertical segments on column $C_{j-1}$ or $C_{j+2}$
    which can minimize the critical areas most;
    if (i + n > max_track)
        break;
    else
        i = i + n;
  }

Figure 4.9: Net bumping procedure.
Procedure Track Assignment with Redundant tracks;
flag = MIN;
for(i = 1; i ≤ |V_v|; i++) {
  if (flag == MAX) {
    Find node v_i, where v_i ∈ TOP(V_v) and W_n(v_i) + AVG(v_i) is maximum;
    T_i = v_i;
    Delete v_i from G_v(V_v, A);
  }
  else {
    Find node v_i, where v_i ∈ TOP(V_v) and W_n(v_i) + AVG(v_i) is minimum;
    T_i = v_i;
    Delete v_i from G_v(V_v, A);
  }
  if (flag == MAX) {
    if (∃v_j ∈ TOP(V_v) and W_e(v_j, v_i) > W_avg and # of redundant tracks > 0)
      Insert a redundant track
    flag = MAX;
  else
    flag = MIN;
  else
    flag = MIN;
}
if (# of redundant tracks > 0) {
  Insert the redundant tracks between the tracks which
  cause largest critical area.
}

Figure 4.10: Track assignment with redundant tracks procedure.

Further preventing the critical areas in a channel. The algorithm, in Figure 4.10, modify
the track assignment algorithm by inserting a redundant track between two tracks and
letting the critical area between them be largest before inserting redundant track. In the
algorithm, W_avg is the average value of the edge weight among current unassigned nodes
in WIG.
CHAPTER 5
EXPERIMENTAL RESULTS

The algorithm has been implemented in C on Sun 3/60 workstations under UNIX operation system. Six channel routing examples in [12] are tested in this paper. In table 5.1, the reduction in the number of vias is shown. Although our primary goal is to minimize critical areas, the results are comparable to those generated by the optimal CVM algorithm [8] which is a pure via minimization algorithm.

Table 5.2 shows the reduction in the number of critical area units by our approach. For the Deutsch's difficult problem, we achieve a 26.22% reduction. Even for a smaller channel with high density such as example 3a, we still get a 8.13% reduction. Also we can see from Table 5.2 that after track assignment, i.e., before via shifting, net floating, net burying, and net bumping, the reduction in critical areas is insignificant and the critical area even increases by a little amount in two cases. Most contributions in minimizing critical areas come from the process of via shifting as well as floating, burying, and
Table 5.1: Reduction on the number of vias.

<table>
<thead>
<tr>
<th>Example</th>
<th># of vias</th>
<th></th>
<th>reduction(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example in [12]</td>
<td>YOR 5</td>
<td>Y&amp;K 22</td>
<td>77.3</td>
</tr>
<tr>
<td>Example 1 in [12]</td>
<td>YOR 40</td>
<td>Y&amp;K 57</td>
<td>30.0</td>
</tr>
<tr>
<td>Example 3a in [12]</td>
<td>YOR 77</td>
<td>Y&amp;K 91</td>
<td>15.4</td>
</tr>
<tr>
<td>Example 3c in [12]</td>
<td>YOR 116</td>
<td>Y&amp;K 125</td>
<td>7.2</td>
</tr>
<tr>
<td>Example 5 in [12]</td>
<td>YOR 124</td>
<td>Y&amp;K 154</td>
<td>19.5</td>
</tr>
<tr>
<td>Deutsch's difficult problem</td>
<td>YOR 235</td>
<td>Y&amp;K 290</td>
<td>19.0</td>
</tr>
</tbody>
</table>

bumping nets. Hence, the combination of via shifting, net floating, net burying, and net bumping procedures can be used as a post processor for other routers and it can improve the quality of the routing results from those routers in terms of yield and cost.

In Table 5.3, the density denotes the ratio of total horizontal segment length on the routing plane to the total length of tracks. The table indicates that critical area reduction is inversely related to the density, i.e., the higher the density is, the harder the minimization will be. Our algorithm achieves a 8.13% reduction for example 3a which has a high density of 0.881.

Before evaluating the yield, we have to choose a yield model. There are several yield models proposed in the literatures. If the defects are randomly distributed, the yield is an exponential function, \( e^{-DA} \), where \( D \) is the density of the spot defects and \( A \) is the total size of the critical areas. However, in the real manufacturing environment, the defects have a tendency toward clustering. Therefore, the more accurate negative
Table 5.2: Reduction on the number of critical area units.

<table>
<thead>
<tr>
<th>Example</th>
<th># of critical area units</th>
<th></th>
<th></th>
<th>reduction(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Y &amp; K</td>
<td>YOR (after track assignment)</td>
<td>YOR</td>
</tr>
<tr>
<td>Example in [12]</td>
<td></td>
<td>47.64</td>
<td>42.07</td>
<td>16.28</td>
</tr>
<tr>
<td>Example 1 in [12]</td>
<td></td>
<td>391.97</td>
<td>409.11</td>
<td>315.70</td>
</tr>
<tr>
<td>Example 3a in [12]</td>
<td></td>
<td>1028.39</td>
<td>1038.82</td>
<td>944.74</td>
</tr>
<tr>
<td>Example 3c in [12]</td>
<td></td>
<td>1485.54</td>
<td>1475.68</td>
<td>1354.54</td>
</tr>
<tr>
<td>Example 5 in [12]</td>
<td></td>
<td>1670.64</td>
<td>1566.69</td>
<td>1268.7</td>
</tr>
<tr>
<td>Deutsch’s difficult problem</td>
<td></td>
<td>4190.43</td>
<td>4061.63</td>
<td>3091.53</td>
</tr>
</tbody>
</table>

Table 5.3: Relationship between channel density and critical area reduction.

<table>
<thead>
<tr>
<th>Example</th>
<th>Density</th>
<th>Critical area reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example [12]</td>
<td>0.483</td>
<td>65.83</td>
</tr>
<tr>
<td>Example 1 in [12]</td>
<td>0.726</td>
<td>19.46</td>
</tr>
<tr>
<td>Example 3a in [12]</td>
<td>0.881</td>
<td>8.13</td>
</tr>
<tr>
<td>Example 3c in [12]</td>
<td>0.775</td>
<td>8.82</td>
</tr>
<tr>
<td>Example 5 in [12]</td>
<td>0.588</td>
<td>24.02</td>
</tr>
<tr>
<td>Deutsch’s difficult problem</td>
<td>0.486</td>
<td>26.22</td>
</tr>
</tbody>
</table>
binomial distribution yield model is used in this paper as shown below:

\[ Y = \left(1 + \frac{\gamma}{\alpha}\right)^{-\alpha} \]

where \( Y \) is the yield, \( \gamma \) is the average number of faults (fatal defects), and \( \alpha \) is a parameter that indicates the degree of clustering [5]. If the value of \( \alpha \) is infinite, the yield will become the exponential yield model. Because we assume that no two or more defects cause the same fault and we consider only bridging faults, we will have \( \gamma = A_{ch} \cdot D_f \), where \( A_{ch} \) is the area of the channel excluding wires and vias, and \( D_f \) is the fault density. The yield of a channel is then

\[ Y_{ch} = \left(1 + \frac{A_{ch}D_f}{\alpha}\right)^{-\alpha} \]

Since the defects will cause faults only when they fall on the critical areas, we have \( \gamma = A_cD_d \), where \( A_c \) is size of the critical areas between wires and vias in the channel and \( D_d \) is the density of defects (fatal or nonfatal) which can cause a bridging fault and therefore,

\[ Y_{ch} = \left(1 + \frac{A_cD_d}{\alpha}\right)^{-\alpha} \]

The width of the channel is the product of the number of tracks and the sum of the wire width and the spacing between wires. The length of the channel is the product of the number of columns and the sum of the wire width and the spacing between wires. In example 5, since \( A_{ch} \) is \( 1.333 \times 10^{-3} cm^2 \) and \( A_c \) is \( 1.776 \times 10^{-4} cm^2 \), \( D_f \) will be \( 37.53/cm^2 \) if \( D_f \) is \( 5.0/cm^2 \). In current technology, a fault density under \( 5/cm^2 \) is normal and therefore, we use two fault(fatal defect) densities \( 2.5/cm^2 \) and \( 5/cm^2 \) in Table 5.4 and Table 5.5, respectively. The realistic clustering parameter \( \alpha \) is around 0.3 to 5 [14], and we use 2 in
this paper. In the current technology, a chip with 30, 50, or more channels is common and the chip yield will be low if the single channel yield is not high enough. Assume that all channels in a chip have approximately the same complexity and therefore, the same yield. For convenience, the yield of the non-channel area in a chip is assumed to be 1. The yield of a chip is then the product of all single channel yields, i.e.,

\[ Y_{\text{chip}} = (Y_{\text{ch}})^{nc} \]

where \( nc \) is the number of channels in a chip. We present the yields for chips with 20, 30, and 50 channels in Table 5.4 and Table 5.5. Only examples 3a, 3c, 5, and Deutsch’s difficult problem are compared because their sizes are more practical for a real chip. Table 5.6 shows the percentage yield improvements for Table 5.4 and Table 5.5. For instance, in example 5, we get an 8.68% yield improvement on a 50-channel chip with the fault density 2.5/cm\(^2\) and a 18.08% improvement with a 5.0/cm\(^2\) fault density. For the Deutsch’s difficult problem, we get a 19.99% improvement on a 50-channel chip with a 2.5/cm\(^2\) fault density and a 43.82% improvement with a 5.0/cm\(^2\) fault density. The results are very significant and encouraging since the sizes of these two examples are more realistic for VLSI chips than those of other smaller examples.
Table 5.4: Chip yields with $D_f = 2.5/cm^2$.

<table>
<thead>
<tr>
<th>Example</th>
<th># of channels per chip</th>
<th>Y &amp; K</th>
<th>YOR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>Example 3a in [12]</td>
<td>97.43</td>
<td>96.17</td>
<td>93.70</td>
</tr>
<tr>
<td>Example 3c in [12]</td>
<td>96.10</td>
<td>94.21</td>
<td>90.53</td>
</tr>
<tr>
<td>Example 5 in [12]</td>
<td>93.56</td>
<td>90.50</td>
<td>84.67</td>
</tr>
<tr>
<td>Deutsch's difficult problem</td>
<td>86.43</td>
<td>80.35</td>
<td>69.44</td>
</tr>
</tbody>
</table>

Table 5.5: Chip yields with $D_f = 5.0/cm^2$.

<table>
<thead>
<tr>
<th>Example</th>
<th># of channels per chip</th>
<th>Y &amp; K</th>
<th>YOR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>Example 3a in [12]</td>
<td>94.93</td>
<td>92.49</td>
<td>87.80</td>
</tr>
<tr>
<td>Example 3c in [12]</td>
<td>92.35</td>
<td>88.75</td>
<td>81.97</td>
</tr>
<tr>
<td>Example 5 in [12]</td>
<td>87.54</td>
<td>81.91</td>
<td>71.70</td>
</tr>
<tr>
<td>Deutsch's difficult problem</td>
<td>74.73</td>
<td>64.61</td>
<td>48.28</td>
</tr>
</tbody>
</table>
Table 5.6: Percentage yield improvement.

<table>
<thead>
<tr>
<th>Example</th>
<th>2.5</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>Example 3a in [12]</td>
<td>1.31</td>
<td>1.97</td>
</tr>
<tr>
<td>Example 3c in [12]</td>
<td>2.01</td>
<td>3.03</td>
</tr>
<tr>
<td>Example 5 in [12]</td>
<td>3.38</td>
<td>5.12</td>
</tr>
<tr>
<td>Deutsch's difficult problem</td>
<td>7.56</td>
<td>11.55</td>
</tr>
</tbody>
</table>
CHAPTER 6

CONCLUSIONS

From the yield analysis, we know that the manufacturing yield is highly dependent on the routing geometry. Inadequate consideration during routing can cause extensive critical areas which introduce faults. To evaluate the yield of a chip, we have to calculate the size of critical areas. A simple and elegant method for computing the size of critical areas on routing plan is proposed in this thesis.

To minimize the critical area, an effective heuristic algorithm is designed. The net merging step merge nets such that long nets become longer and short nets become shorter. In track assignment step, long nets are interlaced with short nets to avoid the introduction of critical areas. By net floating, net burying, and net bumping, we further reduce the critical area on the routing plan. In this thesis, a new channel routing algorithm, YOR, which minimizes routing areas, critical areas, and vias at the same time is presented.
Without sacrificing the total routing areas, YOR generates far less critical areas and vias for the benchmark layouts than the results in [12].

Techniques such as net floating, net burying, net bumping, and via shifting have been shown to be very effective in reducing the number of critical areas and vias. These techniques can be used as a postprocessor for any routing algorithm to improve the quality of routing in terms of yield and cost. Experimental results show that the yields are improved significantly for practical size channels in a VLSI chip.

If the routing area has unused tracks, our algorithm efficiently utilizes the routing area to minimize the critical area. This approach obviously improves the yield compared to those routers which did not notice this merit.

The future works include the application of the techniques in the three layer routing. Since in the three layer routing there is more space for routing, it means more space for detouring the critical areas. However, more layers means higher complexity such that to find an optimum layer assignment for segments of nets is more difficult and needs further research.

In this thesis, we assume the spacing between tracks and columns is uniform but, in some cases, it is not. For example, the mixed signal circuit will need different spacing between different kinds of nets to isolate the noise or interference. Modified YOR for mixed signal circuit will be another research topic.
REFERENCES


