YOR: A Yield Optimizing Routing Algorithm by Minimizing Critical Areas and Vias

Sy-Yen Kuo
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan, R.O.C.

Abstract

The goal of a channel routing algorithm 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. 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. The experimental results show that large reduction in the number of critical areas and significant improvement in yield are achieved.

1 INTRODUCTION

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.

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 and minimizing the number of vias. However, a smaller chip means higher density routing regions in the chip and wires and components will be much closer to each other. Therefore, adjacent wires and devices become more vulnerable to manufacturing defects which reduces yield and increases testing costs. 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 [2]. Some failures can be avoided by appropriately designing in the layout phase. From the results in [3], two types of faults, ***line stuck-at faults*** and ***bridging faults***, are accounted for 58% of total faults in VLSI. 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 [4] 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 [5]. 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.

2 CRITICAL AREAS

The defects considered in this paper are ***spot defects*** which are assumed by most of the literatures in the field of yield modeling[6]. The faults caused by spot defects have been discussed in [2]. The spot defects are caused by extra or missing material during manufacturing process. For example, an extra spot of conducting material or a missing spot of insulating material will cause a bridging fault which is a short between two lines. If a defect sites in an area known as a ***critical area***, this defect will cause a fault. The critical area is dependent on the size of a spot defect. The larger the defect size is, the bigger the critical area will be. The probability of having faults depends on the size of the critical areas and therefore, we have to minimize the critical areas.

0-8186-2645-3/92 $3.00 © 1992 IEEE

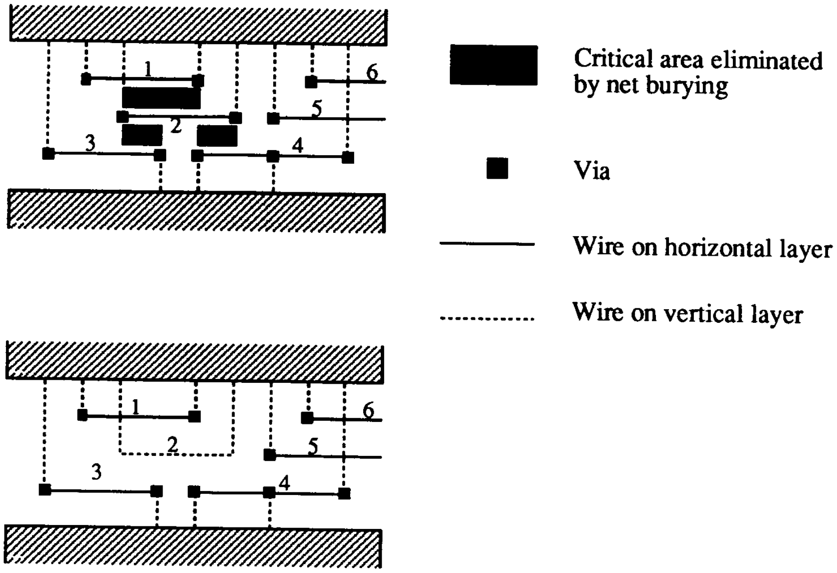

Figure 1: Net burying.

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. The model proposed in [2] 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. The critical areas between conducting wires were considered in DTR [4], but the other two kinds of areas need also be considered.

3 DEFINITIONS

3.1 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 1 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.

3.2 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 2, 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. It is possible that the vertical segment of a net is blocked by another net such that only partial vertical segment of the net can be floated. Similarly, we can have partial net burying.

4 ALGORITHM DESCRIPTIONS

The algorithm presented in Figure 3 is based on the channel routing algorithm proposed by Yoshimura and Kuh[7]. However, we use the WIG(weighted interval graph) instead of an interval graph for net merging and track assignment.

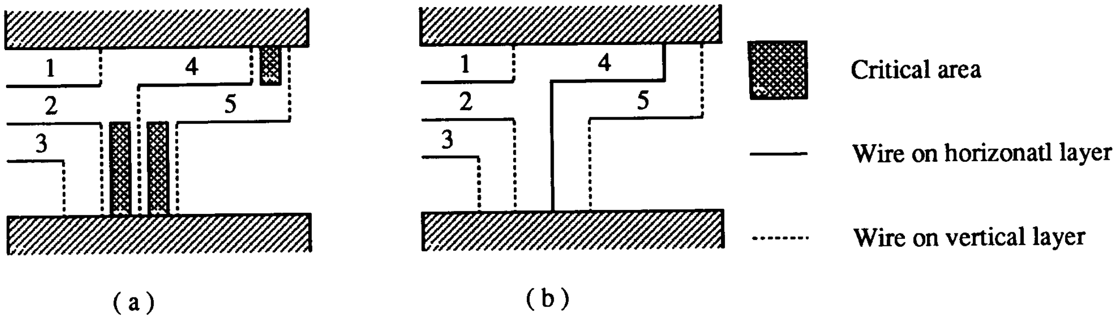

Figure 2: Net floating: (a) before floating (b) after floating.

```
Algorithm YOR($Z_s$, $Z_t$);
1.    $L = \{\}$;
2.    for ($Z = Z_s$ to $Z_t$) {
3.        $L = L+$ {nets which terminate at zone $Z$};
4.        $R = \{$ net which start at zone $Z + 1$ $\}$;
5.        Merge $L$ and $R$ 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 3: The yield optimizing routing algorithm.

The algorithm YOR also uses the zone representation of horizontal segments[7]. The zone representation of horizontal segments classifies the nets into several zones. Each zone contains a set of nets which defines a maximum clique in the interval graph G. A graph G is an interval graph corresponding to a set of nets N, where a vertex in G represents a net in N and an edge exists between node i and node j if there is a horizontal overlap between net i and net j. The algorithm will merge the nets in the current zone to the nets in the next zone, zone by zone.

Since our merging and track assignment approaches are based on those in [7] and their contributions to critical area reduction are insignificant as demonstrated by the experimental results, we will not show the details of these approaches here. In the following, we will describe how to perform via shifting and net bumping.

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.

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. Although bumping a net segment will increase the net length, we can prevent excessive net length increases by setting some constraints. In this paper, 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 generate larger critical area for bumping.

5 EXPERIMENTAL RESULTS

The algorithm has been implemented in C on Sun 3/60 workstations under UNIX operation system. Six

Table 1: Relationship between channel density and critical area reduction.

Example	Density	Critical area reduction (%)
Example [7]	0.483	65.83
Example 1 in [7]	0.726	19.46
Example 3a in [7]	0.881	8.13
Example 3c in [7]	0.775	8.82
Example 5 in [7]	0.588	24.02
Deutsch's difficult problem	0.486	26.22

channel routing examples in [7] are tested in this paper.

Table 1 shows the percentage 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 in Table 1, the density denotes the ratio of total horizontal segment length on the routing plane to the total length of tracks. The figure 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. In the real manufacturing environment, the defects have a tendency toward clustering. Therefore, the more accurate negative binomial distribution yield model is used in this paper as shown below:

$$Y = (1 + \frac{\gamma}{\alpha})^{-\alpha}$$

where Y is the yield, γ is the average number of faults (fatal defects), and α is a parameter that indicates the degree of clustering [5]. Because we assume that no two or more defects cause the same fault and we consider only bridging faults, we will have $\gamma = A_{ch} * D_f$, 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

$$Y_{ch} = (1 + \frac{A_{ch} D_f}{\alpha})^{-\alpha}$$

Since the defects will cause faults only when they fall on the critical areas, we have $\gamma = A_c D_d$, 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 non-fatal) which can cause a bridging fault and therefore,

$$Y_{ch} = (1 + \frac{A_c D_d}{\alpha})^{-\alpha}$$

The width of the channel is the product of the number of tracks and the spacing between wires. The length of the channel is the product of the number of columns and the spacing between wires. 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 2. The realistic clustering parameter α is around 0.3 to 5, 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.,

$$Y_{chip} = (Y_{ch})^{nc}$$

where nc is the number of channels in a chip. Table 2 shows the percentage yield improvements. Only examples 3a, 3c, 5, and Deutsch's difficult problem are compared because their sizes are more practical for a real chip. 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

Table 2: Percentage yield improvement.

Fault density (cm^{-2})	2.5			5		
# of channels per chip Example	20	30	50	20	30	50
Example 3a in [7]	1.31	1.97	3.31	2.64	3.98	6.72
Example 3c in [7]	2.01	3.03	5.10	4.06	6.14	10.45
Example 5 in [7]	3.38	5.12	8.68	6.87	10.49	18.08
Deutsch's difficult problem	7.56	11.55	19.99	15.65	24.36	43.82

are very significant and encouraging since the sizes of these two examples are more realistic for VLSI chips than those of other smaller examples.

6 CONCLUSIONS

A new channel routing algorithm YOR which minimizes routing areas, critical areas, and vias at the same time has been presented. Without sacrificing the total routing areas, YOR generates far less critical areas and vias for the benchmark layouts than the results in [4]. 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. Experimental results show that the yields are improved significantly for practical size channels in a VLSI chip.

References

[1] E. S. Kuh and T. Ohtsuki, "Recent Advances in VLSI Layout," *Proceedings of the IEEE*, Vol. 78, Feb. 1990, pp. 237-263.

[2] Wojciech Maly, "Modeling of Lithograph Related Yield Losses for CAD of VLSI circuits," *IEEE Trans. on Computer-Aided Design*, vol. 3, July 1985, pp. 166-177.

[3] J. P. Shen, W. Maly, and F. J. Ferguson, "Inductive Fault Analysis of MOS Intergrated Circuits," *IEEE design & Test*, Vol. 2, Dec. 1985, pp. 13-26.

[4] A. Pitaksannonkul, S. Thanawastien, C. Lursinsap, and J. A. Gandhi, "DTR: A Defect_Tolerant Routing Algorithm," *26th ACM/IEEE Design Automation conference*, 1989, pp. 247-253.

[5] K. S. The, D. F. Wong, and J. Cong, "Via Minimization by Layout Modification," *26th ACM/IEEE Design Automation Conf.*, 1989, pp. 799-802.

[6] C. H. Stapper, "Modeling of defects in integrated circuit photolithographic patterns," *IBM J. Res. Develop.*, Vol. 28, July 1984, pp. 461-475.

[7] T. Yoshimura and E. S. Kuh, "Efficient Algorithms for Channel Routing," *IEEE Trans. on Computer Aided Design*, Vol. 1, Jan. 1982, pp. 180-190.