Wire Spacing and Metal Filling: A New Solution for Layout Density Control and Manufacturability Improvement

Shuo Zhang  
Wayne Dai  

UCSC-CRL-02-11  
March 4, 2002

Jack Baskin School of Engineering  
University of California, Santa Cruz  
Santa Cruz, CA 95064 USA

ABSTRACT

To reduce fabrication process variation due to chemical-mechanical polishing, layout must be made uniform with respect to density criteria. Currently the layout density control is achieved by metal filling, which adds dummy metal geometries into the layout either at the foundries or by specialized verification tools. But because the added metal inevitably increases layout interconnect capacitance as well as the defect sensitivity, which reduces the manufacturability, the fill amount must be kept as small as possible. This paper presents a new solution for layout density control, named wire spacing and metal filling. Before adding dummy metal into the low-density area, we increase the wire spacing by moving wires from high-density areas toward low-density areas. The experimental results show that compared with stand-alone metal filling only, wire spacing followed by metal filling greatly reduces the total fill amount and also improves the manufacture yield.

Keywords: layout density control, manufacturability, wire spacing, metal filling, topological, TEG
1. Introduction

In recent years, chemical-mechanical polishing (CMP) has emerged as the primary technique for planarizing inter-level dielectrics (ILD). Although CMP is very effective at reducing the as-deposited step height and achieves a measure of global planarization [1], CMP processes are hampered by layout pattern sensitivities which cause certain regions on a chip to have thicker dielectrics layers than others. This thickness difference is due to the differences in the underlying layout topography, mainly the layout metal density [2] (Figure 1). As CMOS technology advances into the 180nm generation and beyond, this CMP process variation has severe impacts on the electrical performance, design reliability and wafer yield.

Attempts to control this CMP process variation includes an exhaustive search and experiment for different process choices, but no practical solution is currently available. Thus, the only viable choice is to control the local metal density [2]. Some works have showed that the ILD thickness increases monotonously with the metal density [3]. So by reducing the difference of local metal density between different parts of the layout, the variation of ILD can be decreased. Currently, foundries typically impose density rules on metal layers and even active layers, which normally constraints the metal density must be within some range in any window of the specified size. Normally those rules are satisfied with a post-layout processing that adds dummy fill geometries, called metal filling. And it is performed either by foundries or by specialized verification tools [3].

Although metal filling can reduce the CMP process variation, it inevitably increases the interconnect capacitance, particularly when the dummy metal is introduced onto the upper layers [2]. Furthermore, adding metal into the layout will cause another manufacture problem: higher sensitivity of manufacture defects which could bring more layout faults and reduces the yield. Therefore, the amount of filled metal should be as small as possible to meet the density rule.

In this paper, we introduce a new solution to reduce the layout density differences, which is called wire spacing and metal filling. Before adding dummy metal into low-density areas, we can increase the wire spacing by moving wires from the high density area toward the low density area. On one hand, less or no dummy metal is needed in sparse areas, resulting in less influence on interconnect delay; and on the other hand, more space is available in dense areas, less defect sensitivity for better yield. Figure 2 illustrates a simple case comparing

Figure 1: Sparse regions polish faster than dense regions resulting in ILD thickness variation.
metal filling only and wire spacing followed by metal filling. Wire spacing is implemented under TEG, a post layout optimization environment. The rest of this paper is organized as follows. We first review some background on layout density control in section 2. Then the details of wire spacing are presented in section 3. Section 4 includes the implementation of implementation as well as a brief introduction if the implementation environment, TEG. Finally the experimental results are reported in section 5.

2 Background

2.1 Layout Density

Layout density control consists of two phases: density analysis and density synthesis. The goal of density analysis is to find the layout windows with maximum or minimum metal density, or determine the density of each window. The density synthesis phase then actually changes the layout geometries in the some area in order to meet the density rules.

Kahng et [3] gave the first formulation of the layout density control problems that arise in post layout optimization for manufacturability. This work and the following works [4] [5] proposed a number of algorithms for density analysis. The general case, e.g., examining all possible windows, is regarded as floating window regime; when only examining windows from some fixed dissection over the layout, it is regarded as fixed-dissection regime. The floating window analysis algorithm can provide accurate maximum-density and minimum-density results, while the fixed-dissection only provide a lower and upper bound of the layout density, but runs much faster.

There were also some works on the layout density synthesis, and most of them focus on metal filling, which increase the metal density in the sparse area. The filling algorithms include Linear Programming [3], Multilevel [6], Monte Carlo [5], Iterative [7] and Hierarchical [8]. Some industry tools are also available to do filling through layout Boolean functions, such as Cadence Dracula or Assura.

Normally there are two styles of filling, grounded or floating metal-fill. And both of them increase the layout interconnect capacitance, in which the grounded one affects delay
3. Wire Spacing for Layout Density Control

Figure 3: Wire Spacing in 2 steps

attributes while the floating one increases coupling/crosstalk attributes [2]. Due to this disadvantage on the interconnect issue, decreasing the metal density in the dense area is more preferred than metal filling. Until now there is only one method available to decrease the metal density [3], which is to slot (partial delete) the large metal geometries. Because slotting could change the cross-section of a power bus, which in turn affects peak current density and reliability, it is even not as favored as filling.

2.2 Bridging Faults

Bridging faults are the shorts between different conductors in the layout and represent a significant amount of faults responsible for chip failure. They are usually caused by spot defects, which occur during the manufacturing process and results in spots of extra or missing material on the wafer. The size of spot defects are comparable to the feature size of the layout such that their presence alter the intended functionality or performance of the circuit. Not all spot defects will cause faults; the presence of faults depends on the area where the defect appears, called the critical area. The larger the critical area, the larger the possibility of the bridging faults. Stapper showed that the critical area increases as the size of spot defects increases, or as the feature spacing decreases [9]. Because the size of spot defects is usually related to the fabrication process, the larger spacing is more preferable for less bridging faults yield. Su [10] showed that it is practical to specified larger wire spacing rules in order to decrease the critical area. And Jee et [11] provided an inductive fault analysis tool, Carafe, which generates critical area measurements to estimate the manufacture yield.

Metal filling for layout density control always increases the critical area. For grounded filling style, every bridging fault with the filling could cause a real chip failure. For floating filling style, it is also possible that multiple bridging faults with the dummy metal could cause a real bridging fault which fails the chip. But wire spacing following by metal filling can diminish this increment or even decrease the critical area; dense areas get larger spacing and sparse areas get less or no filling than metal filling only.

3 Wire Spacing for Layout Density Control

The idea of wire spacing is to increase the wire spacing by move the wire from the high-density area toward the low-density area in order to achieve less density variation, while
kept layout functionality and the design rule. Moving a wire includes moving the wire routing and moving its terminals, such as pins, vias and steiner points. As a post-layout application, like metal filling, wire spacing should keep its influence on the original design as little as possible. So in our approach, we didn’t consider moving the cell pins, which needs to move the cell placement. And for the wire routing, we only adjusted it path for larger spacing, but didn’t reconstruct its routing path. It is possible that rip-up and reroute also works, but for metal density which is a local characteristic related to multiple nets, rerouting a few nets will not be effective considering the global change to the layout.

Our wire spacing approach works in two steps. First, each via, (the steiner points can be treated as a single-layer via), is moved toward the sparse area. Then, for each local area, wires are distributed among the available space. Because the goal of density control is to reduce the density difference between different local windows, even every metal movement in wire spacing could be very small compared with the layout size, the total effect is the global density uniform.

Figure 3 shows a simple example of these two steps. In this example, we assume the terminals along the layout boundary are pins while the inside terminals are vias.

3.1 Via Moving

The target of the via moving is to balance the wiring density around a via, in the other words, to let the areas with more wires have more space. The via moving works as below. For each via in the layout, at first we calculate the local wiring densities around it in different direction. If the difference among these densities are over the limitation, we move the via toward the less-density direction in order to decrease the difference. Each via moving should not change the connectivity of any net.

How to formulate the metal densities around a via and how to determine its new position is related with the density analysis method and layout optimization environment, we will discuss it in the implementation section.

3.2 Wire Distribution

Along with vias, the wires attached to them were also moved from the dense area to the sparse area by via moving. But this step only changed the location of the wire ends,
and most of the routing paths didn’t change. Considering usually most of the layout metal is used for wire paths other than wire ends, it is more important and effective to optimize wire paths for layout density control. So after available space is moved to the dense area, next step is to spread these space among wire paths, which is called wire distribution.

Basically, wire distribution is to increase the spacing between different wires, or between wires and other metal elements, like pins, vias or obstacles. Because this spacing increment is based on the local density analysis, we can not accomplish it by simply changing the design rules, or by appointing larger spacing rules for some specified wires. Figure 4 shows an example that wire distribution achieves better local density uniform than specifying a larger spacing rule.

With these two steps, we can get much less density differences before using metal filling. But the problem resides on how can we implement them. In the post-layout stage, every layout modification is restrained by surrounding geometries, especially, the wiring. So every operation intending to move vias or wires could create a design rule violation. This makes most changes of via moving and wire distribution unacceptable. We solved this problem by implementing wire spacing through a new post-layout optimization method, which is TEG [12]. Based on topological representation of the layout, TEG provides an incremental layout modification environment for the implementation of wire spacing.

4 Implementation of Wire Spacing through TEG

In the reviews of the previous works for layout density control, we discussed that all layout density synthesis methods use filling or slotting. From the layout modification point of view, neither filling, which adds dummy metal into the available space, nor slotting, which partially deletes metal inside large geometries, changes the location of any layout elements in the original layout. But our wire spacing changes the position of the vias and the wire paths for improved density control. This change is the most unique characteristic of wire spacing, and it is also the most difficult part because of layout geometry constraints. This problem was successfully solved by implementing wire spacing through TEG, a post-layout optimization method [12]. In the following, we will first give a brief introduction of TEG, then the implementation details of wire spacing will be explained.
4. Implementation of Wire Spacing through TEG

Figure 6: TEG: Vertex moving procedure

4.1 TEG: A Post Layout Optimization Method

TEG is achieved by the topological representation of the layout, or topological layout. Compared with the typical geometry representation (Figure 5a), in which every layout elements (including pins, vias, steiner points, wires and obstacles, etc) has specified shape and location, topological layout only captures the relative positions and connections of the elements, but no geometry information of wires (Figure 5b). In TEG, a topological layout is represented on a triangulation graph (V, E), in which V includes all the terminals, vias, Steiner points, the vertices of obstacles and layout boundary in the layout; and each wire is represented as a sequence of intersection point between the wire path and graph edges (Figure 5c). The elimination of wiring geometry releases layout modifications from local geometry constraints: whether a modification will create a design rule violation (DRV) doesn’t depend only on the surrounding geometries, but also on the available space beyond local area, even in the whole layout. This makes most layout modifications feasible and practical in post-layout stage.

Based on topological representation, TEG provides a set of operations to support layout modifications. First of all, there is a layout modification procedure for vertex moving. Vertex moving is an essential procedure in TEG, with which via moving, obstacle/cell moving, or even buffer insertion can be achieved. Figure 6 shows an example of vertex moving. In (a) vertex v is to be moved to new position d, and (b) is the moving result. It was proved that only the layout topology in a local region should be updated, which is a set of triangles who contacts with the moving path. As our example, the local region is the polygon (ABCDEFGH). The use of local region make the running time of a vertex moving to be $O(n)$, where $n$ is the number of vertices in the local region.

TEG also provides a topological design rule check (DRC) procedure. According to regular geometry design rules, the DRC is able to verify the topological layout, e.g. to determine whether the latter represents a valid geometry layout without any DRVs. Furthermore, based on the vertex moving, a procedure named DRV solver is available to remove the DRVs found by the DRC procedure so that a valid DRV-free geometry layout can be
generated from the modified topological layout. By using a new layout routability theorem, the topological DRC and DRV solver runs very fast. The experimental results from the real industry IC design showed that practically there running time are $O(n)$, too, where $n$ is the number of vertices in the layout.

In our implementation of metal moving, at first a topological layout is extracted from the original routed layout. At second, the via moving and wire distribution are performed on the topological layout. Then the topological DRC verifies the layout and DRV solver help to remove the problems; finally a geometry layout is generated with the proposed wire spacing.

4.2 Via Moving through TEG

In TEG, a layout is represented by a triangulation graph. Based on this graph, we use the linear wire on each edge density to approximate the local area wire density. In Equation 1, the wire density $\sigma$ is formulated between any two adjacent vertices $a$ and $b$, where $W$ stands for the set of wires which cross the edge between $a$ and $b$, $x$ and $y$ is the vertex position and $r$ is the vertex radii.

$$\sigma(a, b) = \frac{((x_a - x_b)^2 + (y_a - y_b)^2)^{\frac{1}{2}} - (r_a + r_b)}{\sum_{w \in W} \text{width}(w)}$$  \hspace{1cm} (1)

For a via $v$, which has a set of neighbors $A$, we compute the wire density $\sigma$ between each neighbor $a$ and $v$ and the average density $\sigma_{avg}$. So the wire density variation $\delta$ around $v$ is

$$\delta(v) = \frac{\sum_{a \in A} |\sigma(v, a) - \sigma_{avg}|}{|A|}$$  \hspace{1cm} (2)

Then, we can find a new position of $v$ by solving a nonlinear constrained optimization problem to minimize $\delta(v)$. Because moving $v$ infinitely far away from the neighbors will always generate a zero $\delta(v)$, we restrict the new position inside the local area $S$, which is the polygon composed by $\forall a \in |A|$. Figure 7 shows an example of via moving. In practice, a via always crosses a set of layers, and $A$ includes the neighbor vertices from these layers and the effective constraint area is the overlapping of all local areas.
4.3 Wire Distribution through TEG

For a given edge \((a, b)\) in the triangulation graph, the maximum uniform spacing \(\omega(a, b)\) can be determined by Equation 3, where the symbols have the similar meaning as Equation 1.

\[
\omega(a, b) = \frac{((x_a - x_b)^2 + (y_a - y_b)^2)^{\frac{1}{2}} - (r_a + r_b) - \sum_{w \in W} width(w)}{|W|}
\]  

(3)

For each edge \((a, b)\) in the layout, wire distribution can enforce the spacing \(\omega(a, b)\) on the wires crossing this edge by inserting a steiner point into every wire. Considering a new steiner point could add a jog to the wiring, we can set a threshold for the minimum spacing increment for each edge to avoid some low-efficient wire distribution. Moreover, arbitrarily large spacing will neither have a significant impact on better density control while resulting in that extra metal filling is needed in the increased spacing. So a upper bound of \(\omega(a, b)\) is always set by Equation 1 and the specified density rule.

5 Experimental Results

<table>
<thead>
<tr>
<th>Case</th>
<th>Layout size</th>
<th>#Cells</th>
<th>#Nets</th>
<th>#Pins</th>
<th>#Layers</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>1.49e+06</td>
<td>2.9K</td>
<td>3.0K</td>
<td>8.6K</td>
<td>3</td>
</tr>
<tr>
<td>D2</td>
<td>6.70e+05</td>
<td>8.8K</td>
<td>8.8K</td>
<td>47.5K</td>
<td>5</td>
</tr>
<tr>
<td>D3</td>
<td>1.37e+06</td>
<td>14.6K</td>
<td>17.3K</td>
<td>76.8K</td>
<td>5</td>
</tr>
<tr>
<td>D4</td>
<td>3.87e+06</td>
<td>71.2K</td>
<td>36.7K</td>
<td>265K</td>
<td>4</td>
</tr>
<tr>
<td>D5</td>
<td>8.13e+06</td>
<td>88.5K</td>
<td>41.8K</td>
<td>323K</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 1: Experimental examples

Our experimental examples (Table 1) are routed and cell-based design, the scale ranging from 2.9K cells to more than 88K cells. Most of them are the real industry designs. The layout size (in \(\mu m^2\)) and the numbers of cells, nets, pins and metal layers of each example are listed in Table 1. All design examples are in the Cadence LEF/DEF format.

In our experiments, the design rule is to control the layout density within a 10% range in every metal layer. For example, in metal layer 3 of 'D1', the maximum density is 32%, so we need to control the density of every window in this layer between 22% and 32%. At first we perform wire spacing on each example with two different minimum spacing increment thresholds (referred in section 4.3). One threshold is 100%, with which the wire distribution only works at the place where it can increase the wire spacing by at least 100%. The other is 20%, and the wire distribution works more aggressively to increase the wire spacing. The modified layouts are exported back to DEF format. Then, we use Cadence Dracula COVERAGE and GENRECT command to do metal filling on both the original layout and modified layout by wire spacing. The filling rule is based on the original metal density in each layer and we use a same rule on the original and modified layouts for each example. Finally we measure the total metal fill amount in each case and use Carafe to calculate the critical area of the original layouts and filled layouts for yield analysis. Because most layout area is occupied by cells in the bottom metal layer, layout area is occupied by cells, in our experiments we only consider upper metal layers, or the interconnect layers.
5. Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>Wire spacing 100%</th>
<th>Wire spacing 20%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Via mov.</td>
<td>Wire dist.</td>
</tr>
<tr>
<td>D1</td>
<td>81</td>
<td>20</td>
</tr>
<tr>
<td>D2</td>
<td>177</td>
<td>52</td>
</tr>
<tr>
<td>D3</td>
<td>420</td>
<td>120</td>
</tr>
<tr>
<td>D4</td>
<td>1134</td>
<td>407</td>
</tr>
<tr>
<td>D5</td>
<td>1218</td>
<td>405</td>
</tr>
</tbody>
</table>

Table 2: Running time of wire spacing through TEG

![Graph showing running time of wire spacing versus number of pins](image)

Figure 8: Running time of wire spacing versus number of pins

<table>
<thead>
<tr>
<th></th>
<th>Filling only</th>
<th>Wire spacing 100% + filling</th>
<th>Wire spacing 20% + filling</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>1.51e+08</td>
<td>1.33e+08</td>
<td>11.9%</td>
</tr>
<tr>
<td>D2</td>
<td>8.23e+07</td>
<td>7.88e+07</td>
<td>4.3%</td>
</tr>
<tr>
<td>D3</td>
<td>3.00e+08</td>
<td>2.96e+08</td>
<td>1.3%</td>
</tr>
<tr>
<td>D4</td>
<td>4.99e+08</td>
<td>4.83e+08</td>
<td>3.2%</td>
</tr>
<tr>
<td>D5</td>
<td>1.24e+10</td>
<td>1.22e+10</td>
<td>1.6%</td>
</tr>
</tbody>
</table>

Table 3: Amount of metal filling by Dracula

The running time of wire spacing through TEG is listed in Table 2. All numbers are in seconds. Because the different minimum spacing increment thresholds only affect the running of wire distribution, the time for via moving are same for either threshold. The time of via moving and wire distribution also includes the time of DRC and DRV solver on modified layout in each step, respectively. The ‘Other’ stands for the running time to import the original layout into TEG and the time to generated the geometry layout from the wire spacing result. Figure 8 shows the trend of the running time in respect to the number of pins in the layout. We can see this relationship is almost linear.
5. Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>Filling only</th>
<th>Wire spacing 100% + filling</th>
<th>Wire spacing 20% + filling</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Area size</td>
<td>Area size</td>
<td>Area size</td>
<td>Reduc.</td>
</tr>
<tr>
<td>D1</td>
<td>4.55e+09</td>
<td>4.70e+09</td>
<td>4.59e+09</td>
<td>2.3%</td>
</tr>
<tr>
<td>D2</td>
<td>5.56e+09</td>
<td>5.84e+09</td>
<td>5.48e+09</td>
<td>6.3%</td>
</tr>
<tr>
<td>D3</td>
<td>1.95e+10</td>
<td>2.05e+10</td>
<td>1.98e+10</td>
<td>3.4%</td>
</tr>
<tr>
<td>D4</td>
<td>4.53e+10</td>
<td>4.77e+10</td>
<td>4.47e+10</td>
<td>6.3%</td>
</tr>
<tr>
<td>D5</td>
<td>3.02e+10</td>
<td>3.37e+10</td>
<td>3.10e+10</td>
<td>8.0%</td>
</tr>
</tbody>
</table>

Table 4: Critical area measurements by Carafe

Table 3 shows the total fill amount by Dracula and the reduction between metal filling only and wire spacing followed by metal filling. The fill amount is in \( \mu m^2 \). We also compared the change in wire length resulting from wire spacing. We can see, wire spacing with the aggressive threshold 20% always generate larger amount reduction than the higher threshold 100%, but it also results in larger increment in wire length.

Table 4 shows the size of critical area measured by Carafe on the original layouts and layouts after metal filling. We can see, the size of critical area is always increased by the stand-alone filling compared the original unfilled layouts. But with wire spacing, the size of critical area is greatly reduced. Even the results with the higher threshold 100% is near or better than the unfilled layouts in some cases.

Figure 9 and Figure 10 show the filled layouts by metal filling only and by wire spacing followed with metal filling. The wire spacing uses 100% threshold. They are from example D2’, zoom view in the Cadence Virtuoso Layout editor with only metal layer 3 displayed. The shadowed rectangles and polygons (and the small dot-line ones, which are too small for Virtuoso to fill the shadow) are dummy metal generated by Dracula. We can see the fill amount in Figure 9 is obviously less than Figure 10.
Figure 9: Result of metal filling only, ‘D2’ metal layer 2, zoom from Cadence Virtuoso Layout Editor
Figure 10: Result of wire spacing followed by metal filling, ‘D2’ metal layer 2, layout zoom from Cadence Virtuoso Layout Editor
6 Conclusion

In this paper, we proposed a new solution for the layout density control, wire spacing followed by metal filling. Before adding dummy metal into low-density areas, we can increase the wire spacing by moving wires from the high density area toward the low density area. This method successfully reduces the metal fill amount compared with metal filling only. It not only decreases the influence of the fillings on layout interconnect delay, but also improves the manufacture yield in the aspect of less bridging faults. Wire spacing demonstrates the potential of TEG, which is a post layout optimization method based on topological representing of the layout. By TEG, a routed layout can be modified for an optimization target free of the wiring geometry constraints. TEG solves the most difficult obstacle in implementing wire spacing.

References