Coupling-Constrained Dummy Fill for Density Gradient Minimization

Huang-Yu Chen\textsuperscript{1}, Szu-Jui Chou\textsuperscript{2}, Yao-Wen Chang\textsuperscript{3}

\textsuperscript{1}Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan
\textsuperscript{2}Synopsys, Inc, Taipei, Taiwan
\textsuperscript{3}Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan

Abstract

Dummy fill is typically performed by foundries to achieve layout pattern uniformity for chip yield enhancement. However, filling dummies may greatly increase interconnect coupling capacitance and lead to explosion of mask data. Therefore, it is desirable to simultaneously consider the coupling constraints, the number of inserted dummies, and the density gradient during dummy fill. In this paper, we present dummy fill algorithm that adopts a three-stage technique of coupling constraint regulation, gradient-driven multilevel dummy density analysis, and ILP-based fill synthesis for dummy feature minimization. Based on the Gaussian smoothing, the density gradient can be greatly minimized. Experimental results show that our algorithm can reduce the standard deviation of the metal density by up to 37%, the average neighboring density difference by up to 49%, and dummy counts by 41% with a reasonable runtime overhead, which preserves more flexibility for subsequent manufacturability enhancement.

1 Introduction

As nanometer technology advances to 65nm and below, one important yield loss of interconnects comes from the chemical-mechanical polishing (CMP) step in the copper metallization process. A non-uniform feature density distribution on each metal layer causes CMP to over polish or under polish, generating metal dishing and dielectric erosion \cite{9, 11}.

In order to improve CMP quality, dummy fill is a highly recommended technique proposed by foundries. By inserting dummies into layouts, the pattern density variation on each layer can be kept within the density upper and lower bounds. However, in addition to the density variation, the density gradient would also affect the chip planarity, as reported in \cite{1, 6}. Fig. 1 shows the difference between the density variation and the density gradient.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig1}
\caption{(a) and (b) are two wire distributions with the same variation but different gradients, and (c) and (d) are the corresponding density bar graphs of (a) and (b), respectively.}
\end{figure}

In general, dummy fill algorithm has two phases for layout density control: density analysis and fill synthesis. The density analysis determines the available area for dummy features, whereas the fill synthesis determines the number of inserted dummies \cite{4}. The fill synthesis is a key step for gradient-driven dummy fills, since the actual number and positions of dummies would directly affect the density gradient. On the other hand, the density analysis can guarantee the density of inserted dummies under a given density bound, which is critical when considering the coupling constraints. Thus, it is desirable to combine these two phases to form an integrated dummy fill algorithm.

Dummy features may either be connected to power/ground (tied fills) or left floating (floating fills), both of which induce coupling capacitances and affect the timing delay of interconnects. Moreover, dummy fills also dramatically increase the data volume of mask, lengthening the time of mask-making processes and significantly increasing the input data in the following time-consuming reticle enhancement techniques (RETs). There are many previous works presenting dummy fill algorithms, focusing on the wire density control \cite{3, 10}, the gradient variations of the dummy density \cite{5}, or the fill area or lithography costs under coupling constraints \cite{8, 12}. Different from the previous works, our work contains the following distinguished features:

- The first work simultaneously considering the coupling constraints, the dummy counts, and the density gradient.
- A new multilevel framework based on Gaussian smoothing which efficiently solves the gradient minimization problem. The framework can also avoid the processing ordering problem of the sliding window method.
- An integrated flow combining density analysis and fill synthesis, which can ensure that the gradient and the density are well-controlled through the flow.

Experimental results show that our algorithm can reduce the standard deviation of the metal density by up to 37%, the average neighboring density difference by up to 49%, and the number of dummies by up to 41% compared with \cite{12} with a reasonable runtime overhead. The rest of this paper is organized as follows. Section 2 gives the definition of density gradient and the problem formulation. Section 3 presents our multilevel framework to solve the dummy insertion problem. Experimental results are reported in Section 4, and conclusions are given in Section 5.

2 Problem Description

2.1 Definition of Density Gradient

According to the wire-density predictive CMP model characterized in \cite{7}, the post-CMP copper thickness is proportional to the square of wire density within a given tile (region). Thus, it is desirable to minimize the difference among the density of tiles to control the post-CMP variation and minimize the dummy counts. In this work, we define the density gradient as the maximum density difference between adjacent tiles, which conforms to the elementary meaning of the gradient in mathematics.

2.2 Coupling Constraints Regulation (CCR) Problem

We deal with the coupling constraints by adopting the CCR problem definitions of \cite{12}. Suppose the half width of a dummy feature is $w_f$, the minimum spacing between two dummy features is $s_f$, and the minimum spacing between a wire segment and a dummy feature is $s$. For a given routing result on a layer with $N$ wire segments, $S_1$ to $S_N$, each segment $S_i$ has a coupling threshold $C_i$, presenting the maximum allowable induced coupling capacitance of $S_i$. Then, the objective of CCR problem is to find the maximum dummy fill regions (called dummy fill regions) which satisfy all the $s$, $s_f$, and $C_i$ constraints. Note that in this work, we apply the Coupling constrained Dummy Fill (CDF) analysis algorithm \cite{12} for the CCR problem.
2.3 Density Gradient-Driven Dummy Fill Problem

After CCR generates the maximum dummy fill regions, we first transform the layout from slot-based partition into tile-based partition (see Fig. 2), and then a coupling density bound $D_c(t)$ can be computed for each tile $t$, which is an upper bound of the metal density in $t$ and is defined as follows:

**Definition 1** The maximum allowable dummy area of a tile is the area sum of dummies which can be inserted into the dummy fill regions of the tile. The area bound of a tile is the area sum of all segments plus the maximum allowable dummy area. The coupling density bound of a tile is the ratio of the area bound to the area of the tile.

Let $F_u(t)$ and $F_l(t)$ be the respective density upper and lower bounds for a tile $t$ given by foundries, the density upper bound $B_u(t)$ and lower bound $B_l(t)$ of $t$ can be defined as:

$$B_u(t) = \min \{D_c(t), F_u(t)\}$$

$$B_l(t) = \max \{d_s(t), F_l(t)\},$$

where $d_s(t)$ represents the segment density of $t$.

With the density bounds $B_u$ and $B_l$, we can guarantee that the dummy insertion would satisfy both the coupling constraints and foundry density rules. As a result, we do not need to explicitly consider the coupling constraints in the following procedures. Then, the gradient-driven dummy metal insertion problem can be defined as follows:

- Given a routing result partitioned into global tiles, where each tile has specified density upper bound $B_u$ and density lower bound $B_l$, our goal is to minimize the gradient of metal density and the number of dummies.

![Figure 2: Dummy fill region transformation from slot-based partition to tile-based partition. (a) Slot-based one. (b) Tile-based one.](image)

Note that for each tile, the coupling density bound is the maximum density with no coupling violations, but it might not result in the minimum gradient. Thus, the coupling density bound will be treated as an upper bound in the following procedures.

3 Gradient-Driven Dummy Fill Algorithm

In this section, a dummy fill algorithm is presented. First, a multilevel density analysis is presented to decide the density of each tile. Then, minimum number of dummies are inserted to achieve the dummy density.

3.1 Multilevel Dummy Density Analysis

Traditionally, the methods for density analysis and fill synthesis are usually based on discretization; e.g., the layout is partitioned into tiles, and a sliding window is used to compute the density of tiles [5]. However, these methods may incur the "discretization gap" problem and the ordering problem in density analysis.

To remedy this deficiency, we present multilevel density analysis to handle the discretization gap and avoid the ordering problem. We denote an insertion graph of level $i$ as $G_i$, and each $G_i$ will be divided into tiles with different size. A tile of $G_i$ is denoted as $t_i(x, y)$, where $x$ and $y$ are the coordinates, and the bottom-left tile is $t_i(1, 1)$. For convenience, if we do not specify the level of a tile, it is a level-0 tile, as all the tiles mentioned in the previous sections; if we do not specify the coordinates of a tile $t_i$, it can be an arbitrary tile of level $i$. Each $G_i$ contains several non-overlapping windows. A window of $G_i$ is denoted by $w_i(x, y)$, where $x$ and $y$ are the coordinates.

The framework contains two stages of coarsening followed by uncoarsening. In the coarsening stage, we decide the density of each $t_i$ by performing the local gradient minimization within each window independently. This process is called density assignment. Then the $W^2$ tiles of each window are merged into a $t_{i+1}$ tile for the next level $i + 1$. In the uncoarsening stage, we extract the density of each tile. For level $i$, each $t_i$ is recovered to $W^2$ number of $t_{i-1}$. The process repeats until level 0 is reached, and the extracted density of each tile is used to compute the number of dummies, which will be described in Section 3.2.

To express the relation between tiles of different levels, some notations are given. Let $T_i(x, y)$ be the set of level-$i$ tiles contained by window $w_i(x, y)$, such that the set would be merged into the level-$i+1$ tile $t_{i+1}(x, y)$. $T_i(x, y)$ can be defined as $\{t_i(x, y) | [x/W] = \hat{x}, [y/W] = \hat{y}\}$ for all levels, except the top level $L$.

3.1.1 Coarsening Stage

At level 0, the density upper and lower bounds of $t_0$ have already been defined as $B_u(t_0)$ and $B_l(t_0)$, respectively. Suppose for level $i$, $B_u$ and $B_l$ have already been defined. Then, Gaussian smoothing is performed to decide the density of each $t_i$ while minimizing the gradient inside each window. Let $d_s(t_i)$ be the original density of $t_i$ before Gaussian smoothing, and $d_s(t_i)$ be the density after Gaussian smoothing. Then, $d_s(t_i)$ and $d_s(t_i)$ can be computed as follows:

$$d_s(t_i(x, y)) = \begin{cases} 
\frac{d_s(t_{i-1}(x, y))}{\text{avg}(d_s(t_{i-1}(x, y)))}, & \text{if } i = 0 \\
\frac{d_s(t_{i-1}(x, y))}{\text{avg}(d_s(t_{i-1}(x, y)))}, & \text{if } i \geq 1
\end{cases}$$

For density assignment considering local gradient minimization, 2-D Gaussian smoothing is performed on $t_i(x, y)$ and the surrounding tiles, as expressed by the function $G(t_i(x, y))$. Each tile is multiplied by a weight, which is defined as

$$g(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x-x)^2 + (y-y)^2}{2\sigma^2}\right),$$

where $\sigma$ is the standard deviation of the distribution and can be used to adjust the weight and the covering range of Gaussian smoothing. Then, Gaussian smoothing at $t_i(x, y)$ can be computed by

$$G(t_i(x, y)) = \sum (g(x, y)d_s(t_i(x, y))).$$

Since $g(x, y)$ will be truncated to 0 when the distance between $t_i(x, y)$ and $t_{i+1}(x, y)$ is large, only the tiles with a certain covering range, according to the value of $\sigma$, need to be computed. After Gaussian smoothing, if the value of $G(t_i(x, y))$ exceeds the density upper bound $B_u(t_i(x, y))$, $d_s(t_i(x, y))$ will be truncated to $B_u$, and so is the lower bound $B_l$.

After level-$i$ density assignment, the coarsening stage enters the next level $i + 1$, and $W^2$ level-$i$ tiles will be merged into one $t_{i+1}$. The density of $t_{i+1}$ before the density assignment is the average density of those $W^2$ level-$i$ tiles. The density bounds for level-$(i + 1)$ tiles need to be re-computed as follows:

$$B_u(t_{i+1}(x, y)) = d_s(t_{i+1}(x, y)) + \min\{B_u(t_i(x, y)) - d_s(t_i(x, y)), t_i(x, y) \in T_i(x, y)\}$$

$$B_l(t_{i+1}(x, y)) = d_s(t_{i+1}(x, y)) - \min\{d_s(t_i(x, y)) - B_l(t_i(x, y)), t_i(x, y) \in T_i(x, y)\}.$$
The coarsening stage continues until the topmost level $L$ is reached, where the insertion graph $G_L$ can be entirely covered by a single window. For the topmost level, the density values of the level-$L$ tiles are the final value of the insertion, then we perform the uncoarsening stage to extract the density value of the tiles. Fig. 3 illustrates an example with three levels. In level 0 of the coarsening stage, gradient minimization is performed in a $3 \times 3$ window (represented by the red square), and then the average density of these nine level-0 tiles is taken as the initial density of a level-1 tile, which is $(0.2 + 0.2 + 0.2 + 0.2 + 0.2 + 0.2 + 0.3 + 0.3 + 0.3)/9 \simeq 0.23$. Then, level-1 gradient minimization is performed, and the average density of these level-1 tiles is taken as the initial density of a level-2 tile, which is $(0.26 + 0.27 + 0.28 + 0.28 + 0.29 + 0.30 + 0.30 + 0.30 + 0.30)/9 = 0.28$. After level-2 gradient minimization completes, the coarsening stage is finished, and then the flow will enter the uncoarsening stage to extract the final insertion density.

3.1.2 Uncoarsening Stage

After the final density of the topmost level $L$ is calculated, the level-$L$ tiles are decomposed level by level to extract the density of level-0 tiles. Note that the windows in the coarsening stage are not needed in the uncoarsening stage. Since in the coarsening stage, the density value $d_i$ of a tile $t_{i+1}(x, y)$ is the average of the $W^2$ level-$i$ tiles in $T_i(x, y)$, the extracted values $d_u$ of $t_i$ can simply be computed as follows:

$$d_u(t_i(x, y)) = \begin{cases} d_y(t_i(x, y)), & \text{if } i = L \\ d_y(t_i(x, y)) + d_u(t_{i+1}(x, y)) - d_y(t_{i+1}(x, y)), & \forall t_i(x, y) \in T_i(x, y), \text{if } i < L \end{cases}$$

As illustrated in the right part of Fig. 3, after level-2 gradient minimization in the coarsening stage, the density of the bottom-right level-2 tile is changed from 0.28 to 0.31, so the density of each associated level-1 tile is increased by 0.03. For the bottom-right level-1 tile, since its density is changed from 0.23 to 0.31, the density in each associated level-0 tile is increased by 0.08. Finally, the insertion density of each level-0 tile is extracted, and we can use dummy number assignment to decide the number of inserted dummies.

3.2 Dummy Number Assignment

After the multilevel density analysis, the density of each tile has been determined, and thus the density of dummy features in a tile $t$ can be computed as $d_d(t) = d(t) - d_u(t)$. Since the tile must be a level-0 tile and each tile is processed independently, we omit the level subscript and the coordinates for convenience.

The objective of dummy number assignment is to decide the number of dummies inserted in each dummy fill region and minimize the number of total dummies. For a given tile $t$, suppose there are $n$ dummy fill regions $R_1, R_2, \ldots, R_n$. Let $r_i$ be the number of dummies inserted in $R_i$, and $u_i$ be the size of $R_i$. The dummy features in the same dummy fill region will have the same shape and area. Let $a_i$ be the area of one dummy feature in $R_i$. Then, the dummy number assignment (DNA) problem can be formulated as the following ILP:

$$\text{minimize } \sum_{i=1}^{n} r_i$$

subject to

$$d_d(t) a_t - \frac{a_{\text{max}}}{2} \leq \sum_{i=1}^{n} (a_t r_i) \leq d_d(t) a_t + \frac{a_{\text{max}}}{2}$$

$$0 \leq r_i \leq u_i, i = 1, \ldots, n$$

where $a_t$ is the area of the tile, and $a_{\text{max}}$ is the maximum $a_i$ among all dummy fill regions.

In the area constraint as shown in first constraint, $d_d(t) a_t$ is the desirable dummy area of tile $t$. However, since the area of dummy features can just be certain values in a tile, we may fail to find any feasible solution if the area constraint is set to $\sum_{i=1}^{n} a_t r_i = d_d(t) a_t$. For example, if $d_d(t) a_t = 2.5$ and the area of a dummy features is 1 for all $R_i$, there exists no feasible solution for $\sum_{i=1}^{n} r_i = d_d(t) a_t = 2.5$ since $r_i$ must be an integer. Thus, a tolerance must be added in the area constraint to ensure the feasibility.

Theorem 1 With a tolerance of $a_{\text{max}}/2$, the feasibility of the ILP can be guaranteed.

With the objective function, the number of dummies in each tile can be minimized, and so is the total number of dummies of the whole layout.

4 Experimental Results

Our dummy insertion algorithm was implemented in the C++ programming language on a 2 GHz AMD-64 workstation with 8 GB memory. We used the lp_solve package as the ILP solver and performed experiments on two suites of benchmarks, the MCNC and Faraday benchmarks. The routing results for 11 circuits of the Faraday benchmark were obtained from [2], whereas those of five circuits of the Faraday benchmark were generated from Cadence SOC Encounter.

We compared the proposed algorithm with the CDF algorithm [12]. We set the window size $W$ and $\rho$ of Gaussian smoothing as 3 and 1.0, respectively. The number of levels $L$ depends on the size of the layout, which affects the size of the level-0 tiles; that is, the layout was divided into $W^L \times W^L$ tiles. We set the density upper and lower bounds as 80% and 20%, respectively, which are the default values set in most commercial tools. Further, the width and spacing of dummy features were the same as those of wire segments.

Table 1 shows the comparison results of metal density, which is the summation of the segment and dummy density. In the ta-
Table 1: Comparison for density of tiles.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Ours</th>
<th>Coupling Constrained Dummy Fill Analysis Algorithm [12]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#Dummy</td>
<td>Density among All Layers</td>
</tr>
<tr>
<td>Mcc1</td>
<td>1262100</td>
<td>Avg. Max Std. Avg. Max Std. Avg. Max Std. Avg. Max Std.</td>
</tr>
<tr>
<td>Mcc2</td>
<td>2017381</td>
<td>Avg. Max Std. Avg. Max Std. Avg. Max Std. Avg. Max Std.</td>
</tr>
<tr>
<td>Struct</td>
<td>8996228</td>
<td>Avg. Max Std. Avg. Max Std. Avg. Max Std. Avg. Max Std.</td>
</tr>
<tr>
<td>Primary1</td>
<td>7081135</td>
<td>Avg. Max Std. Avg. Max Std. Avg. Max Std. Avg. Max Std.</td>
</tr>
<tr>
<td>Primary2</td>
<td>24892846</td>
<td>Avg. Max Std. Avg. Max Std. Avg. Max Std. Avg. Max Std.</td>
</tr>
<tr>
<td>S378</td>
<td>269876</td>
<td>Avg. Max Std. Avg. Max Std. Avg. Max Std. Avg. Max Std.</td>
</tr>
<tr>
<td>S924</td>
<td>230173</td>
<td>Avg. Max Std. Avg. Max Std. Avg. Max Std. Avg. Max Std.</td>
</tr>
<tr>
<td>S13207</td>
<td>657840</td>
<td>Avg. Max Std. Avg. Max Std. Avg. Max Std. Avg. Max Std.</td>
</tr>
<tr>
<td>S15850</td>
<td>721298</td>
<td>Avg. Max Std. Avg. Max Std. Avg. Max Std. Avg. Max Std.</td>
</tr>
<tr>
<td>S38417</td>
<td>2100467</td>
<td>Avg. Max Std. Avg. Max Std. Avg. Max Std. Avg. Max Std.</td>
</tr>
<tr>
<td>S83844</td>
<td>2100467</td>
<td>Avg. Max Std. Avg. Max Std. Avg. Max Std. Avg. Max Std.</td>
</tr>
<tr>
<td>S5378</td>
<td>269876</td>
<td>Avg. Max Std. Avg. Max Std. Avg. Max Std. Avg. Max Std.</td>
</tr>
<tr>
<td>Risc1</td>
<td>4251368</td>
<td>Avg. Max Std. Avg. Max Std. Avg. Max Std. Avg. Max Std.</td>
</tr>
<tr>
<td>Dsp2</td>
<td>941447</td>
<td>Avg. Max Std. Avg. Max Std. Avg. Max Std. Avg. Max Std.</td>
</tr>
<tr>
<td>Dsp1</td>
<td>1371351</td>
<td>Avg. Max Std. Avg. Max Std. Avg. Max Std. Avg. Max Std.</td>
</tr>
<tr>
<td>Comp.</td>
<td>1.00</td>
<td>Avg. Max Std. Avg. Max Std. Avg. Max Std. Avg. Max Std.</td>
</tr>
</tbody>
</table>

Figure 4: The dummy insertion result of layer 1 for “S5378”. (a) By [12]. (b) By our algorithm.

Figure 5: The metal density distribution of layer 1 for “S5378”. (a) Dummy insertion results by [12]. (b) Dummy insertion results by our algorithm.

5 Conclusion

In this paper, we have developed a dummy fill algorithm considering the density gradient minimization, the coupling constraints, and the number of inserted dummies. Experimental results have shown that the proposed algorithm can reduce the standard deviation of the metal density by up to 37%, and the average neighboring density difference by up to 49%, with a reasonable runtime overhead. In addition, the number of inserted dummies by our algorithm is only 59% compared with [12], which preserves more flexibility for the RET processes.

References