Density Gradient Minimization with Coupling-Constrained Dummy Fill for CMP Control

Huang-Yu Chen\textsuperscript{1}, Szu-Jui Chou\textsuperscript{1}, and Yao-Wen Chang\textsuperscript{1,2}
\textsuperscript{1}Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan
\textsuperscript{2}Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan

ABSTRACT

In the nanometer IC design, dummy fill is often performed to improve layout pattern uniformity and the post-CMP quality. However, filling dummies might greatly increase interconnect coupling capacitance and thus circuit delay, and might also lead to explosion of mask data due to the extra layout patterns. Traditional dummy-fill algorithms try to make each tile (window) density satisfy foundry’s density upper and lower bounds under the coupling constraint. As technology advances, however, it is not sufficient to just keep the pattern density variation of each layer within density bounds. The density gradient, besides the density variation, plays a pivotal role in determining the post-CMP thickness of modern circuit designs. In this paper, we present the first gradient-driven dummy-fill algorithm to address the density gradient and other classical objectives (such as density variation, coupling constraints, dummy count) as well. Our dummy-fill algorithm has the two distinguished features: (1) Gaussian smoothing based gradient-driven multi-level dummy density analysis to minimize density gradient level by level, and (2) ILP-based fill synthesis to insert the fewest dummies within the coupling-violation-free feasible regions while satisfying the density constraints. Experimental results show that our algorithm can achieve promising results by inserting minimal dummies to reduce the density gradient and variation under the coupling constraints with a reasonable runtime overhead.

Categories and Subject Descriptors
B.7.2 [Integrated Circuits]: Design Aids - Layout

General Terms
Algorithms, Designs, Reliability

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

\textsuperscript{*}This work was partially supported by National Science Council of Taiwan under Grant No’s. NSC 97-2221-E-002-237-MY3, NSC 96-2628-E-002-249-MY3, NSC 96-2628-E-002-248-MY3.

Keywords
Chemical-Mechanical Polishing, Density Gradient, Dummy Fill, Manufacturability

1. INTRODUCTION

As IC technology advances to 65nm node and beyond, one important interconnect yield loss comes from the copper metallization in the CMP (chemical-mechanical polishing) process. A non-uniform pattern density distribution on each layer causes CMP to over polish or under polish, generating metal dishing and dielectric erosion [7, 9]. In order to improve CMP quality, dummy fill is a highly accepted process to keep density variation of tiles (windows) on each layer within the density upper and lower bounds recommended by foundries.

For dummy fill, there are some well-known performance considerations to evaluate the filling results, including circuit timing degradation and restrictions on the number of dummies. On one hand, dummy patterns may either connect to power/ground nets (tied fills) or left floating (floating fills), both of which may induce coupling capacitances, affect interconnect delay, and lead to timing violations [13]. On the other hand, those extra dummy patterns increase the mask data size, lengthening the time of mask-making processes and complicating the following time-consuming reticle enhancement techniques, such as OPC (optical proximity correction).

Traditionally, dummy-fill algorithms adopt a two-stage technique of density analysis followed by fill synthesis [2]. The density analysis first determines the available regions for dummy patterns filling, which is a key step for controlling the density of inserted dummies in each tile to satisfy the density bounds under coupling constraints. The fill synthesis then performs dummy insertion into these feasible regions, responsible for the final positions and the number of inserted dummies.

As technology advances, however, it is already not sufficient to just keep the pattern density variation of each tile within density upper and lower bounds. The density gradient (defined as the maximum density difference among a tile and its adjacent tiles), besides the density variation, has emerged as a crucial objective that significantly affects the post-CMP thickness planarity as well as mask distortion, and thus must be considered during dummy fill (see articles from both academia [12] and industry—XYALIS [6], Blaze DFM [10, 11], and VCMP [14]). Figure 1 shows the difference between the density variation and the density gradient.
Density distribution in the tiles of Figures 1 (a) and (b) has the same density variation of 0.0523, while the distribution in (b) results in a smaller density gradient (= 0.4) than that in (a) (= 0.7) and thus can alleviate dummy-fill efforts (by inserting fewer dummies to achieve desired CMP quality).

There are many performance-driven dummy-fill algorithms in the literature, focusing on the wire density control [1, 8], the variations of the dummy density [3], or the fill area or lithography costs under coupling constraints [5, 15]. However, none of the existing works well addresses the density gradient issues.

In this paper, we present the first gradient-driven dummy-fill algorithm to address the density gradient and other classical objectives (such as density variation, coupling constraints, dummy count) as well. Different from the previous works, our dummy-fill algorithm has the two key distinguished ingredients: (1) Gaussian smoothing based gradient-driven multilevel dummy density analysis to minimize density gradient level by level, and (2) ILP-based fill synthesis to insert the fewest dummies within the coupling-violation-free feasible regions while satisfying the density constraints.

In addition, our work also contains the following significant features:

- The first work in the literature that simultaneously considers the coupling constraints, the dummy counts, and the density gradient.
- A novel multilevel framework based on Gaussian smoothing that efficiently solves the gradient minimization problem. In particular, this framework can also avoid the processing ordering problem arising from the popular sliding window method.
- A unified flow that combines density analysis and fill synthesis to facilitate the control of the wire density and gradient through the flow.

Experimental results show that our algorithm can reduce the average density gradient by up to 63%, the standard deviation of the metal density by up to 49%, and the number of dummies by up to 19% under the coupling constraints, compared with a modified CDF algorithm [15] 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

In this section, we first give a formal definition of density gradient and then formulate the dummy metal insertion problem. There are two parts of the problem: coupling constraints regulation and gradient-driven dummy metal insertion. Coupling constraints regulation is to handle the coupling constraints and compute the maximum allowable dummy insertion region, and gradient-driven dummy metal insertion is to assign the dummy density and then insert dummy features.
2.1 Definition of Density Gradient

According to the wire-density predictive CMP model characterized in [4], 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.

Definition 1: The density gradient is the maximum density difference among a tile and its adjacent tiles.

Note that the definition 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 defined in [15]. Let the half width of a dummy feature be \( w_f \), the minimum spacing between two dummy features be \( s_f \), and the minimum spacing between a wire segment and a dummy feature be \( s \). 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 \( C_i \) representing the maximum allowed induced coupling capacitance of \( S_i \). Then, the objective of the 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 [15] 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 partitions into tile-based partitions (see Figure 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:

\[
D_c(t) = \frac{w_f}{c(t) - d(t)}
\]

where \( c(t) \) is the coupling density upper bound of the tile \( t \), and \( d(t) \) is the maximum density with no coupling violations, but it might not result in the minimum gradient. Thus, the coupling-driven dummy-metal insertion problem can be defined as follows.

The Gradient-Driven Dummy-Metal Insertion Problem: Given a routed layout partitioned into global tiles, where each tile has the specified density upper bound \( B_u \) and density lower bound \( B_l \), minimize the density gradient and the number of dummy.

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 novel dummy-fill algorithm is presented. First, a multilevel density analysis is presented to decide the density of each tile. According to the density analysis, the minimum number of dummy is then inserted to achieve the dummy-density requirement. Figure 4 illustrates the overall algorithm flow.

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 [3]. However, these methods may incur the “discretization gap” and the processing ordering problems in density analysis.

To remedy this deficiency, we present multilevel density analysis to handle the discretization gap and avoid the processing ordering problem. We denote an insertion graph of level \( i \) as \( G_i \), and each \( G_i \) is divided into tiles of different sizes. A tile of \( G_i \) is denoted by \( t_i(x, y) \), where \( x \) and \( y \) give the coordinate of the tile, with the bottom-left tile being \( t_i(1, 1) \). For convenience, if we do not specify the level of a tile, it is a level-0 tile; if we do not specify the coordinate of a tile \( t_i \), it can be an arbitrary tile of level \( i \). Each \( G_i \) contains a set of non-overlapping windows. A window of \( G_i \) is denoted by \( w_i(x, y) \), where \( x \) and \( y \) give the coordinate.

The multilevel 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
The process is called density assignment. Then the $W^2(W \times W)$ 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 explained in Section 3.2.

To express the relation between tiles of different levels, some notations are given here. Let $T_i(\hat{x}, \hat{y})$ be the set of level-$i$ tiles contained by window $w_i(\hat{x}, \hat{y})$, such that the set would be merged into the level-$(i+1)$ tile $t_{i+1}(\hat{x}, \hat{y})$; in other words, $T_i(\hat{x}, \hat{y}) = \{t_i(x,y)|x/W = \hat{x}, [y/W] = \hat{y}\}$ 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 $d_g(t_i)$ be the density after Gaussian smoothing. Then, $d_s(t_i)$ and $d_g(t_i)$ can be computed as follows:

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

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{(-\frac{(x-\hat{x})^2+(y-\hat{y})^2}{2\sigma^2})},$$

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 $t_i(\hat{x}, \hat{y})$ can be computed by

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

Since $g(x,y)$ will be truncated to 0 when the distance between $t_i(x,y)$ and $t_i(\hat{x}, \hat{y})$ is large, only the tiles with a certain covering range, according to the value of $\sigma$, need to be computed. The effect of Gaussian smoothing is illustrated in Figure 5, which comes from the experimental result of circuit S5378. After Gaussian smoothing, if the value of $G(t_i(\tilde{x}, \tilde{y}))$ exceeds the density upper bound $B_u(t_i(\hat{x}, \hat{y}))$, $d_g(t_i(\hat{x}, \hat{y}))$ will be truncated to $B_u$, and so is the lower bound $B_l$.

After the level-$i$ density assignment, the coarsening stage enters the next level $i + 1$, and the $W^2$ 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 the level-$(i+1)$ tiles need to be recomputed as follows:

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

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

where $t_i(x,y)$ is the $i$th level of each tile.

![Figure 3: Illustration of the coarsening and uncoarsening stages with three levels. This example focuses on how the density of tiles changes between levels and neglects the detailed computation of the density assignment.](image-url)
corresponding level-1 tile, which is \((0, 0)\).

The coarsening stage continues until the topmost level \(L = 9\) 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 give the final value of the insertion. Then, we extract the density value of the tiles at the uncoarsening stage. Figure 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/bold square), and then the average density of these nine level-0 tiles is taken as the initial density of the corresponding level-1 tile, which is \((0.2 + 0.2 + 0.2 + 0.2 + 0.2 + 0.3 + 0.3 + 0.3 + 0.3)/9 \approx 0.23\). Then, level-1 gradient minimization is performed, and the average density of these level-1 tiles is taken as the initial density of the corresponding level-2 tile, which is \((0.26 + 0.27 + 0.28 + 0.28 + 0.28 + 0.28 + 0.29 + 0.29 + 0.30)/9 \approx 0.28\). After the level-2 gradient minimization, the coarsening stage is finished, and then the uncoarsening stage starts and it extracts 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 \(d_u\) of a tile \(t_i(x, y)\) is the average of the \(W^2\) level-\(i\) tiles in \(T_i(\hat{x}, \hat{y})\), the extracted value of \(t_i(x, y)\), \(d_u(t_i(x, y))\), can be computed as follows:

\[
d_u(t_i(x, y)) =
\begin{cases}
  d_g(t_L(x, y)), & \text{if } i = L \\
  d_g(t_i(x, y)) + d_u(t_{i+1}(\hat{x}, \hat{y})) - d_g(t_{i+1}(\hat{x}, \hat{y})), & \text{if } i < L.
\end{cases}
\]

As illustrated in the right part of Figure 3, after the 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, and thus the density of each corresponding 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 corresponding level-0 tile is increased by 0.08. Finally, the insertion density of each level-0 tile is extracted, and we can perform 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 \(d_d(t) = d_u(t) - d_g(t)\). Since the tile must be a level-0 tile and each tile is processed independently, we omit the level subscript and the coordinates for easier presentation.

#### 3.2.1 ILP Formulation

The objective of the dummy number assignment is to determine 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

![Figure 5: The metal density distribution of layer 1 for the circuit S5378. (a) The distribution before Gaussian smoothing. (b) The distribution after Gaussian smoothing.](image)
dummy-fill region will have the same shape and area. Let \( a_i \) be the area of one dummy feature in \( R_t \). Then, the dummy number assignment (DNA) problem can be formulated as the following ILP:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{n} r_i \\
\text{subject to} & \quad d_d(t)a_i - \frac{a_{\text{max}}}{2} \leq \sum_{i=1}^{n} a_i r_i \leq d_d(t)a_i + \frac{a_{\text{max}}}{2}, \\
& \quad 0 \leq r_i \leq u_i, \quad i = 1, \ldots, n,
\end{align*}
\]

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

In the area constraint for the left inequality of the first constraint, \( d_d(t)a_i \) is the desirable dummy area of tile \( t \). However, since the area of dummy features can just be a certain value in a tile, for example, the deviation of given dummy area from its average is small, the area constraint can be set to \( \sum_{i=1}^{n} a_i r_i = d_d(t)a_i \). For example, if \( d_d(t)a_i = 2.5 \), then we can also set the area constraint as \( \sum_{i=1}^{n} a_i r_i = d_d(t)a_i = 2.5 \) since \( r_i \) must be an integer. Thus, a tolerance must be added in the area constraint to ensure the feasibility, and we have the following theorem.

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.

3.2.2 Solution Space Reduction

If the desirable dummy density \( d_d(t) \) is relatively small compared to the density upper bound, the solution space of ILP can further be reduced. For a dummy-fill region \( R_t \), if the maximum dummy area \( a_i u_i \) exceeds the total desirable dummy area \( d_d(t)a_i \), we can set a new bound \( u_i' = \min(r_i | a_i r_i > d_d(t)a_i) \); that is, there exists no feasible solution when \( r_i > u_i' \), and thus \( u_i \) can be replaced with \( u_i' \). This method can effectively reduce the solution space when \( d_d(t) \) is small and maintain the optimality. However, if \( d_d(t) \) is large, the solution space might be hard to be reduced.

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 the industrial Faraday benchmarks.

For fair comparison, we implemented the modified CDF algorithm [15], named CDFm. Unlike the original CDF algorithm that tries to insert as many dummies as possible to slots under the coupling constraint, the CDFm algorithm also honors the density lower and upper bound rules to control the number of inserted dummies. 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 lower and upper bounds as 20% and 60%, 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 density gradient for both algorithms. In the table, “\#Dummy” shows the number of inserted dummies, “Density among layers” and “Density of layer 1” give the statistics of density gradient among all layers and layer one, respectively.

Compared with the CDFm algorithm, for density gradient among all metal layers, our approach can achieve respective 63%, 34%, and 49% reductions for the average, the maximum, and the standard deviation. Considering the metal one only, ours can also achieve respective 68%, 69%, and 47% reductions in terms of the average, the maximum, and the standard deviation, compared to CDFm.

Note that since our dummy insertion algorithm adopts the dummy fill region computation in [15] as CDFm, both dummy insertion results of ours and CDFm would meet the coupling constraints.

Also note that since our dummy insertion is performed layer by layer, the statistics of a single layer is more significant than that of all layers.

Moreover, the number of dummies inserted by our algorithm is on average only 19% of that inserted by CDFm. This great reduction is mainly contributed from the gradient minimization of our algorithm. If the original density of a region before dummy fills is low and uniformly distributed, our algorithm will not insert any dummy into that region since the density gradient is small, but the CDFm algorithm would still try to insert as many dummies as possible under the coupling constraint. The runtime overhead of our algorithm is only 19% over CDFm, implying that our method can effectively reduce the density variation within an acceptable runtime. The dummy inserted layouts of S3378 for the metal one by both algorithms are also shown in Figure 6.

5. CONCLUSION

We have developed the first density-gradient driven dummy-fill algorithm considering the density gradient and variation minimization, the coupling constraints, and the number of inserted dummies simultaneously. Gaussian smoothing based multilevel dummy density analysis and ILP-based fill synthesis are two key techniques to handle the simultaneous considerations. Experimental results have shown that the proposed algorithm can reduce the average, the maximum, and the standard deviation of metal density and density gradient with a reasonable runtime overhead. In addition, the number of inserted dummies by our algorithm is only 19%, compared with an extension of the state-of-the-art work [15], which preserves more flexibility for the resolution enhancement technique (RET) processes.

6. REFERENCES


Table 1: Comparison for density gradient between tiles.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>#Dummy</th>
<th>Density gradient among layers</th>
<th>Ours</th>
<th>Density gradient among layers</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>avg.</td>
<td>max.</td>
<td>std.</td>
<td>avg.</td>
<td>max.</td>
</tr>
<tr>
<td>Mcc1</td>
<td>1,262,298</td>
<td>7.14%</td>
<td>8.75%</td>
<td>5.87%</td>
<td>5.28%</td>
</tr>
<tr>
<td>Mcc2</td>
<td>20,117,831</td>
<td>4.40%</td>
<td>14.67%</td>
<td>2.39%</td>
<td>3.53%</td>
</tr>
<tr>
<td>Struct</td>
<td>9,004,650</td>
<td>1.41%</td>
<td>5.75%</td>
<td>1.34%</td>
<td>0.56%</td>
</tr>
<tr>
<td>Primary1</td>
<td>7,102,170</td>
<td>2.38%</td>
<td>13.09%</td>
<td>2.54%</td>
<td>1.61%</td>
</tr>
<tr>
<td>Primary2</td>
<td>24,897,686</td>
<td>1.22%</td>
<td>3.97%</td>
<td>1.98%</td>
<td>1.42%</td>
</tr>
<tr>
<td>S5378</td>
<td>269,916</td>
<td>5.38%</td>
<td>15.88%</td>
<td>2.53%</td>
<td>5.38%</td>
</tr>
<tr>
<td>S9234</td>
<td>230,220</td>
<td>6.31%</td>
<td>20.79%</td>
<td>3.18%</td>
<td>6.27%</td>
</tr>
<tr>
<td>S13207</td>
<td>9,004,650</td>
<td>1.41%</td>
<td>5.75%</td>
<td>1.34%</td>
<td>0.56%</td>
</tr>
<tr>
<td>S15850</td>
<td>7,102,170</td>
<td>2.38%</td>
<td>13.09%</td>
<td>2.54%</td>
<td>1.61%</td>
</tr>
<tr>
<td>S38417</td>
<td>2,100,467</td>
<td>2.91%</td>
<td>9.32%</td>
<td>1.42%</td>
<td>2.28%</td>
</tr>
<tr>
<td>S38584</td>
<td>2,460,061</td>
<td>2.80%</td>
<td>9.32%</td>
<td>1.42%</td>
<td>2.28%</td>
</tr>
</tbody>
</table>

Figure 6: The dummy insertion result of the metal one for S5378. (a) By CDFm. (b) By our algorithm. (The red segments and blue rectangles represent wires and dummies in metal one, respectively.)