# Fast Dual Graph Based Hotspot Detection


Andrew B. Kahng,[a,b,c] Chul-Hong Park[a] and Xu Xu[b,c]

[a]UCSD ECE Department, La Jolla, CA
[b]UCSD CSE Department, La Jolla, CA
[c]Blaze DFM, Inc., Sunnyvale, California



## ABSTRACT

As advanced technologies in wafer manufacturing push patterning processes toward lower-$k_1$ subwavelength printing, lithography for mass production potentially suffers from decreased patterning fidelity. This results in generation of many *hotspots*, which are actual device patterns with relatively large CD and image errors with respect to on-wafer targets. Hotspots can be formed under a variety of conditions such as the original design being unfriendly to the RET that is applied, unanticipated pattern combinations in rule-based OPC, or inaccuracies in model-based OPC. When these hotspots fall on locations that are critical to the electrical performance of a device, device performance and parametric yield can be significantly degraded. Previous rule-based hotspot detection methods suffer from long runtimes for complicated patterns. Also, the model generation process that captures process variation within simulation-based approaches brings significant overheads in terms of validation, measurement and parameter calibration.

In this paper, we first describe a novel detection algorithm for hotspots induced by lithographic uncertainty. Our goal is to rapidly detect all lithographic hotspots without significant accuracy degradation. In other words, we propose a filtering method: as long as there are no "false negatives", i.e., we successfully have a superset of actual hotspots, then our method can dramatically reduce the layout area for golden hotspot analysis. The first step of our hotspot detection algorithm is to build a ***layout graph*** which reflects pattern-related CD variation. Given a layout $L$, the ***layout graph*** $G = (V, E_c \cup E_p)$ consists of nodes $V$, corner edges $E_c$ and proximity edges $E_p$. A face in the layout graph includes several close features and the edges between them. Edge weight can be calculated from a traditional 2-D model or a lookup table. We then apply a three-level hotspot detection: (1) edge-level detection finds the hotspot caused by two close features or "$L$-shaped" features; (2) face-level detection finds the pattern-related hotspots which span several close features; and (3) merged-face-level detection finds hotspots with more complex patterns. To find the merged faces which capture the pattern-related hotspots, we propose to convert the layout into a planar graph $G$. We then construct its dual graph $G^D$ and sort the dual nodes according to their weights. We merge the sorted dual nodes (i.e., the faces in $G$) that share a given feature, in sequence. We have tested our flow on several industry testcases. The experimental results show that our method is promising: for a 90nm metal layer with 17 hotspots detected by commercial optical rule check (ORC) tools, our method can detect all of them while the overall runtime improvement is more than 287X.


## 1. INTRODUCTION AND MOTIVATION

Moore's Law continues to drive higher performance with smaller circuit features. Aggressive technology scaling has introduced new variation sources and made process variation control more difficult. For optical lithography, manufacturability is roughly defined by the $k_1$ factor from the Rayleigh equation. Beyond the 45nm CMOS technology node, even using a high-end optical exposure system such as immersion lithography with higher NA, it is necessary to have a $k_1$ factor is lower than 0.35. The primary risk posed by lower $k_1$ is the likelihood of degradation of patterning fidelity on VLSI circuits. Lower $k_1$ could decrease patterning fidelity and result in generation of many hotspots; a hotspot is an actual device pattern which has relatively large CD and image errors with respect to on-wafer targets. Under ultra-low $k_1$ conditions ($k_1 < 0.3$), in particular, many hotspots may arise anywhere. Hotspots can form under a variety of conditions such as the original design being unfriendly to the RET that is applied to the chip, pattern combinations unanticipated by rule-based OPC, or inaccuracies in model-based OPC. When these hotspots fall on locations that are critical to the electrical performance of a device, they can reduce the yield and performance of the device. It is therefore necessary to detect hotspots earlier in the layout design flow.

Park et al.[4] proposed a detection method for critical patterns (hotspots), using a design rule check (DRC) tool. The approach is a rule-based detection which generates lookup tables with line and space parameters. However, for more complex patterns, the number of layout pattern parameter required to enable detection increases. As a result, the speed advantage of the rule-based approach is reduced. Thus the simulation-based approach has occupied the mainstream and has been able to detect hotspots accurately.[5,6] Furthermore, software solutions running on customized hardware platforms have been developed so that aerial image simulation can be carried out quickly.[7] However, hotspots can be changed according to process conditions. The accuracy requirement for hotspot detection strongly depends on qualified optical and process models. Model generation corresponding to process variation represents a significant overhead in terms of validation, measurement and parameter calibration.

In this paper, we describe a novel detection algorithm for hotspots induced by lithographic uncertainty. Our approach utilizes a layout graph which reflects pattern-related CD variation. Our goal is to detect a set of potential lithographic hotspots within minutes without degrading accuracy (i.e., without missing any actual hotspots). The main steps are as follows.

- **Layout Graph Construction.** Given a layout $L$, the *layout graph* $G = (V, E_c \cup E_p)$ consists of nodes $V$, corner edges $E_c$ and proximity edges $E_p$. A face in the layout graph includes several close features and the edges between them. Edge weight can be calculated from a traditional 2-D model or a lookup table.
- **Graph Planarization.** For any two crossing edges, delete one edge with smaller weight.
- **Three-Level Hotspot Detection.** (1) Edge-level detection finds the hotspot caused by two close features or "$L$-shaped" features; (2) face-level detection finds the pattern-related hotspots which span several close features; and (3) merged-face-level detection finds hotspots with more complex patterns. That is, we construct the dual graph $G^D$ and sort the dual nodes according to their weights. We merge the sorted dual nodes (i.e., the faces in $G$) that share the same feature, in sequence.

The remainder of this paper is organized as follows. In Section 2, we describe the problem formulation, dual graph based hotspot detection algorithm and implementation details. Section 3 presents our evaluation flow and experimental results. We conclude in Section 4 with directions for ongoing research.

## 2. DUAL GRAPH BASED HOTSPOT DETECTION

VLSI manufacturing has achieved low defect densities which enable increasingly higher integration of a large number of devices on a single chip and continuous performance increase. However, in nanometer VLSI designs, such low defect density may not be expected for the latest technologies. At 90nm and below, systematic defects, such as CD variations, have major impacts on yield.

DEFINITION 2.1. *Processing* hotspots *are the locations in the design where the magnitude of edge displacement is exceptionally large. In other words, hotspots are printed features whose CD variations are greater than a given threshold value.*

### Problem Formulation

We formulate the fast hotspot detection problem as follows.

**Hotspot Detection Problem**
**Given:** Layout $L$, and threshold of CD variation which defines a hotspot.
**Detect:** Hotspots which may result in large CD variation.
**To Minimize:** The number of undetected hotspots and falsely detected hotspots.

The basic function for detection depends on process variations (i.e., defocus and exposure) and pattern parameters (i.e. width and space). To reduce the number of process conditions for hotspot validation, the effects of pattern complexity have been introduced. As shown in Figure 1, we evaluated patterns with three different complexities: (a) one wide metal line, (b) two wide metal lines, and (c) four wide metal lines. Figure 2 shows that different complexities lead to different CD variations. Also, the CD variation may be affected by different

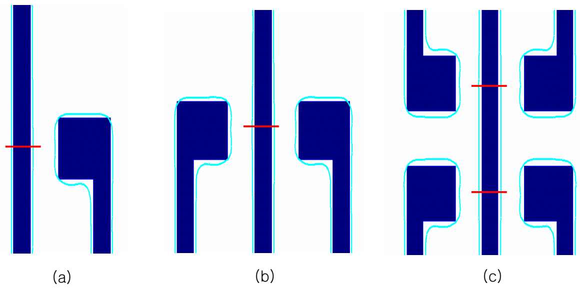


**Figure 1.** Test patterns to evaluate CD variation induced by pattern complexity: red lines show CD measurement location, and two CD values are averaged in the case of (c).

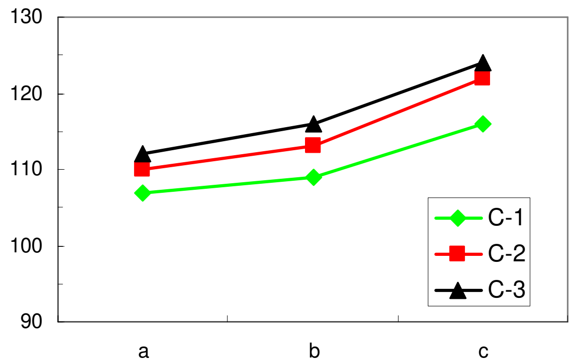


**Figure 2.** Evaluation of CD variation with various test patterns and process conditions. C-1: NA=0.85 and $\sigma$ =0.96/0.76. C-2: NA=0.75 and $\sigma$=0.75/0.55. C-3: NA=0.75 and $\sigma$ = 0.75/0.45.

process conditions. However, the patterns with more complex configuration, e.g., (c) in Figure 1, have larger CD variation than the patterns with simple complexity, e.g., (a) in Figure 1, at any process conditions. Therefore, our key observation is that the higher the pattern complexity, the higher the probability of a hotspot.

We now propose a new graph-based detection which is very fast and accurate. We group pattern-induced CD variations into two cases.

1. **Corner induced CD variation.** As shown in Figure 3(a), two orthogonal connected features form a corner which may lead to "corner rounding" CD variations.
2. **Proximity induced CD variation.** As shown in Figures 3(b) and (c), two close features may lead to "shortening" or "bridging" CD variations.

In lithography, a given hotspot may be the result of a single effect as shown in Figure 4(a), or the combination of several effects in an accumulative way as shown in Figure 4(b) and 4(c). The accumulative property of hotspots make detection and filtering very difficult. In our approach, we try to formulate this accumulative effect with an iterative merging process. As noted above, our proposed hotspot detection flow is as follows.

- **Layout Graph Construction.** Construct the layout graph $G = (V, E_c \cup E_p)$ for a given layout $L$, which consists of nodes $V$, corner edges $E_c$ and proximity edges $E_p$.
- **Graph Planarization:** For any two crossing edges, delete the one edge with the smaller weight.
- **Three-Level Hotspot Detection:** Perform edge-level (Figure 4(a)), face-level (Figure 4(b)) and merged-face-level (Figure 4(c)) detection to find hotspots with complex patterns.

We present each step in detail in the following subsections.

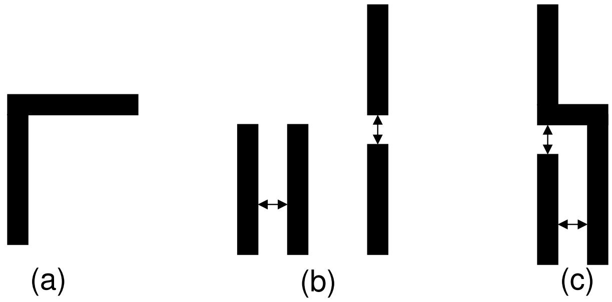


**Figure 3.** Effects lead to hotspots.

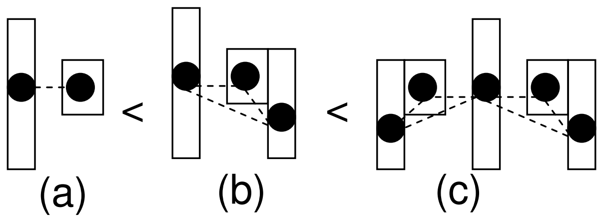


**Figure 4.** Effects lead to hotspots: (a) edge-level, (b) face-level and (c) merged face-level.

## 2.1. Layout Graph Construction

To fast detect the hotspots, the first step of our algorithm is to build a ***layout graph*** which reflects the pattern-related CD variation. Given a layout $L$, the ***layout graph*** $G = (V, E_c \cup E_p)$ consists of nodes $V$, corner edges $E_c$ and proximity edges $E_p$.

1. For every horizontal or vertical line, create a node $v \in V$ located in the middle of the line.
2. For two orthogonal connected lines, connect two corresponding nodes with a corner edge $c \in E_c$ whose weight is a constant.
3. Create a proximity edge $e \in E_p$ between two closely proximate lines having the same direction, where the weight of the edge is a function of the separation distance, overlapped projection length and the widths of the two lines.

Figure 6(a) shows an example of a layout graph for the layout. The layout graph has 7 nodes representing 7 lines, 4 corner edges (dashed edges) and 5 proximity edges (solid edges).

One crucial issue is the edge weighting scheme. We propose both closed-form formula based and lookup table based weighting schemes. In the closed-form formula scheme, we assume that the weights of corner edges are a constant $c$. As shown in Figure 5, the weights of the proximate edges are given by $\frac{w_1 \times w_2 \times l}{d}$, where $w_1$ and $w_2$ are the widths of the two features, $l$ is the length of the overlapped projection, $d$ is the distance between the two features. Intuitively, the approximate effect is more obvious for larger width features with smaller distance and large overlapped projection length. For lookup table based weighting scheme, the weights of the proximate edges are determined by feature widths, spacing, length of the overlapped projection. Although, lookup table based weighting scheme is more accurate, it also brings overhead in parameter tuning. In this paper, we use a closed-form formula based weighing scheme.

## 2.2. Graph Planarization

The next step is to convert the layout graph $G = (V, E_c \cup E_p)$ into its ***dual graph*** $G^D = (V^D, E_c^D \cup E_p^D)$.

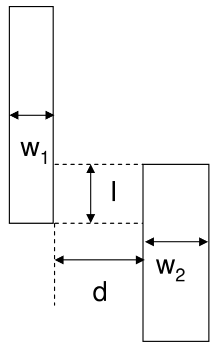


**Figure 5.** Weights of the proximate edges.

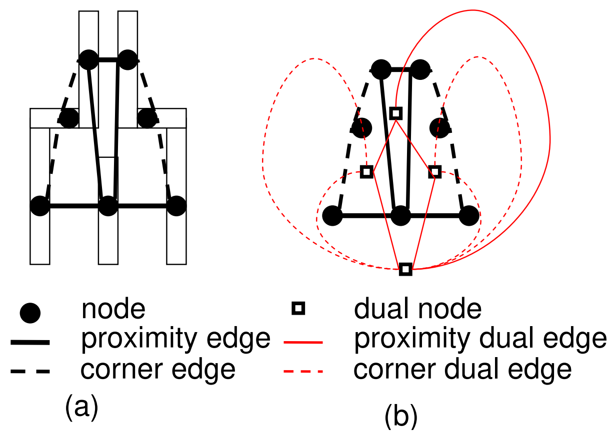


**Figure 6.** An example of a layout graph (a) and its dual graph (b).

DEFINITION 2.2. *The* **dual graph** $G^D$ *of the layout graph* $G$ *is constructed by representing every face* $f$ *of* $G$ *with a dual node* $n$ *whose weight is equal to the sum of the edge weights of* $f$*. An edge* $e$ *which belongs to faces* $f_1$ *and* $f_2$ *in* $G$ *is represented with a dual edge* $e^d = \{n_1, n_2\}$ *in* $G^D$ *with the same weight of* $e$*.*

One fact is that the dual graph $G^D$ exists if $G$ is a *planar* graph, i.e., there is no crossing edges. Therefore, for any two crossing edges in $G$, we need to delete the edge with smaller weight.

## 2.3. Three-Level Hotspot Detection

The intuition behind our hotspot detection method is that we view the hotspot as the result of the combination of several locally related "bad" patterns. With the assumption that the CD variation effect is cumulative, the effect can be reflected by the dual node weight, i.e., the total edge weight of one face. However, a hotspot may also relate to the lines of several faces. Therefore, we need to consider dual nodes merging to capture all possible hotspots. Our proposed iterative dual node merging heuristic is shown in Figure 7. The heuristic starts with the layout graph $G$ construction in Line 1. Then we perform edge-level detection in Lines 2-4. We delete one edge with smaller weight for any pair of crossing edges to make $G$ a planar graph and construct dual graph $G^D$ from $G$. In Lines 7-9, we perform face-level hotspot detection. Finally, we perform merged-face-level detection by sorting dual nodes according to weights and iteratively merging two connected dual nodes. A local wiring density based hotspots filter is used to reduce the number of falsely detected hotspots.

# 3. EXPERIMENTAL RESULTS

**Experimental Setup** We use one benchmark design in our experiments which is the *alu*128 core with 8.7K instances from *Artisan* libraries in a *90nm* technology using *Synopsys Design Compiler v2003.06-SP1*. The chip

| |
|---|
| **Input:** Layout $L$, $\epsilon_0$, $\epsilon_1$, $\epsilon_2$, $d_0$ |
| **Output:** A list of hotspots in $L$ |
| 1. Construct layout graph $G$ from $L$, $S \leftarrow \emptyset$<br>2. **For** all edges $e$ whose weight $> \epsilon_0$ (Edge-level detection)<br>3. $S \leftarrow S \cup \{e\}$<br>4. Delete $e$ from $G$<br>5. Perform graph planarization to delete one of the crossing edges<br>6. Construct dual graph $G^D$ from $G$<br>7. **For** all dual nodes $n$ whose weight $> \epsilon_1$ (Face-level detection)<br>8. $S \leftarrow S \cup \{n\}$<br>9. Delete $n$ from $G^D$<br>10. **While** ($\exists$ dual nodes can be merged) (Merged-face-level detection)<br>11. Sort all dual nodes according to weight<br>12. Merge the two connected dual nodes with the max combined weight<br>13. **If** (the weight of the combined node $n_c > \epsilon_2$)<br>14. $S \leftarrow S \cup \{n_c\}$<br>15. Delete $n_c$ from $G^D$<br>16. **For** all hotspots in $S$<br>17. **If** (local wiring density $< d_0$)<br>18. Remove it from $S$ |

**Figure 7.** Iterative Dual Node Merging Heuristic

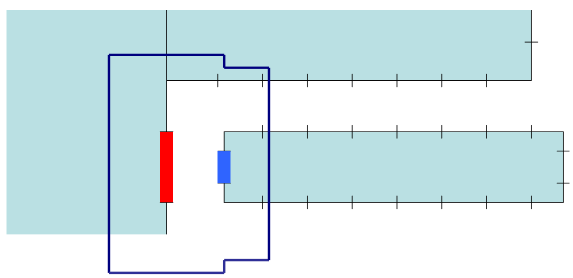

**Figure 8.** An example of a layer marking hotspot patterns.

size is 335$\mu$m $\times$ 285$\mu$m. The synthesized netlists are placed with row utilization of 70% using *Cadence First Encounter v3.3*. The netlists of the design has been obtained from *OpenCores*. The design is trial routed before running timing analysis. On the lithography side, *CalibreOPC* , *CalibreOPCsbar* and *CalibreORC* from *Mentor Graphics Calibre v9.3 5.11* are used for model-based OPC, SRAF generation and optical rule check (ORC), respectively. Simulation is performed with wavelength $\lambda = 193$, numerical aperture NA $= 0.75$, and annular aperture $\sigma = 0.75/0.50$. We used 0 $\mu$m DOF model and 0.35 aerial image threshold for OPC and then evaluated the OPC'ed layer with various simulation conditions as shown in Table 1.

**Experimental Results** We use the layout sizing technique to mark the hotspots and compare simulation-based detection with our dual graph based detection. Simulation-based detection makes fragments on pattern and decide whether each fragment is a hotspot based on magnitude of edge displacement. As a result, there may be several marked layer on a line-end and a corner of a pattern. On the other hand, our graph-based detection marks all patterns affecting hotspot and hence it is difficult to compare hotspots of simulation-based with graph-based methods. We size all layers marked as a hotspot after ORC by 0.5$\mu$m*. Then all sized layer is merged by one layer which will represent a hotspot and is used for comparison of two methods. Figure 8 shows an example of a hotspot marking layer which merged by two layers in the ORC result.

Figure 9 shows the results of hotspot detection; (a) hotspot pattern and (b) no hotspot pattern. The result

*The size amount is reasonable when we consider proximity range 0.6$\mu$m at 90nm technique.

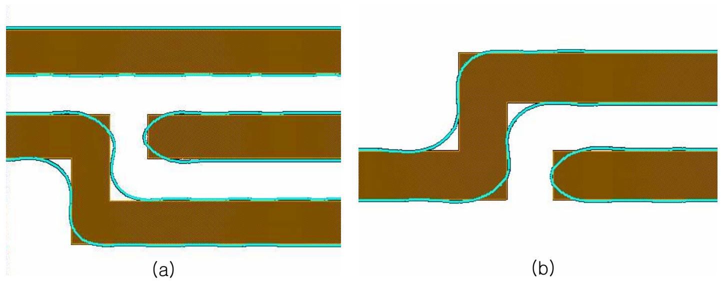


**Figure 9.** Results of hotspot detection: comparison of (a) hotspot patterns versus (b) no hotspot patterns

| Simulated Condition | # Hotspots ORC | # Hotspots Detected | # Hotspots Falsely Detected | Runtime (sec) ORC | Runtime (sec) Dual Graph |
|---|---|---|---|---|---|
| DOF($\mu$m)=0.1, ET=0.36 | 17 | 17 | 13 | 690 | 1.37 |
| DOF($\mu$m)=0.1, ET=0.37 | 21 | 21 | 22 | 690 | 1.52 |
| DOF($\mu$m)=0.1, ET=0.38 | 25 | 25 | 46 | 690 | 2.32 |
| DOF($\mu$m)=0.2, ET=0.38 | 152 | 152 | 1291 | 690 | 4.38 |
| Total | 215 | 215 | 1372 | 2760 | 9.59 |

**Table 1.** Comparison of hotspot detection efficiency of ORC and our proposed method.

of dual graph-based method is exactly matching to simulation-based method. Thus, dual graph-based method can detects a hotspot induced by pattern density which cannot be achieved by rule-based approach.

We implemented our proposed iterative dual node merging heuristic in C++. The values of $\epsilon_0$, $\epsilon_1$, $\epsilon_2$ and $d_0$ in Figure 7 are chosen according to different conditions. For example, the values are chosen as 0.7, 0.36, 0.67 and 0.17 for the first condition. We test our code under four different process conditions. The runtime of our method is more than 287X faster compared with commercial ORC tools. The results are summarized in Table 1. We can see that our proposed hotspot detection method achieves good accuracy (100% of hotspots are detected) with smaller false detection overhead.

## 4. CONCLUSIONS

With the continued shrinkage of minimum feature sizes, hotspots, i.e., printed image with large CD variation, present an important threat for manufacturing yield. Therefore, it becomes more and more important to fast and accurately detect the hotspots in a layout. In this paper, we first describe a novel, fast, dual graph based lithographic hotspot detection algorithm without significant accuracy degradation. The experimental results show that our method is promising: for the metal layer with 17 hotspots detected by commercial ORC tools, our method can detect all of them while the runtime improvement is more than 287X. In the future, we plan to include the fast hotspot detection engine in detailed router to improve the yield. We also want to test our algorithm on more large testcases.

## REFERENCES


1. K. Hashimoto, S. Usui, S. Nojima, S. Tanaka, E. Yamanaka and S. Inoue, "Hot Spot Management in Ultra-low k1 Lithography", 2005, *Proc. International Society for Optical Engineering (SPIE) Symposium on Microlithography*, vol. 6156, pp. 1207-1219.
2. S. Nojima, S. Mimotogi, M. Itoh, O. Ikenaga, S. Hasebe, K. Hashimoto, S. Inoue, M. Goto and I. Mori, "Flexible mask specifications", 2002, *Proc. SPIE 22nd Annual BACUS Symposium on Photomask Technology*, Vol.4889, pp. 187-196.
3. M. Cote and P. Hurat, "Standard Cell Printability Grading and Hot Spot Detection", 2005, *Intl. Symp. on Quality of Electronic Design*, pp. 264-269.

4. C.-H. Park, Y.-H. Kim, J.-S. Park, K.-D. Kim, M.-H. Yoo and J.-T. Kong, “A Systematic Approach to Correct Critical Patterns Induced by the Lithography Process at the Full-Chip Level”, *Proc. International Society for Optical Engineering (SPIE) Symposium on Microlithography*, 1999, vol. 3679, pp. 622-629.
5. E. Sahouria, Y. Granik, N. Cobb and O. Toublan, “Full-Chip Process Simulation for Silicon DRC”, *International Conference on Modeling and Simulation of Mircosystems (MSM)*, 2000, pp. 32-35.
6. M. Cote and P. Hurat, “Layout Printability Optimization Using a Silicon Simulation Methodology”, 2004, *Intl. Symp. on Quality of Electronic Design*, pp. 159-164.
7. Y. Cao, Y.-W. Lu, L. Chen and J. Ye, “Optimized Hardware and Software for Fast, Full Chip Simulation”, 2005, *Proc. International Society for Optical Engineering (SPIE) Symposium on Microlithography*, vol. 5754, pp. 407-414.