

# Lithography Hotspot Detection and Mitigation in Nanometer VLSI

Jhih-Rong Gao, Bei Yu, Duo Ding, and David Z. Pan  
Dept. of ECE, The University of Texas at Austin, Austin, TX 78712  
Email: dpan@mail.utexas.edu

**Abstract**—With continued feature size scaling, even state of the art semiconductor manufacturing processes will often run into layouts with poor printability and yield. Identifying lithography hotspots is important at both physical verification and early physical design stages. While detailed lithography simulations can be very accurate, they may be too computationally expensive for full-chip scale and physical design inner loops. Meanwhile, pattern matching and machine learning based hotspot detection methods can provide acceptable quality and yet fast turn-around-time for full-chip scale physical verification and design. In this paper, we discuss some key issues and recent results on lithography hotspot detection and mitigation in nanometer VLSI.

## I. INTRODUCTION

The continued shrinking of feature size has made IC manufacturing more and more prone to lithography hotspots, i.e., the layouts with poor printability and yield. To address this challenge, various resolution enhancement techniques (RETs), such as optical proximity correction (OPC), source mask co-optimization, and so on, have been proposed to improve pattern printability. However, lithography hotspots still remain a key challenging issue for IC designs in deep sub-wavelength processes (e.g., below 32nm), as the current lithography wavelength is still stuck at 193nm, which is much bigger than the feature size (e.g., 22nm).

In conventional design flow, lithography simulations [1], [2] have been used to identify problematic patterns. Lithography simulation is accurate but extremely computational intensive, especially for full-chip scale. If the lithography hotspot detection needs to be fed into some physical design stage to guide lithography-friendly layout optimization, it would be almost impossible to apply these detailed lithography simulations in the inner loop.

Recently, several hotspot detection approaches have been proposed, mainly based on pattern matching and machine learning techniques to avoid CPU-intensive lithography simulations. The challenges are how to extract critical information of these hotspot patterns and match them in the full-chip scale with high fidelity and low false alarm. There are also studies on integrating hotspot detection into physical design. This paper will discuss some key aspects of these lithography hotspot detection and mitigation methods.

The rest of the paper will be organized as follows. In Section II, we discuss several lithography hotspot

detection techniques. We then discuss hotspot mitigation in Section III, followed by the conclusion in Section IV.

## II. LITHOGRAPHY HOTSPOT DETECTION

### A. Layout Encoding Techniques

A hotspot is caused not only by a particular pattern, but also by the interaction with neighboring patterns inside the lithography influence region. One fundamental step for the hotspot detection in both pattern matching and machine learning methods is to represent layout patterns with certain format that can well describe the layout environment. Several layout encoding methods have been proposed to extract critical layout information from different aspects.

The concept of range pattern [3] is proposed to incorporate process-dependent specifications, and is enhanced in [4] to represent new types of hotspots. A range pattern is a two-dimensional layout of rectangles with additional specifications encoded by strings. Fig. 1 shows an example of range pattern “Staircase.” Each range pattern is associated with a scoring mechanism to reflect the problematic regions according to yield impact. The hotspot patterns are stored in a pre-defined library and the detection process performs string matching to find hotspots. This approach is accurate, but the construction of range patterns relies on a grid-based layout matrix, and may be time-consuming when the number of grids is large.

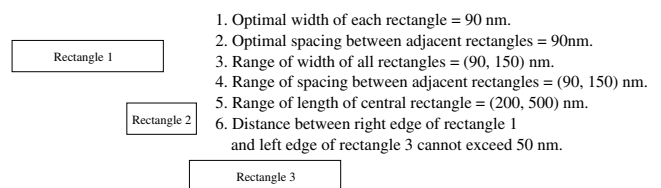


Fig. 1. An example of range pattern staircase [4].

The context characterization [5] cuts a layout pattern into fragments. For each fragment  $F$ , an effective radius  $r$  is defined to cover the neighboring fragments which need to be considered in the context characterization of  $F$  as shown in Fig. 2. A complete representation of  $F$  includes the geometric characteristic of fragments inside  $r$ , including pattern shapes, the distance between patterns, corner information (convex or concave), and so on.

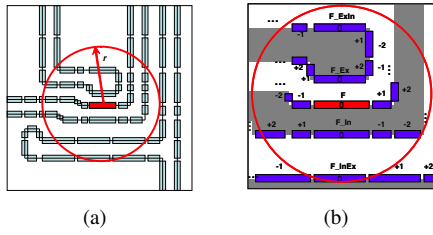


Fig. 2. Fragmentation based hotspot signature extraction [5].

The density-based pattern encoding is introduced in [6], where a layout pattern is represented as a vector of layout density values of its surrounding area. Given a layout clip with predefined grids, the method calculates the layout density covered in each grid. An ordered list of density values then forms the final vector that represents the corresponding layout pattern. Fig. 3 illustrates the process of pattern encoding. The goal of this representation is not to identify the geometrical features that may degrade the printability of a pattern. Instead, it aims at providing a compact representation of layout patterns to enable measurement of pattern similarities.

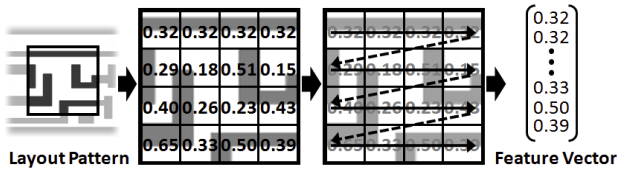


Fig. 3. Density-based pattern representation. [7].

### B. Pattern Matching Based Hotspot Detection

In pattern matching based approaches, a set of known hotspots are pre-characterized and stored in a database. The hotspot detection process involves matching the tested layout patterns with the hotspots in the database. This method is fast and accurate at detecting known patterns, but it lacks the capability of predicting unseen data.

A layout graph is proposed in [8] to reflect pattern-related CD variation. The resulted graph can be used to find hotspots including closed features, L-shaped features and complex patterns. Yu et al. proposed a DRC-based hotspot detection [9] by extracting critical topological features and modeling them as design rules. Therefore, hotspot detection can be viewed as a rule checking process through a DRC engine.

Recently a fuzzy matching model was proposed in [7] which can dynamically tune appropriate fuzzy regions around known hotspots in multi-dimensional space. Fig. 4 shows an example with known layout patterns of hotspots and non-hotspots in a 2-dimensional space. A machine learning method would divide the space into two regions of hotspots and non-hotspots as shown in Fig. 4(a), while a conventional pattern matching approach would construct an individual pattern to match each known hotspot as

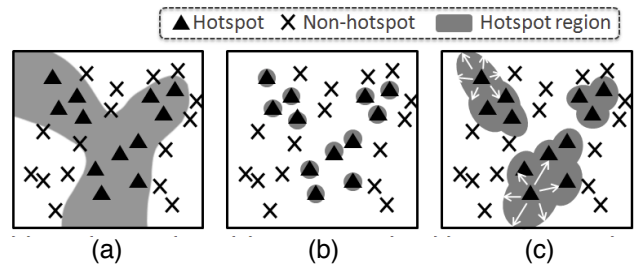


Fig. 4. A 2D-space example of hotspot region decision. (a) Machine learning; (b) Pattern matching; (c) Fuzzy matching model. [7]

shown in (b). The fuzzy matching model in Fig. 4(c) includes groups of hotspots, where the fuzzy region of each group will iteratively grows to provide better detection accuracy.

### C. Machine Learning Based Hotspot Detection

Machine learning techniques construct a regression model based on a set of training data. This method can naturally identify previous unknown hotspots. However, it may generate false alarms, which are not real hotspots. How to improve the detecting accuracy is the main challenge when adopting machine learning techniques.

Many recent approaches utilize support vector machine (SVM) and artificial neural network (ANN) techniques to construct the hotspot detection kernel. In [10], a 2-D distance transform and histogram extraction is performed on pixel-based layout images, which are then used to construct the SVM-based hotspot detection. In [11][12], SVM is employed through extraction and classification of layout density-based metrics. A neural network judgment based detection flow is proposed in [13], where 2-D hotspot patterns are directly used to train an ANN kernel. A hybrid method [14] that adopts both SVM and ANN is presented to further improve the performance.

### D. Hybrid Machine Learning and Patterning Matching

Since both patterning matching and machine learning have pros and cons, it will be is to apply both machine learning models and pattern matching models. In [15], data samples are fed to a pattern matcher first, then machine learning classifiers are used to examine the non-hotspots left by the pattern matcher. Motivated by the fact that different hotspot classifiers have different objectives and strengths, [15] further proposed a unified meta-classifier that enables several classifiers to work together. The meta-classifier is composed of multiple *base classifiers* and *weighting functions*. Each base classifier is an individual hotspot classifier that is optimized under certain performance metric, such as detection accuracy, false-alarms, adaptivity to new unknown designs, etc. Weighting functions are used to control the overall combination of base classifiers, which needs to be optimized to achieve better accuracy and less noise. The construction flow of meta-classification is illustrated in Fig. 5. For each layout

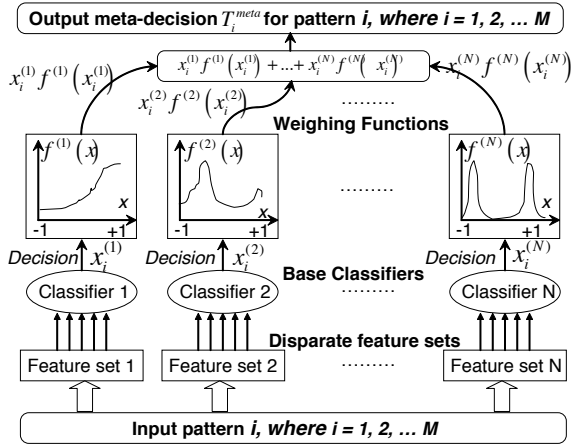


Fig. 5. Meta-classifier construction via a combination of disparate base classifiers [15].

pattern, certain hotspot features are extracted and then fed into each base classifier, which calculates the prediction decision and generates a weight based on the weighing functions. The final meta-decision is based on the weighed sum of base classifiers.

### E. Clustering in Hotspot Detection

Since there are many design rules to guide and restrict physical design, in general the number of non-hotspot patterns greatly outnumber that of real hotspot patterns [16]. The imbalance between hotspot and non-hotspot data is called *imbalanced populations*, and it may cause performance degradation for some hotspot detection approaches. In addition, how to avoid redundant data and reduce the time for detection is an important issue.

In [17][18], the extracted hotspots are classified into clusters by data mining methods. An incremental clustering algorithm [18] is used to group the hotspot snippets into a small number of clusters containing geometrically similar hotspots. Given a set of hotspot patterns, a distance metric is defined to calculate the similarity of different hotspot patterns. All patterns are assigned to a cluster where the distance between the cluster center and the pattern is less than the cluster radius. Once the clusters are determined, it is analyzed to produce a description of this cluster, including a representative hotspot snippet and a radius that characterizes the cluster tightness. The representative hotspot in each cluster is then identified and stored in a hotspot library for future hotspot detection. The clustering method is further extended in [19] using an improved tangent space based distance metric to achieve better accuracy.

## III. LITHOGRAPHY HOTSPOT MITIGATION

### A. Lithography Friendly Placement

For CMOS feature size significantly smaller than the lithography wavelength (193nm), the printability of a standard cell could be well affected by its neighboring

cells. To minimize the interference between adjacent cells, standard cell design itself and the standard placement methodology need to be co-designed to avoid newly generated lithography hotspots between these cells after placement. Dummy poly or metal lines may be inserted to create regular neighborhood patterns between neighboring cells.

As the feature size and pitch become even smaller, double or multiple patterning is needed to extend the 193nm lithography. A grand challenge for standard cell and placement co-optimization is that there may be coloring conflict between adjacent cells. In fact, it is still an open question whether the cells shall be pre-colored (i.e., during the standard cell layout stage) or post-colored (i.e., flat after standard cell placement of the entire chip). Liebmann et al. in [20] proposed some guidelines to enable double patterning friendly standard cell design and placement. Other placement studies toward double patterning are present [21][22]. Recently, Yu et al. [23] proposed a systematic framework to seamlessly integrate triple patterning constraints for standard cell and placement stages.

### B. Lithography Friendly Routing

Lithography hotspot mitigation can be performed at the post-routing stage, e.g., [24]. In [26][27], design rule checker is integrated with the routing engine at the post-routing stage to identify and correct hotspots. First, a set of problematic pattern topologies are transformed into DRC rules. Once placement and routing is done, those pre-built rules are applied to the layout to identify violated pattern region based on a pattern matching based rule checker. These approaches can provide a fast feedback to the router on the hotspot location, and then the router can apply rip-up and reroute to fix the hotspots.

However, fixing hotspots at this post-routing stage has limited flexibility as only limited rip-up and reroute may be performed. With efficient hotspot predictions, it will be interesting to integrate lithography hotspot detection together with routing.

One challenge of lithography-aware routing is that hotspots are difficult to be detected before a real routing

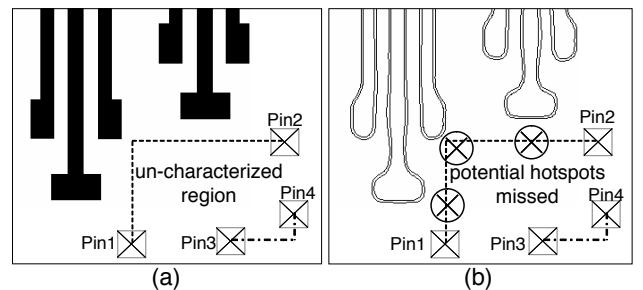


Fig. 6. The hotspot detection challenge in the detailed routing stage [25].

path is obtained. Fig. 6(a) shows a layout region with metal blockages and unrouted pins Pin1-Pin4. Because some nets are not yet routed, there is an un-characterized region where no hotspots would be identified by general hotspot detection methods. Consequently, potential hotspots may be caused by route Pin1-Pin2 as shown in Fig. 6(b). Ding et al. [25] proposed a lithography-friendly detailed routing based on a pre-built hotspot prediction kernel and a routing path prediction kernel. First, the hotspot detection kernel is trained to evaluate the pattern printability based on a set of post-RET data. To overcome the issue of un-characterized regions, the routing path prediction kernel is established using the following steps: (1) explore the possible routing solutions given the available routing resources; (2) perform accurate lithography simulation for the possible layout results; (3) identify preferable routes according to results of hotspots and routing congestion. Because the data that need to be processed for building the routing path prediction kernel is huge, a neural network classifier is constructed to guide the routing engine. The experimental results showed very promising results with this approach.

#### IV. CONCLUSION

Lithography hotspots have a great impact on the manufacturing yield. Identifying these problematic layouts during physical design has become a critical problem. Since full chip lithography simulation is computational expensive, pattern matching and machine learning based hotspot detection are very useful for full-chip scale physical verification/screening and layout optimization. In this paper, we discuss some key issues of the lithography hotspot detection problem, including critical feature extraction, patterning matching, and machine learning based methods. We also discuss hotspot mitigation, e.g., at placement and routing stages. It shall be noted that the accuracy and robustness of hotspot detection, and the integration with physical design still have a lot of room for improvement and future research, in particular for multiple patterning and other emerging lithography technologies.

#### ACKNOWLEDGMENT

This work is supported in part by NSF grants CCF-0644316 and CCF-1218906, SRC task 2414.001, NSFC grant 61128010, IBM, and Oracle.

#### REFERENCES

- [1] J. Kim and M. Fan, "Hotspot detection on Post-OPC layout using full chip simulation based verification tool: A case study with aerial image simulation," *Proc. SPIE 5256, 23rd Annual BACUS Symposium on Photomask Technology*, pp. 919–925, 2003.
- [2] E. Roseboom *et al.*, "Automated full-chip hotspot detection and removal flow for interconnect layers of cell-based designs," *Proc. SPIE 6521, Design for Manufacturability through Design-Process Integration*, p. 65210C, 2007.
- [3] H. Yao, S. Sinha, C. Chiang, X. Hong, and Y. Cai, "Efficient process-hotspot detection using range pattern matching," in *ICCAD*, 2006, pp. 625–632.
- [4] J. Xu, S. Sinha, and C. C. Chiang, "Accurate detection for process-hotspots with vias and incomplete specification," in *ICCAD*, 2007, pp. 839–846.
- [5] D. Ding, A. J. Torres, F. G. Pikus, and D. Z. Pan, "High performance lithographic hotspot detection using hierarchically refined machine learning," in *ASPDAC*, 2011, pp. 775–780.
- [6] J.-Y. Wu, F. G. Pikus, A. Torres, and M. Marek-Sadowska, "Detecting context sensitive hotspots in standard cell libraries," *Proc. SPIE 7275*, p. 727515, 2009.
- [7] S.-Y. Lin, J.-Y. Chen, J.-C. Li, W. Yu Wen, and S.-C. Chang, "A novel fuzzy matching model for lithography hotspot detection," in *DAC*, 2013, pp. 68:1–68:6.
- [8] A. B. Kahng, C.-H. Park, and X. Xu, "Fast dual graph based hotspot detection," *Proc. SPIE 6349, Photomask Technology*, pp. 63 490H–63 490H–8, 2006.
- [9] Y.-T. Yu, Y.-C. Chan, S. Sinha, I. H.-R. Jiang, and C. Chiang, "Accurate process-hotspot detection using critical design rule extraction," in *DAC*, 2012, pp. 1167–1172.
- [10] D. G. Drmanac, F. Liu, and L.-C. Wang, "Predicting variability in nanoscale lithography processes," in *DAC*, 2009, pp. 545–550.
- [11] J.-Y. Wu, F. G. Pikus, A. Torres, and M. Marek-Sadowska, "Rapid layout pattern classification," in *ASPDAC*, 2011, pp. 781–786.
- [12] Y.-T. Yu, G.-H. Lin, I. H.-R. Jiang, and C. Chiang, "Machine-learning-based hotspot detection using topological classification and critical feature extraction," in *DAC*, 2013, pp. 67:1–67:6.
- [13] D. Ding, X. Wu, J. Ghosh, and D. Z. Pan, "Machine learning based lithographic hotspot detection with critical-feature extraction and classification," in *ICICDT*, 2009, pp. 219–222.
- [14] D. Ding, J. A. Torres, and D. Z. Pan, "High performance lithography hotspot detection with successively refined pattern identifications and machine learning," *IEEE Transactions on TCAD*, pp. 1621–1634, 2011.
- [15] D. Ding, B. Yu, J. Ghosh, and D. Z. Pan, "EPIC: Efficient prediction of IC manufacturing hotspots with a unified meta-classification formulation," in *ASPDAC*, 2012, pp. 263–270.
- [16] J. A. Torres, "ICCAD-2012 CAD contest in fuzzy pattern matching for physical verification and benchmark suite," in *ICCAD*, 2012.
- [17] N. Ma *et al.*, "Automatic hotspot classification using pattern-based clustering," *Proc. SPIE 6925, Design for Manufacturability through Design-Process Integration II*, pp. 692 505–692 505–10, 2008.
- [18] J. Ghan *et al.*, "Clustering and pattern matching for an automatic hotspot classification and detection system," *Proc. SPIE 7275, Design for Manufacturability through Design-Process Integration III*, pp. 727 516–727 516–11, 2009.
- [19] J. Guo, F. Yang, S. Sinha, C. Chiang, and X. Zeng, "Improved tangent space based distance metric for accurate lithographic hotspot classification," in *DAC*, 2012, pp. 1173–1178.
- [20] L. Liebmann, D. Pietromonaco, and M. Graf, "Decomposition-aware standard cell design flows to enable double-patterning technology," in *Proc. of SPIE*, vol. 7974, 2011.
- [21] M. Gupta, K. Jeong, and A. B. Kahng, "Timing yield-aware color reassignment and detailed placement perturbation for double patterning lithography," in *ICCAD*, 2009, pp. 607–614.
- [22] J.-R. Gao, B. Yu, R. Huang, and D. Z. Pan, "Self-aligned double patterning friendly configuration for standard cell library considering placement," in *Proc. of SPIE*, 2013.
- [23] B. Yu, X. Xu, J.-R. Gao, and D. Z. Pan, "Methodology for standard cell compliance and detailed placement for triple patterning lithography," in *ICCAD*, 2013.
- [24] J. Mitra, P. Yu, and D. Z. Pan, "RADAR: RET-aware detailed routing using fast lithography simulations," in *DAC*, 2005, pp. 369–372.
- [25] D. Ding, J.-R. Gao, K. Yuan, and D. Z. Pan, "AENEID: a generic lithography-friendly detailed router based on post-RET data learning and hotspot detection," in *DAC*, Jun. 2011, pp. 795–800.
- [26] J. Yang *et al.*, "DRCPlus in a router: Automatic elimination of lithography hotspots using 2d pattern detection and correction," *Proc. SPIE 7641, Design for Manufacturability through Design-Process Integration IV*, p. 76410Q, 2010.
- [27] J.-R. Gao *et al.*, "Self-aligned double patterning compliant routing with in-design physical verification flow," *Proc. SPIE 8684, Design for Manufacturability through Design-Process Integration VII*, p. 868408, 2012.