2017 IEEE 35th International Conference on Computer Design

Post-Layout Perturbation towards Stitch Friendly Layout for Multiple E-Beam Lithography

Sudipta Paul*†, Pritha Banerjee† and Susmita Sur-Kolay*
*Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata, India
Email: sudiptapaul2@gmail.com, ssk@isical.ac.in
†Dept. of Computer Science and Engineering, University of Calcutta, India
Email: banerjee.pritha74@gmail.com

***Abstract*—Soaring distortions for immersion lithography with 193nm wavelength has necessiated the Next Generation Lithography (NGLs) such as Electron Beam Lithography (EBL). While single e-beam lithography suffers from very low throughput, Multiple Electron Beam Lithography (MEBL) improves it by writing with multiple e-beams in parallel, each dedicated to a disjoint region called a vertical stripe. However, layout patterns, in particular routing segments, vias and short polygons crossing over stripe boundary (*stitch line*) causes severe pattern distortion leading to malfunctioning of the chip. We minimize these *stitch unfriendly* patterns at post-layout stage based on perturbation of wire segments by formulating it as a maximum matching problem. Experimental results comprise two variants of perturbations of wire segments. Each variant shows significant minimization of stitch unfriendly patterns, thereby making an already optimized design more MEBL friendly without increasing the wirelength.**

I. Introduction

Advances in IC fabrication technology has scaled down the geometry of process nodes to well below 20 nanometer in the recent past [1]. Next Generation Lithography (NGLs) are slowly becoming an alternative for modern chip manufacturing replacing the earlier resolution enhancement techniques such as optical proximity correction, multiple patterning [2],[3] for immersion lithography. Electron Beam Lithography (EBL) [4] is a mask-less direct write NGL for IC fabrication which can print patterns of sub-$10nm$ resolution. However, it has the limitation of low throughput.

The authors of [5] report Multiple E-Beam Lithography (MEBL) towards increasing the throughput of EBL, where multiple beams are shot to print patterns simultaneously within predefined mutually exclusive collectively exhaustive vertical stripes on the die. Fig. 1(a) shows the vertical stripes, the two boundaries on either side are referred as *stitch lines* in [6] [7]. The patterns intersecting a stripe boundary, called *stitch unfriendly* patterns and shown in Fig. 1(b), are written by two e-beam guns which cause Critical Dimension (CD) distortions as shown in Fig. 1(c). In order to make MEBL a feasible alternative, the key issue is to obtain a MEBL friendly layout efficiently, devoid of stitch unfriendly patterns.

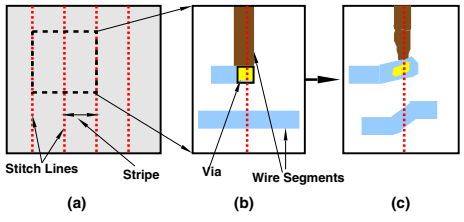

Fig. 1. (a) Vertical stripes on die (b) Patterns intersecting the stripe boundary, (c) Distortion of pattern after fabrication.

Liu et al. proposed a stitch aware routing framework in [6] to minimize the occurrences of *stitch unfriendly* patterns. However, this work performs routing from scratch for all existing designs, leading to an increase in turnaround time for MEBL based IC fabrication. On the contrary, existing highly optimized layouts may be processed efficiently to migrate the existing designs to MEBL. We propose a graph matching formulation to minimize the number of *stitch unfriendly patterns* at the post-layout stage based on perturbation of wire segments. This is the first attempt towards making an existing layout MEBL friendly at the post-layout stage.

The rest of this paper has Section II describing the problem statement. Section III and Section IV discuss the proposed method and the experimental results on Faraday benchmark circuits. The concluding remarks appear in Section V.

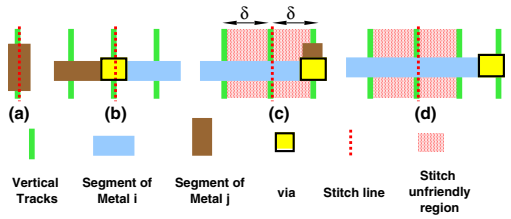

Fig. 2. (a) Routing violation, (b) Via violation, (c) Short polygon, and (d) a horizontal wire segment crossing a stitch line — not stitch unfriendly

II. Preliminaries

Three types of *stitch unfriendly* routing patterns causing most severe CD distortions, are identified in [6] as: (a) routing violation, (b) via violation, and (c) short polygon.

Fig. 2(a) illustrates a *routing violation*, where a vertical wire segment is routed along the stitch line to produce severe distortion of the vertical segment on the wafer. Fig. 2(b) shows a *via violation* which occurs when a via, connecting two wire segments in different metal layers, overlaps a stitch line. It introduces overlay error causing discontinuity among

1063-6404/17 $31.00 © 2017 IEEE
DOI 10.1109/ICCD.2017.71

segments. Fig. 2(c) shows a short polygon where a horizontal segment is critical. A span δ on each side of the stitch line is specified by the foundry as *stitch unfriendly region* [6]. A short polygon is said to be generated when the following condition holds: an end of the horizontal segment crossing a stitch line lies inside the corresponding stitch unfriendly region and that end of the segment has a via. Above condition does not apply to the horizontal segment in Fig. 2(d), so it is not considered as a short polygon and thus not stitch unfriendly.

Here, the routing tracks and wire segments within the stitch unfriendly region are referred as *stitch unfriendly*, and the ones outside these regions are termed as *regular* in this paper.

A. Problem Statement

The motivation for this work is to perturb or reassign a set of already *routed* wire segments in the layout to a set of *regular* tracks in order to minimize the three different types of violations without sacrificing the total wirelength.

A stitch unfriendly vertical segment with movable via (non terminal) at the end can be moved to a regular track if it is not occupied by any other segment. Perturbation of such vertical segments can remove routing violations, via violations and short polygons generated due to the via at the junction of the vertical and a horizontal segment. The above observations lead to the following problem statement:

Given a layout and the positions of stitch lines at uniform intervals aligned to routing tracks, reassign a set of vertical segments to regular tracks such that the total number of routing violations, via violations and short polygons are minimized without increasing the total wirelength.

III. Proposed Method

The overview of the method is illustrated in Algorithm 1.

Algorithm 1: Maximum Matching Based Wire Perturbation

Input : A layout along with the stitch lines positioned at fixed intervals.
Output: A layout with minimized stitch unfriendly patterns.
/* $NumSeg$: Total number of vertical segments */
/* $NumSL$: Total number of Stitch Lines */
1 **for** $s \leftarrow 1$ ***to*** $NumSeg$ **do**
2 Compute $perturbation\_range$ of segment s
3 **for** $SL \leftarrow 1$ ***to*** $NumSL$ **do**
4 Find $stitch\ unfriendly$ vertical segments.
5 Create a $Band$ corresponding to SL.
6 Construct a $Bipartite\ Graph$ corresponding to the candidate segments to be reassigned in $Band$ for SL .
7 Perform $Maximal\ Bipartite\ Matching$ to obtain the new track assignment for each candidate segment in the band.
8 $Reassign$ segments according to the bipartite matching.
9 **return**

A. Perturbation Range

In this work, perturbation essentially means shifting of a vertical wire segment along the x-axis to a *regular* routing track. The *perturbation range* of a vertical segment s with movable vias (steiner points) on both ends of the segment is the x-span along which s can be perturbed, preserving the connectivity without increasing the wirelength. Fig. 3(a)-(d) shows the perturbation range for four primitive patterns as described in [8].

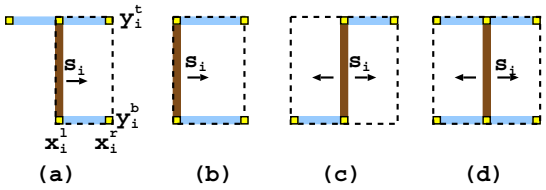

Fig. 3. Four types of routing patterns and their respective perturbation range $r_i = (x_i^l, x_i^r)$ which preserves wirelength and connectivity — arrow indicates the direction of perturbation.

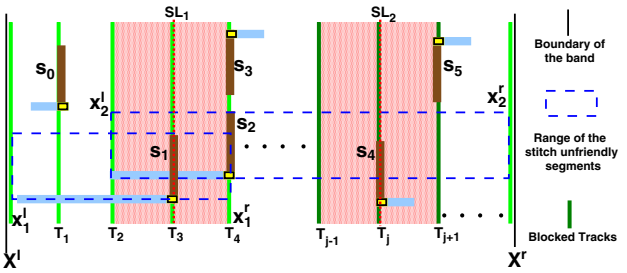

Fig. 4. A *band* = (X^l, X^r) corresponding to stitch line SL_1 — among the segments $s_0, s_1, \ldots, s_5$ in this band, only s_1 and s_2 are stitch unfriendly.

B. Creation of Band

A *band* is defined for each stitch line, as the region between the leftmost x-coordinate and the rightmost x-coordinate of the perturbation ranges of the vertical segments in its stitch unfriendly region. In other words, for a stitch line, the perturbation range for each of the m stitch unfriendly segments is computed. Let the ranges be $R = \{r_1, r_2, \cdots r_m\}$, $r_i = (x_i^l, x_i^r)$. Let X^l be the minimum of the x_i^l's, and X^r be the maximum of the x_i^r's. Then a band corresponding to the stitch line is denoted by $B = (X^l, X^r)$. Fig. 4 illustrates the band corresponding to the stitch line SL_1 placed on vertical routing track T_3. Here, the stitch unfriendly region is defined to be one routing track on either side of this stitch line. Thus, T_2 and T_4 are in stitch unfriendly region, and the stitch unfriendly segments are s_1 and s_2.

The tracks T_{j-1} to T_{j+1} corresponding to stitch line SL_2 on T_j are kept blocked for perturbation of segments in the band corresponding to SL_1 as these are in the stitch unfriendly region of SL_2. Thus, segments cannot be reassigned to T_{j-1} to T_{j+1} and cannot produce any new violations with SL_2. Furthermore, in Fig. 4 violation caused by s_4 are not to be processed in B, because this violation corresponds to SL_2.

C. Maximum matching for MEBL friendly layout

We propose a perturbation method by processing the bands in a left ot right order and obtaining a maximum matching in a bipartite graph $G_B(U_B \cup V_B, E_B)$ defined (see below) for each band B corresponding to each stitch line. The maximum bipartite matching of the graph produces the new assignments of segments that minimizes the three types of violations defined in Section II.

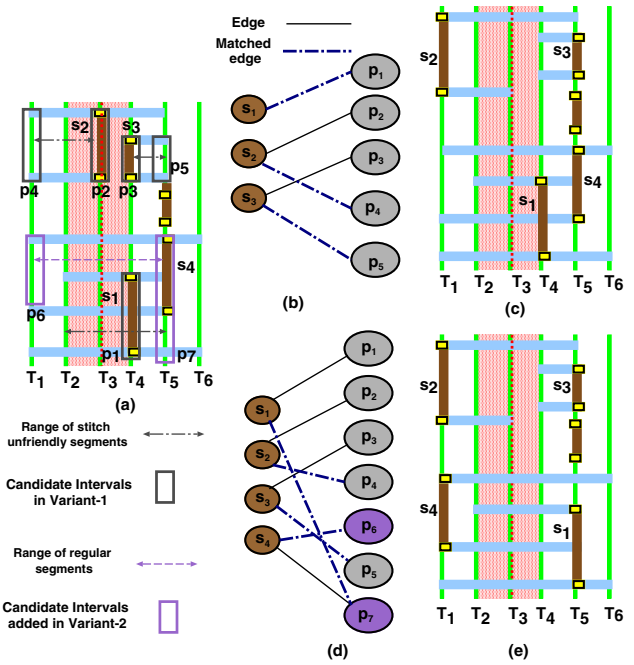

Fig. 5. A portion of a layout within a band: (a) Initial: s_1, s_2, s_3 and s_4 with ranges T_2-T_5, T_1-T_3, T_4-T_5 and T_1-T_5 respectively. $s_1, s_2, s_3 \in S_{U_B}$ are stitch unfriendly and $s_4 \in S_{R_B}$ is a regular segment. (b) Bipartite graph for $Variant$-1: U_B has three vertices only, and V_B has five for the two free intervals $p_4, p_5 \in P_F$ and the three intervals $p_1, p_2, p_3 \in P_C$; (c) Final layout for $Variant$-1 after Bipartite Matching: all violations in the band except for s_1 are removed. (d) Bipartite graph for $Variant$-2: $s_4 \in S_{R_B}$ with overlapping perturbation range with $s_1 \in S_{U_B}$ is added to U_B of $Variant$-1, and V_B corresponds to the seven intervals $p_1 \cdots p_7 \in P_R$ where p_6 and p_7 are due to s_4; (e) Final layout for $Variant$-2: All the violations are removed.

Two variants, namely $Variant$-1 and $Variant$-2 of the bipartite graph G_B for a band B are constructed. The following notations are used to define G:

- $S_{U_B} = \{s_i \mid 1 \leq i \leq m\}$: s_i is a stitch unfriendly segment in band B with non zero perturbation range r_i;
- $p = (y^t, y^b)$: y-span of an interval along the vertical track where p may be free or occupied by a segment

1) Variant-1: In this case, only the vertical wire segments in a band which cause any of the three types of violations corresponding to a stitch line, are perturbed. The rest of the vertical segments within the band are considered as *fixed*, and are not perturbed. Let

- $P_C = \{p_j \mid 1 \leq j \leq m\}$: p_j is the interval occupied by $s \in S_{U_B}$ to be perturbed.
- $P_F = \{p_j \mid 1 \leq j \leq n\}$: p_j is a free interval on a regular track where a segment may be assigned.

So, we define U_B, V_B and E_B as follows:

- $U_B = \{u_i \mid 1 \leq i \leq m\}$: u_i corresponds to $s_i \in S_{U_B}$.
- $V_B = \{v_j \mid 1 \leq j \leq (n+m)\}$: v_j corresponds to $p_j \in \{P_F \cup P_C\}$.
- $E_B = \{e_{ij}\}$ where edge $e_{ij} = (u_i, v_j)$ exists if either condition (i) below holds, or all of (ii), (iii) and (iv) hold.
  - (i) $s_i \in S_{U_B}$ is originally assigned to the occupied interval $p_j \in P_C$ in the given layout. This ensures that if a stitch unfriendly segment cannot be assigned to a regular track, it retains its original assignment;
  - (ii) the y-span of the segment $s_i \in S_{U_B}$ fits into the free interval $p_j \in P_F$ on a different track if s_i is moved horizontally;
  - (iii) free interval $p_j \in P_F$ is within the perturbation range r_i of stitch unfriendly segment $s_i \in S_{U_B}$;
  - (iv) an assignment of $s_i \in S_{U_B}$ to a free interval $p_j \in P_F$ does not induce any new violation.

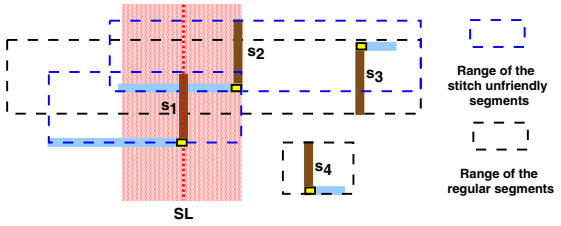

Fig. 6. In $Variant$-2, candidate segments in the band are s_1, s_2, s_3: perturbation ranges of $s_1, s_2 \in S_{U_B}$ overlap with that of regular segment s_3 but not of regular segment s_4.

An example instance in the Fig. 5(a) represents a part of a band. Fig. 5(b)-(c) shows the maximum matching formulation for the layout and its solution for $Variant$-1. Note that the tracks in the perturbation range of a stitch unfriendly segment may be occupied by many fixed segments. Thus, all stitch unfriendly segments in a band may not find a violation free new assignment for MEBL. This necessitates the formulation of our next variant for further reduction of violations.

2) Variant-2: Here, the set of stitch unfriendly segments as well as the set of regular segments that reside in the perturbation range of at least one stitch unfriendly segment are the candidates for perturbations. In Fig. 6, s_1 and s_2 have violations and their perturbation ranges overlap with s_3 but not with s_4. Thus s_1, s_2, s_3 are included in U_B but not s_4. Let

- $S_{R_B} = \{s_k \mid 1 \leq k \leq d\}$: s_k is a regular segment and has overlapping perturbation range with at least one $s \in S_{U_B}$ in band B;
- $p^u = (\max_{s_i \in S_{U_B} \cup S_{R_B}}\{y^t_{s_i}\}, \min_{s_j \in S_{U_B} \cup S_{R_B}}\{y^b_{s_j}\})$: the maximum y-span on a vertical track in which segments $s_i, s_j \in S_{U_B} \cup S_{R_B}$ are to be perturbed and all have mutually overlapping perturbation ranges;
- $P_R = \{p^u_j \mid 1 \leq j \leq q\}$: p^u_j is an interval on a regular track where a segment can be assigned.

Thus, U_B, V_B and E_B for $Variant$-2 are:

- $U_B = \{u_i \mid 1 \leq i \leq (m+d)\}$: u_i corresponds to a segment $s_i \in S_{U_B} \cup S_{R_B}$;
- $V_B = \{v_j \mid 1 \leq j \leq q\}$: v_j corresponds to an interval $p_j \in P_R$;
- $E_B = \{e_{ij}\}$ where edge $e_{ij} = (u_i, v_j)$ if the conditions (v) and (vi) stated below hold together.
  - (v) y-span of segment $s_i \in S_{U_B} \cup S_{R_B}$ fits into interval $p^u_j \in P_R$ on a track if s_i is perturbed horizontally;
  - (vi) an assignment of $s_i \in S_{U_B} \cup S_{R_B}$ to an interval $p^u_j \in P_R$ does not induce any new violation.

TABLE I

RESULTS FOR $Variant$-1 AND $Variant$-2: POST-LAYOUT PERTURBATION OF VERTICAL WIRE SEGMENTS FOR MEBL FRIENDLY LAYOUT

Circuits	Area (μm^2)	#T_Nets	Layer #	#Nets	Initial			$Variant$-1			$Variant$-2			Time (Sec.)	
					#RV	#VV	#SP	#RV	#VV	#SP	#RV	#VV	#SP	$Variant$-1	$Variant$-2
DMA	408 X 408	13256	2	9756	2552	2734	1583	2251	2393	1415	953	1038	189	34	720
			4	3233	578	743	411	496	642	344	180	238	62		
			6	619	67	109	55	49	82	34	28	47	12		
DSP1	706 X 706	28447	2	22622	5180	4912	3293	4454	4221	2874	3484	2989	454	96	1100
			4	6449	1145	1515	864	933	1252	689	834	720	51		
			6	1157	140	234	132	117	198	110	96	80	3		
DSP2	643 X 643	28431	2	22502	5199	5000	3116	4477	4288	2696	3551	3867	553	84	1025
			4	6518	1172	1550	881	971	1295	748	893	800	71		
			6	837	100	162	96	69	119	77	72	68	9		
RISC1	1004 X 1004	34034	2	27914	8108	7781	5234	7068	6809	4588	6510	5610	732	134	1810
			4	8549	1879	2429	1614	1576	2031	1359	1393	1644	88		
			6	1572	165	272	242	130	222	198	110	221	4		
RISC2	960 X 960	34034	2	27696	8018	7752	5237	7014	6785	4640	6600	5891	950	112	1734
			4	8450	1991	2497	1595	1717	2169	1364	1513	1530	112		
			6	1781	198	324	236	162	267	184	148	265	10		
GEOMEAN	-	-	-	-	940	1201	753	777	1003	622	598	662	63	-	-
Normalized	-	-	-	-	1	1	1	0.83	0.84	0.83	0.64	0.55	0.08	-	-

This formulation provides more flexibility to the movement of stitch unfriendly segments leading to an improved MEBL friendly layout compared to $Variant$-1. The result of reassignment of segments obtained by maximum bipartite matching for this formulation is shown in Fig. 5(e). $Variant$-2 removes all violations for the above example.

The time complexity of the entire flow including the preprocessing is $O(m^3d^3)$, where m and d are the number of stitch unfriendly and regular segments respectively.

IV. EXPERIMENTAL RESULTS

The proposed method has been implemented in C on a HP workstation XW8600 with Intel(R) Xeon(R) CPU @ 2.33GHz and 16GB RAM. The Python networkX library [9] is used to solve the maximum bipartite matching problem. The experiments have been performed on a set of Faraday benchmark circuits [10].

Table I presents the circuit specifications and the results of the proposed method for $Variant$-1 and $Variant$-2. The columns named $Circuit$, $Area$, $\#T\_Net$ denote the circuit name, area of the die measured in micrometers, and the total number of nets respectively. Note that routing layers 2, 4 and 6 among six metal layers are for vertical segments. The columns $Layer\#$ and $\#Nets$ report the number of nets that has its segment in a particular layer. The distance between two stitch lines is assumed to be 15 times the pitch distance. The stitch unfriendly region is set to one vertical routing track on both sides of the stitch line. The columns $\#RV$, $\#VV$ and $\#SP$ are the number of routing violations, via violations and short polygons respectively for initial and final layout obtained by the formulations of $Variant$-1 and $Variant$-2 for each layer. The column $Time$ reports CPU time in seconds for each $Variant$ for the proposed method.

The authors of [6] proposed a stitch-aware router for MEBL. For the placed Faraday benchmark circuits, the stitch line violations in their resulting layouts were compared with those in the layouts produced by a baseline router for the same placements. Since the final layouts produced by the baseline router were not available to us, our experimental results on reduction of the same types of violations could not be compared with those in [6]. It is not known whether the placements for the our layouts were same as those used in [6].

In summary, the experimental results show 17.41%, 16.46% and 17.45% reductions in routing violation, via violation and short polygon respectively in $Variant$-1, while $Variant$-2 yields significant reductions of 36.37%, 44.90% and 91.65% respectively in the corresponding violations with no increase in wirelength compared to the initial layout. However, $Variant$-2 has an overhead of CPU time.

V. CONCLUSION

This paper presents a bipartite matching based method to minimize the number of stitch unfriendly patterns in MEBL at the post-layout stage for the first time. The experimental results show significant reduction of violations. This motivates us to address the issue at the earlier stages of physical design as future work.

REFERENCES

[1] H. Y. Chen and Y. W. Chang, "Routing for manufacturability and reliability," *IEEE Circuits and Systems Magazine*, vol. 9, no. 3, pp. 20–31, Third 2009.

[2] S. Park, H. Youn, N. Chung, and J. M. et al., "Model based opc for implant layer patterning considering wafer topography proximity (w3d) effects," *Proc. SPIE*, vol. 8326, pp. 83 260O–83 260O–7, 2012.

[3] B. Yu, K. Yuan, B. Zhang, D. Ding, and D. Z. Pan, "Layout decomposition for triple patterning lithography," in *2011 IEEE/ACM ICCAD*, Nov 2011, pp. 1–8.

[4] N. W. Parker, A. D. Brodie, and J. H. McCoy, "High-throughput NGL electron-beam direct-write lithography system," *Proc. SPIE*, vol. 3997, pp. 713–720, 2000.

[5] B. J. Lin, "Future of multiple-e-beam direct-write systems," *Proc. SPIE*, vol. 8323, pp. 832 302–832 302–11, 2012.

[6] I. J. Liu, S. Y. Fang, and Y. W. Chang, "Stitch-aware routing for multiple e-beam lithography," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 3, pp. 471–482, 2015.

[7] K. Ronse, "E-beam maskless lithography: prospects and challenges," *Proc. SPIE*, vol. 7637, pp. 76 370A–76 370A–7, 2010.

[8] S. Pan and Y. W. Chang, "Crosstalk-constrained performance optimization by using wire sizing and perturbation," in *Computer Design*, 2000, pp. 581–584.

[9] Python package: https://pypi.python.org/pypi/networkx.

[10] http://vlsicad.eecs.umich.edu/bk/iccad04bench/#faraday_bench.