New and Exact Filling Algorithms for Layout Density Control*

Andrew B. Kahng, Gabriel Robins†, Anish Singh† and Alexander Zelikovsky

UCLA Department of Computer Science, Los Angeles, CA 90095-1596
†Department of Computer Science, University of Virginia, Charlottesville, VA 22903-2442

Abstract

To reduce manufacturing variation due to chemical-mechanical polishing and to improve yield, layout must be made uniform with respect to *density* criteria. This is achieved by layout postprocessing to add *fill* geometries, either at the foundry or, for better convergence of performance verification flows, during layout synthesis [10]. This paper proposes a new *min-variation* objective for the synthesis of fill geometries. Within the so-called *fixed-dissection* regime (where density bounds are imposed on a predetermined set of *windows* in the layout), we exactly solve the min-variation objective using a linear programming formulation. We also state criteria for fill pattern synthesis, and discuss additional criteria that apply when fill must be grounded for predictability of circuit performance. We believe that density control for CMP will become an important research topic in the VLSI design-manufacturing interface over the next several years.

1 Introduction

As CMOS technology advances into the 180nm generation and beyond, foundry amortization becomes a dominant business concern, manufacturing cost must increasingly drive design [13]. To maximize yield, process engineers must achieve *predictability* and *uniformity* of manufactured device and interconnect attributes, e.g., dopant concentrations, channel lengths, interconnect dimensions, contact shapes and parasitics, and interlayer dielectric thicknesses. A total *variability budget* for the design is distributed among such attributes [20] [9]. The manufacturing process has an increasingly constraining effect on physical layout design and verification, as the physics of very deep-submicron semiconductor processing makes large process windows and uniform manufacturing difficult [5] [9] [20] [12].

In this paper, we are concerned with manufacturing variation due to *chemical-mechanical polishing* (CMP) [12] [25] [17]. CMP is the procedure by which wafers are polished with a rotating pad and slurry to achieve the planarized surfaces on which succeeding processing steps can build. Key observations are:

- Large pad downforce[1], as well as significant variability due to pad wear. Hence, control of polish depth (i.e., final thickness of the layer being polished) is extremely difficult.
- The elasticity of the polishing pad compounds the variability problem. For example, in oxide polishing of interlayer dielectrics (oxide CMP), the pad conforms to local topography and overpolishes empty oxide areas that have no underlying metal features (a phenomenon called *dishing*); on the other hand, areas with dense underlying metal features are underpolished.
- As noted by such works as [23] [8], a huge fraction of the die's variability budget is used up by the oxide thickness variation. Interlayer dielectric thickness variation of 4000 angstroms is common; [23] [24] [10] observe that this can severely affect estimates of electrical performance.
- The problem of CMP variation is rapidly worsening today, as industry moves to shallow-trench isolation (STI) sub-0.25μm processes, where CMP is used to planarize glass [7] [15] [22]. For such processes, as well as for new inlaid-metal (e.g., damascene copper) processes [6], CMP variation must be even more tightly controlled.

Fundamentally, CMP variation will be controlled if the *local feature density* is controlled. Figure 1 illustrates the local dependency of oxide thickness on feature density, which is roughly monotone: by reducing the variation of local feature densities over the die, the variation of oxide thickness can be reduced.

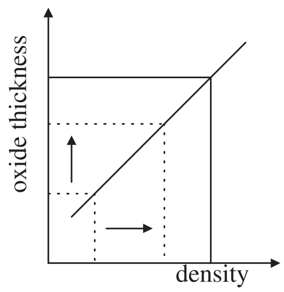

Figure 1: Relationship between oxide thickness and local feature density.

The definition of "local" is determined by the *length scale* at which feature density impacts oxide thickness, and corresponds to the "window size" within which feature density must be controlled. For oxide CMP, this length scale has been estimated to be on the order of 1-3mm, depending on CMP pad material, slurry composition, etc. [18]. Thus, foundries today typically impose *area density rules* for features on active and metal layers. (A typical rule would be of the form: "Within any window of size $w \times w$, the area density of metal features must be between 30% and 70%.") These rules are satisfied with layout postprocessing that adds *fill* geometries; the postprocessing is performed either by foundries or by specialized TCAD/verification tools.[2]

The remainder of this paper is organized as follows. Section 2 develops notation and terminology, drawing in part on those of a recent paper by Kahng et al. [10]. We then give new *min-variation* formulation of the *Filling Problem*. This

* This research was supported by a contract and grant from Cadence Design Systems, Inc.

[1] Typical polish downforces in oxide CMP range from 4 to 10 psi, depending on slurry / oxidizer concentration and process considerations. For 200mm substrates, this results in a total wafer downforce of up to 500 pounds [6].

[2] Kahng et al. [10] give reasons (e.g., performance verification flow) why layout density rules should be satisfied during, instead of after, the layout synthesis. We do not address the question of whether density control by filling belongs in layout synthesis or in layout verification: either way, it is an increasingly critical issue if manufacturing yield is to be maintained in deep-submicron processes.

new formulation is more realistic than those proposed in the previous work of [10]. In Section 3, we present solutions to the Filling Problem. We first restrict the discourse to the *fixed-dissection* regime; this allows us to consider a discrete set of dissections of the layout into $w \times w$ windows, rather than all possible windows. For the min-variation formulation, we give a new *exact* solution using linear programming. Several practical approaches to synthesis of filling patterns are described in Section 4. Finally, Section 5 concludes with a description of our experimental results and ongoing research directions.

2 Notation and Problem Formulations

Following [10], we use the following notation and definitions.

- The input is a *layout* consisting of rectangular *geometries*, with all sides having length a multiple of c (minimum feature width, spacing).
- $n \equiv$ side of the layout region. If the layout region is the entire die, n might be about $50,000 \cdot c$.
- $w \equiv$ fixed *window* size. The window is the moving square area over which the layout density rule applies.
- $k \equiv$ layout complexity, number of input rectangles.
- $U \equiv$ *area density* upper bound, expressed as a real number $0 < U < 1$. Each $w \times w$ region of the layout must contain total area of features $\leq U \cdot w^2$.
- $B \equiv$ *buffer distance.* Fill geometries cannot be introduced within distance B of any layout feature.
- $slack(W) \equiv$ *slack* of a given $w \times w$ window W. $slack(W)$ is the maximum amount of fill area that can be introduced into W.[3]

The terms B and $slack(W)$ are new to our present work. In contrast to the previous work of [10], our notation does not address the issue of density lower bounds (reasons will become clear shortly), nor the issue of perimeter density. We state the Filling Problem as follows.

The Filling Problem. Given a design rule-correct *layout* geometry of k disjoint rectilinear rectangles in an $n \times n$ layout region, minimum feature size c, window size $w < n$, buffer distance B, and area (or perimeter) density lower bound L and upper bound U, add fill geometries to create a *filled layout* that satisfies the following conditions:

(1) circuit function and design rule-correctness are preserved;

(2) no fill geometry is within distance B of any layout feature;

(3) no fill is added into any window that has density $\geq U$ in the original layout;

(4) for any window that has density $< U$ in the original layout, the filled layout density is $\geq L$ and $\leq U$; and

(5) the minimum window density in the filled layout is maximized.

[3] The value of $slack(W)$ will depend on the maximum possible fill pattern density. I.e., total empty area beyond the buffer distance B from any feature should be scaled by the maximum possible fill density to yield the slack of the window. The work of [10] gives some first steps toward synthesis of fill patterns with prescribed densities.

Condition (5) corresponds to what we call the **Min-Variation Formulation**, since it minimizes the difference between minimum and maximum window density in the filled layout. Condition (3) implies that, without loss of generality, no window in the original layout has density $> U$ (otherwise, such a window would have its contents fixed, so that it could not be changed by the filling process).

Notice that our formulation of the Filling Problem is considerably different from those given in [10]. We believe that the min-variation formulation, which directly minimizes CMP variation, more accurately reflects real-world objectives.

3 Solutions

In this section, we develop new exact solutions to the Filling Problem. We first restrict our discussion to the so-called *fixed-dissection* regime, then present new exact algorithms based on LP formulations.

The authors of [10] note that in practice, feature density bounds are enforced only within a fixed set of $w \times w$ windows corresponding to a *dissection* of the layout region. Since bounding the density in windows of a fixed dissection can incur error (i.e., other windows not in the dissection could violate the density bound), a common practice is to enforce density bounds in r^2 *overlapping fixed dissections*, where r determines the "phase shift" w/r by which the dissections are offset from each other.

Definition: A *fixed r-dissection* of the layout is the set of $w \times w$ windows having top-left corners at points $(i \cdot \frac{w}{r}, j \cdot \frac{w}{r})$, for $i, j = 0, 1, ..., r(\frac{n}{w} - 1)$, where r is a divisor of w.

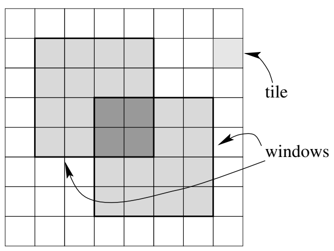

Figure 2: The layout is partitioned by r^2 ($r = 4$) fixed dissections into $\frac{nr}{w} \times \frac{nr}{w}$ *tiles.* Each $w \times w$ window (dark) consists of r^2 tiles. A pair of windows from different dissections may overlap.

We say that a fixed r-dissection divides the layout into $\frac{nr}{w} \times \frac{nr}{w}$ *tiles*, each of size $\frac{w}{r} \times \frac{w}{r}$ (see Figure 2). In other words, each $w \times w$ window in a fixed r-dissection consists of r^2 non-overlapping tiles.[4] The work of [10] showed that a density upper bound for arbitrarily located windows can be achieved by enforcing density upper bounds on all windows in a fixed r-dissection.

Theorem 1 *[10] Suppose all $w \times w$-sized windows with top-left corners at points $(i \cdot \frac{w}{r}, j \cdot \frac{w}{r})$, for $i, j = 0, 1, ..., r(\frac{n}{w} - 1)$, have area density at most U. Then any $w \times w$ window has density at most $\min\{1, U + \frac{1}{r} - \frac{1}{4r^2}\}$, and this bound is tight.* $\square$

[4] As noted in [10], layout density checking with respect to overlapping fixed dissections is the type of analysis most often performed by commercial tools. The Cadence Dracula COVERAGE command [4] is one example; it allows checking of feature area density upper and lower bounds in $w \times w$ windows (e.g., $w = 500\mu m$) that occur at a fixed offset from each other (e.g., $r = 5$ corresponds to an offset of $\frac{w}{r} = 100\mu m$).

For example, if $r = 10$ and all windows of a fixed r-dissection have feature area density at most 75% (i.e., $U = 0.75$), then the density of *any* $w \times w$ window in the layout is at most 85%. Theorem 1 allows us to consider the Filling Problem for only a fixed r-dissection of the layout, i.e., we will analyze density with respect to each $w \times w$-window W that covers exactly r^2 tiles. A desired accuracy of the result is achieved by increasing r.

For any given tile $T = T_{ij}, i, j = 1, \ldots, \frac{nr}{w}$, denote the total feature area inside T as $area(T)$. We define the *slack* of T, $slack(T)$, as the maximum fill amount that can be introduced using a given fill pattern into T without violating the density upper bound U in any window containing T. In other words, the total layout feature area inside T can be increased up to any value between $area(T)$ and $area(T) + slack(T)$, using fill geometries. The slack of T is determined by the total area of metal features inside T *and its neighbor tiles.* The slack of a window W is the sum of the slacks of the tiles that form W (efficient algorithms for slack computation are discussed in the next subsection). Using the concept of slack, the Filling Problem for the fixed-dissection regime can be formulated as follows.

The Filling Problem for a fixed r-dissection. Suppose we are given a fixed r-dissection of the layout into tiles of size $\frac{w}{r} \times \frac{w}{r}$, as well as an $area(T)$ and $slack(T)$ for each tile in the dissection. Then, for each tile T_{ij}, the total fill pattern area $p_{ij} = p(T_{ij})$ to be added to T_{ij} must satisfy

$$0 \leq p_{ij} \leq slack(T_{ij})$$

and

$$\sum_{T_{ij} \in W} p_{ij} \leq \max\{U \cdot w^2 - area(W), 0\} \quad (1)$$

for any fixed dissection $w \times w$-window W.

Then, the **Min-Variation Formulation** seeks to maximize the minimum window density:

$$maximize \quad \left(\min_{ij}(area(T_{ij}) + p_{ij}) \right)$$

A Linear Programming Approach

Consider the linear program:
Maximize M
subject to:

$$p_{ij} \geq 0, \quad i, j = 1, \ldots, \frac{nr}{w} - 1 \quad (2)$$

$$p_{ij} \leq pattern \cdot slack(T_{ij}), \quad i, j = 1, \ldots, \frac{nr}{w} - 1 \quad (3)$$

$$\sum_{s=i}^{i+r-1} \sum_{t=j}^{j+r-1} p_{st} \leq \alpha_{ij} \left(U \cdot w^2 - area_{ij} \right), \quad i, j = 1, \ldots, \frac{nr}{w} - r + 1 \quad (4)$$

$$M \leq area_{ij} + \sum_{s=i}^{i+r-1} \sum_{t=j}^{j+r-1} p_{st}, \quad i, j = 1, \ldots, \frac{nr}{w} - r + 1 \quad (5)$$

where

$$area_{ij} = \sum_{s=i}^{i+r-1} \sum_{t=j}^{j+r-1} area(T_{st})$$

is the area of the (i, j)-th window, and $\alpha_{ij} = 0$ if $area_{ij} > U \cdot w^2$ and 1 otherwise. Also, the pattern-dependent coefficient *pattern* denotes the maximum pattern area which can be embedded in an empty unit square.

The constraints (2) imply that features can only be added, and cannot be deleted from any tile. The slack constraints (3) are computed for each tile. The pattern-dependent coefficient *pattern* denotes the maximum pattern area which can be embedded in an empty unit square. If a tile T_{ij} is originally overfilled, then we set $slack(T_{ij}) = 0$. From the linear programming solution, the values of p_{ij} indicate the fill amount to be inserted in each tile T_{ij}. The constraint (4) says that no window can have density more than U after filling unless it was overfilled initially, i.e., such a window cannot increase its density. The number of variables and the number of constraints in the linear program are both $O((\frac{nr}{w})^2)$. In practice, even for a large die and a user requirement of high accuracy, we might have $n = 15000$, $w = 3000$, $r = 10$, which yields a linear program of tractable size. Equation (5) implies that the auxiliary variable M is the lower bound on all window densities. The linear programming seeks to maximize M, thus achieving the min-variation objective.

Solving the above LP formulation will give the optimal fill amounts to be added to each tile in the fixed r-dissection, as dictated by the Min Variation objective. However, as shown in [10], the LP solution may distribute the fill unevenly among the tiles of a given window. If this is unsatisfactory, various simple fixes can be applied (e.g., partial pre-filling of all tiles, binary search on an upper bound of fill added into each individual tile, etc.) so that the result is more balanced while still being optimal. (Our current implementation sets an upper bound U_t on the tile density in order to achieve a balanced fill pattern.)

Slack Computation

We describe how to efficiently compute slack values for the linear programming formulation from above. To compute the slack, i.e., to determine the total area of k possibly overlapping rectangles, we adopt the "measure of union of rectangles" sweep-line -based technique described in [19]. We begin by sorting all the left and right edges of all of the k rectilinear rectangles according to their x coordinates. Next, we sweep horizontally across these $2k$ edges from left to right, while using a segment tree [2] to keep track of the total length of the sweep line intersected by any of the k rectangles (see Figure 3).

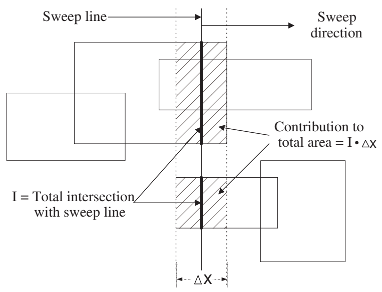

Figure 3: Finding the total area of a union of possibly intersecting rectangles using a sweep line technique.

The time complexity of the sorting step is $O(k \log k)$. Insertions and deletions from the segment tree require $O(\log k)$ time each, and the total time to process all $2k$ segments is therefore $O(k \log k)$. The total time complexity to determine the area of the union of k possibly overlapping rectangles is therefore $O(k \log k)$.

A simple implementation which avoids the usage of segment trees altogether can still have reasonably fast expected time, as follows. We still use the sweep line technique as before, but rather of using a segment tree to store the intersected rectangle, we instead use a simple linked list to store those segments, and then apply the one-dimensional "measure of union of intervals" technique of [11]. The time complexity of this practical implementation is $O(k^2)$ in the worst-case, and the expected time is $O(k \cdot h)$ where h is the average length of this list (i.e., the expected number of rectangles intersected by the sweep line). Thus, on average this method will run in time $O(k\sqrt{k})$ in practice.

4 Synthesis of Filling Patterns

Given the layout geometry along with the parameters of the Filling Problem, we apply the methods of previous sections to analyze density violations, and determine the necessary amounts of fill to be added in each region of the layout. We now discuss criteria for, and actual synthesis of, the fill geometries added into the layout.

Coupling to Long Conductors

Fill patterns should be devised such that all long conductors on adjacent layers have identical coupling capacitance to the inserted fill.[5] There are several practical ways of achieving this, of which one is to "basket-weave" the fill [26]. In other words, the fill pattern should not consist of a regular grid geometries, but instead have some internal offsets that "skew" the pattern. Figure 4 illustrates this concept.

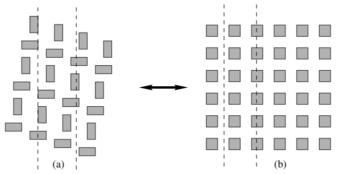

Figure 4: "Basket-weaving" of the fill pattern so that long conductors on adjacent layers will have identical coupling to the fill. With pattern (a), each vertical or horizontal crossover line will have the same overlap capacitance to fill. On the other hand, with fill pattern (b) two crossovers can have different coupling to fill.

Grounded vs. Floating Fill

Grounded fill can be required for *predictable* extracted parasitic values.[6] We seek a grounded fill pattern that requires relatively few edges to specify. We propose to stripe empty areas in the layout using horizontal lines (see Figure 5). Then, we span the horizontal stripes using vertical lines. The width and pitch of the horizontal stripes, and the number of vertical segments, can be easily determined in terms of the required pattern density. Connections to an existing ground distribution network can be made using standard special-net routers. An interesting possibility arises if separate ground planes of metalization are used in between signal layers (as in printed-circuit board construction). If there is a grounded sheet of metal on an adjacent layer, then grounded fill patterns can look similar to floating fill patterns (connections to ground are achieved by vias down to the adjacent layer).

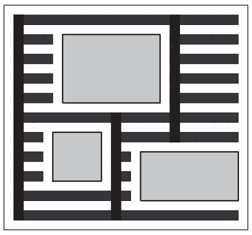

Figure 5: Given a layout, we create a grounded fill pattern by first creating horizontal stripes, and then spanning these stripes using a small number of vertical lines.

5 Conclusions and Ongoing Research Directions

To reduce manufacturing variation due to chemical-mechanical polishing and to improve yield, layout must be made uniform with respect to *density* criteria. This is achieved by layout postprocessing to add *fill* geometries, either at the foundry or, for better convergence of performance verification flows, during layout synthesis [10]. We have proposed a new *min-variation* objective for the synthesis of fill geometries. We believe that this new formulation of the Filling Problem is considerably more realistic and useful than those proposed in the previous work of [10].

Within the so-called *fixed-dissection* regime (where density bounds are imposed on a predetermined set of *windows* in the layout), we have given an exact linear programming solution for the min-variation objective. Finally, we state criteria for fill pattern synthesis, including additional criteria that apply when fill must be grounded for predictability of circuit performance. Our current experimental testbed integrates GDSII Stream input, conversion to CIF format, and internally-developed geometric processing engines.

Our experiments have been run using three metal layers extracted from industry standard-cell layouts. Benchmark L1 is the M2 layer from an 8131-cell design; Benchmark L2 is the M3 layer from a 20577-cell design; and Benchmark L3 is the M2 layer from the same 20577-cell design. The layout dimension N, number of rectangles k, and window size w (w always chosen to equal 1.5mm) for each test case are shown in Table 1.

Industry Test Cases			
Benchmark	layout size	# rectangles	window size
L1	125,000	49,506	31,250
L2	112,000	76,423	28,000
L3	112,000	133,201	28,000

Table 1: Parameters of three industry test cases.

[5] Coupling to same-layer conductors is not a concern, because the buffer distance B is usually quite large, e.g., 10 μm or more.

[6] Structured-custom (microprocessor) designs have strong requirements for predictability, due to aggressive timing tolerances. For such designs, it is better to have larger, but exactly known, coupling capacitances to grounded fill geometries, rather than indeterminate capacitances to floating fill. On the other hand, for ASIC designs where timing is not being pushed too hard, designers seek the simplest fill construction that meets feature density requirements. A secondary reason for studying grounded-fill constructions is that modern parasitic extraction tools do not handle floating capacitors well. If, as suggested in [10], fill synthesis should be performed earlier so as to achieve an accurate performance verification flow during the layout phase, it may be necessary to use grounded fill.

Table 2 shows the running time of the window density analysis in the fixed dissection regime for $r = 2, 4, 8$. The maximum fixed r-dissection window density reported is smaller than maximum window density because not all $w \times w$ windows are taken in account.

Recall that U is the user-specified upper bound on density of a filled window (it is forbidden to increase the density of a window whose initial density is greater than U), and that U_t is the (user-specified) upper bound on density of a filled tile. In our experiments we assume that U and U_t are the maximum window and tile density, respectively. In Table 3, the runtimes for preparing the min-variation LP formulation and solving resulting LP are given separately. The achieved minimum density for fixed r-dissection windows (M) is also reported.

We believe that density control for CMP will become an important research topic in the VLSI design-manufacturing interface over the next several years. Thus, our ongoing research seeks:

- improved heuristics or exact algorithms for the min-variation formulation,
- solutions to the Filling Problem in the continuous (as opposed to fixed-dissection) regime,
- robust implementations of our linear-programming based synthesis methods, and
- calibration of our methods with data and density control requirements from industry partners.

Fixed-Dissection Density Analysis			
Benchmark	r	Max Density	CPU Time
L1	2	.2021	1.3
L1	4	.2125	2.9
L1	8	.2170	9.2
L2	2	.1610	2.1
L2	4	.1791	4.5
L2	8	.1791	14.5
L3	2	.2883	3.6
L3	4	.2895	8.0
L3	8	.2910	25.1

Table 2: Fixed-dissection density analysis results.

Fixed-Dissection LP for Fill Amount and Fill Generation						
Bench-mark	r	generation (sec.)	solution (sec.)	M	Filling (sec.)	Total (sec.)
L1	2	4.3	0.0	.2192	3.3	7.6
L1	4	4.0	0.4	.2192	3.2	7.6
L1	8	10.3	18.3	.2189	3.3	31.9
L2	2	2.8	0.0	.1816	5.2	8.0
L2	4	5.2	1.7	.1704	5.0	11.9
L2	8	15.8	41.5	.1631	5.2	62.5
L3	2	5.2	0.0	.2640	8.3	13.5
L3	4	9.4	0.8	.2606	8.0	18.2
L3	8	27.2	24.4	.2553	8.1	59.7

Table 3: Experimental results showing CPU times to computing optimal fill amounts and fill generation.

6 Acknowledgments

We thank Larry Camilletti and Duane Boning for enlightening discussions. We also gratefully acknowledge a software donation from Artwork Conversions, Inc.

References

[1] R. Bek, C. C. Lin and J. H. Liu, *personal communication*, December 1997.

[2] J. L. Bentley, "Algorithms for Klee's Rectangle Problems", unpublished manuscript, 1977.

[3] D. Boning, W. Moyne, T. Smith, "Run by Run Control of Chemical-Mechanical Polishing", *IEEE Trans. CPMT(C)*, 19 (1996), pp. 307-314.

[4] Cadence Design Systems, Inc. *Dracula Standalone Verification Reference*, November 1997.

[5] L. E. Camilletti, "Implementation of CMP-based Design Rules and Patterning Practices", *1995 IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, pp. 2-4.

[6] L. E. Camilletti, *personal communication*, April 1998.

[7] A. Chatterjee, I. Ali, K. Joyner and D. Mercer et al, "Integration of Unit Processes in a Shallow Trench Isolation Module for a 0.25 μm Complementary Metal-Oxide Semiconductor Technology", *Journal of Vacuum Science and Technology B*, vol.15, (no.6):1936-42, 1997.

[8] R. R. Divecha, B. E. Stine, D. O. Ouma, J. U. Yoon, D. S. Boning, J. E. Chung, O. S. Nakagawa and S. Y. Oh, "Effect of Fine-line Density and Pitch on Interconnect ILD Thickness Variation in Oxide CMP Process", *Proc. CMP-MIC*, p. 29, Santa Clara, Feb. 1997.

[9] W. B. Glendinning and J. N. Helbert, *Handbook of VLSI Microlithography: Principles, Technology, and Applications*, Noyes Publications, 1991.

[10] A. B. Kahng, G. Robins, A. Singh, H. Wang, and A. Zelikovsky, "Filling and Slotting: Analysis and Algorithms", *Proc. ACM/IEEE Intl. Symp. on Physical Design*, April 1998, pp. 95-102.

[11] V. Klee, "Can the Measure of $\cup[a_i, b_i]$ be Computed in Less than $O(n \log n)$ Steps?", *American Mathematical Monthly*, 84(1977), pp. 284-285.

[12] H. Landis, P. Burke, W. Cote, W. Hill, C. Hoffman, C. Kaanta, C. Koburger, W. Lange, M. Leach and S. Luce, "Integration of Chemical-Mechanical Polishing into CMOS Integrated Circuit Manufacturing", *Thin Solid Films*, 220(1992), pp. 1-7.

[13] W. Maly, "Moore's Law and Physical Design of ICs", *special address, ACM/IEEE Intl. Symp. on Physical Design*, April 1998.

[14] K. Miyamoto, J. Nakamura, K. Hatanaka, T. Hashimoto, I. Tamura, T. Maeda, K. Sato and M. Kakumu, "Impact of Pattern Density on Plasma Damage of CMOS LSIs", *Proc. IEDM*, 1996, pp. 739-742.

[15] S. Nag and A. Chatterjee, "Shallow Trench Isolation for Sub-0.25-mu m IC Technologies", *Solid State Technology*, Sept. 1997, vol.40, (no.9):129-30, 132, 134, 136.

[16] O. S. Sakagawa, S.-Y. Oh and G. Ray, "Modeling of Pattern-Dependent On-Chip Interconnect Geometry Variation for Deep-Submicron Process and Design Technology", *Proc. IEDM*, 1997, pp. 137-140.

[17] G. Nanz and L. E. Camilletti, "Modeling of Chemical-Mechanical Polishing: A Review", *IEEE Trans. on Semiconductor Manufacturing* 8(4) (1995), pp. 382-389.

[18] D. Ouma, C. Oji, D. Boning and J. Chung, "Effect of High Relative Speed on Planarization Length in Oxide Chemical Mechanical Polishing", *Proc. CMP-MIC*, Santa Clara, Feb. 1998.

[19] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985.

[20] P. Rai-Choudhury, ed., *Handbook of Microlithography, Micromachining, and Microfabrication, vol. 1: Microlithography*, Bellingham, SPIE Optical Engineering Press, 1997.

[21] Semiconductor Industry Association, *The National Technology Roadmap for Semiconductors*, December 1997.

[22] K. Smekalin, "CMP Dishing Effects in Shallow Trench Isolation", *Solid State Technology*, July 1997, vol.40, (no.7):187-8, 190, 192, 194.

[23] B. E. Stine, D. S. Boning, J. E. Chung, L. Camilletti et al, "The Physical and Electrical Effects of Metal-fill Patterning Practices for Oxide Chemical-Mechanical Polishing Processes", *IEEE Transactions on Electron Devices*, vol.45, (no.3):665-79, 1998.

[24] B. E. Stine, V. Mehrotra, D. S. Boning, J. E. Chung and D. J. Ciplickas, "A Simulation Methodology for Assessing the Impact of Spatial/Pattern Dependent Interconnect Parameter Variation on Circuit Performance", *IEDM Technical Digest*, pp. 133-136, 1997.

[25] M. Tomozawa, "Oxide CMP Mechanisms", *Solid State Technology*, July 1997, pp. 169-175.

[26] K. Wampler and T. Laidig, *personal communication*, Sept. 1997.