A Novel and Unified Full-Chip CMP Model Aware Dummy Fill Insertion Framework With SQP-Based Optimization Method

Junzhe Cai, Changhao Yan, *Member, IEEE*, Yudong Tao, Yibo Lin, *Member, IEEE*,
Sheng-Guo Wang, *Life Senior Member, IEEE*, David Z. Pan, *Fellow, IEEE*,
and Xuan Zeng, *Senior Member, IEEE*

***Abstract*—Dummy filling is widely applied to significantly improve the planarity of topographic patterns for the chemical mechanical polishing process in VLSI manufactures. The main challenge of dummy filling is balancing multiple objectives, such as fill amounts, planarity, parasitic capacitance, etc. An obvious drawback of traditional rule-based dummy filling methods is pattern densities, instead of post-chemical mechanical polishing (CMP) topographies, being included in optimization objectives. Although the quality of post-CMP topography strongly depends on pattern features of layouts, especially the density uniformity, however, experimental results show that chip surface variations are not exactly the same as density variations. In this article, a unified dummy fill insertion optimization framework is proposed, integrated with the multiple starting points-sequential quadratic programming (MSP-SQP) optimization solver, where all objectives are considered without approximation. Inside this framework, a full-chip CMP simulator is first integrated to evaluate the planarity of the chip surface. By selecting the initial points smartly with heuristic prior knowledge, the proposed method can be effectively accelerated. The effectiveness of the proposed algorithm is verified with the average 25.8% improvement of quality compared with rule-based methods.**

***Index Terms*—Chemical mechanical polishing (CMP), design for manufacture, dummy fill insertion, sequential quadratic programming (SQP).**

I. Introduction

Chemical mechanical polishing (CMP) is widely applied to layout planarization in integrated circuit fabrication. The quality of post-CMP topography strongly depends on pattern features of layout, especially the pattern density uniformity [1]. Dummy fill insertion is generally processed to improve layout pattern density uniformity for CMP. However, dummy fill insertion could induce additional parasitic capacitance and deteriorate the circuit performance [2].

To balance the planarization and performance degradation, dummy filling algorithms need to consider amount, capacitance, and density constraints simultaneously [3]. Determining the amount is termed dummy synthesis [4], while determining the locations and shapes of dummies termed dummy insertion [3], [5]–[8].

Methods for dummy synthesis can be roughly classified into two categories: 1) rule-based and 2) model-based. Most research works are based on prior knowledge of the CMP process, termed *rule*, for example, density variation, density gradient, etc. Rare works are based on CMP models because calibrated full-chip CMP simulators are complicated and expensive.

Rule-based dummy filling synthesis was first formulated as a linear programming (LP) problem [9] to balance density variation and fill amount. Covering LP (CLP) was proposed to improve efficiency [10]. Then overlay area was considered for estimating dummy capacitance [11], which was formulated as integer LP (ILP) [12].

In 2014, ICCAD launched a dummy fill contest [4], where multiple metrics, including fill amount, overlay, density variation, line deviation, and outliers, are considered and balanced to formulate the quality of dummy fills. Several methods have been proposed based on this benchmark. An LP-based algorithm considering layout planarity and fill amount was proposed in [13]. An improved ILP algorithm considering all objectives and gradient minimization constraints was proposed in [14] and [15]. We proposed a unified dummy filling framework, but only verified by the rule-based benchmarks [16].

However, rule-based approaches for dummy insertion have an intrinsic drawback that density related rules can be thought as a coarse emulation of CMP models. It is hard to describe the complex behaviors of a full-chip CMP simulator in rules. Therefore, the results of rule-based methods, even based on ICCAD 2014 benchmarks, remain dubious.

Instead, model-based methods can utilize the electro-chemical plating (ECP) and CMP models to avoid the incompleteness of rules. Tian *et al.* [9] proposed the model-based dummy fill placement algorithm using a two-step procedure of global density assignment followed by a local insertion, which achieved excellent results for post-CMP ILD topography reduction. Sinha *et al.* [17] proposed a novel ECP model-based metal filling method to improve chip surface uniformity, which significantly reduced the height variation of chip surface and amount of dummies. Chen *et al.* [18] compared rule- and model-based dummy filling methods, and concluded that the model-based method is not better. However, [18] used the analytical models and applied only for CMP hot-spots. Therefore, their conclusion is conditional on their methods and experiments.

The key step of this article is that model-based methods should NOT lie on semiempirical or analytical ECP/CMP models for efficiency consideration, as most model-based works have done. The calibrated full-chip CMP simulators will provide better quality in dummy filling, with the affordable tradeoff of speed loss. Since the quality of dummy filling determines the post-CMP topography and directly affects the chip yields, it is worthwhile to spend more CPU resources on improving the filling quality.

Manuscript received December 11, 2019; revised March 24, 2020; accepted May 25, 2020. Date of publication June 10, 2020; date of current version February 19, 2021. This work was supported in part by the National Major Science and Technology Special Project of China under Grant 2017ZX01028101-003, and in part by the National Natural Science Foundation of China Research Projects under Grant 61674042, Grant 61974032, Grant 61774045, and Grant 61929102. This article was recommended by Associate Editor L. Behjat. *(Corresponding authors: Changhao Yan; Xuan Zeng.)*

Junzhe Cai, Changhao Yan, Yudong Tao, and Xuan Zeng are with the School of Microelectronics, Fudan University, Shanghai 200433, China (e-mail: 19212020039@fudan.edu.cn; yanch@fudan.edu.cn; 11300720039@fudan.edu.cn; xzeng@fudan.edu.cn).

Yibo Lin is with the Center for Energy-Efficient Computing and Applications, Peking University, Beijing 100871, China (e-mail: yibolin@pku.edu.cn).

Sheng-Guo Wang is with the Department of Engineering Technology, University of North Carolina at Charlotte, Charlotte, NC 28223 USA (e-mail: swang@uncc.edu).

David Z. Pan is with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78712 USA (e-mail: dpan@ece.utexas.edu).

Digital Object Identifier 10.1109/TCAD.2020.3001380

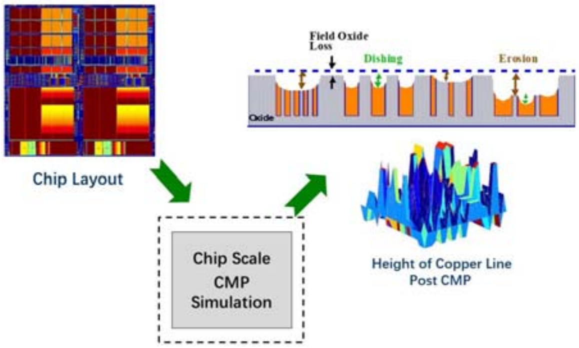

Fig. 1. Interface of a full-chip CMP simulator.

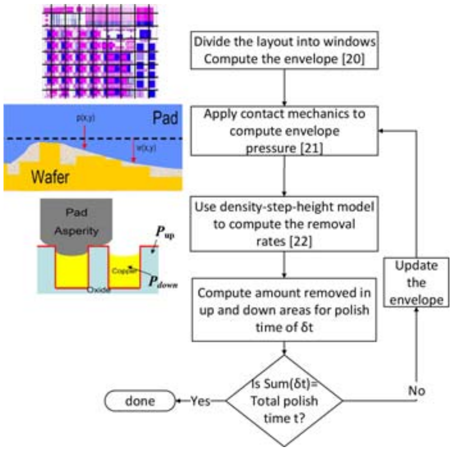
Divide the layout into windows Compute the envelope [20]
Pad
p(x,y)
w(x,y)
Wafer
Apply contact mechanics to compute envelope pressure [21]
Pad Asperity
P_{up}
Copper
P_{down}
Oxide
Use density-step-height model to compute the removal rates [22]
Update the envelope
Compute amount removed in up and down areas for polish time of δt
Is Sum(δt)= Total polish time t?
done
Yes
No

Fig. 2. Framework of a full-chip CMP simulator.

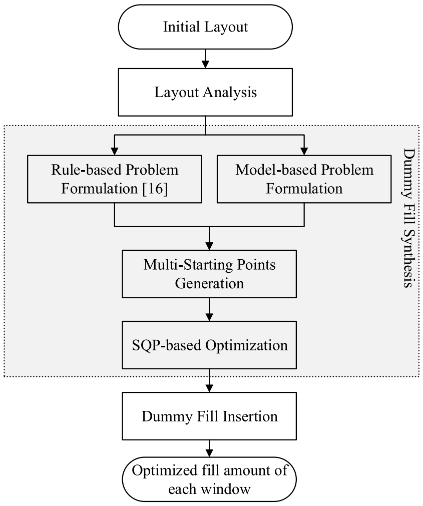
Initial Layout
Layout Analysis
Rule-based Problem Formulation [16]
Model-based Problem Formulation
Dummy Fill Synthesis
Multi-Starting Points Generation
SQP-based Optimization
Dummy Fill Insertion
Optimized fill amount of each window

Fig. 3. Framework of dummy filling algorithm.

In this article, a novel and unified dummy fill insertion framework based on multistarting points and sequential quadratic programming (MSP-SQP) [19] is proposed, where both rule- and model-based methods can be applied. The objectives of the latter are hard to be simplified. Meanwhile, a full-chip CMP simulator, while not simplified closed-form CMP models, is first integrated into the proposed dummy filling framework. The sequential quadratic programming (SQP) solver, along with prior knowledge-based starting point generation methods, is applied to obtain the optimized solution. The experimental results show an average 25.8% improvement in quality compared with rule-based methods.

The remainder of this article is organized as follows. Section II presents the framework of the full-chip CMP simulator. Section III gives the details of the model-based flow of the proposed unified dummy filling framework. Section IV gives the experimental results and Section V concludes this article.

II. Review of Full-Chip CMP Simulator

A typical interface of a full-chip CMP simulator is illustrated in Fig. 1, where the input is a full-chip layout, and the outputs are the predictive *dishing*, *erosion*, and average height profile of the chip surface at post-CMP.

Generally, the framework of a full-chip CMP simulator includes four steps as follows in Fig. 2.

1) Divide the whole chip into uniform grids and compute the envelope heights of the grids [20].
2) Apply contact mechanics and/or fluid mechanics to solve the partial differential equations (PDEs) among polishing pad, chip surface, slurry, and particles inside slurry, and obtain the average pressure of each grid [21].
3) In each grid, apply the density-step height (DSH) model [22] to compute the different pressures of up and down areas, and get the removal rates of both areas.
4) Compute the amounts removed in up and down areas within a unit polish time by the Preston equation [23].

All four steps iterate until a given total polishing time is met. Finally, the chip topography, including *dishing*, *erosion*, and an average height of each grid can be obtained, as illustrated in Fig. 1.

For improving accuracy, most parameters of the full-chip CMP model need to be calibrated on realistic production lines in foundries. Nowadays, full-chip CMP simulators have become the standard tools for overcoming DFM problems in the reference flows of foundries after 45-nm technology node.

III. Unified Framework of Dummy Filling

In this section, the details of the proposed *unified* framework of the dummy filling algorithm are described.

Fig. 3 shows an aerial view of the unified framework of dummy fill insertion algorithms, which mainly includes three phases: 1) layout analysis; 2) dummy fill synthesis; and 3) dummy fill insertion. In layout analysis, a chip is divided into uniformed windows. Dummy fill synthesis tries to answer the problem of how many amounts of dummies needed in each window. To solve this optimization problem, heuristic starting points are generated and optimized by the SQP method [19], a nonlinear optimizer, to obtain better quality. Finally, the locations of all dummies are determined and stored in a GDS file during the dummy fill insertion phase.

The rule-based part was performed on ICCAD 2014 benchmarks and showed vast (22%) filling quality improvements over the top team [16]. In the following, we mainly focus on the model-based part.

A. Intrinsic Problem of Density Rule

Most rule-based methods are based on copper density. Although it is one of the most important factors for the post-CMP height profile, other factors, such as average width, length, perimeter of coppers, etc., may influence the final results. Therefore, although the density rule is widely used and easy to solve, their dummy filling results may suffer from serious topography quality degeneration after real CMP processes.

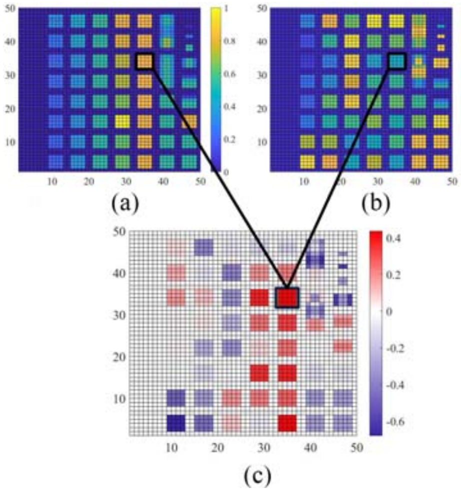

Fig. 4. Difference between density and height of post-CMP. (a) Normalized window density. (b) Normalized window height. (c) Difference of density rule and CMP model.

Fig. 4 shows comparison results of normalized density [Fig. 4(a)] and normalized height [Fig. 4(b)] of chip surface topography with a layout, where the height is generated by a calibrated full-chip CMP simulator. Based on the philosophy of density rules, the chip density will be a good prediction of chip height. However, from the difference between normalized density and normalized height [Fig. 4(c)], we can see that the positions with positive or negative vast differences are considerable, and they may result in severe performance degeneration and yield problem. To avoid ineffective optimization in rule-based methods, it is better to include the results of full-chip CMP simulators from the start.

B. Model-Based Dummy Fill Synthesis

In this section, a model-based dummy fill synthesis method is proposed. The evaluation metrics were modified from [4] to formulate the model-based synthesis problem.

Three objectives are calculated to evaluate the L-layer layout planarity, including height variation σ, line deviation σ^*, and outliers ol

$$\sigma = \sum_{l=1}^{L} \sqrt{\frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} (H_{l,i,j} - \bar{H}_l)^2} \tag{1a}$$

$$\sigma^* = \sum_{l=1}^{L} \sum_{i=1}^{N} \sum_{j=1}^{M} \left| H_{l,i,j} - \bar{H}_{l,j} \right| \tag{1b}$$

$$ol = \sum_{l=1}^{L} \sum_{i=1}^{N} \sum_{j=1}^{M} \max(0, H_{l,i,j} - 3 \cdot \sigma_l) \tag{1c}$$

where σ_l is the height variation of layer l, $\bar{H}_l$ and $\bar{H}_{l,j}$ are the average window height of layer l and of column j in layer l, respectively. Parasitic capacitance is another major concern during dummy fill insertion. Overlay area ov and total fill amount fa are two major factors relevant to performance degradation [11]. Additionally, since file size fs is hard to formulate, though defined as a criterion [4], it is not optimized in this article.

The objective of dummy fill synthesis is to maximize dummy fill quality and can thus be formulated as

$$\max_{x_{l,i,j}} \quad [f(\mathbf{x}) = f_{ov}(\mathbf{x}) + f_{fa}(\mathbf{x}) + f_{\sigma}(\mathbf{H}) + f_{\sigma^*}(\mathbf{H}) + f_{ol}(\mathbf{H})] \tag{2a}$$

$$\text{s.t.} \quad x_{l,i,j} \in [0, s_{l,i,j}] \tag{2b}$$

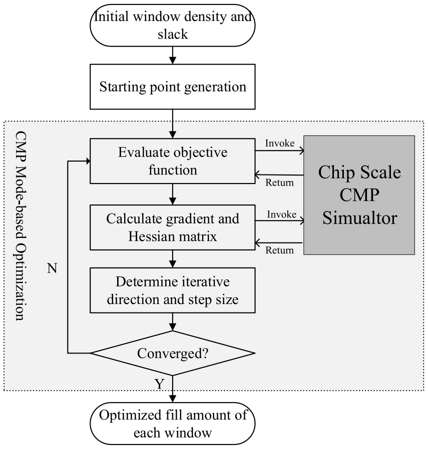
Initial window density and slack
Starting point generation
CMP Mode-based Optimization
Evaluate objective function
Invoke
Return
Chip Scale CMP Simualtor
Calculate gradient and Hessian matrix
Invoke
Return
N
Determine iterative direction and step size
Converged?
Y
Optimized fill amount of each window

Fig. 5. Framework of CMP model-based dummy filling algorithm.

where $f_{ov}, f_{fa}, f_{\sigma}, f_{\sigma^*}$, and f_{ol} refer to the score function of the overlay, fill amount, height variation, line deviation, and outliers in [4], respectively; $x_{l,i,j}$ and $s_{l,i,j}$ refer to the fill amount and slack area in window $W_{l,i,j}$, respectively; $\mathbf{x}$ and $\mathbf{H}$ refer to the vector of fill amount $x_{l,i,j}$ and height $H_{l,i,j}$, respectively.

Since the objective of optimization becomes more complicated, a linear approximation formulation can likely lead to a suboptimal solution. SQP method [19] is applied in our framework for fill amount optimization. With SQP and nonapproximation model-based formulation with a CMP simulator embedded, a better filling quality may be obtained.

Inequality constraints in (2b) are merged into the objective function with a Lagrangian approach [19]. The objective with merged constraints becomes

$$\max_{x_{l,i,j}} \mathcal{L}(\mathbf{x}, \mathbf{v_1}, \mathbf{v_2}) = f(\mathbf{x}) + \mathbf{v_1^T} \cdot (\mathbf{x} - \mathbf{s}) + \mathbf{v_2^T} \cdot (-\mathbf{x}) \tag{3}$$

where $\mathbf{s}$ is the vector of slack $s_{l,i,j}$, $\mathbf{v_1}$ and $\mathbf{v_2}$ are vectors of Lagrangian multipliers for constraint merging, and they are iteratively calculated during SQP optimization.

Fig. 5 shows the flow of the unified dummy fill framework integrated with a calibrated full-chip CMP simulator. In every iteration, surface features such as metal density, metal perimeter, average line width, and so on are calculated and given to the simulator. The simulator utilizes these surface features to produce the average heights, which, instead of the density, will be evaluated by the objective function. The full-chip CMP simulator is regarded as a nonlinear black box, so the gradient cannot be directly derived. The elements of the numerical gradient, i.e., $\nabla f_{i,j,l}$, are calculated as

$$\nabla f_{i,j,l} = \frac{f(\mathbf{x} + \Delta x_{i,j,l}) - f(\mathbf{x})}{\Delta x_{i,j,l}} \tag{4}$$

where a small dummy $\Delta x_{i,j,l}$ is given in each window $W_{i,j,l}$. Evaluation of objective function f needs an invocation of the CMP simulator, and therefore, totally, $N \times M \times L$ invocations are needed in gradient calculation, which dominates the time of the model-based method.

C. Prior Knowledge-Based Starting Point Generation

The model-based algorithm provides accurate post-CMP planarity at the cost of more runtime, because the simulator is invoked for every dimension in the gradient calculation. Therefore, it is crucial to select good starting points, which not only give better optimization results but also significantly reduce the runtime.

TABLE I
COMPARISON OF THE RULE- AND MODEL-BASED METHODS

Design	Type	Method	ΔH	Performance	Variation	Line Deviation	Outliers	File Size	Runtime	Memory	**Quality**	Overall
Case 1 5cm × 5cm 16.4MB	rule-based	Lin [14]	174Å	0.000	0.145	0.445	1.000	0.967	1.000(1s)	0.756	**0.395**	0.504
	rule-based	Tao [16]	174Å	1.000	0.142	0.425	1.000	0.970	0.968(39s)	0.756	**0.640**	0.695
	model-based	Tao+SQP	147Å	1.000	0.390	0.615	1.000	0.974	0.07(18.6min)	0.756	**0.750**	0.648
	model-based	MPKB	147Å	1.000	0.400	0.615	1.000	0.989	0.985(18s)	0.756	**0.753**	0.788
	model-based	MPKB+SQP	139Å	1.000	0.595	0.750	1.000	0.989	0(1.5h)	0.756	<u>**0.835**</u>	0.706
Case 2 6.7cm × 6.3cm 948.7MB	rule-based	Lin [14]	283Å	0.000	0.425	0.525	0.333	0.683	1.000(1s)	0.707	**0.343**	0.459
	rule-based	Tao [16]	283Å	1.000	0.445	0.530	0.38	0.714	0.915(1.7min)	0.707	**0.610**	0.660
	model-based	Tao+SQP	271Å	1.000	0.447	0.591	0.387	0.714	0(36.7min)	0.707	**0.627**	0.536
	model-based	MPKB	219Å	1.000	0.455	0.530	0.960	0.874	0.973(32s)	0.707	**0.731**	0.766
	model-based	MPKB+SQP	219Å	1.000	0.457	0.550	0.973	0.874	0(3.0h)	0.707	<u>**0.739**</u>	0.627

```
Algorithm 1 Dummy Fill Insertion
Input: Fill amount $x^k_{l,i,j}$ of each type in each window [16]
Output: Locations of all dummies
1: Classify slack into four types [16]
2: Cut the slack patterns into rectangles by the method proposed in [24].
3: Shrink slacks to satisfy all DRC rules
4: Insert dummies into each type of slacks of each $W_{l,i,j}$ in order of their areas
5: Insert a proper small dummy to minimize fill amount gap
```

A straightforward idea is to select the rule-based optimum as a starting point, since the rule-based optimization requires much less time than that of the model-based method. Therefore, the layout will be optimized first by the rule-based algorithm in [16], and the results will be the starting points of the model-based SQP optimization.

Another starting point is the model-based prior knowledge-based (MPKB) method, which is motivated by [14]. A target layer density td_l is first determined in advance for each layer, which represents the expected density of each window. After td_l is determined, a trivial solution to fill dummies for maximum density uniformity can be obtained

$$x_{l,i,j} = \begin{cases} 0, & \text{if } td_l < \rho_{l,i,j} \\ s_{l,i,j}, & \text{if } td_l > \rho_{l,i,j} + s_{l,i,j} \\ td_l - \rho_{l,i,j}, & \text{otherwise} \end{cases} \tag{5}$$

where $\rho_{l,i,j}$ is the pattern density of window $W_{l,i,j}$. Equation (5) shows a simple strategy to minimize density variation, which is filling dummies to let window density become as near as target layer density td_l. A linear search of target layer density is performed on each layer. Then, all solutions are evaluated by the model-based objective function. The solution with the best planarity is chosen as the starting point.

D. Dummy Fill Insertion

After optimizing the fill amounts in each slack type in each window, the dummy fill insertion phase will determine all positions of fills without violating DRC rules. The dummy fill insertion flow is briefly shown in Algorithm 1.

IV. EXPERIMENTAL RESULTS

We implement the proposed algorithm in C/C++ language. All the experiments of our algorithm are performed on a 2.67-GHz Linux server with 64 CPU cores. The full-chip CMP simulator is developed by our team, and calibrated under a 45-nm process of a foundry. The accuracy of our full-chip CMP simulator is compared and matched with the CMP Predictor [25], a commercial full-chip CMP simulator by Cadence.

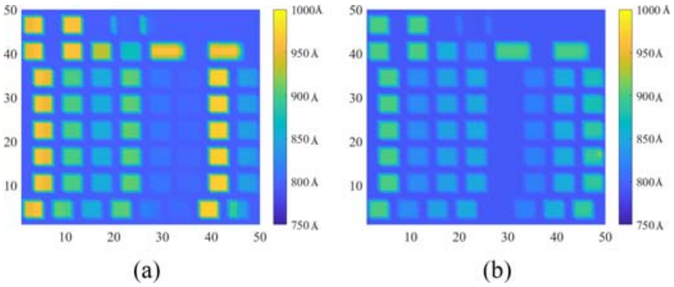

Fig. 6. Post-CMP height distributions of two methods. (a) Post-CMP height distribution of rule-based method. (b) Post-CMP height distribution of model-based method.

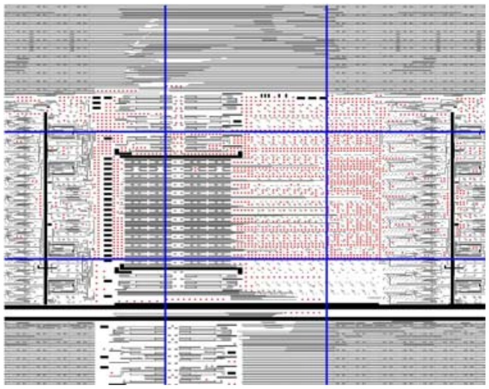

Fig. 7. Visualization of the local dummies on case 2.

Table I shows the comparison results of the proposed rule- and model-based algorithms in two layouts. Case 1 is a CMP test design on 45-nm technology node with chip size 5 cm × 5 cm and file size 16.4 MB. Case 2 is a field programmable gate array (FPGA) design with chip size 6.7 cm × 6.3 cm and file size 948.7 MB. The window sizes of the CMP simulator and dummy filling are both 100 μm × 100 μm. In the column *Method*, Lin and Tao are rule-based methods in [14] and [16], respectively. MPKB is the result of the proposed initial filling method. Tao+SQP and MPKB+SQP are the results of using Tao and MPKB's results as starting points and then optimizing by the proposed SQP optimization algorithm. ΔH is the maximum height variations of layouts with dummies, which is commonly used to evaluate the surface roughness.

For both designs, the filling qualities of model-based methods are much better than rule-based ones. For case 1, the best rule-based quality score is 0.640. However, all three model-based methods are better than that. The highest score is 0.835 by the model-based prior knowledge with SQP (MPKB+SQP), which is far superior (30.5%) to the best rule-based methods (our rule-based method Tao *et al.* [16]). The same results can be found in case 2, where the best model-based algorithm MPKB+SQP has a 21.1% quality improvement. So, the average filling improvement by MPKB+SQP can reach 25.8% in two cases.

The same conclusion can be drawn from the height variations ΔH, where 174 Å versus 139 Å of case 1 and 283 Å versus 219 Å of case 2 are from the best rule- and model-based methods, respectively. A more detailed comparison of the post-CMP height distribution from the best rule- and model-based methods is shown in Fig. 6, which intuitively shows the surface uniformity improvement by the model-based one. Fig. 7 shows local dummies in the 3 × 3 windows of case 2 in detail, where the black represents the metal patterns and the red represents the dummies.

Certainly, the tradeoff of high filling quality by the model-based methods is the long runtime for invoking the CMP simulator repeatedly. All the runtime scores of model-based methods are nearly zero, and therefore, the improvement of overall scores seems not so significant. However, filling a full chip layout (case 2) can be executed in parallel, taking 1.5 to 3 h with 64 CPU cores. The computing demands are modest relative to the benefits derived.

V. Conclusion

This article proposed a novel and full-chip CMP simulator aware dummy fill insertion framework, which is suitable for complicate optimization objectives. With the introduced full-chip CMP simulator, a model-based algorithm with a unified optimization framework is proposed. The experimental results show that the unified framework is effective for the model-based multiple objectives. To obtain better quality, SQP optimization requires more runtime than other proposed algorithms, while the cost is affordable. The prior-knowledge-based starting point generation methods are raised for the model-based algorithm to improve both performance and efficiency. Most importantly, the model-based optimization algorithm can significantly improve the quality of post-CMP topography.

References

[1] A. B. Kahng and K. Samadi, "CMP fill synthesis: A survey of recent studies," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 1, pp. 3–19, Jan. 2008.

[2] W.-S. Lee, K.-H. Lee, J.-K. Park, T.-K. Kim, Y.-K. Park, and J.-T. Kong, "Investigation of the capacitance deviation due to metal-fills and the effective interconnect geometry modeling," in *Proc. 4th Int. Symp. Qual. Electron. Design*, 2003, pp. 373–376.

[3] T. Lan *et al.*, "Timing-aware fill insertions with design-rule and density constraints," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2019, pp. 1–8.

[4] R. O. Topaloglu, "ICCAD-2014 CAD contest in design for manufacturability flow for advanced semiconductor nodes and benchmark suite," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2014, pp. 367–368.

[5] A. B. Kahng and R. O. Topaloglu, "Performance-aware CMP fill pattern optimization," in *Proc. VMIC*, 2007, pp. 3–9.

[6] B. Yang and S. Sridharan, "ICCAD-2018 CAD contest in timing-aware fill insertion," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2018. [Online]. Available: http://iccad-contest.org/2018/Problem_C/2018ICCADContest_ProblemC.pdf

[7] B. Jiang *et al.*, "FIT: Fill insertion considering timing," in *Proc. 56th ACM/IEEE Design Autom. Conf. (DAC)*, 2019, pp. 1–6.

[8] S. Yu, C. Kao, C. Huang, and I. H. Jiang, "Equivalent capacitance guided dummy fill insertion for timing and manufacturability," in *Proc. 25th Asia South Pac. Design Autom. Conf. (ASP-DAC)*, 2020, pp. 133–138.

[9] R. Tian, D. F. Wong, and R. Boone, "Model-based dummy feature placement for oxide chemical-mechanical polishing manufacturability," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 7, pp. 902–910, Jul. 2001.

[10] C. Feng, H. Zhou, C. Yan, J. Tao, and X. Zeng, "Efficient approximation algorithms for chemical mechanical polishing dummy fill," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 3, pp. 402–415, Mar. 2011.

[11] A. B. Kahng and R. O. Topaloglu, "A DoE set for normalization-based extraction of fill impact on capacitances," in *Proc. 8th Int. Symp. Qual. Electron. Design (ISQED)*, 2007, pp. 467–474.

[12] H. Xiang, L. Deng, R. Puri, K. Chao, and M. D. F. Wong, "Fast dummy-fill density analysis with coupling constraints," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 4, pp. 633–642, Apr. 2008.

[13] C. Liu *et al.*, "An effective chemical mechanical polishing filling approach," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, 2015, pp. 44–49.

[14] Y. Lin, B. Yu, and D. Z. Pan, "High performance dummy fill insertion with coupling and uniformity constraints," in *Proc. 52nd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, 2015, pp. 1–6.

[15] Y. Lin, B. Yu, and D. Z. Pan, "High performance dummy fill insertion with coupling and uniformity constraints," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 9, pp. 1532–1544, Jul. 2017.

[16] Y. Tao, C. Yan, Y. Lin, S.-G. Wang, D. Z. Pan, and X. Zeng, "A novel unified dummy fill insertion framework with SQP-based optimization method," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2016, pp. 1–8.

[17] S. Sinha, J. Luo, and C. Chiang, "Model based layout pattern dependent metal filling algorithm for improved chip surface uniformity in the copper process," in *Proc. Asia South Pac. Design Autom. Conf.*, 2007, pp. 1–6.

[18] X. Chen, S. Li, and J. Zhang, "A contrast between rule-based and model-based dummy metal fill in ASIC design," in *Proc. Int. Conf. Intell. Control Inf. Process.*, 2010, pp. 601–606.

[19] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta Numerica*, vol. 4, no. 1, pp. 1–51, Jan. 1995.

[20] T. E. Gbondo-Tugbawa, "Chip-scale modeling of pattern dependencies in copper chemical mechanical polishing processes," Ph.D. dissertation, Dept. Elect. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, 2002.

[21] C. Feng, C. Yan, J. Tao, X. Zeng, and W. Cai, "A contact-mechanics-based model for general rough pads in chemical mechanical polishing processes," *J. Electrochem. Soc.*, vol. 156, no. 7, pp. H601–H611, 2009.

[22] H. Cai *et al.* "Modeling of pattern dependencies in the fabrication of multilevel copper metallization," Ph.D. dissertation, Dept. Elect. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, 2007.

[23] L. M. Cook, "Chemical processes in glass polishing," *J. Noncrystalline Solids*, vol. 120, nos. 1–3, pp. 152–171, 1990.

[24] K. D. Gourley and D. M. Green, "A polygon-to-rectangle conversion algorithm," *IEEE Comput. Graph. Appl.*, vol. 3, no. 1, pp. 31–36, Jan. 1983.

[25] *Cadence Corp. CMP Predictor*. Accessed: Mar. 24, 2020. [Online]. Available: https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/silicon-signoff/cmp-predictor.html