SONIC: Smart Optimization for Neural-Integrated CMP with Timing-Aware Fills

Jiajun Tan[1], Qichao Ma[1], Yiming Du[1], Yiming Gan[2], Ling Lang[1†],
Yibo Lin[1], Ming Zhu[3], Zongwei Wang[1], Yimao Cai[1†]
[1]School of Integrated Circuits, Beijing Advanced Innovation Center for Integrated Circuits, Peking University, Beijing, China
[2]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[3]School of Integrated Circuits, Anhui University, Hefei, China
javin@stu.ahu.edu.cn, qma@pku.edu.cn, yimingdu@stu.ahu.edu.cn, ganyming@ict.ac.cn, lingliang@pku.edu.cn,
yibolin@pku.edu.cn, zhuming@ahu.edu.cn, wangzongwei@pku.edu.cn, caiyimao@pku.edu.cn

***Abstract*—Dummy fill insertion is essential for CMP uniformity but remains challenging due to the nonlinear CMP process, the large optimization space, and timing degradation caused by parasitic coupling. We propose SONIC, a differentiable CMP-driven dummy fill optimization framework that employs a neural CMP simulator to directly optimize planarization objectives using gradient-based methods. SONIC further integrates a timing-aware fill insertion strategy to mitigate coupling capacitance near critical nets. Experimental results demonstrate that SONIC achieves competitive planarization quality with up to 1830× runtime speedup over a full-chip CMP simulator. Compared with the state-of-the-art model-based method, SONIC reduces height variation, line deviation, and outliers by up to 86.16%, 90.10%, and 51.61%, respectively, while achieving a 77.67% runtime reduction and lowering coupling capacitance by 13.05%.**

***Index Terms*—Chemical–Mechanical Polishing (CMP), Dummy Fill Optimization, Neural CMP Modeling, Timing-Aware Design**

I. Introduction

Chemical–mechanical polishing (CMP) is a critical manufacturing step to ensure interconnect planarity in modern integrated circuits. Dummy fill insertion is widely adopted to balance layout density and improve CMP uniformity [1], [2]. However, effective dummy fill optimization remains challenging due to the nonlinear relationship between layout density and post-CMP topography, the extremely high dimensionality of density variables in full-chip designs, and additional constraints such as timing closure. Existing dummy fill optimization methods include rule-based, model-based and neural-network-based approaches. Rule-based methods insert fills according to predefined density rules and heuristics, but lack process awareness and optimization flexibility, often resulting in suboptimal CMP uniformity [3]. Model-based methods embed simplified analytical CMP models into iterative optimization loops, however, they suffer from prohibitive runtime and poor scalability for full-chip optimization [4], [5]. Recent neural-network-based methods significantly accelerate CMP prediction [6], [7], but most of them focus on height prediction accuracy rather than directly optimizing planarization objectives, and timing impact is often ignored [8], [9]. Motivated by these limitations, we propose SONIC, a differentiable CMP-driven dummy fill optimization framework that directly optimizes planarization metrics while incorporating timing awareness. As illustrated in Fig. 1, SONIC combines a neural CMP simulator, gradient-based density optimization, and a timing-aware fill insertion strategy to enable fast, scalable, and timing-conscious dummy fill optimization for large-scale layouts.

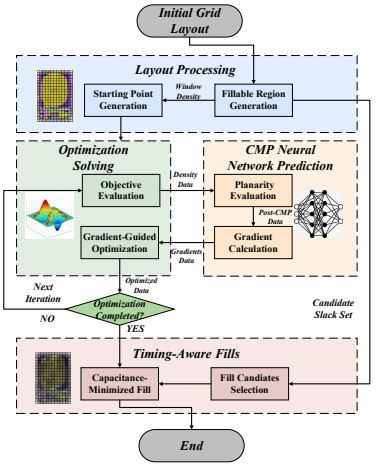

Fig. 1: SONIC framework: layouts are partitioned, fillable regions identified, optimized by neural network predictions, and final fills inserted to balance planarity and timing.

II. Framework of SONIC

A. SE-ResUNet Neural CMP Simulator

SONIC partitions the layout into fixed-size windows, each characterized by its local metal density. Dummy fill optimization is formulated as a continuous optimization problem, where window densities are adjusted to improve post-CMP surface planarity, measured by planarization-related metrics such as height variation, line deviation, and outlier, while satisfying density bounds and design rule constraints. This formulation enables SONIC to handle large-scale designs with tens of thousands of density variables in a unified optimization framework.

To efficiently evaluate CMP effects during optimization, SONIC employs a neural CMP simulator to predict post-CMP height from window density inputs. As shown in Fig. 2, the CMP simulator is implemented using an SE-ResUnet architecture that integrates residual connections, squeeze-and-excitation modules, and Sobel-gradient-based feature enhancement. The Sobel module improves sensitivity to local density transitions,

978-3-9826741-1-7/DATE26/© 2026 EDAA

while SE layers adaptively reweight channel-wise features, enabling more accurate modeling of CMP-induced height variation. This design allows the simulator to provide reliable planarization feedback during iterative optimization.

B. Gradient-Driven Optimization and Timing-Aware Fills

Unlike prior approaches that decouple CMP modeling from dummy fill optimization, SONIC tightly integrates the CMP simulator into a differentiable optimization loop. Planarization-related loss functions are evaluated on the predicted height map and backpropagated through the neural CMP simulator to compute gradients with respect to window densities. These gradients are then exploited by an L-BFGS-B solver to iteratively update density variables under bound constraints on window densities, avoiding the scalability limitations of SQP-based methods and enabling rapid convergence.

To mitigate timing degradation introduced by dummy fills, SONIC incorporates timing awareness into the fill realization stage. Critical nets are identified based on timing slack analysis, and buffer regions are enforced to prevent fill insertion near timing-sensitive signal paths. In addition, fill candidates are ranked to minimize coupling capacitance, allowing SONIC to reduce parasitic impact while preserving planarization quality. The overall optimization and fill insertion flow is summarized in Fig. 1.

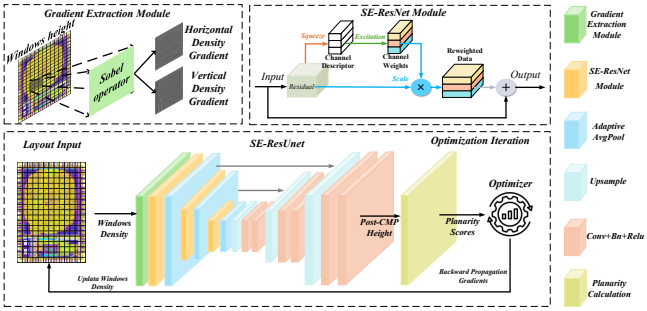

Fig. 2: SE-ResUNet takes layout density as input to predict post-CMP height, evaluates planarity metrics, and iteratively optimizes the layout.

III. Experimental Results

The SONIC framework is implemented in PyTorch and Python and evaluated on ICCAD 2014 benchmark [5] layouts using a Linux server with an Intel Xeon Gold 6346 CPU and eight NVIDIA GeForce RTX 4090 GPUs. Post-CMP height is validated using Siemens Calibre 2019, and coupling capacitance is extracted with the ICCAD 2018 Simple Cap Extractor [10]. Experiments are performed on six open-source designs from two technology nodes: three asap7 (7 nm) layouts using the first three metal layers and three SKY130 (130 nm) layouts using the first four metal layers, covering AES and RISC-V cores, cryptographic accelerators, and SoC designs.

As shown in Fig. 3, the 130,nm-trained model achieves an average relative error of 0.36%, with over 90% of windows below 0.41%. Each prediction takes only 0.0176,s, yielding up to a 1830× speedup over the full-chip CMP simulator. On unseen layouts, the model maintains an average error of 0.72%, indicating strong generalization.

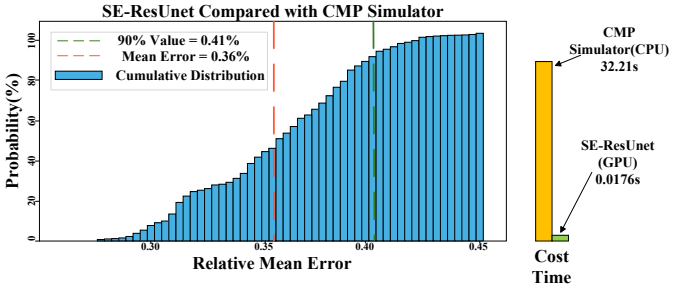

Fig. 3: SE-ResUnet Prediction Accuracy.

Training and testing datasets are generated from these layouts, and SE-ResUNet models are pre-trained using node-specific datasets and evaluated on corresponding test sets. SONIC is evaluated against rule-based and model-based dummy fill methods across all six designs. As summarized in Table I, SONIC reduces height variation, line deviation, and outliers by up to 88.94%, 95.53%, and 65.34% compared to the rule-based approach, and by 86.16%, 90.10%, and 51.61% compared to the model-based method. In addition, SONIC reduces coupling capacitance by an average of 11.64% over the rule-based method and 13.05% over the model-based method, benefiting from its timing-aware dummy fill adjustment. These results demonstrate that directly optimizing planarity metrics, while explicitly accounting for timing-critical regions, enables SONIC to improve both global and local CMP uniformity and effectively mitigate parasitic coupling across technology nodes.

TABLE I: OPTIMIZATION RESULTS ON SIX DESIGNS

Design	Method	Height Variation(Å)	Line Deviation(Å)	Outliers(Å)	Coupling Capacitance(nF)	Runtime(s)
Case A (7nm)	Rule-based [4]	613.34	158749.09	13125.16	7.13e-3	241.06
	Model-based [7]	459.73	104938.16	22806.92	8.42e-3	473.54
	SONIC	**3.93**	**1817.72**	**1.51**	**6.58e-3**	**67.88**
Case B (7nm)	Rule-based [4]	522.46	205731.31	19980.95	7.25e-3	829.33
	Model-based [7]	415.63	139092.80	31086.00	8.51e-3	4026.70
	SONIC	**3.75**	**2667.31**	**0.74**	**7.07e-3**	**83.84**
Case C (7nm)	Rule-based [4]	932.25	581326.13	0	1.97e-2	205.44
	Model-based [7]	800.18	506929.01	0	1.92e-2	1361.51
	SONIC	**5.54**	**4331.55**	0	**1.66e-2**	**53.28**
Case D (130nm)	Rule-based [4]	311.12	291572.89	16696.83	41.68	**282.79**
	Model-based [7]	239.51	233899.41	**3130.00**	43.00	826.41
	SONIC	**66.47**	**24490.67**	6467.15	**37.58**	400.64
Case E (130nm)	Rule-based [4]	151.23	92997.35	16797.54	423.98	3186.24
	Model-based [7]	149.89	84023.67	18229.49	419.65	2313.06
	SONIC	**43.15**	**20092.57**	**4422.96**	**415.16**	**672.11**
Case F (130nm)	Rule-based [4]	405.94	2321019.48	71869.02	1551.42	6301.88
	Model-based [7]	244.98	636070.48	213216.60	1203.20	8200.39
	SONIC	**58.94**	**130329.38**	**30835.48**	**1054.87**	**2963.14**

IV. Conclusion

This paper presents SONIC, a differentiable CMP-driven dummy fill optimization framework that enables fast, scalable, and timing-aware layout optimization. By directly optimizing planarization metrics using a neural CMP simulator and integrating timing awareness into dummy fill insertion, SONIC achieves superior runtime efficiency and competitive planarization quality. These results demonstrate the practicality of SONIC for large-scale, timing-sensitive designs.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (62341407, 62322401 and 62406008), Beijing Municipal Science and Technology Program (Z24110000422401) and in part by the "111" Project (B18001). †Corresponding Author.

REFERENCES

[1] A. Kahng, G. Robins, A. Singh, and A. Zelikovsky, "Filling algorithms and analyses for layout density control," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 4, pp. 445–462, 1999.

[2] A. B. Kahng and K. Samadi, "Cmp fill synthesis: A survey of recent studies," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 1, pp. 3–19, 2008.

[3] Y. Lin, B. Yu, and D. Z. Pan, "High performance dummy fill insertion with coupling and uniformity constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 9, pp. 1532–1544, 2017.

[4] 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 *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2016, pp. 1–8.

[5] R. O. Topaloglu, "Iccad-2014 cad contest in design for manufacturability flow for advanced semiconductor nodes and benchmark suite," in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2014, pp. 367–368.

[6] J. Cai, C. Yan, Y. Ma, B. Yu, D. Zhou, and X. Zeng, "Neurfill: Migrating full-chip cmp simulators to neural networks for model-based dummy filling synthesis," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*, 2021, pp. 187–192.

[7] Z. Chen, J. Cai, C. Yan, Z. Bi, Y. Ma, B. Yu, W. Hu, D. Zhou, and X. Zeng, "pneurfill: Enhanced neural network model-based dummy filling synthesis with perimeter adjustment," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 43, no. 2, pp. 667–680, 2024.

[8] B. Jiang, X. Zhang, R. Chen, G. Chen, P. Tu, W. Li, E. F. Young, and B. Yu, "Fit: Fill insertion considering timing," in *2019 56th ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–6.

[9] X. Bai, Z. Zhu, P. Li, J. Chen, T. Lan, X. Li, J. Yu, W. Zhu, and Y.-W. Chang, "Timing-aware fill insertions with design-rule and density constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 10, pp. 3529–3542, 2022.

[10] 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://iccadcontest.org/2018/Problem_C/2018ICCADContest_ProblemC.pdf