

GRAIN: A Design-Intent-Driven Analog Layout Migration Framework

Bingyang Liu^{*†}, Haoning Jiang[‡], Haoyi Zhang^{*}, Xiaohan Gao^{*†}, Zichen Kong^{*},
Xiyuan Tang^{*}, David Z. Pan[†], Yibo Lin^{*}

^{*}Peking University, Beijing, China

[†]UT Austin, Austin, TX, USA

[‡]SUSTech, Shenzhen, China

Abstract—Migrating a validated analog layout across technology nodes remains labor-intensive. Recent automatic migration methods often miss multi-level design intent embedded in expert layouts and may suffer from routing-induced LVS violations and unstable placement behaviors. We present GRAIN, a design-intent-driven analog layout migration framework that performs constraint-aware hierarchical placement migration to preserve multi-level placement behaviors, and uses guide-based routing that decouples similarity from legality via a maze router to reliably produce LVS-clean layouts. Experiments on real designs migrated from 65 nm to 40 nm and 28 nm show that, compared to a recent representative analog layout migration framework, GRAIN delivers 100% LVS-clean layouts without manual fixes and reduces area and wirelength by 13.8% and 29.2% on average, while also yielding post-layout metrics closer to the schematic.

I. INTRODUCTION

The continuous scaling of semiconductor manufacturing processes has amplified the demand for efficient analog layout migration. Migration transfers an existing, validated layout to a new process while preserving performance and the original design intent. This is especially challenging for analog circuits, whose behavior is highly sensitive to device matching, parasitics, and other layout-dependent effects. As a result, industrial migration flows remain largely manual and labor-intensive.

Automated analog layout migration has been explored, but existing methods still have shortcomings. Early template-based methods [1], [2] cannot capture realistic routing behavior and layout-dependent effects [3]. More recent works have widely adopted constraint-graph (CG) [3]–[6] and topology slicing tree [7], [8] in placement migration. For routing migration, heuristic methods including Constrained Delaunay Triangulation [4] and Cartesian Detection Lines [3], [5] have also been proposed. These approaches share two key issues: (i) CG-based placement is sensitive to graph construction order, thus unstable; (ii) existing routing migration methods frequently fail Layout-Versus-Schematic (LVS) check. Both the issues make it difficult for subsequent refinements after the automatic migration.

To address these challenges, we present GRAIN, a design-intent-driven analog layout migration framework that aims to make migration stable, LVS-clean, and intent-preserving. GRAIN combines a hierarchical clustering algorithm for constraint-aware placement migration, and a guide-based routing algorithm for routing migration. An experiment on real analog designs migrated from 65 nm to 40 nm and 28 nm,

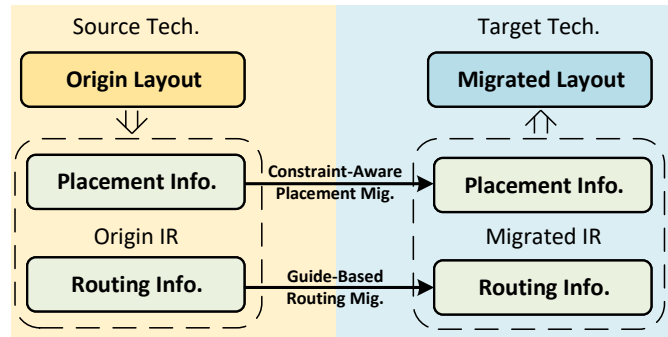


Fig. 1. Overview of the GRAIN framework. Our algorithms extract multi-level design intent from the source layout, progressively building an intermediate representation (IR). This abstracted intent is then migrated to the target technology, and the final layout is reconstructed from the migrated IR.

demonstrates 100% LVS-clean results without manual fixes, 13.8% and 29.2% average reduction on area and wirelength, respectively, and smaller schematic-to-layout performance degradation under post-layout simulation.

II. GRAIN FRAMEWORK

The GRAIN framework facilitates analog layout migration by preserving multi-level design intent through an explicit Intermediate Representation (IR). Instead of manipulating fine-grained GDSII polygons directly, GRAIN extracts design intent into an IR, migrates this abstraction to the target technology, and finally reconstructs the layout. This IR serves as a bridge, carrying placement constraints, including symmetry and well-sharing, and routing topologies across migration.

A. Constraint-aware Placement Migration

To overcome the instability of traditional methods, we employ a hierarchical strategy designed to preserve design intent in the migration process, including symmetry, alignment, and well-sharing behaviors. The first step is constraint-aware clustering. We utilize the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [9] to capture spatial patterns such as dense device groups and reserved guard ring regions. Rather than its traditional use for noise filtering, DBSCAN is specifically leveraged here for its ability to represent irregular clusters. This is critical for analog layouts where well-sharing and guard ring behaviors often manifest as non-convex, density-dependent shapes that standard rectangular partitioning cannot capture. We define a precomputed pairwise distance matrix where the spatial proximity between devices is weighted by a constraint strength coefficient, so that strongly constrained pairs are effectively pulled closer in the clustering space. This ensures that devices sharing strong

B.Liu and X.Gao was with Peking University when this work was conducted. They are now with the University of Texas at Austin.

Corresponding authors: Yibo Lin (yibolin@pku.edu.cn)

design constraints are grouped together, maintaining their relative structural integrity during migration. Then a recursive hierarchical compaction is performed. The migration follows a bottom-up hierarchical process. Within each cluster, intra-cluster compaction moves devices toward the cluster center while strictly following the symmetry and alignment constraints through coordinated movement. Subsequently, inter-cluster compaction treats each cluster as a higher-level macro-block for recursive compaction. The process continues until all elements merge into a single, compact region, yielding a final migrated placement that balances design intent preserving and spatial efficiency.

B. Guide-based Routing Migration

To ensure LVS-clean reconstruction while preserving the original routing style, we introduce a guide-based approach that decouples layout similarity from LVS correctness.

The source routing is abstracted into a 3D-grid IR, where each net is decomposed into 2-pin segments connected by pins and Steiner points. Instead of migrating absolute coordinates, we record the relative position of each pin to its device bounding box. After placement migration, the Steiner points are reconstructed via an axis-aligned affine transformation. This transformation preserves the relative routing structure of the original net, forming a routing guide that represents the abstracted routing intent in the target design.

Then, the final routing is performed by a maze router using the migrated guide as a soft constraint. This allows the framework to prioritize LVS/DRC correctness while staying as close as possible to the original topology. The routing cost for a point p_i is defined as:

$$\text{Cost}(p_i) = \text{BaseCost}(p_i) + \lambda \cdot \text{Distance}(p_i, P_{\text{guide}}) \quad (1)$$

where BaseCost contains components including obstacle penalty, wire length, and other constraints [10], [11], and the penalty term $\lambda \cdot \text{Distance}$ biases the path toward the guide P_{guide} . This cost function biases the routing path toward the migrated guide while ensuring LVS correctness.

III. EXPERIMENTAL RESULTS

We evaluate the GRAIN framework by migrating two 65nm designs, a folded-cascade OTA and a comparator (COMP), to 40nm and 28nm technologies. To isolate the contributions of our algorithms, we compare GRAIN against baseline [3] and two hybrid variants: (1) [3] fix, which uses the baseline’s placement with the same maze router [10], [11] to achieve LVS-cleanliness and compare the placement migration; (2) [3] + Ours, which combines the baseline’s placement with our complete guide-based routing to compare the routing migration. Performance is measured by the degradation from schematic to post-layout simulation in the target technology. We employed the router proposed by [10], [11] as the routing engine in our framework.

The performance metrics for the migration on OTA and COMP are summarized in Table I and Table II. A main observation is that the baseline [3] consistently fails LVS checks and exhibits significant instability, yielding hundreds of varying placement results that necessitate manual filtering.

In contrast, GRAIN provides a deterministic and robust flow that is LVS-clean across all cases.

As for the performance, GRAIN prevents the performance collapse seen in other methods at the 28nm node, where parasitics dominate. GRAIN maintains critical metrics—such as CMRR, PSRR, and phase margin—closest to schematic values. Besides, GRAIN also achieves an average reduction of 13.8% in area and 29.2% in wire length compared to the baseline. The results of two hybrid baselines [3] fix and [3]+ Ours show that though the LVS issue of baseline is fixed, it cannot recover performance loss caused by suboptimal placement. These results suggest that the layouts migrated by GRAIN more closely resemble the compact and high-performance layouts achieved by manual designs.

TABLE I
OTA: GEOMETRIC METRICS AND POST-LAYOUT PERFORMANCE

Tech.	Method	Area (μm^2)	WL (μm)	LVS	Gain (dB)[Δ]	-3dB (kHz)[Δ]	CMRR (dB)	PSRR (dB)	PM ($^\circ$)[Δ]
65nm	Sch.	N/A	N/A	N/A	70.5	63.2	> 120	> 120	69.3
	Manual	2001	524	✓	62.8 (-7.7)	34.8 (-45%)	85.9	80.8	81.8 (+12.5)
40nm	Sch.	N/A	N/A	N/A	55.8	109.6	> 120	> 120	77.7
	[3]	1392	893	✗	-	-	-	-	-
	[3] fix	1392	698	✓	53.8 (-2.0)	56.9 (-48%)	62.6	77.2	81.2 (+3.4)
	[3]+Ours	1392	567	✓	54.9 (-0.9)	58.9 (-48%)	62.1	71.6	86.8 (+9.1)
	Ours	1384	521	✓	55.5 (-0.3)	67.1 (-39%)	63.1	74.9	86.7 (+8.6)
28nm	Sch.	N/A	N/A	N/A	69.5	273.3	> 120	> 120	68.9
	[3]	520	573	✗	-	-	-	-	-
	[3] fix	520	434	✓	27.1 (-42)	1112 (+307%)	38.4	26.8	50.9 (-18.0)
	[1]+Ours	520	402	✓	41.2 (-28)	1318 (+383%)	38.1	35.0	31.7 (-37.2)
	Ours	362	306	✓	58.8 (-10.8)	224.6 (+17.8%)	44.1	38.9	78.9 (+9.3)

Notes: (i) N/A means not applicable to schematic; (ii) - means unavailable due to LVS check failure; (iii) > 120 dB means CMRR/PSRR reaching the upper limit of simulation accuracy.

TABLE II
COMP: GEOMETRIC METRICS AND POST-LAYOUT PERFORMANCE

Tech.	Method	Area (μm^2)	WL (μm)	LVS	t_r (ns)[Δ]	t_f (ns)[Δ]	V_{os}/V_{FS} (%) [Δ]	V_{hys}/V_{FS} (%) [Δ]
65nm	Sch.	N/A	N/A	N/A	19.8	8.25	0.92	5.48
	Manual	262.3	124.8	✓	20.3 (+0.5)	10.1 (+1.9)	1.49 (+0.6)	5.65 (+0.2)
40nm	Sch.	N/A	N/A	N/A	16.0	8.11	1.02	5.96
	[3]	166.0	260.6	✗	-	-	-	-
	[3] fix	166.0	208.4	✓	18.5 (+2.5)	9.77 (+1.7)	1.65 (+0.6)	7.93 (+2.0)
	[3]+Ours	166.0	154.3	✓	18.6 (+2.6)	9.54 (+1.4)	1.65 (+0.6)	8.19 (+2.2)
Ours	136.0	146.6	✓	18.5 (+2.5)	9.23 (+1.1)	1.60 (+0.6)	6.80 (+0.8)	
28nm	Sch.	N/A	N/A	N/A	3.99	2.17	0.14	5.60
	[3]	115.6	182.4	✗	-	-	-	-
	[3] fix	115.6	166.9	✓	4.77 (+0.8)	2.91 (+0.7)	0.60 (+0.5)	4.24 (+1.4)
	[3]+Ours	115.6	128.7	✓	4.58 (+0.6)	2.87 (+0.7)	0.39 (+0.3)	3.45 (+2.2)
	Ours	77.14	107.8	✓	4.42 (+0.4)	2.80 (+0.6)	0.38 (+0.2)	4.25 (+1.4)

Notes: See shared notes under Table I.

IV. CONCLUSION

In this paper, we presented GRAIN, a design-intent-driven analog layout migration framework that achieves 100% LVS-clean migrations and improves geometric compactness and post-layout performances closer to the schematic across real 65 nm to 40 nm and 28 nm designs. Our evaluation currently covers a small set of designs; future work will extend GRAIN to larger benchmarks and additional technologies to further improve robustness.

REFERENCES

- [1] S. Hammouda *et al.*, “Chameleon art: A non-optimization based analog design migration framework,” in *Proceedings of the 43rd annual Design Automation Conference*, 2006, pp. 885–888.
- [2] S. H. Batterywala *et al.*, “Cell swapping based migration methodology for analog and custom layouts,” in *9th International Symposium on Quality Electronic Design (isqed 2008)*. IEEE, 2008, pp. 450–455.
- [3] H.-Y. Chi *et al.*, “A style-based analog layout migration technique with complete routing behavior preservation,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, pp. 2556–2567, 2021.
- [4] P.-C. Pan *et al.*, “A fast prototyping framework for analog layout migration with planar preservation,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, pp. 1373–1386, 2015.
- [5] H.-Y. Chi *et al.*, “Achieving routing integrity in analog layout migration via cartesian detection lines,” in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2019, pp. 1–6.
- [6] M. P.-H. Lin *et al.*, “Achieving analog layout integrity through learning and migration,” in *Proceedings of the 39th International Conference on Computer-Aided Design*, 2020, pp. 1–8.
- [7] Y.-P. Weng *et al.*, “Fast analog layout prototyping for nanometer design migration,” in *2011 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2011, pp. 517–522.
- [8] M. P.-H. Lin *et al.*, “Augmenting slicing trees for analog placement,” in *2012 International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. IEEE, 2012, pp. 57–60.
- [9] M. Ester *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.
- [10] H. Zhang *et al.*, “Sageroute: Synergistic analog routing considering geometric and electrical constraints with manual design compatibility,” in *DATE*, 2023, pp. 1–6.
- [11] H. Zhang *et al.*, “Sageroute2.0: Hierarchical analog and mixed signal routing considering versatile routing scenarios,” in *DATE*, 2024, pp. 1–6.