

# Breaking Standard Cell Margin Constraints for Area-Efficient VLSI Design

Junghyun Yoon<sup>1</sup>, Jooyeon Jeong<sup>2</sup>, and Heechun Park<sup>1</sup>

<sup>1</sup>Department of Electrical Engineering, Ulsan National Institute of Science and Technology (UNIST), Ulsan, Korea

<sup>2</sup>Department of Electrical and Computer Engineering, University of California, Los Angeles, CA, USA  
yoon\_jh@unist.ac.kr, jooyeon37@ucla.edu, h.park@unist.ac.kr

**Abstract**—In standard-cell-based VLSI design, fixed margins at cell boundaries are necessary to prevent short violations between adjacent transistors carrying different signals. However, these margins are redundant for most abutted cell pairs and incur non-negligible area overhead when accumulated across the chip. In this paper, we present a novel VLSI design optimization framework that eliminates redundant margins by strategically merging adjacent cells into *margin-free cells* (MF-cells), which preserve the same functionality with reduced area due to the removal of inter-cell margins. Precisely, we identify optimal cell pairs for merging from an initial standard-cell-based placement using a maximum weighted matching (MWM) algorithm. Each identified pair is replaced with an MF-cell and placed at an optimal position using a placement algorithm that minimizes wirelength and routing congestion. Compared to the conventional standard-cell-based design, we achieve on average 3.9% reduction in total cell area and 4.7% reduction in full chip area, leading to 2.7% reduction in total wire length and 2.1% improvement in timing performance while maintaining comparable power consumption. Our framework is a practical approach to achieve meaningful area and timing improvements, which is fully compatible with commercial standard-cell-based VLSI design flow.

## I. INTRODUCTION

Standard-cell-based design is the most widely used approach in modern VLSI implementation, using pre-defined logic gates (*standard cells*) to build various functions efficiently. Optimizing standard cells is essential for improving full-chip power-performance-area (PPA) in this design paradigm. Since a single chip contains hundreds of thousands to millions of these cells, even small optimizations in individual cells can significantly impact overall chip efficiency. For example, the contact-over-active-gate (COAG) technique reduces cell footprint by allowing gate contacts in the active region, eliminating unnecessary connections and routing overhead [1]–[3]. Moreover, the required diffusion breaks between different signals have been reduced from two contact poly pitches (CPPs) in double diffusion break (DDB) to one CPP with single diffusion break (SDB) technique, which reduces the cell footprint and entire chip area. In the advanced process technology, the concept of buried power rail (BPR) has been introduced [4], [5] to place the power and ground rails below the device layer, which can utilize the metal tracks as in-cell routing resources or reduce the standard cell height [6].

However, these techniques focused on optimizing individual cells and did not address the constraint *between* standard cells, i.e., inter-cell boundaries. The conventional design rule requires a fixed margin between any two standard cells (e.g., 2

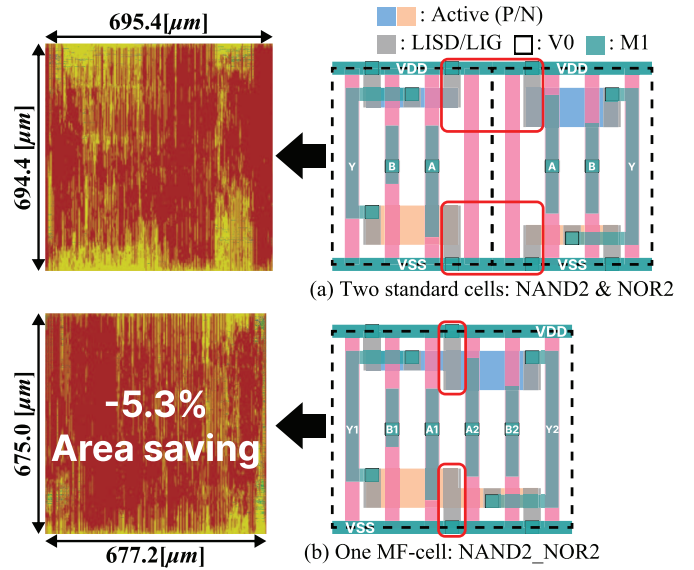


Fig. 1: (a) Two independent NAND2 and NOR2 gates placed adjacent with DDB margins, occupying 8 CPPs total. (b) The corresponding MF-cell merges both gates with diffusion chaining, requiring only 6 CPPs while maintaining functionality. Applying MF-cells to TATE benchmark achieves 5.3% chip area saving.

CPPs for DDB), regardless of the actual layout of these cells. While these fixed margins allow flexible cell placement in the VLSI design flow, they add up to significant area waste across millions of cells on a chip.

The key insight is that when two cells share the same power and ground connections, this spacing becomes unnecessary—the transistors can be directly connected through shared diffusion, removing the margin between cells completely. Fig. 1(a) shows two adjacent cells taking up 8 CPPs with DDB margins. We propose **margin-free cells** (MF-cells) that merge adjacent cells by connecting their power and ground through shared diffusion, removing the diffusion breaks between them while keeping the same function, as shown in Fig. 1(b). This approach reduces the area to 6 CPPs, achieving a 25% area saving. Required MF-cells can be generated instantly, thanks to the state-of-the-art automatic standard cell generator (e.g., [7]–[10]).

In this paper, we present a VLSI placement framework that utilizes MF-cells to optimize chip design. Starting from a high-utilization placement, we identify optimal cell pairs to

merge and replace them with MF-cells. This approach reduces the total cell area within the compact floorplan, achieving the desired utilization while shortening wires and improving timing performance. Our key contributions are as follows:

- We introduce the concept of MF-cells for VLSI design, which eliminates redundant inter-cell margins.
- We develop an efficient placement framework integrating: (1) a maximum weighted matching (MWM) [11] algorithm to identify optimal cell pairs for merging and (2) a congestion-aware MF-cell placement algorithm.
- We demonstrate through comprehensive experiments that our approach achieves chip area and wirelength reduction while improving timing performance and maintaining power consumption.

## II. RELATED WORKS AND MOTIVATION

### A. Physical Design-Level Optimization

While various optimization approaches have been proposed at the synthesis and library level [12]–[15], they did not address the fundamental area inefficiencies that emerge during physical design. These inefficiencies primarily arise at cell boundaries, where mandatory design rule margins between adjacent cells consume chip area. Recently, transistor-level approaches [16], [17] maximize diffusion sharing without standard cells, achieving significant area reduction but facing severe scalability challenges due to design complexity. The work in [18] attempted to merge adjacent cells (called *megacells* in the literature) but could only create a few hand-designed types using simple heuristic clustering, limiting its applicability to various cell merging cases, circuit-specific patterns, and lacking systematic optimization.

### B. Automated Cell Generation: Enabling MF-Cell Approach

Automatic standard cell generators now enable physical design optimization through design technology co-optimization (DTCO). The work in [7] created automatic cell layout generation for the ASAP7 [19] PDK using satisfiability modulo theories (SMT)-based transistor placement and routing. Recently, [8] achieved 95-98% routing success rates using dynamic programming and SMT solvers, while [9] focused on runtime efficiency through heuristic approaches. These tools eliminate the bottleneck of manual standard cell design by enabling automated standard cell generation during physical design, which is a key enabler for our MF-cell approach.

Previous cell merging approaches [12]–[15] modify logic synthesis or early physical design, restricting design freedom in place-and-route (P&R) and degrading quality. In contrast, our MF-cell-based design methodology addresses the margin removal problem at the **physical design stage after placement** to maintain the flexibility and scalability of the conventional standard-cell-based design flow and achieve area savings by removing unnecessary inter-cell margins based on actual placement adjacency.

Addressing the margin elimination problem at the physical design stage rather than at the logic level enables two operational modes: on-demand MF-cell generation after placement and the use of pre-defined MF-cells for frequently occurring

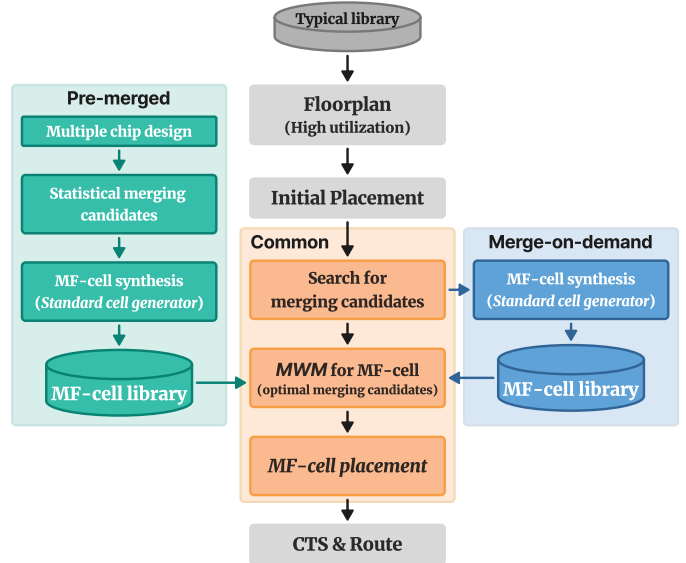


Fig. 2: Proposed MF-cell-based placement optimization framework.

cell pairs. The integration of a fully automated standard cell generator into the framework allows these MF-cells to be created instantly, supporting fast runtime and strong scalability for large-scale designs.

## III. MF-CELL-BASED PLACEMENT OPTIMIZATION

### A. Overview

In this section, we propose two MF-cell-based placement optimization frameworks as illustrated in Fig. 2. The primary framework leverages automatic standard cell generators to enable *merge-on-demand* capability. This framework collects information about adjacent cell pairs from the initial placement results of the target design and generates corresponding MF-cells for frequently occurring pairs that offer area savings. Based on the generated MF-cell library, optimal merging candidates are identified from the adjacent cell pairs to maximize MF-cell integration.

For design environments where the cell library must be pre-defined, frequently occurring adjacent cell pairs are statistically analyzed across multiple benchmark designs. Initial floorplan and placement are conducted for several benchmark designs with a typical standard cell library to collect statistical data on these adjacent pairs. The pairs that save area when merged are selected, and an MF-cell is designed for each pair to construct an MF-cell library in advance. In the main design flow for the target benchmark, the *pre-merged* MF-cell library is applied from the stage of identifying optimal merging candidates.

Both frameworks begin with a high-utilization placement and selectively replace adjacent cell pairs with MF-cells. This replacement significantly reduces total cell area by eliminating redundant inter-cell margins, enabling the desired design density within a smaller chip area while preserving design quality. The key difference lies in when and how the MF-cell library is constructed: *merge-on-demand* generates MF-cells during the

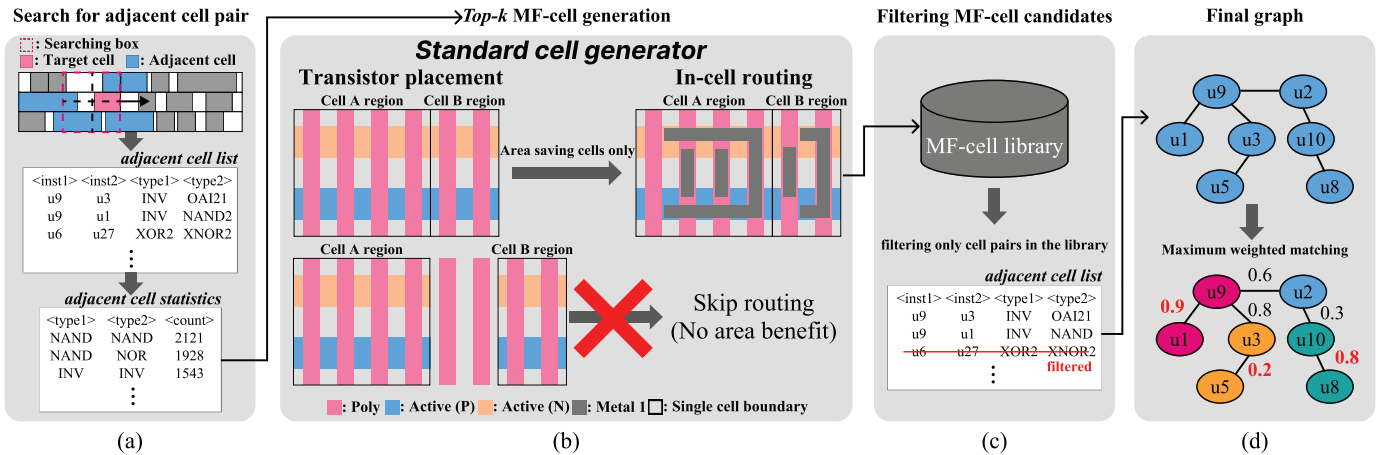


Fig. 3: The merge-on-demand framework finds merging candidates by (a) searching for adjacent cells, (b) generating MF-cells for top- $k$  frequent cell pairs, (c) creating an MF-cell library, and (d) constructing an adjacency graph with weighted edges and applying the MWM algorithm to select optimal cell pairs, where red numbers indicate selected pairs. The pre-merged framework conducts (b) before (a) using statistical merging candidate data.

TABLE I: Comparison between standard cells (top) and MF-cells (bottom). #CPP, total WL, M2, aRPA, wRPA, and IOC indicate the number of contact poly pitches, total wire length ( $\mu\text{m}$ ), M2 wire length, average remaining pin access, worst-case remaining pin access, and inaccessibility of cell, respectively. For MF-cells, #CPP column shows “ $X \rightarrow Y$ ” where  $X$  is the sum of CPPs from both cells (i.e., without merging) and  $Y$  is the reduced value after merging. RPA and IOC are defined in [20].

Cell Type	WL	M2	#CPP	aRPA	wRPA	IOC
INVx1	0.171	0	3	2.75	2.5	0
INVxp33	0.270	0	3	3.33	1.67	0
NAND2xp33	0.306	0	4	3.07	1.2	0
NOR2xp33	0.288	0	4	3.07	1.2	0
MF-cells (Merged Cell Combinations)						
INVx1 + INVxp33	0.386	0	6 $\rightarrow$ 4	3.08	2.0	0
INVxp33 + INVxp33	1.016	0	6 $\rightarrow$ 4	3.8	2.6	0
INVxp33 + NAND2xp33	0.767	0	7 $\rightarrow$ 5	3.68	2.53	0
INVxp33 + NOR2xp33	0.767	0	7 $\rightarrow$ 5	3.68	2.53	0
NAND2xp33 + NAND2xp33	0.572	0	8 $\rightarrow$ 6	3.51	2.53	0
NAND2xp33 + NOR2xp33	0.572	0	8 $\rightarrow$ 6	3.51	2.53	0
NOR2xp33 + NOR2xp33	0.572	0	8 $\rightarrow$ 6	3.47	3.0	0

target design optimization, while pre-merged uses a pre-built library from cross-benchmark analysis.

### B. MF-cell Synthesis

The main idea of the MF-cell-based placement lies in the generation of optimized MF-cells, which is made possible by recent advancements in automatic standard cell generation. In brief, a heuristic-based standard cell generator based on the approach presented in [9] is implemented, adapting it for MF-cells by treating each MF-cell as a complex standard cell and generating it either on-demand or pre-merged. The MF-cell generator operates in two main stages: transistor placement and in-cell routing. During transistor placement, the generator optimizes the arrangement of transistors to minimize area while meeting design rules. The in-cell routing stage then connects these transistors using available metal layers.

Since two cells are merged into a single unit without margins, a higher density of pins is concentrated within a

relatively smaller area, potentially creating pin accessibility issues. The work in [9] addressed pin accessibility problems by adjusting pin sizes, observing that smaller pin sizes lead to worse pin accessibility. They introduced a methodology that maximizes pin sizes even at the cost of slightly increased total wire length. This approach is adopted to ensure that pin accessibility issues do not occur in the generated MF-cells. Table I shows a comparison between typical cells and MF-cells generated by the standard cell generator. All MF-cells have an inaccessibility of cell (IOC) [20] value of 0, confirming no pin accessibility issues in this approach. Additionally, each MF-cell’s CPP (i.e., cell width) is smaller than the combined CPPs of two separate typical cells.

In the MF-cell-based placement frameworks, the MF-cell generator receives a list of adjacent cell pairs either from the target design (merge-on-demand framework) or from the statistical results of several benchmarks (pre-merged framework) as inputs, and performs transistor placement for each pair. Cells with area savings compared to two separated cells proceed to the in-cell routing stage to generate final layouts. The library generation stages then execute including DRC, LVS, and PEX verification, characterization to generate Liberty files (i.e., delay and power information), and abstract generation for library exchange format (LEF) files, which form the complete MF-cell library.

### C. Search for the MF-cell Candidates

Fig. 3 illustrates the comprehensive process of finding merging candidates for MF-cells from the initial placement result with standard cells. The process consists of four stages to systematically identify optimal cell pairs for merging.

The first stage involves collecting adjacent cell pairs from the initial placement, as shown in Fig. 3(a). For each target cell, a user-specified bounding box is defined and all cells within this region are collected to identify potential merging candidates. The bounding box size is typically set to be less than the size of the flip-flop (e.g., 26 CPPs for ASAP7 PDK)

to preserve the placement order and minimize disruption. Flip-flops are excluded from merging candidates, as their positional displacements significantly impact timing paths. After collecting all adjacent cells within the bounding boxes, comprehensive statistics on the frequency of each cell pair appearing in the design are compiled.

In the second stage, illustrated in Fig. 3(b), the top- $k$  most frequently occurring cell pairs ( $k=100$  in the implementation) are selected based solely on their frequency of appearance and processed through the standard cell generator. The generator attempts to merge each selected cell pair at the netlist level, performing transistor placement to determine feasibility. During this transistor placement stage, pairs that would not provide area savings when merged can be immediately identified. Only those pairs that achieve margin reduction and demonstrate area benefits proceed to the in-cell routing stage, as pairs without area savings cannot form true MF-cells. Note that this stage is specific to the merge-on-demand framework. In the case of the pre-merged framework, this stage is performed beforehand without access to placement data, relying instead on statistically extracted merging candidate information.

The third stage, depicted in Fig. 3(c), involves creating the MF-cell library. After completing the in-cell routing for area saving cell pairs, the complete library containing all successfully created MF-cells is generated. This library serves as the reference for the subsequent filtering process. The initial merging candidates collected in the first stage (Fig. 3(a)) are then revisited and filtered to retain only those pairs for which MF-cells were successfully generated in the library. This filtering ensures that only feasible merging candidates are considered in the final optimization.

The final stage, shown in Fig. 3(d), constructs an adjacency graph to determine the optimal merging strategy. In this graph representation, vertices correspond to standard cells, while edges connect cells that can potentially be merged into MF-cells. The edge weight calculation considers the cell-to-cell distance and the presence of connectivity between the cells, ensuring that closely placed and connected cells receive higher weights. Specifically, the weight is calculated based on an inverse relationship with the Manhattan distance between cells, normalized by the maximum possible distance within the search region, to encourage physically adjacent pairs for merging. Additionally, when two cells are positioned on the same row, the weight is further increased to reflect the increased feasibility and benefits of merging cells that share the same horizontal placement level. Furthermore, when two cells have direct connections through shared nets, an additional connectivity bonus is added to favor connected pairs. The edge weight can be formulated as:

$$w_{i,j} = \left(1 - \frac{d_{\text{Manhattan}}(i,j)}{d_{\text{max}}}\right) \times \alpha_{\text{same\_row}} + \beta_{\text{connected}}, \quad (1)$$

where  $d_{\text{Manhattan}}(i,j)$  is the Manhattan distance between cells  $i$  and  $j$ ,  $d_{\text{max}}$  is the maximum distance within the search region,  $\alpha_{\text{same\_row}}$  is applied when both cells are on the same row (otherwise 1), and  $\beta_{\text{connected}}$  is a connectivity bonus

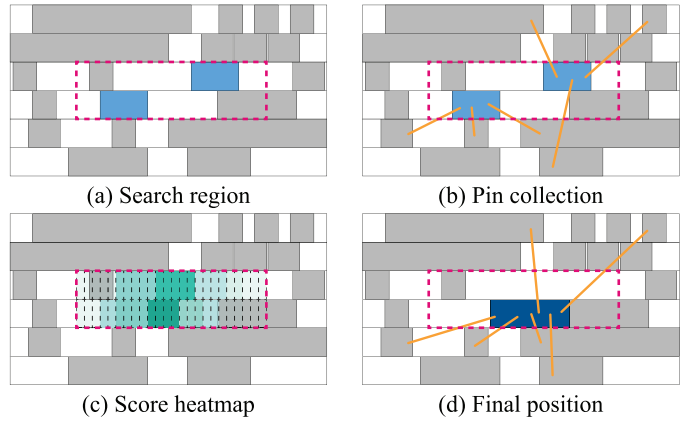


Fig. 4: MF-cell placement optimization process: (a) Search region determination based on the bounding box of two cells to be merged. (b) Collection of all pins connected to either cell for HPWL calculation. (c) Evaluation of all site-grid positions within the search region, with darker colors indicating lower objective scores (better positions). (d) Final placement of the MF-cell at the optimal position with the lowest objective score.

added when the two cells share nets. In our implementation,  $\alpha_{\text{same\_row}} = 10$  and  $\beta_{\text{connected}} = 1$ .

Once the weighted graph is constructed, the MWM algorithm is applied to identify the optimal set of merging candidates. The selected pairs (e.g., the red numbers in Fig. 3(d)) represent the final merging candidates that will proceed to the MF-cell placement stage.

#### D. MF-cell Placement

Before converting the optimal merging candidates to MF-cells, additional filtering constraints are applied to ensure robust physical implementation. Pairs in which the total number of pins connected to the two cells exceeds a threshold (15 in our experiments) are excluded from merging, to prevent timing degradation due to excessive fanout and increased wirelength.

The algorithm then proceeds to determine the best placement position for each remaining MF-cell. The placement algorithm minimizes wirelength and routing congestion while preserving the quality of the initial placement. As illustrated in Fig. 4(a), the search region for each MF-cell is defined as the bounding box enclosing both original cells with an additional margin to accommodate the MF-cell dimensions. By constraining the search within this region, excessive displacement that could degrade the placement quality is prevented while maintaining spatial locality. The algorithm then evaluates candidate positions within this region at site-grid resolution—the discrete grid points where cells can be legally placed—ensuring that all potential positions align with the placement requirements.

For each candidate position, an objective function that incorporates three critical factors is evaluated: wirelength, routing congestion, and cell overlap. The wirelength component is calculated using half-perimeter wirelength (HPWL). As shown in Fig. 4(b), since an MF-cell merges two original cells into one, all pins connected to either original cell are collected to compute the HPWL for each candidate position. The routing congestion is estimated using the RUDY (Rectangular Uniform

wire DensitY) [21] metric. The overlap penalty ensures that MF-cells avoid overlapping with existing cells, maintaining the legality of the placement. In summary, the objective function for determining the optimal position  $(x_{opt}, y_{opt})$  is formulated as:

$$(x_{opt}, y_{opt}) = \arg \min_{(x,y) \in S} (w_{hpwl} \cdot \text{HPWL}(x, y) + w_{rudy} \cdot \text{RUDY}(x, y) + w_{overlap} \cdot \text{Overlap}(x, y)), \quad (2)$$

where  $S$  represents the set of valid site-grid positions within the search region. The weight factors are set as  $w_{hpwl} = 8$ ,  $w_{rudy} = 2$ , and  $w_{overlap} = 20$  based on empirical tuning. These weights reflect the relative importance of each component, with the values normalized to account for their different scales.

The optimization uses an exhaustive search algorithm that evaluates all candidate positions within the search region. As visualized in Fig. 4(c), each site-grid position is evaluated and assigned a score. While this exhaustive search might seem computationally intensive, the constrained search space and site-grid discretization keep the number of candidates manageable. The algorithm selects the position with the minimum cost, as shown in Fig. 4(d), where the MF-cell is placed at the optimal location with the lowest score. This deterministic approach guarantees finding the optimal solution within the defined search space, ensuring consistent and reliable results.

#### E. CTS & Route

After generating the placement result with MF-cells, any remaining cell overlaps are resolved using the EDA tool’s refine placement capability to ensure placement legality. The design then proceeds with the remaining clock tree synthesis (CTS) and routing stages as in the conventional physical design flow. The MF-cell placement maintains compatibility with standard EDA tools, allowing seamless integration into existing design flows. Through this approach, the framework achieves area reduction with additional benefits in timing performance while maintaining power efficiency comparable to conventional physical design flow.

### IV. EXPERIMENTAL RESULTS

#### A. Experimental Setup

Our MF-cell-based placement optimization framework leverages commercial EDA tools for the main flow (Synopsys Design Compiler for logic synthesis and Cadence Innovus for P&R), using the open-source ASAP7 [19] PDK, and benchmark circuits from OpenCores [22]. The standard cell generator is implemented in Python, which has been refined to also work as an MF-cell generator, and we additionally apply adjustments (e.g., transistor folding) to the result. For library generation and verification, we use Siemens EDA Calibre for DRC/LVS/PEX, Synopsys PrimeLib for characterization to generate Liberty files, and Cadence Abstract Generator for LEF files. Here we generate new SPICE netlists for MF-cells that reflect their internal structures to perform LVS verification, and **all generated MF-cells successfully passed DRC/LVS**

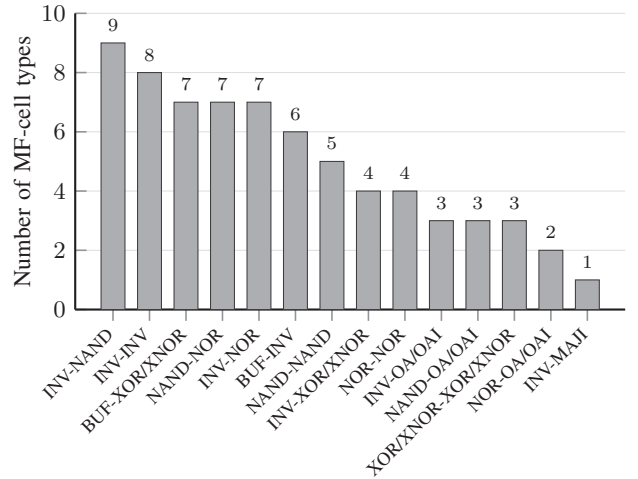


Fig. 5: Distribution of cell type combinations in the MF-cell library sorted by frequency of occurrence (total: 69 MF-cell types). The data is collected from the top- $k$  appearance lists of seven OpenCores benchmarks (vga\_lcd, nova, ldpc, jpeg, aes128, Tate, ecg).<sup>1</sup>

**using the verification flow** described in Section III-B. In the MF-cell placement stage, the MWM algorithm for finding merging candidates is implemented in C++, and our MF-cell placement optimization algorithm is implemented in Python. All interfaces between our algorithm and the commercial CAD tools are implemented in Tcl scripts.

For the MF-cell library preparation in the pre-merged framework, we collect adjacent cell pair lists from the placement results of seven benchmarks (vga, nova, ldpc, jpeg, aes128, Tate, ecg). We then select cell pairs that are included in the top- $k$  ( $k=100$  in our work) appearance lists for more than or equal to four benchmark designs, and generate the corresponding MF-cells to construct the MF-cell library. Fig. 5 shows statistics for cell pairs that appeared in the top- $k$  appearance list of seven benchmarks. We observe that INV-based combinations are the most frequent (24 combinations), followed by NAND-based combinations (14 combinations). This suggests that INV, NAND, and their combinations with other cell types are most commonly placed adjacent to each other in typical designs, making them promising candidates for MF-cell merging.

#### B. Assessments of MF-cells

Table II demonstrates the effectiveness of the MF-cell approach by comparing the characteristics of individual cells versus their behavior when integrated into MF-cells. MF-cells achieve 23-26.6% area reduction by eliminating redundant margins between adjacent cells. Furthermore, this area benefit comes with minimal performance impact. Timing variations remain under 1% , while power consumption often improves, with INV-based and NAND-based MF-cells reducing power

<sup>1</sup>The figure shows statistics for cell type combinations only. In practice, each combination (e.g., INV-NAND) includes multiple distinct MF-cells with different drive strengths (e.g., INVx1-NANDx1, INVx2-NANDx1, INVx1-NANDx2, etc.).

TABLE II: Comparison between standard cells within MF-cells and when used individually.

MF-cell type	Reference	Timing [ps]			Power [nW]			Area
		Alone	MF-Cell	Avg. Change	Alone	MF-Cell	Avg. Change	Avg. Change
INVxp33 + INVxp33	INVxp33		38.21			98.55		
INVxp33 + NAND2xp33			38.26			96.41		
INVxp33 + NOR2xp33		38.06	38.22	+0.22↑	102.4	97.59	-3.38↓	-26.6%↓
INVxp33 + OAI21xp33			38.34			101.00		
INVxp33 + XNOR2xp5			38.39			98.75		
NAND2xp33 + NAND2xp33	NAND2xp33		40.52			169.6		
NAND2xp33 + OAI1xp33			40.44			167.2		
NAND2xp33 + NOR3xp33		40.51	40.51	-0.03↓	169.7	170.8	-1.00↓	-23.0%↓
NAND2xp33 + OAI22xp33			40.46			167.2		
NOR2xp33 + NOR2xp33	NOR2xp33		39.19			203.5		
NOR2xp33 + NOR2xp67		39.14	39.18	+0.05↑	194.5	189.8	+1.60↑	-23.3%↓
NOR2xp33 + NOR3xp33			39.21			195.0		

TABLE III: Performance, power, and area comparison of typical standard-cell placement versus MF-cell-based pre-merged and merge-on-demand frameworks.

Design		Typical	Pre-merged	Merge-on-demand
ECG at 1.43GHz	#Total cells	84026	75151 (-10.6%)	66309 (-21.1%)
	#MF-cells	0	8545	17323
	Chip area [ $\mu\text{m}^2$ ]	222916.1	213732.4 (-4.1%)	208045.3 (-6.7%)
	Std. cell area [ $\mu\text{m}^2$ ]	150927.3	146695.1 (-2.8%)	142766.4 (-5.4%)
	Density	68	69	68
	WL [ $\mu\text{m}$ ]	1656200.7	1600414.1 (-3.4%)	1562470.7 (-5.7%)
	#DRV	6	2	3
	WNS [ps]	-69.02	-56.58	-35.84
	TNS [ns]	-21.11	-21.47 (+2%)	-6.17 (-71%)
	Eff. freq [GHz]	1.30	1.32 (+2%)	1.36 (+5%)
	Cell Power [mW]	25.11	24.98 (-0.5%)	24.96 (-0.6%)
Net Power [mW]	11.98	11.78 (-1.7%)	11.89 (-0.8%)	
Total Power [mW]	37.10	36.77 (-0.9%)	36.86 (-0.6%)	
TATE at 0.83GHz	#Total cells	189628	166341 (-12.3%)	146287 (-22.9%)
	#MF-cells	0	22975	42986
	Chip area [ $\mu\text{m}^2$ ]	482896.9	469630.3 (-2.7%)	457131.6 (-5.3%)
	Std. cell area [ $\mu\text{m}^2$ ]	333870.3	322907.3 (-3.3%)	313848.1 (-6.0%)
	Density	69	69	69
	WL [ $\mu\text{m}$ ]	3279720.3	3266321.1 (-0.4%)	3212241.0 (-2.1%)
	#DRV	17	10	73
	WNS [ps]	-111.43	-76.54	-74.64
	TNS [ns]	-8.29	-6.48 (-22%)	-7.72 (-7%)
	Eff. freq [GHz]	0.76	0.78 (+3%)	0.79 (+4%)
	Cell Power [mW]	32.78	32.82 (+0.1%)	32.63 (-0.5%)
Net Power [mW]	14.82	14.84 (+0.1%)	14.75 (-0.5%)	
Total Power [mW]	47.62	47.68 (+0.1%)	47.39 (-0.5%)	
JPEG at 0.71GHz	#Total cells	241251	212065 (-12.1%)	203599 (-15.6%)
	#MF-cells	0	29062	37006
	Chip area [ $\mu\text{m}^2$ ]	675901.0	649236.3 (-3.9%)	640851.4 (-5.2%)
	Std. cell area [ $\mu\text{m}^2$ ]	511131.2	497705.9 (-2.6%)	493874.5 (-3.4%)
	Density	76	77	77
	WL [ $\mu\text{m}$ ]	3062694.7	2941051.6 (-4.0%)	3053576.2 (-0.3%)
	#DRV	14	2	12
	WNS [ps]	-48.41	-50.22	-44.62
	TNS [ns]	-4.20	-4.60 (+10%)	-5.86 (+40%)
	Eff. freq [GHz]	0.69	0.69	0.69
	Cell Power [mW]	36.00	36.02 (+0.1%)	35.77 (-0.7%)
Net Power [mW]	20.89	20.67 (-1.0%)	20.75 (-0.7%)	
Total Power [mW]	56.91	56.72 (-0.3%)	56.54 (-0.6%)	

by 3.38nW and 1.00nW respectively. Even NOR-based combinations, which show a slight 1.60nW increase, remain within acceptable ranges. These results confirm that MF-cells directly integrate with existing design flows while preserving circuit functionality, enabling significant area reduction without severe timing or power degradation.

### C. Chip-level Evaluation of MF-cell Placement Optimization

Table III presents the chip-level results of the MF-cell-based placement optimization frameworks across three benchmark

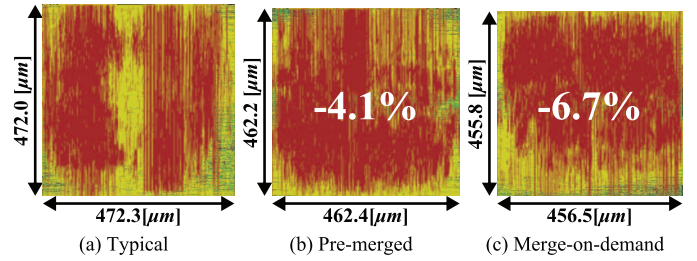


Fig. 6: Final chip layouts of the ECG benchmark: (a) Typical standard-cell placement, (b) Pre-merged framework with 4.1% core area reduction, and (c) Merge-on-demand framework with 6.7% core area reduction. The layouts visually demonstrate the area savings achieved through MF-cell integration.

designs. The pre-merged framework achieves reliable chip area and wirelength reductions while maintaining timing performance comparable to typical standard-cell-based placement. In contrast, the merge-on-demand framework demonstrates superior area efficiency with more aggressive cell merging and WNS improvements across all benchmarks due to reduced path lengths, resulting in enhanced performance, i.e., effective frequency. Both frameworks maintain power consumption at comparable rates while achieving area and performance improvements. The merged cells show no pin accessibility issues, and RUDY congestion-aware placement generally keeps DRV counts manageable. We claim from the table that, the pre-merged framework provides stability and manual optimization opportunities through pre-built MF-cell libraries, and the merge-on-demand framework applies MF-cells more aggressively by generating them on-demand to achieve higher design quality improvements.

Fig. 6 illustrates the actual chip layouts for the ECG benchmark, demonstrating the progressive area reduction achieved by the frameworks. These results validate that MF-cell integration successfully achieves area and performance improvements while maintaining power consumption.

## V. CONCLUSION

This paper presented a novel MF-cell-based systematic placement optimization framework that eliminates redundant margins between adjacent cells through strategic diffusion chaining. We proposed two complementary approaches: a pre-merged framework using statistical analysis of frequent cell pairs, and a merge-on-demand framework that generates MF-cells during the placement flow. Both use MWM algorithms for candidate selection and HPWL/RUDY-based placement optimization. Experimental results demonstrate substantial improvements, with the aggressive merge-on-demand approach achieving up to 6.7% chip area reduction, and the conservative pre-merged framework providing up to 4.1% area improvements. Both approaches enhanced overall design quality by delivering better performance and maintaining comparable power consumption relative to their standard-cell-based counterparts. Our MF-cell-based optimization offers a practical solution for higher design density while ensuring compatibility with existing EDA tools and verification flows.

## ACKNOWLEDGMENT

This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) RS-2024-00359696, and RS-2024-00405495; and in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) Grant funded by the Korea government (MSIT) RS-2025-02218027. The EDA tool was supported by the IC Design Education Center (IDEC), Korea.

## REFERENCES

- [1] H. Park *et al.*, “Challenges on DTCO Methodology Towards Deep Submicron Interconnect Technology,” in *International SoC Design Conference (ISOCC)*, 2021.
- [2] X. Wang *et al.*, “Design-Technology Co-Optimization of Standard Cell Libraries on Intel 10nm Process,” in *International Electron Devices Meeting (IEDM)*, 2018.
- [3] C. Auth *et al.*, “A 10nm high performance and low-power CMOS technology featuring 3rd generation FinFET transistors, Self-Aligned Quad Patterning, contact over active gate and cobalt local interconnects,” in *International Electron Devices Meeting (IEDM)*, 2017.
- [4] J.-S. Yoon *et al.*, “Performance, Power, and Area of Standard Cells in Sub 3 nm Node Using Buried Power Rail,” *IEEE Transactions on Electron Devices (TED)*, 2022.
- [5] S. Shaji *et al.*, “A Comparative Study on Front-Side, Buried and Back-Side Power Rail Topologies in 3nm Technology Node,” in *International Symposium on Low Power Electronics and Design (ISLPED)*, 2023.
- [6] T. Kim *et al.*, “NS3K: A 3-nm Nanosheet FET Standard Cell Library Development and its Impact,” *IEEE Transactions on Very Large Scale Integration Systems (TVLSI)*, 2023.
- [7] D. Lee *et al.*, “SP&R: SMT-Based Simultaneous Place-and-Route for Standard Cell Synthesis of Advanced Nodes,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2021.
- [8] H. Cho *et al.*, “Standard Cell Layout Generator Amenable to Design Technology Co-Optimization in Advanced Process Nodes,” in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2024.
- [9] J. Yoon and H. Park, “Design-Technology Co-Optimization with Standard Cell Layout Generator for Pin Configurations,” in *International Symposium on Quality Electronic Design (ISQED)*, 2024.
- [10] J. Yoon and H. Park, “SMT-Based Optimal Transistor Folding and Placement for Standard Cell Layout Generation,” in *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2026.
- [11] R. Duan and S. Pettie, “Linear-Time Approximation for Maximum Weight Matching,” *J. ACM*, 2014.
- [12] T. Liang *et al.*, “AutoCellLibX: Automated Standard Cell Library Extension Based on Pattern Mining,” in *arXiv*, 2022.
- [13] R. Fu *et al.*, “TeMACLE: A Technology Mapping-Aware Area-Efficient Standard Cell Library Extension Framework,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2025.
- [14] M. I. Dewan and D. H. Kim, “NP-Separate: A New VLSI Design Methodology for Area, Power, and Performance Optimization,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2020.
- [15] Y. Cai *et al.*, “Enhancing ASIC Technology Mapping via Parallel Supergate Computing,” in *International Symposium of Electronics Design Automation (ISEDA)*, 2024.
- [16] C.-H. Hsu *et al.*, “TransPlace: A Scalable Transistor-Level Placer for VLSI Beyond Standard-Cell-Based Design,” in *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2024.
- [17] C.-H. Hsu *et al.*, “TransRoute: A Novel Hierarchical Transistor-Level Routing Framework Beyond Standard-Cell Methodology,” in *Design Automation Conference (DAC)*, 2025.
- [18] C.-P. Lu *et al.*, “Late Breaking Results: Design Dependent Mega Cell Methodology for Area and Power Optimization,” in *Design Automation Conference (DAC)*, 2020.
- [19] L. T. Clark *et al.*, “ASAP7: A 7-nm finFET predictive process design kit,” *Microelectronics Journal*, 2016.
- [20] J. Seo *et al.*, “Pin Accessibility-driven Cell Layout Redesign and Placement Optimization,” in *Design Automation Conference (DAC)*, 2017.
- [21] P. Spindler and F. M. Johannes, “Fast and Accurate Routing Demand Estimation for Efficient Routability-driven Placement,” in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2007.
- [22] Oliscience, “Opencores.” [Online]. Available: <https://opencores.org>