

# Optimizing Multibit Flip-Flop Banking via Agile In-Placement PPA Co-Optimization

Huan-Yuan Chen<sup>1</sup>, Yu-Ruei Lin<sup>1</sup>, Mark Po-Hung Lin<sup>1,2</sup>, Hung-Ming Chen<sup>1,2</sup>

*Institute of Electronics, National Yang Ming Chiao Tung University, Hsinchu, Taiwan<sup>1</sup>*

*Institute of Pioneer Semiconductor Innovation, National Yang Ming Chiao Tung University, Hsinchu, Taiwan<sup>2</sup>*

ryan.chen.1104@gmail.com, {yrlin719.ee12, mphlin, hmchen}@nycu.edu.tw

**Abstract**—Multibit flip-flop (MBFF) banking has been widely adopted to reduce dynamic power, simplify clock tree structures, and minimize layout area. In our analysis, we reveal that early-stage banking, despite potential initial timing degradation, enables more extensive flip-flop merging, with subsequent placement refinements mitigating timing violations. Experimental results, benchmarked against the first-place winner from the 2024 ICCAD CAD Contest and most recent SOTA, demonstrate that our approach delivers competitive performance while providing enhanced design flexibility and superior power reduction.

**Index Terms**—Multibit flip-flop (MBFF), MBFF banking/clustering, PPA co-optimization

## I. INTRODUCTION

Power reduction remains a critical challenge in modern integrated circuit design, particularly as clock power constitutes a substantial portion of total power consumption. Multibit flip-flop (MBFF) banking which merges multiple single-bit flip-flops (SBFFs) into a single MBFF, has been widely adopted. Existing MBFF banking methods applied during the physical design stage can be broadly classified into two main categories [2]: the *Timing-Feasible Region-Based Approach* and the *Clustering-Based Approach*. The timing-feasible region-based approach merges flip-flops only when their feasible timing regions overlap, thereby ensuring that no timing degradation occurs [2]–[10]. The clustering-based approach groups flip-flops based on spatial proximity, aiming to minimize displacement and the associated timing degradation [11]–[15].

We can see in an alternative way: MBFF banking can be applied at three stages of the physical design flow. 1) Pre-Placement Banking [16], [17]: this type of MBFF banking is typically performed during the logic synthesis stage. At this point, flip-flops are aggressively merged to maximize power savings. However, the absence of physical layout information can often lead to significant timing degradation. 2) In-Placement Banking [9], [18]: during the placement stage, in-placement MBFF banking leverages detailed physical layout information to merge flip-flops, enabling significantly more accurate timing estimates compared to pre-placement methods. 3) Post-Placement Banking [2]–[8], [10], [19]: most existing studies adopt MBFF banking during the post-placement stage, as the conservative strategies applied at this point help ensure strict adherence to timing constraints.

While existing methods offer meaningful insights, most of them follow fixed trade-off strategies that limit their ability to take full advantage of opportunities across different stages of the design flow. Motivated by the potential of stage-specific MBFF banking, we propose a configurable framework that dynamically adjusts trade-offs among timing, power, and area throughout the design flow. The key contributions of this work are summarized here. First, we conduct a comprehensive analysis of the advantages and limitations of MBFF banking across the pre-placement, in-placement, and post-placement stages, demonstrating that early-stage banking enables more aggressive power savings. Second, we find that performing incremental banking and debanking across multiple physical design stages allows the design to benefit from the strengths of each stage with incremental MBFF adjustments during

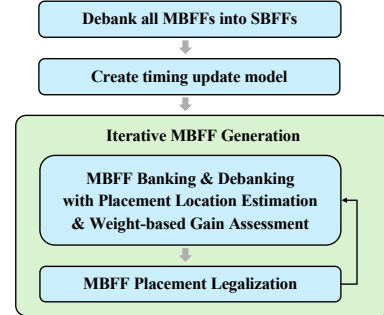


Fig. 1. The proposed weight-based MBFF banking and debanking framework.

iterative in-placement and post-placement optimization. Third, our experimental evaluation, benchmarked against the first-place winner from the 2024 ICCAD CAD Contest [1] and most recent result [22], demonstrates the effectiveness of the proposed method.

## II. PROBLEM FORMULATION

Given the initial placement, netlist and cell library, the goal is to determine MBFF clustering and placement that minimizes the following objective function:

$$Cost = \sum_{f_i \in F} \left( \alpha \cdot TNS(f_i) + \beta \cdot Power(f_i) + \gamma \cdot Area(f_i) \right) + \lambda \cdot D \quad (1)$$

where  $TNS(f_i)$  represents the total negative slack of the flip-flop  $f_i$ ;  $D$  stands for the number of bins that violate the maximum density constraint threshold,  $D_{max}$ .

MBFF placement constraints include the following: (1) All flip-flop cells, after banking and debanking operations, must be placed on-site; (2) Combinational logic cells are fixed and must not be moved; (3) No overlap between cells is permitted; (4) Functional equivalence must be maintained.

## III. PROPOSED ALGORITHMS

The overall flow of our approach is illustrated in Fig. 1. We begin by debanking all flip-flops into single-bit flip-flops to simplify the initial configuration. An efficient and fast timing update model is then employed to estimate timing degradation and update the critical path. The flow then proceeds to the iterative MBFF generation phase, where we prioritize the creation of larger MBFFs first, as they typically offer greater power and area benefits. Each round of MBFF generation consists of three key stages: MBFF banking and debanking, MBFF placement, and MBFF legalization.

One key step for efficient evaluation is timing update model. When banking flip-flops, it is crucial to evaluate the timing impact introduced by their movements. However, during the MBFF banking and debanking process, the total number of flip-flop modifications and timing updates can exceed tens of thousands. To overcome this challenge, our timing update model maintains local critical path information at each cell node.

TABLE I  
COMPARISON WITH THE TOP-3 WINNERS OF THE 2024 ICCAD CAD CONTEST BENCHMARKS [1].

Case	1 <sup>st</sup> Place			2 <sup>nd</sup> Place			3 <sup>rd</sup> Place			Our Result		
	Total Cost	Ratio	Run Time (s)	Total Cost	Ratio	Run Time (s)	Total Cost	Ratio	Run Time (s)	Total Cost	Ratio	Run Time (s)
T1	739235861	1.004	2.66	743001624	1.009	8.39	742644610	1.008	1.91	736602486	1	3.28
T2	744231	0.983	37.00	830259	1.097	17.00	895557	1.184	3.95	756559	1	2.25
T3	727971140	1.001	3.01	728870931	1.002	9.54	736856923	1.013	2.50	727757372	1	3.06
H1	31602979	1.021	19.60	31507917	1.018	12.80	41260003	1.333	5.02	30946401	1	2.91
H2	13408414	1.165	36.40	13863333	1.204	14.70	13486380	1.172	4.80	11510067	1	2.72
H3	56316079	1.006	4.24	55941500	0.999	17.70	55998164	1.001	3.65	55973041	1	2.87
H4	728497353	1.001	2.61	729387591	1.002	9.56	736804470	1.012	2.36	727798594	1	3.02
Average	68341032	1.024	15.07	69727618	1.045	12.81	73191104	1.097	3.45	66718785	1	2.87

Specifically, for each combinational logic gate,  $g_i$ , we record changes in the local critical path using half-perimeter wirelength as an efficient proxy for timing estimation. When the critical path timing of a gate is updated, the change is propagated to the next stage in the circuit.

To enable near-optimal MBFF banking, we introduce three guiding principles that address trade-offs across timing, power, and area. First is MBFF cell selection considering pin distribution. We incorporate pin distribution as a critical factor in MBFF selection. We first evaluate multiple MBFF types and choose the one with the lowest cost based on a weighted combination of power and area. Then, leveraging the specific pin configuration of the selected MBFF, we use an R-tree data structure [21] to efficiently identify nearby SBFFs whose spatial positions align well with the MBFF's pin layout. This pin-aware grouping ensures a better match between flip-flop locations and MBFF pin assignments, leading to improved timing characteristics and more efficient layout utilization.

Second is the estimation of MBFF placement locations. In both in-placement and post-placement MBFF banking scenarios, combinational logic cells are typically fixed and cannot be relocated. This constraint significantly limits the flexibility for MBFF legalization and is a major contributor to timing degradation after banking. Since there is often a mismatch between the ideal MBFF placement location and the actual legal placement region, relying on ideal positions for initial MBFF estimation can lead to misleading evaluations, as it ignores potential displacement caused by surrounding combinational logic cells. To address this issue, after selecting the constituent SBFF members of an MBFF, we identify the closest legal placement site to the ideal location. This site must be aligned to placement rows and free of overlaps with existing combinational logic cells. Any location that would result in a density violation is discarded. This filtering process effectively reduces flip-flop congestion and ensures that the initial MBFF placement estimate is both feasible and physically aware.

Third, we perform weight-based gain assessment for MBFF banking: to evaluate whether forming the MBFF is beneficial. Our weighted cost function that simultaneously considers timing, area, and power is as follows.

$$\begin{aligned}
 Gain_{mbff} = & \beta \times \left( Power(f_{mbff}) - \sum_{f_i \in F_{mbff}} Power(f_i) \right) \\
 & + \gamma \times \left( Area(f_{mbff}) - \sum_{f_i \in F_{mbff}} Area(f_i) \right) \\
 & - \alpha \times t_{degraded}
 \end{aligned} \tag{2}$$

$f_{mbff}$  denotes the set of SBFFs grouped into the MBFF, and  $t_{degraded}$  represents the timing degradation estimated by our timing update model after the banking operation. The gain

function quantifies the improvement achieved by forming an MBFF as compared to retaining the original SBFF configuration. A positive gain indicates that forming the MBFF improves the overall design quality, while a negative gain suggests that debanking is more favorable.

After completing the preceding procedures, we proceed to the final stage of MBFF placement legalization. We introduce two targeted strategies aimed at minimizing overall displacement and improving legalization quality: Prevention of Excessive Displacement and Consideration of Regional Density.

#### IV. EXPERIMENTAL RESULTS

We implement the proposed algorithm in C++ and conducted experiments on a Linux-based system equipped with a 2.5 GHz Intel i7-11700 CPU. For performance evaluation, we use benchmark circuits provided by the 2024 ICCAD CAD Contest [1].

Our major experimental results are summarized in Table I. According to the official cost metric provided by the contest chair, our method consistently outperforms the first-place solution across most benchmarks, with a particularly notable improvement of approximately 16% on the Hidden2 testcase. Although the cost difference on Testcase2 is marginal, our algorithm achieves a runtime over 16X faster than that of the first-place method. We also compare the results with [22] and gain 0.2% cost reduction in average. In another comparison, while the first-place method achieves approximately 16% lower power consumption, it incurs a significantly higher total negative slack (TNS), which is 88% greater than our result, nearly doubling our reported value. It is important to note that although negative slack may be temporarily tolerated during intermediate optimization stages, the final design must ultimately eliminate all negative slack to meet practical timing constraints.

To demonstrate the configurability of our MBFF banking framework, by adjusting the timing weight in the cost function, the framework generates solutions with varying degrees of flip-flop banking. As the timing weight increases, the resulting Pareto front depicts a set of non-dominated solutions with progressively lower TNS, highlighting the framework's ability to prioritize timing when required. In addition, under near-constant TNS conditions, fine-tuning the relative weights of power and area enables effective control over the trade-off between these two metrics. This flexibility allows the framework to adapt to a wide range of design objectives, making it suitable for diverse optimization scenarios.

#### V. CONCLUSION

In this work, we present a configurable weight-based MBFF banking framework designed for the in-placement and post-placement stages of the IC physical design flow. Experimental results demonstrate that our method delivers competitive performance while providing enhanced design flexibility and superior power reduction.

## REFERENCES

- [1] S.-W. Yang, T.-H. Chen, J.-W. Hsu, T.-W. Li, and C.-F. Cindy Shen “(Invited Paper) 2024 ICCAD CAD Contest Problem B: Power and Timing Optimization Using Multibit Flip-Flop,” Proc. ICCAD. 2024.
- [2] Y.-Y. Chen, H.-Y. Wu, I. H.-R. Jiang, C.-H. Tsai, and C.-C. Wu. 2024. Slack Redistributed Register Clustering with Mixed-Driving Strength Multi-bit Flip-Flops. In Proceedings of the 2024 International Symposium on Physical Design (ISPD ’24), ACM.
- [3] Y.-T. Chang, C.-C. Hsu, M. P.-H. Lin, Y.-W. Tsai and S.-F. Chen. “Post-placement power optimization with multi-bit flip-flops,” 2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2010, pp. 218-223.
- [4] S.-H. Wang, Y.-Y. Liang, T.-Y. Kuo and W.-K. Mak, “Power-Driven Flip-Flop Merging and Relocation,” in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 31, no. 2, pp. 180-191, Feb. 2012.
- [5] I. H.-R. Jiang, C.-L. Chang and Y.-M. Yang, “INTEGRA: Fast Multibit Flip-Flop Clustering for Clock Power Saving,” in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 31, no. 2, pp. 192-204, Feb. 2012.
- [6] S. S.-Y. Liu, W.-T. Lo, C.-J. Lee, and H.-M. Chen. 2013. Agglomerative-based flip-flop merging and relocation for signal wirelength and clock tree optimization. ACM Trans. Des. Autom. Electron. Syst. 18, 3, Article 40 (July 2013).
- [7] Y.-T. Shyu, J.-M. Lin, C.-P. Huang, C.-W. Lin, Y.-Z. Lin and S.-J. Chang. “Effective and Efficient Approach for Power Reduction by Using Multi-Bit Flip-Flops,” in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 21, no. 4, pp. 624-635, April 2013.
- [8] C.-C. Tsai, Y. Shi, G. Luo, and I. H.-R. Jiang. 2013. FF-bond: multi-bit flip-flop bonding at placement. In Proceedings of the 2013 ACM International symposium on Physical Design (ISPD ’13). ACM, 147–153.
- [9] M. P.-H. Lin, C.-C. Hsu and Y.-C. Chen. “Clock-Tree Aware Multibit Flip-Flop Generation During Placement for Power Optimization,” in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 2, pp. 280-292, Feb. 2015.
- [10] M.-Y. Liu, Y.-C. Lai, W.-K. Mak and T.-C. Wang, “Generation of Mixed-Driving Multi-Bit Flip-Flops for Power Optimization,” 2022 IEEE/ACM International Conference On Computer Aided Design (ICCAD), pp. 1-9.
- [11] G. Wu, Y. Xu, D. Wu, M. Ragupathy, Y.-Y. Mo and C. Chu, “Flip-flop clustering by weighted K-means algorithm,” 2016 53rd ACM/EDAC/IEEE Design Automation Conference (DAC).
- [12] A. B. Kahng, J. Li and L. Wang, “Improved flop tray-based design implementation for power reduction,” 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 1-8.
- [13] Y.-C. Chang, T.-W. Lin, I. H.-R. Jiang, and G.-J. Nam. 2019. Graceful Register Clustering by Effective Mean Shift Algorithm for Power and Timing Balancing. In Proceedings of the 2019 International Symposium on Physical Design (ISPD ’19), ACM, 11–18.
- [14] C.-C. Huang, G. Tellez, G.-J. Nam and Y.-W. Chang, “Latch Clustering for Timing-Power Co-Optimization,” 2020 57th ACM/IEEE Design Automation Conference (DAC), pp. 1-6.
- [15] A. B. Kahng, S. Kundu, and S. Thumathy. 2024. Scalable Flip-Flop Clustering Using Divide and Conquer For Capacitated K-Means. In Proceedings of the Great Lakes Symposium on VLSI 2024 (GLSVLSI ’24). ACM, 177–184.
- [16] D. Yi and T. Kim, “Allocation of multi-bit flip-flops in logic synthesis for power optimization,” 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD).
- [17] Y. Cai, L. Zhu, and X. Guo. 2025. Revisit MBFF: Efficient Early-Stage Multi-bit Flip-Flops Clustering with Physical and Timing Awareness. In Proceedings of the 30th Asia and South Pacific Design Automation Conference (ASPDAC ’25), 1230–1236.
- [18] W. Hou, D. Liu and P.-H. Ho, “Automatic register banking for low-power clock trees,” 2009 10th International Symposium on Quality Electronic Design, 2009, pp. 647-652.
- [19] L. Cherif, M. Chentouf, J. Benallal, M. Darmi, R. Elgouri and N. Hmina, “Usage and impact of multi-bit flip-flops low power methodology on physical implementation,” 2018 4th International Conference on Optimization and Applications (ICOA), 2018, pp. 1-5.
- [20] J. Jeong and T. Kim. 2024. Binding Multi-bit Flip-flop Cells through Design and Technology Co-optimization. In Proceedings of the 61st ACM/IEEE Design Automation Conference (DAC ’24).
- [21] A. Guttman. 1984. R-trees: a dynamic index structure for spatial searching. SIGMOD Rec. 14, 2 (June 1984), 47–57.
- [22] C.-Y. Li, C.-C. Su, Z.-W. Chen, S.-H. Chen, and Y.-W. Chang, “Late Breaking Results: Multi-Objective Multi-Bit Flip-Flop Placement Considering Pre-Placed Cells,” 2025 ACM/IEEE Design Automation Conference (DAC ’25).