



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Lucas de Lacerda Tinoco

Timing-Aware Enhancements to MBFF Clustering in OpenROAD

Florianópolis
2026

Lucas de Lacerda Tinoco

Timing-Aware Enhancements to MBFF Clustering in OpenROAD

Trabalho de Conclusão de Curso do Curso de Graduação em Ciência da Computação do Centro Tecnológico da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. José Luís Almada Güntzel, Dr.

Florianópolis

2026

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Tinoco, Lucas de Lacerda
Timing-Aware Enhancements to MBFF Clustering in
OpenROAD / Lucas de Lacerda Tinoco ; orientador, José Luís
Almada Güntzel, 2026.
77 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Ciências da Computação, Florianópolis, 2026.

Inclui referências.

1. Ciências da Computação. 2. EDA. 3. VLSI. 4. Projeto
Físico de Circuitos Integrados. 5. Otimização de Potência e
Temporização. I. Güntzel, José Luís Almada. II. Universidade
Federal de Santa Catarina. Graduação em Ciências da
Computação. III. Título.

Lucas de Lacerda Tinoco

Timing-Aware Enhancements to MBFF Clustering in OpenROAD

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Ciência da Computação” e aprovado em sua forma final pelo Curso de Graduação em Ciência da Computação.

Florianópolis, 10 de julho de 2026.

Banca Examinadora:

Prof. José Luís Almada Güntzel, Dr.
Universidade Federal de Santa Catarina

Tiago Augusto Fontana, Dr.
Cadence Design Systems, Austin, TX, USA

Renan Oliveira Netto, Dr.
Cadence Design Systems, San Jose, CA, USA

À minha — muito viva — tia-avó Cleide Souza Gomes.

Foi ela quem, semana após semana, me levava para desenhar na Escola Daniel Azulay. (E me levava também para tomar um milk-shake de Ovomaltine.)

Hoje, desenho chips.

De certa forma, tudo começou ali.

ACKNOWLEDGEMENTS

I am blessed to have had the opportunity to study at an outstanding university. First and foremost, I thank God for making this journey possible. His mercy alone would have been more than enough, yet He graciously blessed me with opportunities far beyond what I deserved. All glory belongs to Him.

I would also like to express my deepest gratitude to my wife, who has stood by my side through all the ups and downs from the beginning to the end of this work. From early morning wake-ups to countless late nights, your love and patience never failed me. I love you so very much, minha Dudinha!

I am also grateful to my family, who has supported me every step of the way and never allowed me to give up. You taught me how to keep all the balls in the air, just like a true juggler, even when life became overwhelming.

My sincere thanks go to my advisor, Dr. José Luís Güntzel, for his continuous guidance and enthusiasm throughout this work.

I am also grateful to Dr. Tiago Augusto Fontana and Dr. Renan Oliveira Netto for generously taking the time to review this text and provide valuable suggestions that greatly improved its quality.

I would like to thank all my colleagues at Chipus Microelectronics and the members of the Embedded Computing Lab (ECL), who are always willing to discuss ideas and are equally passionate about advancing microelectronics research and development in Brazil.

I also thank all the professors at the Universidade Federal de Santa Catarina who contributed significantly to my academic and personal development throughout my undergraduate studies.

I am grateful to Andrew B. Kahng, Jiajia Li, Lutong Wang, Sayak Kundu, and Shreyas Thumathy, the authors of FTray and SFTray, for developing such influential work on MBFF clustering, which served as the foundation for this work.

Finally, I thank all the contributors to the OpenROAD project for their outstanding work and for their commitment to democratizing chip design and enabling open-source innovation in physical design automation.

ABSTRACT

Clock networks represent a major source of power consumption in modern Very-Large-Scale Integration (VLSI) designs. Multi-Bit Flip-Flops (MBFFs) can reduce clock-network power by merging groups of Single-Bit Flip-Flops (SBFFs) into shared structures with common clock drivers. However, determining how to cluster SBFFs while preserving timing quality remains a challenging optimization problem. Among the state-of-the-art MBFF clustering techniques, the Scalable Divide-and-Conquer Flop-Tray (SFTray) algorithm available in the OpenROAD framework performs a Minimum-Cost Flow (MCF)-based assignment driven exclusively by Manhattan displacement, while timing information is considered only in later optimization stages. This work investigates the incorporation of continuous timing-criticality information into the SFTray clustering flow. A slack-derived criticality metric is introduced and combined with normalized displacement to define a timing-aware assignment cost for the MCF formulation. Additionally, timing criticality is incorporated into the Linear Programming (LP) tray-refinement stage to further improve timing awareness during clustering. The proposed modifications were implemented within OpenROAD and evaluated using nine benchmark circuits synthesized and physically implemented with the ASAP7 7 nm technology enablement. Experimental results demonstrate that the proposed formulations are able to better explore the optimization space in designs with high sequential-cell density, achieving improved power and timing trade-offs while preserving the area and clock-network benefits of MBFF clustering. These results indicate that timing criticality can be effectively exploited throughout multiple stages of the clustering process, leading to higher-quality MBFF solutions in open-source physical-design flows.

Keywords: EDA, VLSI, Physical Design, Power and Timing Optimization of Clock Network, MBFF Clustering, Open Source.

RESUMO

As redes de distribuição de clock representam uma das principais fontes de consumo de potência em projetos modernos de circuitos integrados VLSI. Os Flip-Flops Multi-Bit (MBFFs) reduzem esse consumo ao agrupar múltiplos Flip-Flops Single-Bit (SBFFs) em estruturas compartilhadas com buffers de clock comuns. Entretanto, determinar como realizar esse agrupamento preservando a qualidade temporal do circuito permanece um problema de otimização desafiador. Entre as técnicas de agrupamento MBFF de estado da arte, o algoritmo Scalable Divide-and-Conquer Flop-Tray (SFTray), disponível na plataforma OpenROAD, realiza a atribuição de flip-flops por meio de uma formulação de Fluxo de Custo Mínimo (MCF) baseada exclusivamente no deslocamento Manhattan, incorporando informações temporais apenas em etapas posteriores do processo de otimização. Este trabalho investiga a incorporação de informações contínuas de criticidade temporal ao fluxo de agrupamento do SFTray. Para isso, é introduzida uma métrica de criticidade derivada do slack, combinada ao deslocamento normalizado para compor uma função de custo temporalmente consciente na etapa de atribuição baseada em MCF. Além disso, a criticidade temporal é incorporada à etapa de refinamento baseada em Programação Linear (LP), ampliando a influência das informações temporais ao longo do processo de agrupamento. As modificações propostas foram implementadas diretamente no OpenROAD e avaliadas em nove circuitos benchmark sintetizados e implementados fisicamente utilizando a tecnologia ASAP7 de 7 nm. Os resultados experimentais demonstram que as formulações propostas exploram de forma mais eficiente o espaço de otimização em projetos com elevada densidade de elementos sequenciais, obtendo melhores compromissos entre potência e temporização sem comprometer os benefícios de área e redução da rede de clock proporcionados pelos MBFFs. Os resultados indicam que a criticidade temporal pode ser explorada de maneira eficaz em múltiplas etapas do processo de agrupamento, contribuindo para soluções MBFF de maior qualidade em fluxos de projeto físico de código aberto.

Palavras-chave: EDA, VLSI, Projeto Físico de Circuitos Integrados, Otimização de Potência e Temporização, Rede de Clock, Agrupamento MBFF, OpenROAD.

LIST OF FIGURES

Figure 1	– Clock-network routing comparison for the <i>ethmac</i> benchmark generated with OpenROAD framework (Project, 2025). Left: non-clustered implementation (NC). Right: MBFF-clustered implementation using the proposed timing-aware approach. The reduction in clock sinks achieved through MBFF clustering leads to a simpler clock distribution structure and shorter clock routing.	14
Figure 2	– RTL-to-GDSII flow.	17
Figure 3	– MBFF Clustering illustration.	20
Figure 4	– Example of a <i>Minimum-Cost Flow</i> (MCF) network. Assuming a unit flow must be sent from s to t , the path $s \rightarrow B \rightarrow t$ yields the minimum total cost of 3.	21
Figure 5	– Example of a <i>Linear Programming</i> (LP) problem with two decision variables. The shaded region represents the feasible solution space defined by the constraints, while the dashed line represents the objective function. The highlighted point corresponds to the optimal solution. . .	23
Figure 6	– Overview of the SFTray divide-and-conquer flow and the MCF assignment graph.	32
Figure 7	– Overview of the SFTray divide-and-conquer flow and the MCF assignment graph used in the proposed timing-aware formulation.	35
Figure 8	– Example of flip-flop criticality computation. The criticality of a flip-flop is defined as the largest normalized timing violation among all negative-slack paths in which it participates as either a launch or capture endpoint.	37

LIST OF ALGORITHMS

1	Flip-Flop Criticality Computation	36
2	Timing-Aware Edge-Cost Computation for MCF Assignment	38

LIST OF TABLES

Table 1 – Comparison of representative works on flip-flop and MBFF clustering.	31
Table 2 – Benchmark circuits used in the experimental evaluation and their sequential-cell composition before MBFF clustering.	41
Table 3 – Worst Negative Slack (WNS) and Total Negative Slack (TNS) comparison for $\alpha = 8$ and $\beta = 0.1$	42
Table 4 – Comparison of MBFF distribution, Clock Power, Switching Power, Total Power and Area for $\alpha = 8$ and $\beta = 0.1$. NC rows present absolute values, while SFTray, MCF-Crit, and MCF+LP-Crit show absolute values and percentage variation relative to NC (except for #1b, #2b and #4b, which are the number of 1-bit flip-flops, 2-bit flip-flops and 4-bit flip-flops, respectively). A negative percentage is always better. Bold values indicate that MCF-Crit or MCF+LP-Crit achieved a better result than SFTray. All metrics were extracted after detailed routing.	45
Table 5 – Runtime comparison between NC, SFTray, MCF-Crit, and MCF+LP-Crit.	47
Table 6 – Worst Negative Slack (WNS) and Total Negative Slack (TNS) comparison for varying α values with $\beta = 0.1$ (Part 1).	54
Table 7 – Worst Negative Slack (WNS) and Total Negative Slack (TNS) comparison for varying α values with $\beta = 0.1$ (Part 2).	55
Table 8 – Worst Negative Slack (WNS) and Total Negative Slack (TNS) comparison for varying α values with $\beta = 0.1$ (Part 3).	56
Table 9 – Worst Negative Slack (WNS) and Total Negative Slack (TNS) comparison for varying α values with $\beta = 0.1$ (Part 4).	57
Table 10 – Worst Negative Slack (WNS) and Total Negative Slack (TNS) comparison for varying α values with $\beta = 0.1$ (Part 5).	58
Table 11 – Comparison of MBFF clustering strategies for varying α values with $\beta = 0.1$ (Part 1).	60
Table 12 – Comparison of MBFF clustering strategies for varying α values with $\beta = 0.1$ (Part 2).	61
Table 13 – Comparison of MBFF clustering strategies for varying α values with $\beta = 0.1$ (Part 3).	62
Table 14 – Comparison of MBFF clustering strategies for varying α values with $\beta = 0.1$ (Part 4).	63
Table 15 – Comparison of MBFF clustering strategies for varying α values with $\beta = 0.1$ (Part 5).	64

LIST OF ABBREVIATIONS AND ACRONYMS

CTS	<i>Clock Tree Synthesis</i>
DRC	<i>Design Rule Check</i>
EDA	<i>Electronic Design Automation</i>
FTray	<i>Flop-Tray Technique</i>
GDSII	<i>Graphic Data System II</i>
HDL	<i>Hardware Description Language</i>
IC	<i>Integrated Circuit</i>
ILP	<i>Integer Linear Programming</i>
LEC	<i>Logic Equivalence Check</i>
LP	<i>Linear Programming</i>
LVS	<i>Layout-Versus-Schematic</i>
MBFF	<i>Multi-Bit Flip-Flop</i>
MBFFs	<i>Multi-Bit Flip-Flops</i>
MCF	<i>Minimum-Cost Flow</i>
PPA	<i>Power, Performance, and Area</i>
QoR	<i>Quality of Results</i>
RTL	<i>Register Transfer Level</i>
SAT	<i>Boolean Satisfiability Problem</i>
SBFF	<i>Single-Bit Flip-Flop</i>
SBFFs	<i>Single-Bit Flip-Flops</i>
SFTray	<i>Scalable Divide-And-Conquer Flop-Tray Technique</i>
TNS	<i>Total Negative Slack</i>
VLSI	<i>Very-Large-Scale Integration</i>
WNS	<i>Worst Negative Slack</i>

CONTENTS

1	INTRODUCTION	13
2	BACKGROUND	16
2.1	RTL-TO-GDSII FLOW	16
2.1.1	Logic Synthesis	16
2.1.2	Physical Synthesis	17
2.2	MBFF CLUSTERING	19
2.3	MIN-COST FLOW AND LINEAR PROGRAMMING	20
3	RELATED WORK	24
3.1	WEIGHTED K-MEANS	24
3.2	EFFECTIVE MEAN SHIFT	25
3.3	MIXED-DRIVING MBFF	26
3.4	EARLY-STAGE MBFF	26
3.5	MULTI-OBJECTIVE MBFF	27
3.6	FLOP-TRAY (FTRAY)	28
3.7	SCALABLE DIVIDE-AND-CONQUER FLOP-TRAY (SFTRAY)	29
3.8	SUMMARY	29
4	THE SCALABLE DIVIDE-AND-CONQUER FTRAY APPROACH	32
5	ENHANCEMENTS TO THE SFTRAY TECHNIQUE	35
5.1	MCF-CRIT: ADDING NORMALIZED TIMING-CRITICALITY TO THE MCF	35
5.2	MCF+LP-CRIT: ADDING TIMING-CRITICALITY TO LP	38
6	EXPERIMENTAL RESULTS	40
6.1	METHODOLOGY	40
6.2	RESULTS AND DISCUSSION	41
6.3	RUNTIME ANALYSIS	46
7	CONCLUSION	48
	REFERENCES	50
	APPENDIX A – COMPARISON OF MBFF CLUSTERING STRATEGIES FOR VARYING α VALUES WITH $\beta = 0.1$. TIMING-RELATED ANALYSIS.	53
	APPENDIX B – COMPARISON OF MBFF CLUSTERING STRATEGIES FOR VARYING α VALUES WITH $\beta = 0.1$. FF DISTRIBUTION, POWER AND AREA RELATED ANALYSIS.	59
	APPENDIX C – SOURCE CODE	65
	APPENDIX D – WORK PAPER IN SBC FORMAT	66

1 INTRODUCTION

Clock networks are among the largest contributors to power consumption in modern *Very-Large-Scale Integration* (VLSI) circuits, often accounting for up to 30% to 50% of the total dynamic power budget (Wu *et al.*, 2016). Dynamic power consists mostly of the switching power, described as

$$P_{switching} = \alpha C V_{DD}^2 f, \quad (1)$$

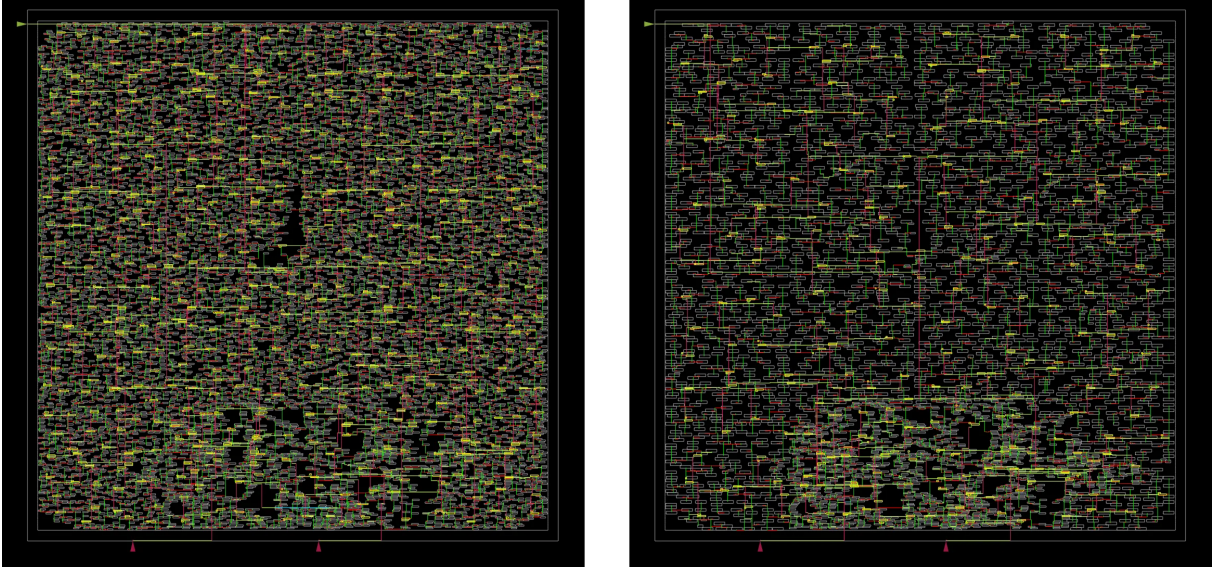
where α is the switching activity, C is the switched capacitance, V_{DD} is the supply voltage, and f is the clock frequency (Weste; Harris, 2011). Unlike ordinary signal nets, the clock network switches every clock cycle, making its switching activity $\alpha = 1$. Also, the capacitance associated with the clock network is the sum of the capacitances of the clock-routing wires, the clock input capacitances of all sequential elements (primarily flip-flops), and the internal capacitances of the clock buffers inserted during clock-tree synthesis. This capacitive load becomes very large, contributing to a high dynamic power.

To mitigate this issue, contemporary standard-cell libraries provide *Multi-Bit Flip-Flops* (MBFFs), which integrate several *Single-Bit Flip-Flops* (SBFFs) into a shared physical structure with common clock drivers and routing resources. By reducing the number of clock sinks and shortening clock-tree wiring, *Multi-Bit Flip-Flop* (MBFF) clustering can substantially decrease switching power while also improving routability and area utilization (Kahng; Li; Wang, 2016; Cai; Zhu; Guo, 2025). Figure 1 illustrates this effect by comparing the clock-network routing of a design before and after MBFF clustering.

Despite its widespread adoption in commercial *Electronic Design Automation* (EDA) flows, MBFF clustering remains a challenging optimization problem. The clustering process seeks to reduce power and preserve timing, while minimizing displacement and satisfying the legalization constraints. Furthermore, questions persist regarding the optimal stage in the design flow – synthesis or placement – to perform this clustering for the best *Quality of Results* (QoR). As a consequence, a significant body of research has been devoted to developing efficient clustering strategies, including approaches based on Weighted K-Means (Wu *et al.*, 2016), Mean Shift (Chang *et al.*, 2019), mixed-driving MBFF generation (Liu *et al.*, 2022), and multi-objective optimization techniques (Li *et al.*, 2025). This continued academic interest is further evidenced by recent competitions, such as the ICCAD CAD Contest 2024 (Yang; Hsu; Lee, *et al.*, 2025) and ICCAD CAD Contest 2025 (Yang; Hsu; Cheng, *et al.*, 2025).

Among the state-of-the-art approaches, the FTray technique (Kahng; Li; Wang, 2016) and its scalable divide-and-conquer extension, SFTray (Kahng; Kundu; Thumathy, 2024), stand out due to their integration of capacitated clustering with timing-aware optimization. SFTray formulates the flip-flop assignment problem using a Minimum-Cost Flow (MCF) model combined with Linear Programming (LP) and Integer Linear Programming

Figure 1 – Clock-network routing comparison for the *ethmac* benchmark generated with OpenROAD framework (Project, 2025). Left: non-clustered implementation (NC). Right: MBFF-clustered implementation using the proposed timing-aware approach. The reduction in clock sinks achieved through MBFF clustering leads to a simpler clock distribution structure and shorter clock routing.



Source: Elaborated by the author with OpenROAD (Project, 2025)

(ILP) refinement stages. Furthermore, its divide-and-conquer strategy enables the efficient application of MBFF clustering to large-scale designs, leading to its integration into the open-source RTL-to-GDSII framework OpenROAD (Kahng, 2022).

However, although timing information is considered during the final stages of the original SFTray optimization flow, its treatment remains relatively limited. The MCF assignment stage relies exclusively on geometric displacement, measured through Manhattan distance, while timing information is only incorporated later through timing-related penalties in the ILP formulation. As a result, timing relationships between flip-flops do not directly influence the early clustering decisions that shape the subsequent optimization stages.

This work investigates the incorporation of timing criticality information into earlier stages of the SFTray flow. More specifically, a continuous timing-criticality model derived from path slack information is introduced and integrated into the MCF assignment formulation through a cost function that combines normalized displacement and normalized timing criticality. Additionally, an extended version of the approach incorporates timing criticality information into the LP refinement stage, allowing timing awareness to influence both assignment and tray-position optimization.

The proposed modifications were implemented directly in the SFTray infrastructure available in OpenROAD, preserving compatibility with a complete open-source physical

design flow. Experimental evaluations were conducted using the ASAP7 technology enablement and a set of benchmark circuits synthesized with OpenROAD. The results compare the original SFTray implementation and the proposed timing-aware formulations in terms of power consumption, timing quality, area utilization, clock-network characteristics, and MBFF distribution.

The remainder of this work is organized as follows. Chapter 2 presents the theoretical background related to integrated-circuit physical design and MBFF clustering. Chapter 3 exposes MBFF Clustering different strategies implement to achieve better QoR using MBFF cells. Chapter 4 describes the original SFTray algorithm. Chapter 5 presents the proposed timing-aware modifications. Chapter 6 details the experimental methodology and discusses its results. Finally, Chapter 7 presents the conclusions and future research directions.

2 BACKGROUND

This chapter presents the necessary theoretical knowledge to understand the current work. We first detail the modern digital *Register Transfer Level* (RTL)-to-*Graphic Data System II* (GDSII) Flow, outlining the major stages from initial behavioral description to final physical implementation, including Logic Synthesis and Physical Synthesis. We then detail the key steps within Physical Synthesis: Floorplanning, Placement (Global and Detailed), *Clock Tree Synthesis* (CTS), and Routing (Global and Detailed). Then, we introduce and define the critical optimization technique at the core of this research, MBFF Clustering, discussing its goals and its application within the overall design flow. Finally, we explain two main algorithmic techniques used in *Flop-Tray Technique* (FTray)/*Scalable Divide-And-Conquer Flop-Tray Technique* (SFTray): MCF and LP.

2.1 RTL-TO-GDSII FLOW

The RTL-to-GDSII flow, as briefly mentioned in the introduction, is the process of inputting a hardware description – via a *Hardware Description Language* (HDL), such as VHDL, Verilog or SystemVerilog – and outputting a format that a foundry (as *Integrated Circuit* (IC)s manufacturers are referred to) can interpret, run tests and begin the manufacturing process: the GDSII format.

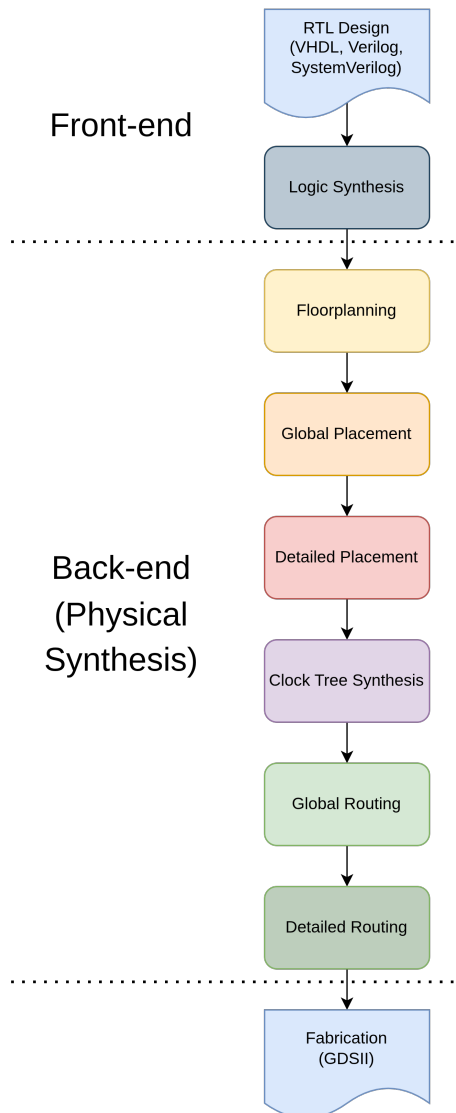
Although the term "RTL-to-GDSII" may give the impression that this process is a simple "input/output" process, actually it is a complex step-by-step sequence that often loops back to previous stages and generally involves several engineers specialized in each or many phases of the flow. In Figure 2 depicts the whole flow in an academic perspective.

It all starts with an RTL description. Front-end engineers – as the industry calls them – write VHDL, Verilog or SystemVerilog describing a hardware to be implemented according to given requirements handled by a client. Those professionals need not only to write, but run tests and validate the description to make sure it is good enough to be inputted in the flow. It may loop back, and oftenly does, to the RTL description phase, but the goal is to have a version the more mature as possible to facilitate the next step, the Logic Synthesis.

2.1.1 Logic Synthesis

Logic Synthesis is "the process of converting HDL into low-level circuit elements" (Kahng; Lienig, *et al.*, 2011). Given the written HDL file and a(a couple of) technology file(s), Front-end engineers (often along with Back-end engineers) run a Logic Synthesis tool to "map the described functionality to specific elements such as standard cells, and a list of signal nets, or netlist" (Kahng; Lienig, *et al.*, 2011). But it not only involves this

Figure 2 – RTL-to-GDSII flow.



Source: Adapted from Kahng, Lienig, *et al.* (2011)

technology mapping step, it also tackles *Power, Performance, and Area* (PPA) constraints by optimizing the circuit before and after mapping. The final output is a so called Gate Level Netlist, usually a description language file but with real standard cells instances, instead of generic blocks.

2.1.2 Physical Synthesis

Given a Gate Level Netlist, next step is to place and route this circuit in a Physical Layout. This process is named by academia as Physical Synthesis and the industry often calls it Implementation or Place & Route, as it naturally expresses the task. There are many stages involved in Physical Synthesis which, for the sake of didactics, could be

summarized in the following six ones:

1. Floorplanning: As an architect draws a layout of a house – a floor plan – in the EDA context we are interested in sketching a preliminary layout of a chip. Thus, Floorplanning determines the positions of the input and output (I/O) pins (Pin assignment), defines positions and shapes of the blocks (Macro placement) and routes the power distribution network (Power planning). (Kahng; Lienig, *et al.*, 2011)
2. Global Placement: Placement in general is the process of placing the standard cells (i.e., the gates and flip-flops layouts) within each block of the layout. This is the initial phase of cell positioning that seeks to determine the general locations of all these instances. Global placement is a multi-objective optimization problem, primarily focused on minimizing the total length of connections (wirelength) between elements. (Kahng; Lienig, *et al.*, 2011) In order to cope with the high complexity of the task, Global Placement disregards cell dimensions, which results in a solution with cell overlaps and misalignments with respect to the placement grid.
3. Detailed Placement: Following the global stage, Detailed Placement refines the object locations. This step ensures that all cells are precisely moved to legal cell sites and strictly enforces nonoverlapping constraints. The resulting accurate locations enable further crucial analysis, including more reliable estimates of circuit delay for timing optimization and power estimates. (Kahng; Lienig, *et al.*, 2011)
4. CTS: Clock trees play a vital role in modern synchronous designs and significantly impact the circuit's performance and power consumption (Wu *et al.*, 2016; Kahng; Lienig, *et al.*, 2011). This stage is dedicated to constructing a network that distributes the clock signal from its source (root) to all sequential elements (sinks) across the chip. A key requirement is low skew, ensuring the clock signal arrives at every sequential gate simultaneously. CTS involves an initial tree construction, which is then followed by clock buffer insertion and subsequent skew optimizations to minimize the timing differences and maintain signal integrity. (Kahng; Lienig, *et al.*, 2011).
5. Global Routing: Routing, the process of determining the signal paths of the nets on the layout, is also divided into two steps: Global and Detailed. Global routing first involves partitioning the chip into routing regions, which are often represented by gcells (global cells) or more general structures like channels and switchboxes. The process then systematically searches for region-to-region paths for all signal nets. During this process, pins with the same electric potential are connected using abstract wire segments, typically aiming to minimize the total routed length or optimizing other objectives. To meet these optimization objectives, the calculated route of each net must be short. (Kahng; Lienig, *et al.*, 2011)

6. Detailed Routing: Following global routing, Detailed Routing assigns specific route segments of signal nets to precise routing tracks, vias, and metal layers while remaining consistent with the given global routes. The core objective of this stage is to respect all design rules (*Design Rule Check* (DRC)). Given that the routing region of a single gcell is orders of magnitude smaller than the entire chip (e.g., 10×10 routing tracks), the detailed routing of one gcell can be performed independently of its neighbors, provided the connections are properly maintained across the boundaries. (Kahng; Lienig, *et al.*, 2011)

After all these stages come to an end, it is time to generate a GDSII file, a format that the foundry can parse and then use to proceed with fabrication. Of course this process of sending the GDSII to the foundry – also known as the tape out – is only done when the designers are confident that the layout fully express the design. And to check that, there are two steps that were not mentioned but are of extreme importance: *Logic Equivalence Check* (LEC) and *Layout-Versus-Schematic* (LVS) checking.

LEC is a critical formal verification technique used in the flow to ensure that the functional behavior of a digital design is conserved throughout various transformation steps (e.g., from RTL to synthesized netlist and from synthesized netlist to physical netlist). Unlike simulation, which uses input stimuli (test vectors), LEC uses mathematical proof based on formal methods to provide a higher level of confidence in the design’s consistency. The core principle of LEC relies on Boolean computational algebra, specifically the concept of Satisfiability (*Boolean Satisfiability Problem* (SAT)), which checks if there is any input pattern that would cause the reference design and the implementation design to produce different outputs. If such a difference, or logical non-equivalence, is found, the check fails. (Mishra, 2022)

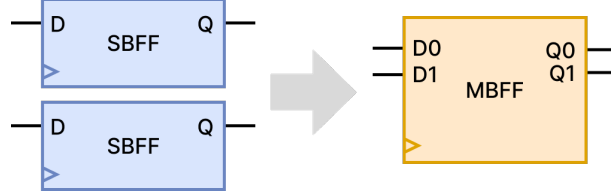
LVS checking is a crucial step in the integrated circuit (IC) design flow used to verify the functional integrity of the final physical layout. The process works by reverse engineering a netlist directly from the finished layout data. This derived netlist is then formally compared against the original, verified golden netlist (typically produced during logic synthesis or initial circuit design). LVS ensures that the physical realization of the circuit exactly matches the intended design schematic, confirming that no errors were introduced during the placement, routing, or other physical design steps. (Kahng; Lienig, *et al.*, 2011)

2.2 MBFF CLUSTERING

MBFF clustering is a technique used to reduce power consumption within a circuit’s clock distribution network. This is achieved by mapping several individual single-bit flip-flops into a MBFF (for which there is a layout in the standard cell library), where each flip-flop occupies a dedicated slot as illustrated in Figure 3. The resulting benefit is lower

power consumption, primarily because the clustering significantly reduces the overall number of clock sink pins and shortens the routed clock wirelength (Kahng; Kundu; Thumathy, 2024).

Figure 3 – MBFF Clustering illustration.



Source: Adapted from Kahng, Kundu, and Thumathy (2024).

This process can be executed at various points in the design flow, but typically during synthesis or placement. Despite being a mature feature in modern EDA tools and a subject of decades of research, there is currently no consensus on the optimal stage for this conversion to consistently yield the best QoR (Cai; Zhu; Guo, 2025).

MBFF clustering is frequently referred to by several alternate names, including register clustering or simply flip-flop clustering. However, the latter term should be used with caution, as some works that use the term "flip-flop clustering" do not actually merge SBFFs into pre-designed MBFF. Instead, they focus on grouping SBFFs spatially (relocation), a technique that may be applied when the target standard cell library lacks MBFFs cells. We can say that MBFF clustering is a flip-flop clustering technique but not the other way around.

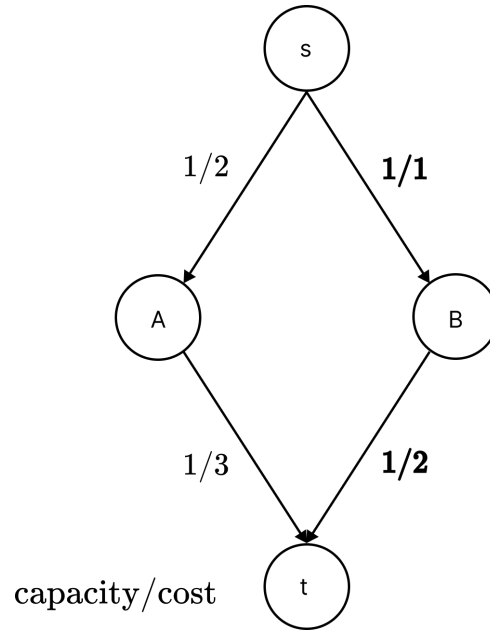
2.3 MIN-COST FLOW AND LINEAR PROGRAMMING

The *Minimum-Cost Flow* (MCF) problem is a network optimization problem defined on a directed graph whose edges have both capacities and costs (Ahuja; Magnanti; Orlin, 1993). The objective is to determine how to send flow through the network while respecting the capacity limits of each edge and minimizing the total transportation cost. MCF can be viewed as a generalization of other well-known graph optimization problems. Unlike the shortest path problem, which considers only costs, and the maximum flow problem, which considers only capacities, the MCF problem takes both aspects into account simultaneously.

Fig. 4 illustrates a simple MCF network. Flow is routed from the source node to the sink node through intermediate nodes. Each edge is associated with a pair (u/c) , where u denotes the maximum amount of flow that can traverse the edge (capacity) and c represents the cost incurred per unit of flow. The objective of the optimization is to determine how much flow should be sent through each path while satisfying all capacity

constraints and minimizing the total accumulated transportation cost. In this example, a unit flow can be routed either through the path $s \rightarrow A \rightarrow t$, which incurs a total cost of $2 + 3 = 5$, or through the path $s \rightarrow B \rightarrow t$, which incurs a total cost of $1 + 2 = 3$. Since both paths satisfy the capacity constraints for a unit flow, the latter corresponds to the minimum-cost solution. Because of its flexibility, MCF is widely used in transportation systems, communication networks, scheduling, assignment problems, and many other optimization applications (Ahuja; Magnanti; Orlin, 1993).

Figure 4 – Example of a MCF network. Assuming a unit flow must be sent from s to t , the path $s \rightarrow B \rightarrow t$ yields the minimum total cost of 3.



Source: Elaborated by the author

Another important optimization technique frequently employed in EDA is *Linear Programming* (LP). LP is a mathematical optimization framework in which a set of decision variables is chosen to maximize or minimize a linear objective function while satisfying a collection of linear equality and inequality constraints (Vanderbei, 2020). In its general form, a LP problem can be expressed as

$$\begin{aligned} & \text{minimize/maximize} && c^T x \\ & \text{subject to} && Ax \leq b, \end{aligned} \tag{2}$$

where x is a vector of decision variables, c defines the objective coefficients, and the matrix A together with vector b specifies the problem constraints. LP formulations are widely used in engineering and operations research because many resource-allocation, scheduling, and geometric optimization problems can be naturally represented using linear relationships.

One of the major advantages of LP is that it can often be solved efficiently for practical optimization problems. Over the years, a variety of highly optimized commercial and open-source solvers have been developed, including Gurobi (Gurobi Optimization, LLC, 2025) and IBM CPLEX (IBM Corporation, 2025). As a result, LP has become a widely used optimization technique in many engineering domains, including several stages of modern EDA flows.

Fig. 5 illustrates a simple LP problem involving two decision variables, x_1 and x_2 . The problem can be formulated as

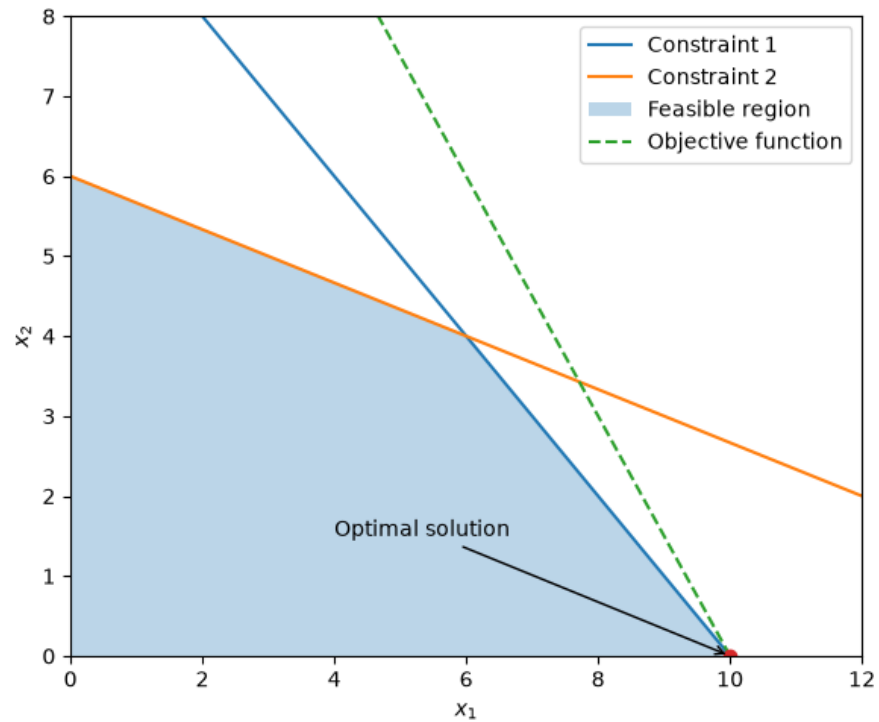
$$\begin{aligned}
 &\text{maximize} && 3x_1 + 2x_2 \\
 &\text{subject to} && x_1 + x_2 \leq 10, \\
 &&& x_1 + 3x_2 \leq 18, \\
 &&& x_1 \geq 0, \\
 &&& x_2 \geq 0.
 \end{aligned} \tag{3}$$

The feasible region, shown as the shaded area, corresponds to the set of points satisfying all constraints simultaneously. The dashed line represents the objective function $3x_1 + 2x_2 = 30$. In a maximization problem, this line is shifted in the direction of increasing objective value until it reaches the last feasible point. In this example, the optimal solution is located at the extreme point $(10, 0)$, highlighted in the figure.

A closely related formulation is *Integer Linear Programming* (ILP), in which some or all decision variables are restricted to integer values. This additional requirement allows ILP to model discrete decisions such as selection, assignment, and placement problems.

MCF, LP, and ILP can be applied in conjunction in large-scale optimization problems. The FTray and SFTray techniques employ all three optimization techniques during the MBFF clustering flow. Initially, an MCF formulation is used to assign flip-flops to candidate MBFFs while minimizing a cost function. Subsequently, an LP formulation refines MBFF positions to improve clustering quality and reduce displacement. Finally, an ILP formulation selects the final set of MBFFs to be instantiated, ensuring that overlap and capacity constraints are satisfied before the resulting clusters are mapped into concrete MBFF cells. Chapter 4 presents a detailed description of the SFTray methodology and explains how these optimization techniques are integrated throughout its clustering flow.

Figure 5 – Example of a LP problem with two decision variables. The shaded region represents the feasible solution space defined by the constraints, while the dashed line represents the objective function. The highlighted point corresponds to the optimal solution.



Source: Elaborated by the author

3 RELATED WORK

The following sections review representative and influential works on flip-flop clustering, ranging from early clustering methodologies to recent state-of-the-art approaches for *Multi-Bit Flip-Flop* (MBFF) generation and optimization.

3.1 WEIGHTED K-MEANS

The work presented by Wu *et al.* (2016) proposes a flip-flop clustering and relocation technique aimed at reducing the overall chip power consumption. The core idea is to form clusters of flip-flops that can be placed near a single clock buffer and connected by a simple routing structure, significantly reducing clock network wirelength and achieving power savings. The technique operates in two main steps: flip-flop clustering and flip-flop relocation.

The flip-flop clustering step utilizes a modified K-means algorithm, called weighted K-means, to assign flip-flops to specific clusters. This proposed modification introduces a weight to the cost function of each cluster based on its current size. This mechanism discourages the assignment of flip-flops to already large clusters, thereby ensuring more balanced clustering results. The objective function is designed to minimize the sum of Manhattan distance displacements between flip-flops and their cluster centers, as well as the total number of clusters (K). Furthermore, the clustering process enforces hard constraints on the maximum allowable displacement for each flip-flop to prevent timing degradation, and limits the maximum number of flip-flops per cluster to satisfy the drive strength requirements of the clock buffer.

In the subsequent flip-flop relocation step, flip-flops are physically moved into legal locations while accounting for placement blockages. The flip-flops within a specific cluster are arranged into a configuration that reserves an empty column over the clock buffer. Clock buffers are then inserted at candidate legal locations, and flip-flops are sequentially relocated to minimize displacement, with the order of movement prioritizing timing-critical flip-flops. Finally, the surrounding combinational cells are legalized to accommodate the changes, while the positions of the newly relocated flip-flops remain fixed.

The technique was evaluated on industrial designs, achieving 9.4% savings in total switching power compared to a non-clustering flow and 4.8% savings compared to an industrial window-based clustering flow. Furthermore, the weighted K-means approach demonstrated 28.3% better total flip-flop displacement compared to the window-based flow, indicating less disruption to the original placement and easier timing closure. (Wu *et al.*, 2016)

It is important to note that, despite being a significant contribution to flip-flop clustering for power optimization, this work does not involve merging SBFFs into physical MBFF cells. Instead, it focuses exclusively on the clustering and spatial relocation of

single-bit flip-flops to share clock buffers, making it a technology-agnostic technique that does not require a specialized MBFF library.

3.2 EFFECTIVE MEAN SHIFT

The work presented by Chang *et al.* (2019) addresses the problem of clock power reduction in modern IC design. The core motivation is that traditional clustering methods often suffer from scalability issues or sensitivity to initialization and outliers, leading to significant placement disruption and timing violations. In contrast, the Effective Mean Shift approach is designed to be robust, efficient, and capable of naturally discovering clusters based on the density of the register distribution.

The algorithm employs a variable bandwidth selection strategy, where the bandwidth is determined by the local density (specifically, the distance to the M -th nearest neighbor) and the timing criticality of the register. This approach allows critical registers to have tighter search windows, minimizing their displacement to prevent timing degradation, while non-critical registers can shift further to form larger clusters.

To enhance computational efficiency and stability, the method only considers "effective neighbors" (identified via K-Nearest Neighbors) during the shift vector computation. By ignoring distant registers and those located beyond the maximum allowable displacement, the algorithm speeds up the process significantly and prevents outliers from distorting the clustering results.

The core of the clustering process involves shifting registers to local density maxima. The algorithm iteratively shifts registers "uphill" on the constructed density surface until they converge to a local peak. Consequently, registers that converge to the same peak are initially grouped into the same cluster.

Following the initial grouping, a post-processing step is applied to refine the clusters. This involves merging nearby peaks to correct approximation errors and subsequently reassigning registers to cluster centers using a stable matching approach. Finally, the clusters are relocated to the median coordinates of their members to further minimize total displacement and improve timing.

Experimental results on 2015 CAD Contest (Viswanathan *et al.*, 2015) benchmarks show that the proposed method outperforms the weighted K-means approach (Wu *et al.*, 2016). It achieves 75% reduction in clock routed wirelength, 46% decrease in clock buffer usage, and 26% savings in clock sink power compared to a non-clustered flow. In comparison to weighted K-means, it offers comparable power savings but with significantly better timing preservation and massive runtime speedups (up to 215x in parallel execution).

3.3 MIXED-DRIVING MBFF

Liu *et al.* (2022) address a specific inefficiency in traditional MBFF generation: the reliance on uniform-driving strength. In standard approaches, if a single bit within an MBFF violates timing constraints, the entire cell must be sized up (e.g., from 1x to 2x driving strength) or decomposed, which unnecessarily increases power consumption for the bits that did not require the higher drive strength. To resolve this, the authors propose a technique for generating mixed-driving MBFFs, where individual bits within the same multi-bit cell can possess different driving capabilities (e.g., a 4-bit MBFF where two bits are 1x and two bits are 2x). The proposed flow operates in two distinct phases to maximize power reduction.

The algorithm begins with an aggressive merging phase designed to maximize the merging rate, thereby minimizing the total clock sink count. This process involves calculating timing-feasible regions for flip-flops under various potential driving strengths and modeling valid merge candidates using rectangle intersection graphs. To select the optimal non-conflicting MBFF candidates, the method employs a cost function that evaluates factors such as merging flexibility, area reduction, and estimated signal wirelength. Notably, this stage explicitly permits the creation of high-driving MBFFs and the inclusion of empty bits within the cells to prioritize the overall reduction of the clock network architecture.

Following the initial merging, the design undergoes legalization and timing analysis. The second phase then refines the design through a timing-driven downsizing process. The algorithm iterates through the generated MBFFs to identify specific bits that possess sufficient timing slack. These identified bits are subsequently "downsized" to lower driving strengths, effectively converting uniform high-driving MBFFs into mixed-driving MBFFs to maximize power savings without introducing timing violations.

Integrated with a commercial Place & Route tool, the approach was evaluated against the Mean Shift algorithm. The results demonstrated significant improvements, including an 11.1% reduction in clock sinks, a 6.1% reduction in flip-flop power, and a 4.9% decrease in normalized total power compared to the uniform-driving baseline. The study highlights that even a limited library of mixed-driving cells can yield substantial power benefits. The issue is that not every library has mixed-driving cells, limiting this technique's range.

3.4 EARLY-STAGE MBFF

Cai, Zhu, and Guo (2025) propose a MBFF clustering methodology applied at the pre-placement stage to maximize clustering flexibility while overcoming the traditional lack of physical and timing awareness. By introducing early-stage estimations, the authors aim to resolve the trade-off between clustering aggressively for power savings and preserving the *Quality of Results* (QoR) in terms of timing and routability.

The proposed flow introduces three key steps after logic synthesis. First, a flip-flop Distance Estimation is performed using a fast coarse placement to identify physically proximate flip-flop pairs, filtering out candidates that would lead to excessive wirelength. Second, a Signal Path Timing Estimation utilizes a pre-trained Natural Language Processing (NLP) based timing predictor to estimate input/output delays for each flip-flop, providing critical timing context absent in standard pre-placement flows. Finally, an MBFF Clustering Algorithm constructs a weighted conflict graph where vertices represent potential mergeable pairs and weights reflect power savings. A greedy selection process then chooses pairs based on a cost function that balances power reduction with timing impact, specifically penalizing pairs with large differences in input-output slack to preserve the efficacy of downstream useful skew optimization.

Cai, Zhu, and Guo (2025) state that experimental evaluations on benchmark circuits demonstrate this approach achieves superior performance compared to a commercial tool without and with MBFF clustering after an initial placement and state-of-the-art pre-placement methods. It delivers an average improvement of 22.5% in *Worst Negative Slack* (WNS) and 33.91% in *Total Negative Slack* (TNS) over commercial tools, while further reducing total power by approximately 3%. The technique is applicable only to 2-bit flip-flops and the extension to flip-flops with more than 2 bits is a future work.

3.5 MULTI-OBJECTIVE MBFF

Li *et al.* (2025) address a critical limitation in existing MBFF clustering methodologies: the neglect of pre-placed cells (fixed gates) during the clustering and legalization phases. Previous approaches often cluster *Single-Bit Flip-Flop* (SBFF)s based solely on proximity or simple timing metrics, without accounting for the significant displacement required to legalize the resulting MBFFs around fixed obstacles. This oversight leads to excessive displacement, causing wirelength increases and severe timing degradation. To solve this, the authors propose a comprehensive MBFF placement methodology that simultaneously optimizes timing, power, area, and bin utilization, specifically designed to handle the constraints imposed by pre-placed cells.

The proposed technique operates in an iterative loop, progressively clustering from larger bit-widths down to smaller ones (e.g., from 4-bit to 1-bit). This process involves three key steps. First, a flip-flop Relocation step employs a timing-driven force model to pull logically related flip-flops closer together, pre-conditioning their positions to minimize timing violations before clustering. Second, a Clustering and Legalization step aggressively groups SBFFs into k-bit MBFFs using a scoring function that penalizes clustering candidates with high displacement sensitivity or low timing slack, ensuring that only robust candidates are merged. Crucially, this step integrates legalization awareness to avoid placing MBFFs in congested regions blocked by pre-placed cells. Finally, an MBFF Declustering step evaluates the formed MBFFs using a multi-objective function.

Any MBFF causing excessive negative slack or failing to meet power/area targets is decomposed back into SBFFs to recover timing.

Experimental results on the 2024 CAD Contest at ICCAD benchmarks (Yang; Hsu; Lee, *et al.*, 2025) demonstrate that this methodology outperforms the top three contest winners. By effectively managing the trade-off between clustering for power and declustering for timing, the approach achieves the best overall score — a composite metric of TNS, power, area, and bin density violations — improving upon the first-place team by 1.2% and the third-place team by 8.8%. However, contest-related techniques may not always translate to a robust QoR when tested against real designs and standard-cell libraries, since timing models are often inaccurate.

3.6 FLOP-TRAY (FTRAY)

Kahng, Li, and Wang (2016) propose a post-placement optimization technique for generating MBFFs, referred to as "flop trays", to minimize clock network power while mitigating the placement and routing congestion often associated with logical clustering. This work explicitly accounts for the physical dimensions and aspect ratios (AR) of large flip-flop trays, supporting sizes up to 64-bit.

The core methodology utilizes an iterative algorithm called Capacitated K-Means Clustering, which alternates between two specific formulations. First, a *Minimum-Cost Flow* (MCF) formulation maps SBFFs to specific slots within candidate flip-flop trays, strictly adhering to capacity constraints. Second, a *Linear Programming* (LP) formulation updates the physical location of the flip-flop trays to minimize the total displacement of the constituent flip-flops.

To finalize the configuration, an *Integer Linear Programming* (ILP) formulation is used to select the optimal subset of candidate trays from a library of mixed sizes, such as 4, 8, 16, 32, and 64-bit. The objective function minimizes a weighted sum of total power, total displacement, and notably, the relative displacement of timing-critical start-end flip-flop pairs to preserve timing integrity.

Experimental results on 28nm FDSOI benchmarks demonstrate that this approach achieves up to 32% reduction in total power and 90% reduction in clock power compared to single-bit implementations. Furthermore, it outperforms commercial logical clustering flows by up to 16% in total power reduction while incurring significantly lower datapath power overheads. (Kahng; Li; Wang, 2016)

Although *Flop-Tray Technique* (FTray) incorporates timing information in its final ILP-based tray selection stage, timing awareness is introduced only through penalty terms associated with critical flip-flop pairs. Consequently, timing criticality does not directly influence the earlier clustering decisions performed by the MCF and LP stages, where assignments and tray locations are determined primarily according to displacement-related objectives.

3.7 SCALABLE DIVIDE-AND-CONQUER FLOP-TRAY (SFTRAY)

Kahng, Kundu, and Thumathy (2024) propose a scalable, divide and conquer approach to MBFF clustering that builds upon the foundational "FlopTray" capacitated k-means methodology. The core innovation is a heuristic that addresses the severe runtime scalability issues of the original FTray algorithm, which relies on solving large MCF instances. By recursively splitting the set of flip-flops into smaller sub-problems (default size $U = 500$) using k-means++ and a Silhouette score-guided merging strategy, the proposed method reduces the complexity of the MCF instances from $O(L^2)$ edges to $O(U^2)$ edges, where L is the total number of flip-flops.

Within each sub-problem, the flow executes the FTray algorithm: an iterative process alternating between MCF to assign flip-flops to MBFF slots and LP to optimize MBFF locations, minimizing a weighted sum of total power, displacement, and timing critical path degradation. This is followed by an ILP step to finalize the selection of instantiated MBFFs. The authors also recalibrate the ILP weighting parameters (α and β) specifically for the ASAP7 7nm research enablement to ensure fair evaluation in modern technology nodes.

Experimental results on designs ranging from 0.5K to 64K clusterable flip-flops demonstrate that this divide-and-conquer strategy achieves massive speedups – up to $400\times$ faster than the original FTray on larger designs like VGA – while incurring only a minor average total power degradation of 1.12%. Furthermore, the method outperforms Mean Shift algorithm, achieving up to 6.18% reduction in total power consumption and improved timing metrics.

Scalable Divide-And-Conquer Flop-Tray Technique (SFTray) serves as the baseline implementation adopted in OpenROAD and can be regarded as one of the current state-of-the-art approaches for post-placement MBFF clustering. Despite preserving the timing-aware optimization technique of FTray, SFTray inherits the same limitation regarding the treatment of timing criticality. Timing information is considered only through discrete penalties in the final ILP objective, whereas the MCF assignment stage remains entirely displacement-driven and the LP refinement stage does not explicitly account for the severity of timing violations. As a result, the clustering process may fail to distinguish between flip-flops participating in highly critical timing paths and those associated with less critical violations.

3.8 SUMMARY

Initial approaches focused on the fundamental algorithmic challenge of grouping spatially correlated instances. Wu *et al.* (2016) established a baseline using a Weighted K-means algorithm to cluster flip-flops for shared clock buffering, balancing cluster sizes to satisfy drive strength limits. However, this technique focuses on "virtual" clustering for

clock network simplification rather than the physical replacement of cells into MBFFs.

Subsequent works advanced the methodology by incorporating explicit physical awareness and rigorous mathematical formulations. Chang *et al.* (2019) introduced Effective Mean Shift to better handle register distribution density and outliers compared to K-means, explicitly targeting physical MBFF creation. Kahng, Li, and Wang (2016) (FTray) pushed the boundaries of cell complexity, handling large aspect ratios and bit-counts (up to 64-bit) using MCF and ILP, albeit with high runtime costs. This scalability issue was later addressed by Kahng, Kundu, and Thumathy (2024), who proposed a Divide and Conquer heuristic to make the rigorous FTray approach computationally viable for large industrial designs.

Overall, the literature demonstrates significant progress in MBFF clustering, ranging from geometric clustering techniques to sophisticated multi-objective optimization frameworks that jointly consider power, timing, area, and legalization constraints. Among these approaches, FTray and, particularly, SFTray stand out due to their rigorous optimization technique based on MCF, LP, and ILP formulations. While FTray established the foundation for optimization-driven MBFF clustering, SFTray introduced a divide-and-conquer strategy that overcomes the scalability limitations of the original approach and enabling its integration into OpenROAD. Table 1 summarizes the representative works reviewed in this chapter.

However, despite the timing-aware nature of the original FTray and SFTray formulations, the treatment of timing criticality remains relatively coarse. Timing information is introduced primarily through discrete penalties in the final ILP objective function, while the earlier clustering stages rely predominantly on displacement-based objectives. In particular, the MCF assignment cost is determined exclusively by Manhattan distance, without explicitly considering timing criticality, and the LP refinement stage does not differentiate between highly critical and non-critical flip-flops.

These observations suggest that there is still room to improve the integration of timing information throughout the clustering process. This work investigates this opportunity by extending the SFTray optimization flow with timing-aware formulations that incorporate slack-based information directly into the clustering stages.

Table 1 – Comparison of representative works on flip-flop and MBFF clustering.

Work	Clust. type		Flow Stage	Benchmark / Tech.	Open Source	Main Idea
Weighted Means	K-	FF	Post-placement	Industrial	No	Weighted K-means clustering balancing cluster sizes while limiting displacement.
Effective Shift	Mean	MBFF	Post-placement	ICCAD'15	Yes	Mean-shift clustering with adaptive bandwidths to naturally handle density variations.
FlopTray (FTray)		MBFF	Post-placement	28nm FDSOI	Yes (OpenROAD)	Optimization-based clustering using MCF, LP and ILP formulations.
Scalable (SFTray)	FTray	MBFF	Post-placement	ASAP7	Yes (OpenROAD)	Recursive divide-and-conquer strategy to improve the scalability of FTray.
Mixed-Driving MBFF		MBFF	Post-placement	ICCAD'15	No	Allows different drive strengths within the same MBFF cell to further reduce power.
Early-Stage MBFF		MBFF	Pre-placement	IWLS'05 / OpenCores	No	Performs MBFF clustering before placement using estimated timing and physical information.
Multi-Objective MBFF		MBFF	Post-placement	ICCAD'24	No	Jointly optimizes power, timing, area, congestion and legalization.

4 THE SCALABLE DIVIDE-AND-CONQUER FTRAY APPROACH

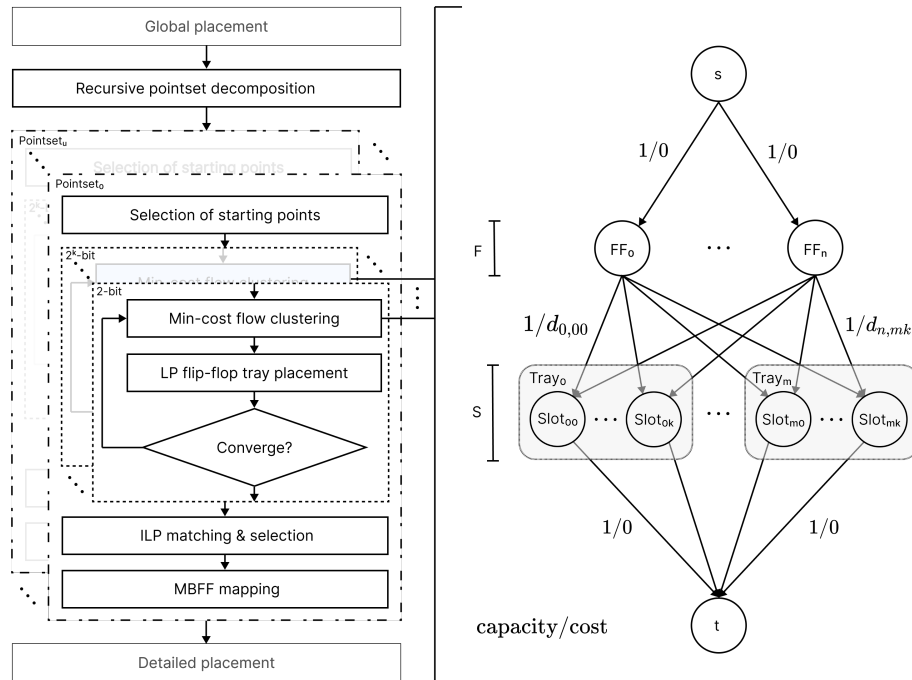
SFTray (Kahng; Kundu; Thumathy, 2024) extends the original FTray technique (Kahng; Li; Wang, 2016) by introducing a scalable divide-and-conquer strategy for MBFF clustering. Figure 6 illustrates the SFTray flow adopted in this work. On the left, the design is recursively decomposed into localized pointsets, allowing candidate MBFF sizes to be explored independently within each partition. This decomposition significantly reduces the computational complexity of the clustering problem and enables the application of FTray to large-scale designs. The right-hand side of the figure illustrates the corresponding MCF assignment graph used to map flip-flops to candidate tray slots.

Starting from a globally placed design containing only SBFFs, the method recursively partitions the design into localized subproblems and, within each partition, iteratively performs two optimization stages:

1. MCF-based assignment of flip-flops to MBFF tray slots;
2. LP-based refinement of tray locations.

A tray is a candidate placement structure representing a potential MBFF instance and its available bit positions (slots). During optimization, flip-flops are assigned to tray slots, and selected trays are later mapped to MBFF cells in the final design.

Figure 6 – Overview of the SFTray divide-and-conquer flow and the MCF assignment graph.



Source: Adapted from Kahng, Li, and Wang (2016)

At the core of each iteration, SFTray constructs the assignment graph illustrated on the right-hand side of Fig. 6. The graph consists of a set F of flip-flop nodes, a set S of candidate tray-slot nodes, a supersource s , and a supersink t . Edges from s to flip-flops and from tray slots to t have unit capacity and zero cost. Between the two sides, each feasible flip-flop-to-slot pair $(i, j) \in F \times S$ is connected by an edge with capacity one and an associated assignment cost.

In the original SFTray formulation, this cost corresponds solely to the Manhattan distance between the current position of flip-flop i and candidate tray slot j :

$$C(i, j) = d(i, j) \quad (4)$$

where $d(i, j)$ denotes the Manhattan distance between the two locations. Consequently, the MCF stage determines a capacity-feasible assignment that minimizes the total displacement required to cluster SBFFs into candidate MBFF trays.

Given the assignment produced by the MCF stage, SFTray applies a Linear Programming (LP) refinement step to determine improved tray locations. While the MCF formulation determines which flip-flops are assigned to each tray slot, the tray positions themselves remain adjustable. The objective of the LP stage is therefore to relocate the trays so that the total displacement of the assigned flip-flops is minimized.

Following Kahng, Kundu, and Thumathy (2024), the LP formulation can be expressed as

$$\min \sum_l d_l \quad (5)$$

subject to

$$d_l = |X_i + x'_{ij} - x_l| + |Y_i + y'_{ij} - y_l| \quad \forall h_l \quad (6)$$

where (X_i, Y_i) denotes the center location of tray i , (x'_{ij}, y'_{ij}) represents the relative position of slot j within the tray, and (x_l, y_l) corresponds to the location of flip-flop h_l . Variable d_l represents the Manhattan displacement between a flip-flop and its assigned tray slot, while D denotes the total displacement accumulated over all assigned flip-flops. Constraint (6) computes the displacement associated with each assigned flip-flop. Consequently, the LP stage determines tray locations that minimize the total displacement induced by the current clustering solution.

The MCF assignment and LP refinement stages are executed iteratively. After the LP solver updates tray positions, a new MCF assignment problem is constructed using the refined locations, potentially producing improved clustering solutions. The process continues until no further displacement reduction is achieved between consecutive iterations.

After all candidate MBFF sizes have been processed, an Integer Linear Programming (ILP) formulation selects the final set of MBFF instances by minimizing

$$\min (\alpha \cdot W + D + \beta \cdot R) \quad (7)$$

where W denotes MBFF total power consumption,

$$\sum_l d_l$$

represents total displacement, and R corresponds to the relative displacement of timing-critical launch-capture flip-flop pairs. Parameters α and β control the trade-off between power minimization and timing preservation.

Although timing information enters the flow through the final ILP selection stage, the MCF-based clustering that precedes it is driven exclusively by geometric displacement. Consequently, timing relationships between flip-flops are not explicitly considered during the iterative assignment phase, potentially limiting the influence of timing information on early clustering decisions. This observation motivates the timing-aware enhancement proposed in the next chapter.

5 ENHANCEMENTS TO THE SFTRAY TECHNIQUE

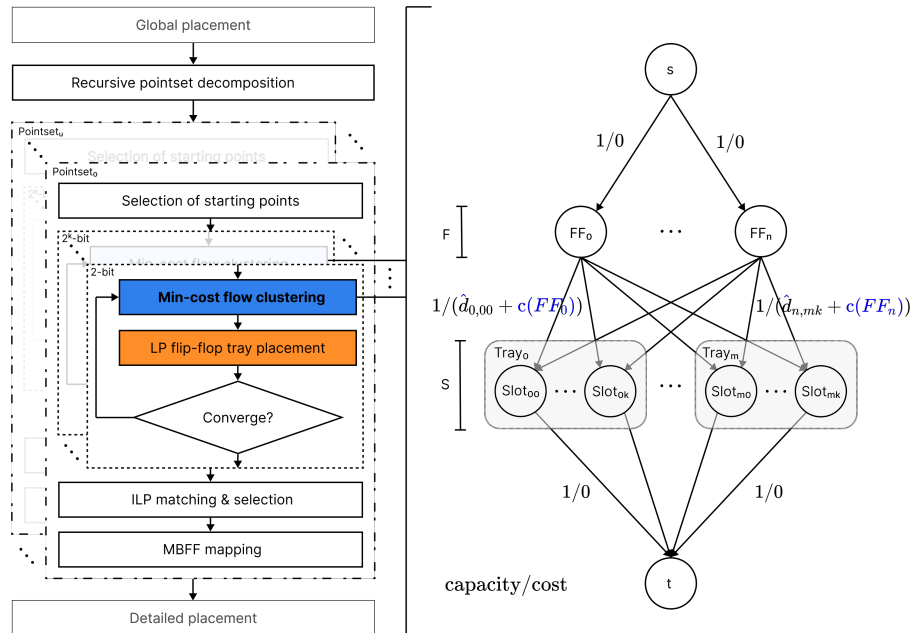
This chapter presents two timing-aware enhancements to the SFTray technique. The first one targets the MCF assignment stage, while the second one modifies the LP-based tray-refinement stage. Both modifications leverage the same flip-flop criticality metric derived from static timing analysis. For clarity, hereinafter the version that modifies only the MCF assignment stage will be referred to as MCF-Crit, whereas the version that modifies both the MCF and LP-based tray-refinement stage will be referred to as MCF+LP-Crit.

Figure 7 highlights the location of the proposed modifications within the SFTray flow. While the overall divide-and-conquer strategy, ILP selection, and MBFF mapping stages remain unchanged, timing-criticality information is incorporated into both iterative optimization stages of SFTray. In MCF-Crit (in blue) the MCF assignment cost is augmented with a timing-criticality component. In MCF+LP-Crit (in blue and orange), the LP-based tray-refinement objective is also modified to prioritize the displacement reduction of timing-critical flip-flops.

5.1 MCF-CRIT: ADDING NORMALIZED TIMING-CRITICALITY TO THE MCF

The timing criticality assigned to a flip-flop is derived from the set of violating timing paths in which the flip-flop participates. For a given flip-flop i , we consider the set $P^-(i)$ of all violating timing paths for which i appears as either the launch or capture

Figure 7 – Overview of the SFTray divide-and-conquer flow and the MCF assignment graph used in the proposed timing-aware formulation.



Source: Adapted from Kahng, Li, and Wang (2016)

endpoint. The criticality of flip-flop i is defined as

$$c(i) = \max_{p \in P^-(i)} \left(\frac{|\text{slack}(p)|}{|\text{WNS}_{\text{FF}}|} \right) \quad (8)$$

where $\text{slack}(p)$ denotes the slack of timing path p , WNS_{FF} is the worst negative slack among all analyzed timing paths from launch to capture flip-flops, and $P^-(i)$ contains only paths with negative slack. Consequently, the most critical timing violation in which a flip-flop participates determines its final criticality score. By construction, criticality values are normalized to the interval $[0, 1]$, where a value of one corresponds to participation in a path whose slack is equal to the global worst negative slack.

Algorithm 1 summarizes the procedure used to compute flip-flop criticalities. The algorithm first determines the global WNS_{FF} across all violating timing paths from launch to capture flip-flops (lines 4-7) and then assigns to each flip-flop the largest normalized timing violation among all paths in which it participates (lines 8-14). Since criticality values are propagated to both launch and capture flip-flops, timing information is distributed throughout the set of sequential elements involved in violating paths.

Figure 8 illustrates the proposed criticality computation with an example. In this example, flip-flop FF_i participates in three timing paths: two of them violate timing constraints, with slacks of -54.84ps and -57.65ps , while the third path exhibits positive slack and is therefore excluded from the criticality calculation. Given a global WNS_{FF} of -86.88ps , the criticality of FF_i is determined by the most severe violating path, resulting in a criticality score of 0.664. Thus, only violating paths contribute to the criticality score, and the largest normalized violation determines the final value assigned to the flip-flop.

Algorithm 1: Flip-Flop Criticality Computation

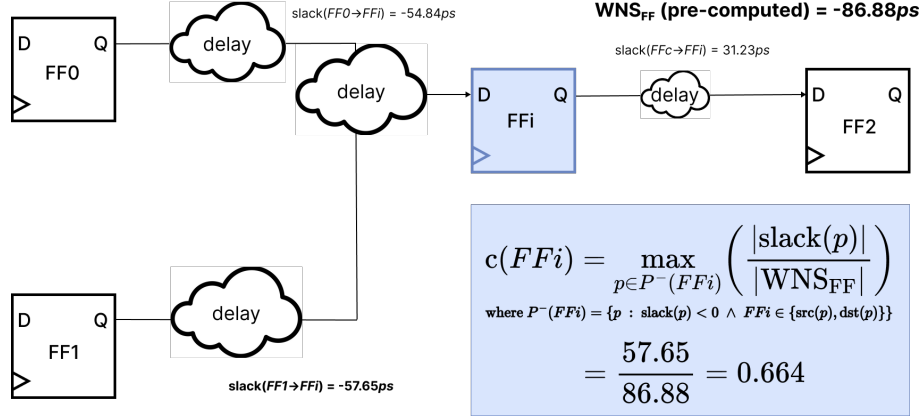
Input: Set of timing paths P from launch to capture flip-flops and their slacks
Output: Criticality value $c(i)$ for each flip-flop

```

1 foreach flip-flop  $i$  do
2   |  $c(i) \leftarrow 0$ ;
3 end
4  $\text{WNS}_{\text{FF}} \leftarrow 0$ ;
5 foreach timing path  $p \in P$  do
6   |  $\text{WNS}_{\text{FF}} \leftarrow \min(\text{WNS}_{\text{FF}}, \text{slack}(p))$ ;
7 end
8 foreach timing path  $p(i, j) \in P$  do
9   | if  $\text{slack}(p) < 0$  then
10    |    $\text{crit} \leftarrow \frac{|\text{slack}(p)|}{|\text{WNS}_{\text{FF}}|}$ ;
11    |    $c(i) \leftarrow \max(c(i), \text{crit})$ ;
12    |    $c(j) \leftarrow \max(c(j), \text{crit})$ ;
13    | end
14 end
15 return  $c(i)$  for all flip-flops;

```

Figure 8 – Example of flip-flop criticality computation. The criticality of a flip-flop is defined as the largest normalized timing violation among all negative-slack paths in which it participates as either a launch or capture endpoint.



Source: Elaborated by the author

As discussed in Chapter 4, the original SFTray formulation assigns a cost equal to the Manhattan distance $d(i, j)$ between flip-flop i and tray slot j . We replace this cost with a normalized, criticality-augmented formulation:

$$C(i, j) = \hat{d}(i, j) + c(i) \quad (9)$$

where the normalized displacement is defined as

$$\hat{d}(i, j) = \frac{d(i, j)}{d_{\max}} \quad (10)$$

and $d_{\max}$ is the maximum Manhattan distance observed across all flip-flop-to-slot candidates in the current partition. Normalization maps both terms to the interval $[0, 1]$, ensuring that displacement and timing criticality contribute at comparable magnitudes.

The proposed formulation increases the influence of timing-critical flip-flops on the global MCF optimization objective. Consequently, assignments involving highly critical flip-flops become more significant during the capacity-constrained optimization process, allowing timing information to affect clustering decisions at an earlier stage of the SFTray flow.

Having computed flip-flop criticalities, algorithm 2 summarizes the modified MCF assignment procedure. First, the maximum displacement observed among all feasible flip-flop-slot pairs is computed (line 1) to be used to normalize Manhattan distances (lines 5-6). Subsequently, the criticality associated with each flip-flop is incorporated into the assignment costs of all candidate slots belonging to that flip-flop (line 7). The resulting cost matrix is then provided to the MCF solver (line 10), which determines the final flip-flop-to-slot assignments (line 11).

The MCF-Crit version corresponds to adding the timing-criticality solely to the

Algorithm 2: Timing-Aware Edge-Cost Computation for MCF Assignment

Input: Flip-flops F , candidate tray slots S
Output: Cost matrix C used by the MCF solver

- 1 Compute

$$d_{\max} = \max_{(i,j) \in F \times S} d(i, j)$$
- 2 **foreach** *flip-flop* $f_i \in F$ **do**
- 3 Compute criticality $c(i)$ using Alg. 1;
- 4 **foreach** *candidate slot* $s_j \in S$ **do**
- 5 $d(i, j) \leftarrow$ Manhattan distance between f_i and s_j ;
- 6 $\hat{d}(i, j) \leftarrow d(i, j)/d_{\max}$;
- 7 $C(i, j) \leftarrow \hat{d}(i, j) + c(i)$;
- 8 **end**
- 9 **end**
- 10 Solve the resulting MCF problem using costs $C(i, j)$;
- 11 Extract flip-flop-to-slot assignments from the flow solution;

MCF assignment stage. LP refinement, ILP selection, legalization, and MBFF mapping remain unchanged with respect to the original SFTray flow. A second timing-aware enhancement (MCF+LP-Crit), targeting the LP refinement stage, is described in the following section.

5.2 MCF+LP-CRIT: ADDING TIMING-CRITICALITY TO LP

The second enhancement proposed in this work targets the LP-based tray-placement refinement stage described in Chapter 4. As previously stated in Equations 5 and 6, the LP stage determines tray locations by minimizing the total displacement of the flip-flops assigned during the MCF stage.

Although this formulation effectively reduces the overall displacement, it treats all flip-flops equally regardless their timing significance. Consequently, displacement reductions achieved for non-critical flip-flops contribute to the objective in the same manner as reductions achieved for timing-critical flip-flops.

To address this limitation, the criticality metric defined in Equation 8 is incorporated directly into the LP objective function. The proposed formulation minimizes a criticality-weighted displacement:

$$\min D_{\text{crit}} \tag{11}$$

with

$$D_{\text{crit}} = \sum_i (1 + c(i)) (d_x^i + d_y^i) \tag{12}$$

where $c(i)$ denotes the criticality associated with flip-flop i .

The factor $(1 + c(i))$ assigns larger weights to highly critical flip-flops while preserving a non-zero contribution from non-critical ones. Since criticality values are normalized to the interval $[0, 1]$, each displacement term receives a weight between 1 and 2.

As a consequence, the LP solver prioritizes reducing the displacement of flip-flops involved in severe timing violations, potentially preserving timing quality during tray relocation. The remaining LP constraints remain identical to those of the original SFTray formulation, ensuring that tray locations continue to satisfy the same geometric feasibility requirements.

Together, the proposed MCF and LP enhancements propagate timing information throughout the entire iterative optimization loop of SFTray. The MCF stage influences how flip-flops are assigned to candidate tray slots, whereas the LP stage favors tray placements that reduce the displacement of timing-critical flip-flops. Consequently, timing information affects both clustering and tray-location refinement before the final ILP-based tray-selection stage. This contrasts with the original SFTray formulation, in which timing awareness is introduced only during the final tray-selection process.

6 EXPERIMENTAL RESULTS

6.1 METHODOLOGY

This chapter presents the experimental methodology adopted to evaluate the proposed timing-aware MBFF clustering approach (MCF-Crit and MCF+LP-Crit), as well as the corresponding results obtained using a complete physical-design flow. The implementation was developed within the OpenROAD framework and evaluated using a set of open-source benchmark circuits synthesized with the ASAP7 technology library. The objective of the experiments is to compare the original SFTray implementation with the proposed enhancements in terms of timing quality, power consumption, area, and MBFF clustering effectiveness.

The proposed enhancements were implemented in C++ within the SFTray clustering engine available in OpenROAD (Kahng, 2022). Following the methodology adopted by the original SFTray work (Kahng; Kundu; Thumathy, 2024), the experiments employ the ASAP7 predictive FinFET standard-cell library with MBFF cells. The MBFF library, clustering infrastructure, and implementation flow provided by SFTray were preserved to ensure a direct comparison between the original SFTray and the proposed implementation.

All experiments were executed on a workstation equipped with an AMD Ryzen 9 9950X processor operating at up to 5.75 GHz, 32 GB of RAM, running Ubuntu 24.04.4 LTS.

To ensure realistic timing evaluation, all reported metrics were collected after detailed routing, except timing metrics that were also evaluated after global placement. Since timing closure results from the interaction among placement, clock-tree synthesis, routing, and post-route optimization stages, evaluating the designs mainly after detailed routing provides a more representative assessment of the overall impact of MBFF clustering on final design quality.

The benchmark circuits used in this work are summarized in Table 2. They cover a diverse set of digital designs, ranging from small arithmetic and control circuits, such as *gcd* and *uart*, to larger communication and processor-oriented designs, including *ethmac*, *cva6*, and *ibex*. The selected benchmarks also exhibit significant variation in sequential-cell density, with flip-flops representing between 3.3% and 17.3% of the total number of cell instances. Since MBFF clustering directly targets sequential elements, this characteristic strongly influences the number of clustering opportunities available within each design.

Designs such as *ethmac* and *mock-alu* contain a relatively large proportion of flip-flops and therefore offer greater opportunities for MBFF clustering. In contrast, designs such as *aes* contain a smaller fraction of sequential elements and consequently provide more limited clustering potential. Four design configurations were evaluated:

- **NC**: original non-clustered OpenROAD implementation;

Table 2 – Benchmark circuits used in the experimental evaluation and their sequential-cell composition before MBFF clustering.

Design	#cells	#FFs	FF %
ethmac	60948	10546	17.3%
mock-alu	13454	1413	10.5%
riscv32i	10879	1056	9.7%
uart	854	79	9.3%
ibex	21734	1939	8.9%
jpeg	57382	4383	7.6%
gcd	468	35	7.5%
cva6	117754	8465	7.2%
aes	17106	562	3.3%

- **SFTray**: SFTray implementation available in OpenROAD;
- **MCF-Crit**: SFTray enhanced with the timing-aware formulation proposed in MCF.
- **MCF+LP-Crit**: SFTray enhanced with the timing-aware formulation proposed in MCF and LP.

Following the exploration methodology proposed by Kahng, Kundu, and Thumathy (2024), multiple values of the parameter α were experimentally evaluated in the range $[2, 256]$. The complete exploration results are reported in Appendix B. Although larger values occasionally achieved superior results for individual metrics on specific benchmarks, they also tended to increase timing degradation and result variability. In contrast, $\alpha = 8$ consistently provided significant power reductions while maintaining stable timing behavior and competitive area improvements across the benchmark suite. Consequently, $\alpha = 8$ was selected as the best overall trade-off between clustering effectiveness, power reduction, and timing preservation. The parameter β was fixed at 0.1, following the recommendation as adopted in the original SFTray formulation (Kahng; Kundu; Thumathy, 2024).

6.2 RESULTS AND DISCUSSION

Table 3 depicts the results for WNS and TNS after Global Placement (GP) and Detailed Routing (DR) in picoseconds and also shows the difference $\Delta(DR - GP)$ between WNS or TNS within the same version. It is important to note that for the MBFF clustering techniques, WNS and TNS are in absolute terms along with their percentage relative to NC. If baseline’s WNS or TNS is negative, a smaller (more negative) percentage means improvement. Otherwise, if baseline’s WNS or TNS is positive – the case for riscv32i and jpeg – a bigger (more positive) percentage means improvement and every negative percentage that does not exceed -100% is not harmful for WNS, since preserves a positive slack. Values in bold indicate that MCF-Crit or MCF+LP-Crit achieved a better WNS or

Table 3 – Worst Negative Slack (WNS) and Total Negative Slack (TNS) comparison after Global Placement (GP) and Detailed Routing (DR) for $\alpha = 8$ and $\beta = 0.1$. All values are in picoseconds (ps). NC represents the non-clustered baseline. For the MBFF clustering techniques, WNS and TNS are in absolute terms along with their percentage relative to NC. If baseline’s WNS/TNS is negative, a smaller (more negative) percentage means improvement. Otherwise, if baseline’s WNS/TNS is positive a bigger (more positive) percentage means improvement (and every negative percentage that does not exceed -100% is not harmful for WNS, since preserves a positive slack). Bold values indicate that MCF-Crit or MCF+LP-Crit achieved a better WNS or TNS timing value than SFTray. $\Delta(DR - GP)$ is simply the difference between WNS/TNS within the same version.

Design	Version	WNS (ps)			TNS (ps)		
		GP	DR	$\Delta(DR - GP)$	GP	DR	$\Delta(DR - GP)$
ethmac	NC	-86.88	-86.27	+0.62	-1492.25	-1464.06	+28.19
	SFTray	-86.88 (+0.00%)	-82.04 (-4.90%)	+4.84	-1492.25 (+0.00%)	-1397.42 (-4.55%)	+94.83
	MCF-Crit	-86.88 (+0.00%)	-81.38 (-5.66%)	+5.50	-1492.25 (+0.00%)	-1369.36 (-6.47%)	+122.89
	MCF+LP-Crit	-86.88 (+0.00%)	-81.38 (-5.66%)	+5.50	-1492.25 (+0.00%)	-1369.36 (-6.47%)	+122.89
mock-alu	NC	-315.57	-283.77	+31.81	-21899.50	-12987.00	+8912.50
	SFTray	-343.18 (+8.75%)	-302.11 (+6.46%)	+41.07	-35900.10 (+63.93%)	-16891.60 (+30.07%)	+19008.50
	MCF-Crit	-315.58 (+0.00%)	-281.32 (-0.86%)	+34.26	-35672.40 (+62.89%)	-15439.70 (+18.89%)	+20232.70
	MCF+LP-Crit	-342.13 (+8.41%)	-300.12 (+5.76%)	+42.01	-35997.10 (+64.37%)	-16783.30 (+29.23%)	+19213.80
uart	NC	-64.65	-28.18	+36.47	-1601.70	-636.38	+965.32
	SFTray	-65.08 (+0.66%)	-22.87 (-18.84%)	+42.21	-1796.85 (+12.18%)	-624.99 (-1.79%)	+1171.86
	MCF-Crit	-65.16 (+0.79%)	-19.69 (-30.13%)	+45.47	-1722.20 (+7.52%)	-525.07 (-17.49%)	+1197.13
	MCF+LP-Crit	-65.16 (+0.79%)	-19.69 (-30.13%)	+45.47	-1722.20 (+7.52%)	-525.07 (-17.49%)	+1197.13
ibex	NC	-29.78	-22.04	+7.74	-10320.60	-166.71	+10153.89
	SFTray	-142.44 (+378.30%)	-43.83 (+98.86%)	+98.61	-114973.00 (+1014.01%)	-1005.98 (+503.42%)	+113967.02
	MCF-Crit	-142.33 (+377.94%)	-35.90 (+62.88%)	+106.43	-114834.00 (+1012.67%)	-432.02 (+159.14%)	+114401.98
	MCF+LP-Crit	-142.33 (+377.94%)	-35.79 (+62.38%)	+106.54	-114806.00 (+1012.40%)	-400.23 (+140.07%)	+114405.77
ged	NC	5.94	-16.70	-22.64	0.00	-79.20	-79.20
	SFTray	-14.61 (-345.93%)	-29.24 (+75.05%)	-14.63	-88.69	-215.44 (+172.03%)	-126.75
	MCF-Crit	-17.74 (-398.70%)	-30.51 (+82.64%)	-12.77	-52.27	-226.54 (+186.04%)	-174.28
	MCF+LP-Crit	-17.74 (-398.70%)	-28.32 (+69.57%)	-10.58	-139.97	-171.45 (+116.49%)	-31.48
cva6	NC	-38.08	-21.31	+16.76	-105.85	-146.32	-40.47
	SFTray	-38.08 (+0.00%)	-22.99 (+7.86%)	+15.09	-105.85 (+0.00%)	-253.49 (+73.24%)	-147.64
	MCF-Crit	-38.08 (+0.00%)	-15.31 (-28.15%)	+22.76	-105.85 (+0.00%)	-101.33 (-30.75%)	+4.52
	MCF+LP-Crit	-64.80 (+70.20%)	4.75 (-122.30%)	+69.56	-154.34 (+45.81%)	0.00 (-100.00%)	+154.34
aes	NC	-27.10	-10.26	+16.84	-637.80	-103.07	+534.73
	SFTray	-40.56 (+49.63%)	-33.98 (+231.19%)	+6.58	-3648.14 (+471.99%)	-2052.00 (+1890.90%)	+1596.14
	MCF-Crit	-40.34 (+48.84%)	-36.24 (+253.24%)	+4.10	-3793.26 (+494.74%)	-2017.41 (+1857.34%)	+1775.85
	MCF+LP-Crit	-42.08 (+55.27%)	-32.88 (+220.56%)	+9.20	-4009.93 (+528.71%)	-2041.88 (+1881.08%)	+1968.05
riscv32i	NC	-135.54	21.09	+156.63	-66001.90	0.00	+66001.90
	SFTray	-133.13 (-1.77%)	18.42 (-12.66%)	+151.55	-66935.80 (+1.41%)	0.00	+66935.80
	MCF-Crit	-131.54 (-2.95%)	17.89 (-15.16%)	+149.44	-66921.30 (+1.39%)	0.00	+66921.30
	MCF+LP-Crit	-131.54 (-2.95%)	17.89 (-15.16%)	+149.44	-66921.30 (+1.39%)	0.00	+66921.30
jpeg	NC	-23.88	16.63	+40.52	-210.86	0.00	+210.86
	SFTray	-30.35 (+27.07%)	20.99 (+26.18%)	+51.34	-361.36 (+71.37%)	0.00	+361.36
	MCF-Crit	-28.45 (+19.13%)	10.27 (-38.24%)	+38.73	-341.76 (+62.08%)	0.00	+341.76
	MCF+LP-Crit	-27.60 (+15.57%)	17.11 (+2.87%)	+44.72	-339.80 (+61.15%)	0.00	+339.80

TNS value compared to SFTray. The design order follows the FF % ordering from Table 2, except for riscv32i and jpeg that are separated because of their positive WNS after DR – one of the reasons in which they’ll be discussed separately.

For the three designs with the highest flip-flop densities (*ethmac*, *mock-alu*, and *uart*), all clustering techniques produced similar WNS values immediately after GP. After DR, however, *ethmac* and *uart* achieved better final WNS than the NC baseline for all clustering approaches. In contrast, *mock-alu* showed only a modest improvement, with MCF-Crit being the only version to slightly outperform NC, while SFTray and MCF+LP-Crit remained close to, but below, the baseline. TNS exhibited a similar trend. Although *mock-alu* experienced a substantial degradation immediately after GP, much of this loss

was recovered during DR, but without surpassing the NC implementation. Conversely, both *ethmac* and *uart* achieved better final TNS than the baseline.

The *cva6* benchmark exhibited a similar behavior. WNS remained comparable after GP for SFTray and MCF-Crit, whereas MCF+LP-Crit introduced a larger initial degradation. Nevertheless, both enhancements recovered a significant amount of slack during DR, with MCF+LP-Crit ultimately achieving positive WNS. SFTray also recovered timing but remained slightly below the NC implementation. Regarding TNS, all techniques exhibited similar values after GP, but only the proposed formulations achieved substantial improvements after routing. MCF+LP-Crit completely eliminated negative slack (TNS = 0), while MCF-Crit improved TNS by 30.75% relative to NC. In contrast, SFTray was unable to improve upon the baseline.

Different behavior was observed for *ibex*, *gcd*, and *aes*. For these designs, the clustering techniques produced worse WNS immediately after GP and were unable to recover sufficient slack during DR to outperform the NC implementation. In particular, *gcd* experienced additional WNS degradation between GP and DR. Similar trends were observed for TNS. The *ibex* and *aes* benchmarks exhibited substantially larger negative TNS values after GP, and although DR reduced these violations, the clustered implementations remained worse than NC. The *gcd* design also developed negative TNS after GP, which further deteriorated during routing. Among all benchmarks, *aes* exhibited the largest TNS degradation.

Comparing MCF-Crit with the original SFTray implementation, the proposed enhancement consistently outperforms it for five designs (*ethmac*, *mock-alu*, *uart*, *ibex*, and *cva6*) when considering the final WNS and TNS after DR. For *aes*, MCF-Crit also achieves a slight improvement in final TNS, although both implementations still exhibit significant timing violations.

MCF+LP-Crit also improves upon SFTray in terms of final WNS and TNS for all seven benchmark circuits whose NC implementation exhibits negative WNS after DR. The most significant improvement is observed for *cva6*, where the proposed formulation completely eliminates timing violations, achieving a positive final WNS and a TNS of zero. Overall, these results indicate that the proposed enhancements preserve the timing behavior of the original SFTray flow while providing additional opportunities for timing improvement, particularly in designs with higher sequential-cell densities.

For the *riscv32i* and *jpeg* benchmarks, although the overall design exhibited a negative WNS after GP, the criticality computation did not identify any timing-violating paths whose launch and capture endpoints were both flip-flops. Consequently, all flip-flops received zero criticality scores, and the timing-aware term in the proposed cost function had no influence on the clustering process.

Nevertheless, MCF-Crit and MCF+LP-Crit still achieved slightly better WNS and TNS than the original SFTray implementation after GP. Since the only difference in

the MCF formulation under these conditions is the normalization of the flip-flop-to-slot distance within each partition, these results suggest that distance normalization itself may have a positive effect on the assignment process. Furthermore, after DR, all three clustering techniques achieved positive WNS, resulting in a final TNS of zero for both benchmarks.

Although these observations indicate that the normalization of the distance term may contribute to improved clustering decisions independently of timing criticality, the individual contributions of distance normalization and criticality weighting cannot be isolated from the current experiments. A more detailed investigation of their respective impacts remains an interesting direction for future work.

Table 4 summarizes the results for MBFF distribution, Clock Power, Switching Power, Total Power, and Area. The NC rows present absolute values, whereas SFTray, MCF-Crit, and MCF+LP-Crit report the corresponding percentage variation relative to NC. Columns #1b, #2b, and #4b indicate the final number of 1-bit, 2-bit, and 4-bit flip-flops, respectively.

As expected, the benefits of MBFF clustering strongly depend on the sequential content of the design. Circuits with higher flip-flop densities consistently exhibit larger power reductions after clustering. For example, *ethmac*, whose flip-flops account for 17.3% of all instances (Table 2), achieves approximately 39% reduction in total power, whereas *aes*, with only 3.3% flip-flops, shows negligible power improvement regardless of the clustering strategy.

Compared with SFTray, MCF-Crit improved both switching and total power in five of the nine evaluated benchmarks (*ethmac*, *mock-alu*, *riscv32i*, *uart*, and *jpeg*), all of which correspond to designs with relatively high flip-flop densities. Clock power was also reduced in seven benchmarks, while area was improved in five (*ethmac*, *mock-alu*, *uart*, *ibex*, and *cva6*). These results suggest that incorporating the proposed enhancement during the MCF assignment stage can improve clustering decisions without affecting MBFF conversion effectiveness.

The LP enhancement produced identical results to MCF-Crit for *ethmac*, *uart*, and *riscv32i*, indicating that, for these designs, the LP refinement converged to the same solution obtained after the MCF stage. For the remaining benchmarks, however, the additional optimization modified the clustering solution to varying degrees.

Among these cases, *mock-alu* and *jpeg* benefited the most consistently from the LP enhancement. In both circuits, MCF+LP-Crit achieved lower switching and total power than MCF-Crit while also maintaining or improving area. The most noticeable changes were observed for *cva6*. Although MCF-Crit slightly increased switching and total power relative to NC, MCF+LP-Crit reversed this behavior, reducing switching power by 3.32% and total power by 1.93%. It also further reduced clock power and the number of clock buffers, indicating that the additional LP optimization can substantially influence the

Table 4 – Comparison of MBFF distribution, Clock Power, Switching Power, Total Power and Area for $\alpha = 8$ and $\beta = 0.1$. NC rows present absolute values, while SFTray, MCF-Crit, and MCF+LP-Crit show absolute values and percentage variation relative to NC (except for #1b, #2b and #4b, which are the number of 1-bit flip-flops, 2-bit flip-flops and 4-bit flip-flops, respectively). A negative percentage is always better. Bold values indicate that MCF-Crit or MCF+LP-Crit achieved a better result than SFTray. All metrics were extracted after detailed routing.

Design	Version	#1b	#2b	#4b	FF %	Clk. P (mW)	Swit. P (mW)	Total P (mW)	Area (μm^2)
ethmac	NC (abs.)	10546	0	0	17.3%	51.000	24.878	111.321	7935.720
	SFTray	1396	31	2272	6.9%	21.500 (-57.84%)	13.523 (-45.64%)	67.949 (-38.96%)	7748.320 (-2.36%)
	MCF-Crit	1380	35	2274	6.9%	21.300 (-58.24%)	13.437 (-45.99%)	67.652 (-39.23%)	7745.510 (-2.40%)
	MCF+LP-Crit	1380	35	2274	6.9%	21.300 (-58.24%)	13.437 (-45.99%)	67.652 (-39.23%)	7745.510 (-2.40%)
mock-alu	NC (abs.)	1413	0	0	10.5%	6.680	27.989	58.655	1632.380
	SFTray	59	9	334	3.2%	2.430 (-63.62%)	26.323 (-5.95%)	52.034 (-11.29%)	1613.040 (-1.18%)
	MCF-Crit	63	11	332	3.3%	2.410 (-63.92%)	26.201 (-6.39%)	51.808 (-11.67%)	1602.060 (-1.86%)
	MCF+LP-Crit	49	8	337	3.2%	2.410 (-63.92%)	26.146 (-6.58%)	51.728 (-11.81%)	1601.260 (-1.91%)
uart	NC (abs.)	79	0	0	9.3%	0.257	0.806	2.330	95.207
	SFTray	0	0	20	2.6%	0.052 (-79.77%)	0.706 (-12.46%)	2.082 (-10.67%)	92.977 (-2.34%)
	MCF-Crit	1	1	19	2.7%	0.071 (-72.26%)	0.703 (-12.79%)	2.079 (-10.78%)	92.058 (-3.31%)
	MCF+LP-Crit	1	1	19	2.7%	0.071 (-72.26%)	0.703 (-12.79%)	2.079 (-10.78%)	92.058 (-3.31%)
ibex	NC (abs.)	1939	0	0	8.9%	2.920	12.254	25.188	2541.780
	SFTray	1691	2	61	8.2%	2.720 (-6.85%)	12.087 (-1.36%)	24.613 (-2.28%)	2529.090 (-0.50%)
	MCF-Crit	1693	1	61	8.1%	2.710 (-7.19%)	12.313 (+0.48%)	25.255 (+0.27%)	2555.220 (+0.53%)
	MCF+LP-Crit	1697	1	60	8.2%	2.730 (-6.51%)	12.086 (-1.38%)	24.621 (-2.25%)	2531.020 (-0.42%)
gcd	NC (abs.)	35	0	0	7.5%	0.099	0.323	0.918	48.347
	SFTray	5	3	6	3.1%	0.043 (-56.98%)	0.287 (-11.15%)	0.818 (-10.89%)	48.172 (-0.36%)
	MCF-Crit	11	2	5	4.0%	0.050 (-48.99%)	0.295 (-8.86%)	0.836 (-8.94%)	48.129 (-0.45%)
	MCF+LP-Crit	3	0	8	2.4%	0.057 (-42.11%)	0.293 (-9.44%)	0.852 (-7.27%)	49.251 (+1.87%)
cva6	NC (abs.)	8465	0	0	7.2%	17.500	46.961	132.122	17586.400
	SFTray	6686	668	374	6.6%	17.200 (-1.71%)	47.191 (+0.49%)	131.893 (-0.17%)	17886.800 (+1.71%)
	MCF-Crit	6681	706	380	6.6%	17.100 (-2.29%)	47.846 (+1.88%)	132.515 (+0.30%)	17915.700 (+1.87%)
	MCF+LP-Crit	6683	687	378	6.6%	16.700 (-4.57%)	45.403 (-3.32%)	129.574 (-1.93%)	17888.500 (+1.72%)
aes	NC (abs.)	562	0	0	3.3%	2.500	98.881	166.145	1838.640
	SFTray	106	12	108	1.3%	1.280 (-48.80%)	99.148 (+0.27%)	166.447 (+0.18%)	1853.060 (+0.78%)
	MCF-Crit	102	12	109	1.3%	1.160 (-53.60%)	99.244 (+0.37%)	166.735 (+0.36%)	1854.810 (+0.88%)
	MCF+LP-Crit	94	12	111	1.3%	1.130 (-54.80%)	99.147 (+0.27%)	166.340 (+0.12%)	1852.560 (+0.76%)
riscv32i	NC (abs.)	1056	0	0	9.7%	2.560	8.428	34.073	2691.150
	SFTray	56	4	248	3.1%	0.958 (-62.58%)	7.738 (-8.19%)	31.411 (-7.81%)	2659.700 (-1.17%)
	MCF-Crit	56	4	248	3.1%	0.936 (-63.44%)	7.657 (-9.14%)	31.210 (-8.40%)	2657.030 (-1.27%)
	MCF+LP-Crit	56	4	248	3.1%	0.936 (-63.44%)	7.657 (-9.14%)	31.210 (-8.40%)	2657.030 (-1.27%)
jpeg	NC (abs.)	4383	0	0	7.6%	9.800	56.825	119.187	6436.980
	SFTray	365	49	980	2.6%	4.000 (-59.18%)	54.254 (-4.53%)	109.118 (-8.45%)	6357.010 (-1.24%)
	MCF-Crit	301	41	1000	2.5%	3.980 (-59.39%)	54.200 (-4.62%)	109.032 (-8.52%)	6358.380 (-1.22%)
	MCF+LP-Crit	307	38	1000	2.5%	3.960 (-59.59%)	54.166 (-4.68%)	108.804 (-8.71%)	6354.260 (-1.29%)

final clustering solution in large sequential designs.

For *ibex* and *aes*, the LP enhancement largely restored the power and area characteristics originally achieved by SFTray while preserving most of the improvements introduced by MCF-Crit. The behavior of *gcd*, however, illustrates the multi-objective nature of the problem. Although MCF+LP-Crit produced a denser MBFF distribution, it traded part of the timing improvement achieved by MCF-Crit for modest degradations in power and area.

Overall, MCF+LP-Crit outperformed SFTray in switching and total power for six benchmarks, one more than MCF-Crit, while producing comparable clock-power reductions. These results suggest that extending the proposed enhancement to the LP refinement stage can provide additional opportunities for improving power, area, and clock-network characteristics when tray locations remain sufficiently flexible after the MCF assignment

stage. The magnitude of these gains, however, remains benchmark dependent.

6.3 RUNTIME ANALYSIS

Table 5 presents the runtime comparison between the non-clustered baseline (NC), the original SFTray implementation, and the two timing-aware formulations proposed in this work. The reported runtimes correspond to the global placement stage (GP) and to the complete RTL-to-GDSII flow. Each configuration was executed five times, and the table reports the mean runtime together with the corresponding standard deviation.

As expected, enabling MBFF clustering generally increases the runtime of the global placement stage relative to the non-clustered flow, since additional optimization procedures must be performed during placement. However, the results indicate that the proposed timing-aware enhancements do not introduce a substantial computational overhead compared to the original SFTray implementation. For most benchmarks, the GP runtimes of MCF-Crit and MCF+LP-Crit remain close to those of SFTray, with differences typically limited to a few percentage points.

The reported standard deviations further support this observation. In most benchmarks, the runtime differences among SFTray, MCF-Crit, and MCF+LP-Crit are comparable to the natural run-to-run variability of the flow. For example, the GP runtime of *cva6* differs by less than 2% among the three clustered configurations, while the standard deviations range from approximately 7 to 11 seconds. Similar behavior can be observed for *ibex*, *jpeg*, and *aes*, indicating that the proposed modifications do not significantly alter the computational complexity of the clustering stage.

Some benchmarks exhibit seemingly unusual runtime behavior. In particular, *riscv32i* presents lower average GP runtimes for all clustered configurations relative to the NC flow. However, the NC configuration also exhibits an exceptionally large standard deviation of 29.3 seconds, exceeding its mean runtime. This variability suggests that the observed reduction is primarily attributable to execution-time fluctuations rather than to a systematic speedup introduced by clustering. Since the clustering procedure is executed after global placement optimization, there is no algorithmic reason to expect a consistent reduction in GP runtime.

Likewise, the benchmark *gcd* exhibits relatively large percentage increases in GP runtime. Nevertheless, its absolute execution times remain extremely small, ranging from approximately 1.6 to 1.8 seconds. Therefore, differences of only a few tenths of a second translate into seemingly large percentage variations. Consequently, the reported overhead for *gcd* is not practically significant.

A similar trend can be observed for the total runtime of the complete flow. Although some variability exists across designs, the overall runtime of the proposed methods remains within the same range as that of SFTray. For several benchmarks, including *aes*, *ethmac*, and *mock-alu*, the timing-aware formulations achieve total runtimes comparable to or

Table 5 – Runtime comparison between NC, SFTray, MCF-Crit, and MCF+LP-Crit. NC rows show mean runtime in seconds with standard deviation, while the remaining rows show percentage variation relative to NC together with the corresponding runtime standard deviation in seconds.

Design	Version	GP Runtime (s)	Total Runtime (s)
aes	NC	10.51 ± 0.47	238.62 ± 2.89
	SFTray	$+24.73\% \pm 0.22$	$+42.07\% \pm 1.12$
	MCF-Crit	$+28.37\% \pm 0.40$	$+29.84\% \pm 2.38$
	MCF+LP-Crit	$+30.10\% \pm 0.45$	$+29.30\% \pm 3.03$
cva6	NC	205.77 ± 7.46	2210.45 ± 225.71
	SFTray	$+0.28\% \pm 6.83$	$+13.84\% \pm 680.55$
	MCF-Crit	$-0.47\% \pm 6.77$	$+11.88\% \pm 12.58$
	MCF+LP-Crit	$+0.06\% \pm 11.39$	$+12.63\% \pm 56.16$
ethmac	NC	53.22 ± 5.59	695.88 ± 47.84
	SFTray	$+55.83\% \pm 37.67$	$+20.11\% \pm 84.44$
	MCF-Crit	$+16.81\% \pm 4.43$	$+16.03\% \pm 4.61$
	MCF+LP-Crit	$+21.53\% \pm 2.78$	$+17.39\% \pm 8.47$
gcd	NC	1.61 ± 0.57	36.38 ± 2.00
	SFTray	$+2.85\% \pm 0.38$	$+25.03\% \pm 0.47$
	MCF-Crit	$+6.82\% \pm 0.55$	$+32.15\% \pm 0.90$
	MCF+LP-Crit	$+10.30\% \pm 0.75$	$+32.51\% \pm 1.02$
ibex	NC	18.81 ± 0.96	561.89 ± 25.87
	SFTray	$+11.35\% \pm 0.99$	$+2.68\% \pm 3.52$
	MCF-Crit	$+14.56\% \pm 1.31$	$+17.26\% \pm 3.42$
	MCF+LP-Crit	$+10.52\% \pm 0.54$	$+17.17\% \pm 0.82$
jpeg	NC	51.69 ± 2.02	672.56 ± 59.62
	SFTray	$+4.70\% \pm 6.14$	$+8.45\% \pm 9.40$
	MCF-Crit	$+5.23\% \pm 1.31$	$+9.52\% \pm 7.04$
	MCF+LP-Crit	$+10.27\% \pm 4.09$	$+9.59\% \pm 2.76$
mock-alu	NC	9.30 ± 0.63	308.13 ± 98.78
	SFTray	$+40.91\% \pm 0.12$	$+1.92\% \pm 22.08$
	MCF-Crit	$+42.52\% \pm 0.55$	$-5.74\% \pm 2.84$
	MCF+LP-Crit	$+47.42\% \pm 0.69$	$-6.09\% \pm 1.98$
riscv32i	NC	24.43 ± 29.29	331.93 ± 107.56
	SFTray	$-34.84\% \pm 1.57$	$+8.44\% \pm 69.62$
	MCF-Crit	$-43.03\% \pm 1.12$	$-2.33\% \pm 1.73$
	MCF+LP-Crit	$-36.74\% \pm 0.55$	$-1.27\% \pm 3.70$
uart	NC	1.44 ± 0.30	41.72 ± 2.04
	SFTray	$+93.74\% \pm 0.74$	$+33.90\% \pm 1.09$
	MCF-Crit	$+68.98\% \pm 0.46$	$+47.08\% \pm 1.04$
	MCF+LP-Crit	$+94.44\% \pm 0.61$	$+46.90\% \pm 1.22$

slightly lower than those of the original implementation. These results suggest that the modified optimization objectives do not substantially increase the computational effort required by subsequent physical-design stages.

Overall, the results demonstrate that incorporating timing-criticality information into the MCF assignment stage and the LP refinement stage has a negligible impact on runtime. Therefore, the improvements in power, timing, and physical-design metrics discussed in the previous section are achieved without introducing a meaningful computational penalty relative to the original SFTray flow.

7 CONCLUSION

The increasing importance of power optimization in modern VLSI systems has motivated the widespread adoption of MBFF clustering techniques, particularly due to their ability to reduce clock-network power consumption. Although several MBFF clustering approaches have been proposed in the literature, many existing solutions either rely on partial design flows or consider timing information only during the final stages of optimization.

This work investigated the incorporation of timing-awareness into the SFTray MBFF clustering technique available in OpenROAD. Initially, a comprehensive review of the state of the art was conducted, covering existing clustering techniques and the optimization methodologies employed by FTray and SFTray. Based on the identified limitations, two timing-aware enhancements were proposed. The first modification, referred to as MCF-Crit, incorporates flip-flop timing criticality into the MCF assignment stage. The second modification, referred to as MCF+LP-Crit, extends timing awareness to the LP-based tray-refinement stage.

The proposed approaches were implemented directly within the OpenROAD framework and evaluated using a complete RTL-to-GDSII flow with the ASAP7 technology library. Experimental results demonstrated that timing information can be incorporated into the iterative clustering process without compromising the primary benefits of MBFF clustering. The MCF-Crit formulation improved power, timing, area, or clock-network metrics in several benchmark circuits, particularly in designs with higher flip-flop densities. Furthermore, the MCF+LP-Crit formulation provided additional improvements in selected benchmarks, indicating that timing information can be effectively exploited not only during flip-flop assignment but also during tray-placement refinement.

Overall, the results indicate that timing-awareness is beneficial when introduced earlier in the SFTray optimization flow rather than being restricted to the final ILP-based tray-selection stage. The proposed modifications preserve clustering effectiveness while improving the balance between power optimization and timing preservation, demonstrating the potential of timing-aware clustering strategies within open-source physical-design flows.

As future work, several research directions can be explored. One possibility is the development of adaptive mechanisms capable of automatically selecting suitable values for the SFTray parameters α and β based on design characteristics, such as flip-flop density, timing slack distribution, and circuit size. Another promising direction is the investigation of alternative criticality models that consider complete path information, fanout structure, or path sensitivity metrics instead of relying solely on WNS normalization. In addition, the individual contributions of the proposed distance normalization and criticality weighting mechanisms could be investigated independently to better understand their respective impacts on the clustering process. Another interesting direction would be

to analyze the impact of the proposed enhancements on power immediately after Global Placement, allowing the benefits introduced by clustering itself to be distinguished from those resulting from subsequent optimization stages. A more detailed investigation of the benchmark circuits exhibiting significant timing degradation after clustering would also be valuable. In particular, analyzing the critical paths responsible for large WNS degradations, such as those observed in the *aes* benchmark, could provide insights into the limitations of the current formulations and guide the development of more robust clustering strategies. Finally, applying the proposed methodology to additional technology libraries and larger industrial-scale designs would provide further insight into its scalability and general applicability.

REFERENCES

- AHUJA, Ravindra K.; MAGNANTI, Thomas L.; ORLIN, James B. **Network Flows: Theory, Algorithms, and Applications**. [S. l.]: Prentice Hall, 1993.
- CAI, Yichen; ZHU, Linyu; GUO, Xinfei. 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). New York, NY, USA: ACM, 2025. 7 pages. ASPDAC '25, January 20–23, 2025, Tokyo, Japan. DOI: 10.1145/3658617.3697725.
- CHANG, Ya-Chu *et al.* 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). New York, NY, USA: ACM, 2019. 8 pages. ISPD '19, April 14–17, 2019, San Francisco, CA, USA. DOI: 10.1145/3299902.3309753.
- GUROBI OPTIMIZATION, LLC. **Gurobi Optimizer Reference Manual**. [S. l.], 2025. Accessed: 2026-06-22. Disponível em: <https://www.gurobi.com/documentation/>.
- IBM CORPORATION. **IBM ILOG CPLEX Optimization Studio Documentation**. [S. l.], 2025. Accessed: 2026-06-22. Disponível em: <https://www.ibm.com/docs/en/icos>.
- KAHNG, A.; LIENIG, J., *et al.* **"VLSI Physical Design: From Graph Partitioning to Timing Closure"**. [S. l.]: Springer Netherlands, 2011. DOI: 10.1007/978-90-481-9591-6.
- KAHNG, Andrew B. Leveling Up: A Trajectory of OpenROAD, TILOS and Beyond. *In*: PROCEEDINGS of the 2022 International Symposium on Physical Design. Virtual Event, Canada: Association for Computing Machinery, 2022. (ISPD '22), p. 73–79. DOI: 10.1145/3505170.3511479.
- KAHNG, Andrew B.; KUNDU, Sayak; THUMATHY, Shreyas. Scalable Flip-Flop Clustering Using Divide and Conquer For Capacitated K-Means. *In*: PROCEEDINGS of the Great Lakes Symposium on VLSI 2024. Clearwater, FL, USA: ACM, June 2024. (GLSVLSI '24), 8 pages. DOI: 10.1145/3649476.3658744. Disponível em: <https://doi.org/10.1145/3649476.3658744>.
- KAHNG, Andrew B.; LI, Jiajia; WANG, Lutong. Improved Flop Tray-Based Design Implementation for Power Reduction. *In*: PROCEEDINGS of the 35th International Conference on Computer-Aided Design. Austin, Texas: ACM, Nov. 2016. (ICCAD '16). DOI: 10.1145/2966986.2967047.
- LI, Cheng-Yen *et al.* Late Breaking Results: Multi-Objective Multi-Bit Flip-Flop Placement Considering Pre-Placed Cells. *In*: 2025 62nd ACM/IEEE Design Automation

Conference (DAC). New York, NY, USA: IEEE, 2025. Late Breaking Results. DOI: 10.1109/DAC63849.2025.11133043.

LIU, Meng-Yun *et al.* Generation of Mixed-Driving Multi-Bit Flip-Flops for Power Optimization. *In: PROCEEDINGS of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD '22)*. New York, NY, USA: ACM, 2022. 9 pages. ICCAD '22, October 30–November 3, 2022, San Diego, CA, USA. DOI: 10.1145/3508352.3549473.

MISHRA, Priyambada. **Understanding Logic Equivalence Check (LEC) Flow and Its Challenges and Proposed Solution**. eInfochips. Mar. 2022. Disponível em: <https://www.design-reuse.com/article/61332-understanding-logic-equivalence-check-lec-flow-and-its-challenges-and-proposed-solution/>. Acesso em: 1 dez. 2025.

PROJECT, The OpenROAD. **OpenROAD**. [S. l.]: GitHub, 2025. <https://github.com/The-OpenROAD-Project/OpenROAD>.

VANDERBEI, Robert J. **Linear Programming: Foundations and Extensions**. 5. ed. [S. l.]: Springer, 2020.

VISWANATHAN, Natarajan *et al.* Overview of the 2015 CAD Contest at ICCAD. *In: PROCEEDINGS of the IEEE/ACM International Conference on Computer-Aided Design*. Austin, TX, USA: IEEE Press, 2015. (ICCAD '15), p. 910–911.

WESTE, Neil H.; HARRIS, David Money. **CMOS VLSI Design a Circuit and Systems Perspective**. [S. l.]: Addison-Wesley, 2011.

WU, Gang *et al.* Flip-flop Clustering by Weighted K-means Algorithm. *In: PROCEEDINGS of the 53rd Annual Design Automation Conference (DAC '16)*. New York, NY, USA: ACM, 2016. article 82, 6 pages. DAC '16, June 05–09, 2016, Austin, TX, USA. DOI: 10.1145/2897937.2898025.

YANG, Sheng-Wei; HSU, Jhih-Wei; CHENG, Yu-Hsuan, *et al.* Invited Paper: 2025 ICCAD CAD Contest Problem B: Power and Timing Optimization Using Multibit Flip-Flop. *In: p. 1–6*. DOI: 10.1109/ICCAD66269.2025.11240779.

YANG, Sheng-Wei; HSU, Jhih-Wei; LEE, Ting-Wei, *et al.* 2024 ICCAD CAD Contest Problem B: Power and Timing Optimization Using Multibit Flip-Flop. *In: PROCEEDINGS of the 43rd IEEE/ACM International Conference on Computer-Aided Design*. Newark Liberty International Airport Marriott, New York, NY, USA: Association for Computing Machinery, 2025. (ICCAD '24). DOI: 10.1145/3676536.3689911. Disponível em: <https://doi.org/10.1145/3676536.3689911>.

Appendix

**APPENDIX A – COMPARISON OF MBFF CLUSTERING STRATEGIES FOR
VARYING α VALUES WITH $\beta = 0.1$. TIMING-RELATED ANALYSIS.**

Table 6 – Worst Negative Slack (WNS) and Total Negative Slack (TNS) comparison after Global Placement (GP) and Detailed Routing (DR) for varying α values with $\beta = 0.1$ (Part 1). All values are in picoseconds (ps). NC rows present absolute values, while SFTray, MCF-Crit, and MCF+LP-Crit show absolute values followed by percentage variation relative to NC in parentheses. Bold values indicate that MCF-Crit or MCF+LP-Crit achieved a better WNS or TNS timing value than SFTray.

Design	Version	GP	WNS (ps) DR	$\Delta(DR - GP)$	GP	TNS (ps) DR	$\Delta(DR - GP)$
ethmac	NC (abs.)	-86.88	-86.27	+0.62	-1492.25	-1464.06	+28.19
	SFTray ($\alpha=2$)	-86.88 (+0.00%)	-82.41 (-4.47%)	+4.47	-1492.25 (+0.00%)	-1412.63 (-3.51%)	+79.62
	MCF-Crit ($\alpha=2$)	-86.88 (+0.00%)	-86.45 (+0.21%)	+0.43	-1492.25 (+0.00%)	-1469.30 (+0.36%)	+22.95
	MCF+LP-Crit ($\alpha=2$)	-86.88 (+0.00%)	-86.45 (+0.21%)	+0.43	-1492.25 (+0.00%)	-1469.30 (+0.36%)	+22.95
	SFTray ($\alpha=4$)	-86.88 (+0.00%)	-83.62 (-3.06%)	+3.26	-1492.25 (+0.00%)	-1390.76 (-5.01%)	+101.49
	MCF-Crit ($\alpha=4$)	-86.88 (+0.00%)	-79.83 (-7.45%)	+7.05	-1492.25 (+0.00%)	-1345.03 (-8.13%)	+147.22
	MCF+LP-Crit ($\alpha=4$)	-86.88 (+0.00%)	-79.83 (-7.45%)	+7.05	-1492.25 (+0.00%)	-1345.03 (-8.13%)	+147.22
	SFTray ($\alpha=8$)	-86.88 (+0.00%)	-82.04 (-4.90%)	+4.84	-1492.25 (+0.00%)	-1397.42 (-4.55%)	+94.83
	MCF-Crit ($\alpha=8$)	-86.88 (+0.00%)	-81.38 (-5.66%)	+5.50	-1492.25 (+0.00%)	-1369.36 (-6.47%)	+122.89
	MCF+LP-Crit ($\alpha=8$)	-86.88 (+0.00%)	-81.38 (-5.66%)	+5.50	-1492.25 (+0.00%)	-1369.36 (-6.47%)	+122.89
	SFTray ($\alpha=16$)	-86.88 (+0.00%)	-84.55 (-1.99%)	+2.34	-1492.25 (+0.00%)	-1430.47 (-2.29%)	+61.78
	MCF-Crit ($\alpha=16$)	-86.88 (+0.00%)	-83.72 (-2.95%)	+3.16	-1492.25 (+0.00%)	-1421.33 (-2.92%)	+70.92
	MCF+LP-Crit ($\alpha=16$)	-86.88 (+0.00%)	-83.72 (-2.95%)	+3.16	-1492.25 (+0.00%)	-1421.33 (-2.92%)	+70.92
	SFTray ($\alpha=67$)	-86.88 (+0.00%)	-83.39 (-3.34%)	+3.49	-1492.25 (+0.00%)	-1408.02 (-3.83%)	+84.23
	MCF-Crit ($\alpha=67$)	-86.88 (+0.00%)	-86.36 (+0.12%)	+0.52	-1492.25 (+0.00%)	-1407.76 (-3.85%)	+84.49
	MCF+LP-Crit ($\alpha=67$)	-86.88 (+0.00%)	-86.36 (+0.12%)	+0.52	-1492.25 (+0.00%)	-1407.76 (-3.85%)	+84.49
	SFTray ($\alpha=130$)	-86.88 (+0.00%)	-83.39 (-3.34%)	+3.49	-1492.25 (+0.00%)	-1408.02 (-3.83%)	+84.23
	MCF-Crit ($\alpha=130$)	-86.88 (+0.00%)	-86.36 (+0.12%)	+0.52	-1492.25 (+0.00%)	-1407.76 (-3.85%)	+84.49
	MCF+LP-Crit ($\alpha=130$)	-86.88 (+0.00%)	-86.36 (+0.12%)	+0.52	-1492.25 (+0.00%)	-1407.76 (-3.85%)	+84.49
	SFTray ($\alpha=193$)	-86.88 (+0.00%)	-83.39 (-3.34%)	+3.49	-1492.25 (+0.00%)	-1408.02 (-3.83%)	+84.23
	MCF-Crit ($\alpha=193$)	-86.88 (+0.00%)	-82.61 (-4.24%)	+4.27	-1492.25 (+0.00%)	-1358.10 (-7.24%)	+134.15
	MCF+LP-Crit ($\alpha=193$)	-86.88 (+0.00%)	-82.61 (-4.24%)	+4.27	-1492.25 (+0.00%)	-1358.10 (-7.24%)	+134.15
	SFTray ($\alpha=256$)	-86.88 (+0.00%)	-83.39 (-3.34%)	+3.49	-1492.25 (+0.00%)	-1408.02 (-3.83%)	+84.23
	MCF-Crit ($\alpha=256$)	-86.88 (+0.00%)	-82.61 (-4.24%)	+4.27	-1492.25 (+0.00%)	-1358.10 (-7.24%)	+134.15
	MCF+LP-Crit ($\alpha=256$)	-86.88 (+0.00%)	-82.61 (-4.24%)	+4.27	-1492.25 (+0.00%)	-1358.10 (-7.24%)	+134.15
mock-ahu	NC (abs.)	-315.57	-283.77	+31.81	-21899.50	-12987.00	+8912.50
	SFTray ($\alpha=2$)	-315.58 (+0.00%)	-286.84 (+1.08%)	+28.73	-28570.40 (+30.46%)	-13865.00 (+6.76%)	+14705.40
	MCF-Crit ($\alpha=2$)	-315.57 (+0.00%)	-285.84 (+0.73%)	+29.73	-27191.90 (+24.17%)	-13841.00 (+6.58%)	+13350.90
	MCF+LP-Crit ($\alpha=2$)	-315.57 (+0.00%)	-277.19 (-2.32%)	+38.38	-27400.10 (+25.12%)	-13777.10 (+6.08%)	+13623.00
	SFTray ($\alpha=4$)	-337.28 (+6.88%)	-311.20 (+9.67%)	+26.08	-33936.80 (+54.97%)	-15466.40 (+19.09%)	+18470.40
	MCF-Crit ($\alpha=4$)	-329.44 (+4.39%)	-303.24 (+6.86%)	+26.20	-33124.90 (+51.26%)	-15768.10 (+21.41%)	+17356.80
	MCF+LP-Crit ($\alpha=4$)	-333.42 (+5.66%)	-301.99 (+6.42%)	+31.43	-32641.40 (+49.05%)	-16673.40 (+28.39%)	+15968.00
	SFTray ($\alpha=8$)	-343.18 (+8.75%)	-302.11 (+6.46%)	+41.07	-35900.10 (+63.93%)	-16891.60 (+30.07%)	+19008.50
	MCF-Crit ($\alpha=8$)	-315.58 (+0.00%)	-281.32 (-0.86%)	+34.26	-35672.40 (+62.89%)	-15439.70 (+18.89%)	+20232.70
	MCF+LP-Crit ($\alpha=8$)	-342.13 (+8.41%)	-300.12 (+5.76%)	+42.01	-35997.10 (+64.37%)	-16783.30 (+29.23%)	+19213.80
	SFTray ($\alpha=16$)	-343.18 (+8.75%)	-309.00 (+8.89%)	+34.18	-35771.20 (+63.34%)	-15575.60 (+19.93%)	+20195.60
	MCF-Crit ($\alpha=16$)	-315.70 (+0.04%)	-282.29 (-0.52%)	+33.41	-36549.10 (+66.89%)	-16326.60 (+25.71%)	+20222.50
	MCF+LP-Crit ($\alpha=16$)	-342.25 (+8.45%)	-320.35 (+12.89%)	+21.90	-35945.90 (+64.14%)	-17224.90 (+32.63%)	+18721.00
	SFTray ($\alpha=67$)	-343.18 (+8.75%)	-304.46 (+7.29%)	+38.72	-35792.20 (+63.44%)	-17393.30 (+33.93%)	+18398.90
	MCF-Crit ($\alpha=67$)	-338.28 (+7.20%)	-298.40 (+5.16%)	+39.88	-36502.20 (+66.68%)	-15416.70 (+18.71%)	+21085.50
	MCF+LP-Crit ($\alpha=67$)	-342.25 (+8.45%)	-315.96 (+11.34%)	+26.29	-35949.60 (+64.16%)	-17340.40 (+33.52%)	+18609.20
	SFTray ($\alpha=130$)	-343.18 (+8.75%)	-304.46 (+7.29%)	+38.72	-35792.20 (+63.44%)	-17393.30 (+33.93%)	+18398.90
	MCF-Crit ($\alpha=130$)	-315.70 (+0.04%)	-292.39 (+3.04%)	+23.31	-36552.80 (+66.91%)	-17619.10 (+35.67%)	+18933.70
	MCF+LP-Crit ($\alpha=130$)	-342.25 (+8.45%)	-315.96 (+11.34%)	+26.29	-35949.60 (+64.16%)	-17340.40 (+33.52%)	+18609.20
	SFTray ($\alpha=193$)	-343.18 (+8.75%)	-304.46 (+7.29%)	+38.72	-35792.20 (+63.44%)	-17393.30 (+33.93%)	+18398.90
	MCF-Crit ($\alpha=193$)	-315.70 (+0.04%)	-281.00 (-0.97%)	+34.69	-36080.80 (+64.76%)	-16077.00 (+23.79%)	+20003.80
	MCF+LP-Crit ($\alpha=193$)	-342.25 (+8.45%)	-315.96 (+11.34%)	+26.29	-35949.60 (+64.16%)	-17340.40 (+33.52%)	+18609.20
	SFTray ($\alpha=256$)	-343.18 (+8.75%)	-304.46 (+7.29%)	+38.72	-35792.20 (+63.44%)	-17393.30 (+33.93%)	+18398.90
	MCF-Crit ($\alpha=256$)	-338.28 (+7.20%)	-298.40 (+5.16%)	+39.88	-36502.20 (+66.68%)	-15416.70 (+18.71%)	+21085.50
	MCF+LP-Crit ($\alpha=256$)	-342.25 (+8.45%)	-315.96 (+11.34%)	+26.29	-35949.60 (+64.16%)	-17340.40 (+33.52%)	+18609.20

Table 7 – Worst Negative Slack (WNS) and Total Negative Slack (TNS) comparison after Global Placement (GP) and Detailed Routing (DR) for varying α values with $\beta = 0.1$ (Part 2). All values are in picoseconds (ps). NC rows present absolute values, while SFTray, MCF-Crit, and MCF+LP-Crit show absolute values followed by percentage variation relative to NC in parentheses. Bold values indicate that MCF-Crit or MCF+LP-Crit achieved a better WNS or TNS timing value than SFTray.

Design	Version	WNS (ps)			TNS (ps)		
		GP	DR	$\Delta(DR - GP)$	GP	DR	$\Delta(DR - GP)$
uart	NC (abs.)	-64.65	-28.18	+36.47	-1601.70	-636.38	+965.32
	SFTray ($\alpha=2$)	-64.65 (+0.00%)	-22.50 (-20.13%)	+42.14	-1622.63 (+1.31%)	-598.38 (-5.97%)	+1024.25
	MCF-Crit ($\alpha=2$)	-64.65 (+0.00%)	-28.99 (+2.90%)	+35.65	-1701.69 (+6.24%)	-762.91 (+19.88%)	+938.78
	MCF+LP-Crit ($\alpha=2$)	-64.65 (+0.00%)	-28.99 (+2.90%)	+35.65	-1701.69 (+6.24%)	-762.91 (+19.88%)	+938.78
	SFTray ($\alpha=4$)	-65.08 (+0.66%)	-25.92 (-7.99%)	+39.15	-1639.30 (+2.35%)	-681.38 (+7.07%)	+957.92
	MCF-Crit ($\alpha=4$)	-65.16 (+0.79%)	-21.49 (-23.74%)	+43.67	-1711.21 (+6.84%)	-698.30 (+9.73%)	+1012.91
	MCF+LP-Crit ($\alpha=4$)	-65.16 (+0.79%)	-21.49 (-23.74%)	+43.67	-1711.21 (+6.84%)	-698.30 (+9.73%)	+1012.91
	SFTray ($\alpha=8$)	-65.08 (+0.66%)	-22.87 (-18.84%)	+42.21	-1796.85 (+12.18%)	-624.99 (-1.79%)	+1171.86
	MCF-Crit ($\alpha=8$)	-65.16 (+0.79%)	-19.69 (-30.13%)	+45.47	-1722.20 (+7.52%)	-525.07 (-17.49%)	+1197.13
	MCF+LP-Crit ($\alpha=8$)	-65.16 (+0.79%)	-19.69 (-30.13%)	+45.47	-1722.20 (+7.52%)	-525.07 (-17.49%)	+1197.13
	SFTray ($\alpha=16$)	-65.08 (+0.66%)	-22.87 (-18.84%)	+42.21	-1796.85 (+12.18%)	-624.99 (-1.79%)	+1171.86
	MCF-Crit ($\alpha=16$)	-65.16 (+0.79%)	-19.69 (-30.13%)	+45.47	-1722.20 (+7.52%)	-525.07 (-17.49%)	+1197.13
	MCF+LP-Crit ($\alpha=16$)	-65.16 (+0.79%)	-19.69 (-30.13%)	+45.47	-1722.20 (+7.52%)	-525.07 (-17.49%)	+1197.13
	SFTray ($\alpha=67$)	-65.08 (+0.66%)	-22.87 (-18.84%)	+42.21	-1796.85 (+12.18%)	-624.99 (-1.79%)	+1171.86
	MCF-Crit ($\alpha=67$)	-65.16 (+0.79%)	-21.02 (-25.39%)	+44.14	-1720.57 (+7.42%)	-665.15 (+4.52%)	+1055.42
	MCF+LP-Crit ($\alpha=67$)	-65.16 (+0.79%)	-21.02 (-25.39%)	+44.14	-1720.57 (+7.42%)	-665.15 (+4.52%)	+1055.42
	SFTray ($\alpha=130$)	-65.08 (+0.66%)	-22.87 (-18.84%)	+42.21	-1796.85 (+12.18%)	-624.99 (-1.79%)	+1171.86
	MCF-Crit ($\alpha=130$)	-65.16 (+0.79%)	-21.02 (-25.39%)	+44.14	-1720.57 (+7.42%)	-665.15 (+4.52%)	+1055.42
	MCF+LP-Crit ($\alpha=130$)	-65.16 (+0.79%)	-21.02 (-25.39%)	+44.14	-1720.57 (+7.42%)	-665.15 (+4.52%)	+1055.42
	SFTray ($\alpha=193$)	-65.08 (+0.66%)	-22.87 (-18.84%)	+42.21	-1796.85 (+12.18%)	-624.99 (-1.79%)	+1171.86
	MCF-Crit ($\alpha=193$)	-65.16 (+0.79%)	-21.02 (-25.39%)	+44.14	-1720.57 (+7.42%)	-665.15 (+4.52%)	+1055.42
	MCF+LP-Crit ($\alpha=193$)	-65.16 (+0.79%)	-21.02 (-25.39%)	+44.14	-1720.57 (+7.42%)	-665.15 (+4.52%)	+1055.42
	SFTray ($\alpha=256$)	-65.08 (+0.66%)	-22.87 (-18.84%)	+42.21	-1796.85 (+12.18%)	-624.99 (-1.79%)	+1171.86
	MCF-Crit ($\alpha=256$)	-65.16 (+0.79%)	-21.02 (-25.39%)	+44.14	-1720.57 (+7.42%)	-665.15 (+4.52%)	+1055.42
	MCF+LP-Crit ($\alpha=256$)	-65.16 (+0.79%)	-21.02 (-25.39%)	+44.14	-1720.57 (+7.42%)	-665.15 (+4.52%)	+1055.42
ibex	NC (abs.)	-29.78	-22.04	+7.74	-10320.60	-166.71	+10153.89
	SFTray ($\alpha=2$)	-29.78 (+0.00%)	-19.90 (-9.73%)	+9.88	-10567.70 (+2.39%)	-166.11 (-0.36%)	+10401.59
	MCF-Crit ($\alpha=2$)	-29.78 (+0.00%)	-38.10 (+72.86%)	-8.32	-10564.70 (+2.37%)	-124.14 (-25.54%)	+10440.56
	MCF+LP-Crit ($\alpha=2$)	-50.92 (+70.99%)	-23.41 (+6.21%)	+27.51	-24528.50 (+137.67%)	-125.30 (-24.84%)	+24403.20
	SFTray ($\alpha=4$)	-143.35 (+381.38%)	-35.30 (+60.17%)	+108.05	-115846.00 (+1022.47%)	-614.55 (+268.63%)	+115231.45
	MCF-Crit ($\alpha=4$)	-142.33 (+377.94%)	-50.29 (+128.20%)	+92.04	-114583.00 (+1010.24%)	-3120.06 (+1771.52%)	+111462.94
	MCF+LP-Crit ($\alpha=4$)	-142.33 (+377.94%)	-43.61 (+97.88%)	+98.72	-114595.00 (+1010.35%)	-2207.23 (+1223.97%)	+112387.77
	SFTray ($\alpha=8$)	-142.44 (+378.30%)	-43.83 (+98.86%)	+98.61	-114973.00 (+1014.01%)	-1005.98 (+503.42%)	+113967.02
	MCF-Crit ($\alpha=8$)	-142.33 (+377.94%)	-35.90 (+62.88%)	+106.43	-114834.00 (+1012.67%)	-432.02 (+159.14%)	+114401.98
	MCF+LP-Crit ($\alpha=8$)	-142.33 (+377.94%)	-35.79 (+62.38%)	+106.54	-114806.00 (+1012.40%)	-400.23 (+140.07%)	+114405.77
	SFTray ($\alpha=16$)	-142.44 (+378.30%)	-48.53 (+120.21%)	+93.90	-115043.00 (+1014.69%)	-552.70 (+231.53%)	+114490.30
	MCF-Crit ($\alpha=16$)	-142.33 (+377.94%)	-40.86 (+85.38%)	+101.47	-114889.00 (+1013.20%)	-831.35 (+398.67%)	+114057.65
	MCF+LP-Crit ($\alpha=16$)	-142.33 (+377.94%)	-40.86 (+85.38%)	+101.47	-114909.00 (+1013.39%)	-831.35 (+398.67%)	+114077.65
	SFTray ($\alpha=67$)	-142.44 (+378.30%)	-44.32 (+101.07%)	+98.12	-115066.00 (+1014.92%)	-992.79 (+495.51%)	+114073.21
	MCF-Crit ($\alpha=67$)	-142.33 (+377.94%)	-37.44 (+69.88%)	+104.89	-114907.00 (+1013.38%)	-711.06 (+326.52%)	+114195.94
	MCF+LP-Crit ($\alpha=67$)	-142.33 (+377.94%)	-46.58 (+111.36%)	+95.75	-114932.00 (+1013.62%)	-1673.22 (+903.65%)	+113258.78
	SFTray ($\alpha=130$)	-142.44 (+378.30%)	-44.32 (+101.07%)	+98.12	-115066.00 (+1014.92%)	-992.79 (+495.51%)	+114073.21
	MCF-Crit ($\alpha=130$)	-142.33 (+377.94%)	-37.44 (+69.88%)	+104.89	-114907.00 (+1013.38%)	-711.06 (+326.52%)	+114195.94
	MCF+LP-Crit ($\alpha=130$)	-142.33 (+377.94%)	-46.58 (+111.36%)	+95.75	-114932.00 (+1013.62%)	-1673.22 (+903.65%)	+113258.78
	SFTray ($\alpha=193$)	-142.44 (+378.30%)	-44.32 (+101.07%)	+98.12	-115066.00 (+1014.92%)	-992.79 (+495.51%)	+114073.21
	MCF-Crit ($\alpha=193$)	-142.33 (+377.94%)	-37.44 (+69.88%)	+104.89	-114907.00 (+1013.38%)	-711.06 (+326.52%)	+114195.94
	MCF+LP-Crit ($\alpha=193$)	-142.33 (+377.94%)	-46.58 (+111.36%)	+95.75	-114932.00 (+1013.62%)	-1673.22 (+903.65%)	+113258.78
	SFTray ($\alpha=256$)	-142.44 (+378.30%)	-44.32 (+101.07%)	+98.12	-115066.00 (+1014.92%)	-992.79 (+495.51%)	+114073.21
	MCF-Crit ($\alpha=256$)	-142.33 (+377.94%)	-37.44 (+69.88%)	+104.89	-114907.00 (+1013.38%)	-711.06 (+326.52%)	+114195.94
	MCF+LP-Crit ($\alpha=256$)	-142.33 (+377.94%)	-46.58 (+111.36%)	+95.75	-114932.00 (+1013.62%)	-1673.22 (+903.65%)	+113258.78

Table 8 – Worst Negative Slack (WNS) and Total Negative Slack (TNS) comparison after Global Placement (GP) and Detailed Routing (DR) for varying α values with $\beta = 0.1$ (Part 3). All values are in picoseconds (ps). NC rows present absolute values, while SFTray, MCF-Crit, and MCF+LP-Crit show absolute values followed by percentage variation relative to NC in parentheses. Bold values indicate that MCF-Crit or MCF+LP-Crit achieved a better WNS or TNS timing value than SFTray.

Design	Version	WNS (ps)			TNS (ps)		
		GP	DR	$\Delta(DR - GP)$	GP	DR	$\Delta(DR - GP)$
gcd	NC (abs.)	5.94	-16.70	-22.64	0.00	-79.20	-79.20
	SFTray ($\alpha=2$)	-14.61 (-345.93%)	-25.59 (+53.19%)	-10.98	-34.57	-154.41 (+94.96%)	-119.84
	MCF-Crit ($\alpha=2$)	-14.61 (-345.93%)	-22.93 (+37.29%)	-8.33	-34.57	-122.62 (+54.83%)	-88.05
	MCF+LP-Crit ($\alpha=2$)	-14.61 (-345.93%)	-25.59 (+53.19%)	-10.98	-34.57	-154.41 (+94.96%)	-119.84
	SFTray ($\alpha=4$)	-14.61 (-345.93%)	-30.94 (+85.20%)	-16.33	-34.57	-242.93 (+206.73%)	-208.36
	MCF-Crit ($\alpha=4$)	-14.61 (-345.93%)	-34.88 (+108.79%)	-20.27	-34.57	-247.70 (+212.76%)	-213.13
	MCF+LP-Crit ($\alpha=4$)	-14.61 (-345.93%)	-31.88 (+90.83%)	-17.27	-47.60	-218.59 (+176.01%)	-171.00
	SFTray ($\alpha=8$)	-14.61 (-345.93%)	-29.24 (+75.05%)	-14.63	-88.69	-215.44 (+172.03%)	-126.75
	MCF-Crit ($\alpha=8$)	-17.74 (-398.70%)	-30.51 (+82.64%)	-12.77	-52.27	-226.54 (+186.04%)	-174.28
	MCF+LP-Crit ($\alpha=8$)	-17.74 (-398.70%)	-28.32 (+69.57%)	-10.58	-139.97	-171.45 (+116.49%)	-31.48
	SFTray ($\alpha=16$)	-15.88 (-367.40%)	-26.98 (+61.52%)	-11.10	-140.28	-188.90 (+138.52%)	-48.62
	MCF-Crit ($\alpha=16$)	-17.74 (-398.70%)	-31.10 (+86.20%)	-13.36	-52.27	-256.17 (+223.46%)	-203.90
	MCF+LP-Crit ($\alpha=16$)	-17.74 (-398.70%)	-28.32 (+69.57%)	-10.58	-139.97	-171.45 (+116.49%)	-31.48
	SFTray ($\alpha=67$)	-15.88 (-367.40%)	-26.98 (+61.52%)	-11.10	-140.28	-188.90 (+138.52%)	-48.62
	MCF-Crit ($\alpha=67$)	-17.74 (-398.70%)	-31.10 (+86.20%)	-13.36	-52.27	-256.17 (+223.46%)	-203.90
	MCF+LP-Crit ($\alpha=67$)	-17.74 (-398.70%)	-28.32 (+69.57%)	-10.58	-139.97	-171.45 (+116.49%)	-31.48
	SFTray ($\alpha=130$)	-15.88 (-367.40%)	-26.98 (+61.52%)	-11.10	-140.28	-188.90 (+138.52%)	-48.62
	MCF-Crit ($\alpha=130$)	-17.74 (-398.70%)	-31.10 (+86.20%)	-13.36	-52.27	-256.17 (+223.46%)	-203.90
	MCF+LP-Crit ($\alpha=130$)	-17.74 (-398.70%)	-28.32 (+69.57%)	-10.58	-139.97	-171.45 (+116.49%)	-31.48
	SFTray ($\alpha=193$)	-15.88 (-367.40%)	-26.98 (+61.52%)	-11.10	-140.28	-188.90 (+138.52%)	-48.62
	MCF-Crit ($\alpha=193$)	-17.74 (-398.70%)	-31.10 (+86.20%)	-13.36	-52.27	-256.17 (+223.46%)	-203.90
	MCF+LP-Crit ($\alpha=193$)	-17.74 (-398.70%)	-28.32 (+69.57%)	-10.58	-139.97	-171.45 (+116.49%)	-31.48
	SFTray ($\alpha=256$)	-15.88 (-367.40%)	-26.98 (+61.52%)	-11.10	-140.28	-188.90 (+138.52%)	-48.62
	MCF-Crit ($\alpha=256$)	-17.74 (-398.70%)	-31.10 (+86.20%)	-13.36	-52.27	-256.17 (+223.46%)	-203.90
	MCF+LP-Crit ($\alpha=256$)	-17.74 (-398.70%)	-28.32 (+69.57%)	-10.58	-139.97	-171.45 (+116.49%)	-31.48
cva6	NC (abs.)	-38.08	-21.31	+16.76	-105.85	-146.32	-40.47
	SFTray ($\alpha=2$)	-38.08 (+0.00%)	-21.99 (+3.19%)	+16.08	-105.85 (+0.00%)	-166.39 (+13.72%)	-60.54
	MCF-Crit ($\alpha=2$)	-38.08 (+0.00%)	-20.15 (-5.44%)	+17.92	-105.85 (+0.00%)	-211.30 (+44.41%)	-105.45
	MCF+LP-Crit ($\alpha=2$)	-64.80 (+70.20%)	4.74 (-122.25%)	+69.55	-154.34 (+45.81%)	0.00 (-100.00%)	+154.34
	SFTray ($\alpha=4$)	-38.08 (+0.00%)	-17.22 (-19.21%)	+20.86	-105.85 (+0.00%)	-113.92 (-22.15%)	-8.06
	MCF-Crit ($\alpha=4$)	-38.08 (+0.00%)	-28.40 (+33.24%)	+9.68	-105.85 (+0.00%)	-434.40 (+196.88%)	-328.55
	MCF+LP-Crit ($\alpha=4$)	-64.80 (+70.20%)	-1.18 (-94.49%)	+63.63	-154.34 (+45.81%)	-1.18 (-99.20%)	+153.17
	SFTray ($\alpha=8$)	-38.08 (+0.00%)	-22.99 (+7.86%)	+15.09	-105.85 (+0.00%)	-253.49 (+73.24%)	-147.64
	MCF-Crit ($\alpha=8$)	-38.08 (+0.00%)	-15.31 (-28.15%)	+22.76	-105.85 (+0.00%)	-101.33 (-30.75%)	+4.52
	MCF+LP-Crit ($\alpha=8$)	-64.80 (+70.20%)	4.75 (-122.30%)	+69.56	-154.34 (+45.81%)	0.00 (-100.00%)	+154.34
	SFTray ($\alpha=16$)	-38.08 (+0.00%)	-27.07 (+27.03%)	+11.00	-105.85 (+0.00%)	-242.44 (+65.69%)	-136.59
	MCF-Crit ($\alpha=16$)	-38.08 (+0.00%)	-11.06 (-48.12%)	+27.02	-105.85 (+0.00%)	-55.91 (-61.79%)	+49.95
	MCF+LP-Crit ($\alpha=16$)	-64.80 (+70.20%)	-0.20 (-99.05%)	+64.60	-154.34 (+45.81%)	-0.20 (-99.86%)	+154.14
	SFTray ($\alpha=67$)	-38.08 (+0.00%)	-17.05 (-20.02%)	+21.03	-105.85 (+0.00%)	-221.44 (+51.33%)	-115.58
	MCF-Crit ($\alpha=67$)	-38.08 (+0.00%)	-32.15 (+50.83%)	+5.93	-105.85 (+0.00%)	-537.15 (+267.10%)	-431.30
	MCF+LP-Crit ($\alpha=67$)	-64.80 (+70.20%)	-2.60 (-87.79%)	+62.20	-154.34 (+45.81%)	-2.60 (-98.22%)	+151.74
	SFTray ($\alpha=130$)	-38.08 (+0.00%)	-24.69 (+15.86%)	+13.38	-105.85 (+0.00%)	-259.55 (+77.38%)	-153.70
	MCF-Crit ($\alpha=130$)	-38.08 (+0.00%)	-23.59 (+10.70%)	+14.48	-105.85 (+0.00%)	-134.09 (-8.36%)	-28.24
	MCF+LP-Crit ($\alpha=130$)	-64.80 (+70.20%)	-5.39 (-74.72%)	+59.42	-154.34 (+45.81%)	-5.39 (-96.32%)	+148.96
	SFTray ($\alpha=193$)	-38.08 (+0.00%)	-21.43 (+0.57%)	+16.64	-105.85 (+0.00%)	-131.02 (-10.46%)	-25.17
	MCF-Crit ($\alpha=193$)	-38.08 (+0.00%)	-24.82 (+16.45%)	+13.26	-105.85 (+0.00%)	-141.20 (-3.50%)	-35.35
	MCF+LP-Crit ($\alpha=193$)	-64.80 (+70.20%)	-3.42 (-83.94%)	+61.38	-154.34 (+45.81%)	-4.00 (-97.27%)	+150.34
	SFTray ($\alpha=256$)	-38.08 (+0.00%)	-15.40 (-27.76%)	+22.68	-105.85 (+0.00%)	-114.46 (-21.78%)	-8.61
	MCF-Crit ($\alpha=256$)	-38.08 (+0.00%)	-15.86 (-25.58%)	+22.22	-105.85 (+0.00%)	-93.88 (-35.84%)	+11.97
	MCF+LP-Crit ($\alpha=256$)	-64.80 (+70.20%)	-17.28 (-18.90%)	+47.52	-154.34 (+45.81%)	-31.86 (-78.22%)	+122.48

Table 9 – Worst Negative Slack (WNS) and Total Negative Slack (TNS) comparison after Global Placement (GP) and Detailed Routing (DR) for varying α values with $\beta = 0.1$ (Part 4). All values are in picoseconds (ps). NC rows present absolute values, while SFTray, MCF-Crit, and MCF+LP-Crit show absolute values followed by percentage variation relative to NC in parentheses. Bold values indicate that MCF-Crit or MCF+LP-Crit achieved a better WNS or TNS timing value than SFTray.

Design	Version	GP	WNS (ps) DR	$\Delta(DR - GP)$	GP	TNS (ps) DR	$\Delta(DR - GP)$
aes	NC (abs.)	-27.10	-10.26	+16.84	-637.80	-103.07	+534.73
	SFTray ($\alpha=2$)	-33.30 (+22.85%)	-23.19 (+126.02%)	+10.11	-1415.89 (+122.00%)	-699.81 (+578.98%)	+716.08
	MCF-Crit ($\alpha=2$)	-31.48 (+16.13%)	-14.96 (+45.85%)	+16.51	-1463.77 (+129.50%)	-367.06 (+256.13%)	+1096.71
	MCF+LP-Crit ($\alpha=2$)	-37.28 (+37.54%)	-31.61 (+208.09%)	+5.67	-1830.63 (+187.02%)	-973.04 (+844.06%)	+857.59
	SFTray ($\alpha=4$)	-41.07 (+51.52%)	-31.94 (+211.34%)	+9.13	-2902.79 (+355.13%)	-1835.35 (+1680.70%)	+1067.44
	MCF-Crit ($\alpha=4$)	-38.64 (+42.57%)	-36.51 (+255.91%)	+2.13	-3235.11 (+407.23%)	-1884.14 (+1728.04%)	+1350.97
	MCF+LP-Crit ($\alpha=4$)	-38.76 (+43.00%)	-26.32 (+156.60%)	+12.43	-3198.21 (+401.44%)	-1753.60 (+1601.38%)	+1444.61
	SFTray ($\alpha=8$)	-40.56 (+49.63%)	-33.98 (+231.19%)	+6.58	-3648.14 (+471.99%)	-2052.00 (+1890.90%)	+1596.14
	MCF-Crit ($\alpha=8$)	-40.34 (+48.84%)	-36.24 (+253.24%)	+4.10	-3793.26 (+494.74%)	-2017.41 (+1857.34%)	+1775.85
	MCF+LP-Crit ($\alpha=8$)	-42.08 (+55.27%)	-32.88 (+220.56%)	+9.20	-4009.93 (+528.71%)	-2041.88 (+1881.08%)	+1968.05
	SFTray ($\alpha=16$)	-64.14 (+136.65%)	-35.32 (+244.29%)	+28.82	-4513.05 (+607.60%)	-2639.89 (+2461.28%)	+1873.16
	MCF-Crit ($\alpha=16$)	-44.33 (+63.58%)	-32.89 (+220.58%)	+11.45	-4330.04 (+578.90%)	-2117.12 (+1954.08%)	+2212.92
	MCF+LP-Crit ($\alpha=16$)	-60.94 (+124.82%)	-39.85 (+288.41%)	+21.09	-4539.18 (+611.69%)	-2723.15 (+2542.07%)	+1816.03
	SFTray ($\alpha=67$)	-64.14 (+136.65%)	-39.26 (+282.66%)	+24.88	-5310.03 (+732.56%)	-2472.98 (+2299.34%)	+2837.05
	MCF-Crit ($\alpha=67$)	-44.57 (+64.44%)	-51.57 (+402.67%)	-7.00	-4357.79 (+583.25%)	-2726.21 (+2545.03%)	+1631.58
	MCF+LP-Crit ($\alpha=67$)	-60.94 (+124.82%)	-43.83 (+327.26%)	+17.10	-4640.11 (+627.52%)	-2570.11 (+2393.58%)	+2070.00
	SFTray ($\alpha=130$)	-64.14 (+136.65%)	-39.26 (+282.66%)	+24.88	-5310.03 (+732.56%)	-2472.98 (+2299.34%)	+2837.05
	MCF-Crit ($\alpha=130$)	-44.57 (+64.44%)	-51.57 (+402.67%)	-7.00	-4357.79 (+583.25%)	-2726.21 (+2545.03%)	+1631.58
	MCF+LP-Crit ($\alpha=130$)	-60.94 (+124.82%)	-43.83 (+327.26%)	+17.10	-4640.11 (+627.52%)	-2570.11 (+2393.58%)	+2070.00
	SFTray ($\alpha=193$)	-64.14 (+136.65%)	-39.26 (+282.66%)	+24.88	-5310.03 (+732.56%)	-2472.98 (+2299.34%)	+2837.05
	MCF-Crit ($\alpha=193$)	-44.57 (+64.44%)	-51.57 (+402.67%)	-7.00	-4357.79 (+583.25%)	-2726.21 (+2545.03%)	+1631.58
	MCF+LP-Crit ($\alpha=193$)	-60.94 (+124.82%)	-43.83 (+327.26%)	+17.10	-4640.11 (+627.52%)	-2570.11 (+2393.58%)	+2070.00
	SFTray ($\alpha=256$)	-64.14 (+136.65%)	-39.26 (+282.66%)	+24.88	-5310.03 (+732.56%)	-2472.98 (+2299.34%)	+2837.05
	MCF-Crit ($\alpha=256$)	-44.57 (+64.44%)	-51.57 (+402.67%)	-7.00	-4357.79 (+583.25%)	-2726.21 (+2545.03%)	+1631.58
	MCF+LP-Crit ($\alpha=256$)	-60.94 (+124.82%)	-43.83 (+327.26%)	+17.10	-4640.11 (+627.52%)	-2570.11 (+2393.58%)	+2070.00
riscv32i	NC (abs.)	-135.54	21.09	+156.63	-66001.90	0.00	+66001.90
	SFTray ($\alpha=2$)	-135.54 (+0.00%)	15.57 (-26.17%)	+151.11	-66176.80 (+0.26%)	0.00	+66176.80
	MCF-Crit ($\alpha=2$)	-135.03 (-0.37%)	2.87 (-86.40%)	+137.90	-66167.50 (+0.25%)	0.00	+66167.50
	MCF+LP-Crit ($\alpha=2$)	-135.03 (-0.37%)	2.87 (-86.40%)	+137.90	-66167.50 (+0.25%)	0.00	+66167.50
	SFTray ($\alpha=4$)	-135.03 (-0.37%)	-10.97 (-152.01%)	+124.07	-66546.30 (+0.82%)	-957.34	+65588.96
	MCF-Crit ($\alpha=4$)	-135.03 (-0.37%)	5.30 (-74.85%)	+140.34	-66514.20 (+0.78%)	0.00	+66514.20
	MCF+LP-Crit ($\alpha=4$)	-135.03 (-0.37%)	5.30 (-74.85%)	+140.34	-66514.20 (+0.78%)	0.00	+66514.20
	SFTray ($\alpha=8$)	-133.13 (-1.77%)	18.42 (-12.66%)	+151.55	-66935.80 (+1.41%)	0.00	+66935.80
	MCF-Crit ($\alpha=8$)	-131.54 (-2.95%)	17.89 (-15.16%)	+149.44	-66921.30 (+1.39%)	0.00	+66921.30
	MCF+LP-Crit ($\alpha=8$)	-131.54 (-2.95%)	17.89 (-15.16%)	+149.44	-66921.30 (+1.39%)	0.00	+66921.30
	SFTray ($\alpha=16$)	-133.13 (-1.77%)	18.83 (-10.71%)	+151.96	-66937.50 (+1.42%)	0.00	+66937.50
	MCF-Crit ($\alpha=16$)	-131.54 (-2.95%)	27.31 (+29.49%)	+158.85	-66940.00 (+1.42%)	0.00	+66940.00
	MCF+LP-Crit ($\alpha=16$)	-131.54 (-2.95%)	27.31 (+29.49%)	+158.85	-66940.00 (+1.42%)	0.00	+66940.00
	SFTray ($\alpha=67$)	-133.13 (-1.77%)	13.34 (-36.75%)	+146.47	-66942.90 (+1.43%)	0.00	+66942.90
	MCF-Crit ($\alpha=67$)	-131.54 (-2.95%)	13.88 (-34.18%)	+145.42	-66939.70 (+1.42%)	0.00	+66939.70
	MCF+LP-Crit ($\alpha=67$)	-131.54 (-2.95%)	13.88 (-34.18%)	+145.42	-66939.70 (+1.42%)	0.00	+66939.70
	SFTray ($\alpha=130$)	-133.13 (-1.77%)	13.34 (-36.75%)	+146.47	-66942.90 (+1.43%)	0.00	+66942.90
	MCF-Crit ($\alpha=130$)	-131.54 (-2.95%)	13.88 (-34.18%)	+145.42	-66939.70 (+1.42%)	0.00	+66939.70
	MCF+LP-Crit ($\alpha=130$)	-131.54 (-2.95%)	13.88 (-34.18%)	+145.42	-66939.70 (+1.42%)	0.00	+66939.70
	SFTray ($\alpha=193$)	-133.13 (-1.77%)	13.34 (-36.75%)	+146.47	-66942.90 (+1.43%)	0.00	+66942.90
	MCF-Crit ($\alpha=193$)	-131.54 (-2.95%)	13.88 (-34.18%)	+145.42	-66939.70 (+1.42%)	0.00	+66939.70
	MCF+LP-Crit ($\alpha=193$)	-131.54 (-2.95%)	13.88 (-34.18%)	+145.42	-66939.70 (+1.42%)	0.00	+66939.70
	SFTray ($\alpha=256$)	-133.13 (-1.77%)	13.34 (-36.75%)	+146.47	-66942.90 (+1.43%)	0.00	+66942.90
	MCF-Crit ($\alpha=256$)	-131.54 (-2.95%)	13.88 (-34.18%)	+145.42	-66939.70 (+1.42%)	0.00	+66939.70
	MCF+LP-Crit ($\alpha=256$)	-131.54 (-2.95%)	13.88 (-34.18%)	+145.42	-66939.70 (+1.42%)	0.00	+66939.70

Table 10 – Worst Negative Slack (WNS) and Total Negative Slack (TNS) comparison after Global Placement (GP) and Detailed Routing (DR) for varying α values with $\beta = 0.1$ (Part 5). All values are in picoseconds (ps). NC rows present absolute values, while SFTray, MCF-Crit, and MCF+LP-Crit show absolute values followed by percentage variation relative to NC in parentheses. Bold values indicate that MCF-Crit or MCF+LP-Crit achieved a better WNS or TNS timing value than SFTray.

Design	Version	WNS (ps)			TNS (ps)		
		GP	DR	$\Delta(DR - GP)$	GP	DR	$\Delta(DR - GP)$
jpeg	NC (abs.)	-23.88	16.63	+40.52	-210.86	0.00	+210.86
	SFTray ($\alpha=2$)	-23.88 (+0.00%)	19.84 (+19.28%)	+43.73	-269.14 (+27.64%)	0.00	+269.14
	MCF-Crit ($\alpha=2$)	-23.88 (+0.00%)	19.19 (+15.38%)	+43.08	-215.95 (+2.41%)	0.00	+215.95
	MCF+LP-Crit ($\alpha=2$)	-23.88 (+0.00%)	16.34 (-1.78%)	+40.22	-218.54 (+3.64%)	0.00	+218.54
	SFTray ($\alpha=4$)	-25.88 (+8.34%)	14.47 (-13.00%)	+40.35	-326.86 (+55.01%)	0.00	+326.86
	MCF-Crit ($\alpha=4$)	-25.92 (+8.51%)	18.31 (+10.11%)	+44.23	-317.46 (+50.56%)	0.00	+317.46
	MCF+LP-Crit ($\alpha=4$)	-27.60 (+15.57%)	17.67 (+6.21%)	+45.27	-315.28 (+49.52%)	0.00	+315.28
	SFTray ($\alpha=8$)	-30.35 (+27.07%)	20.99 (+26.18%)	+51.34	-361.36 (+71.37%)	0.00	+361.36
	MCF-Crit ($\alpha=8$)	-28.45 (+19.13%)	10.27 (-38.24%)	+38.73	-341.76 (+62.08%)	0.00	+341.76
	MCF+LP-Crit ($\alpha=8$)	-27.60 (+15.57%)	17.11 (+2.87%)	+44.72	-339.80 (+61.15%)	0.00	+339.80
	SFTray ($\alpha=16$)	-30.35 (+27.07%)	14.00 (-15.84%)	+44.35	-387.06 (+83.57%)	0.00	+387.06
	MCF-Crit ($\alpha=16$)	-28.45 (+19.13%)	19.36 (+16.39%)	+47.81	-354.02 (+67.89%)	0.00	+354.02
	MCF+LP-Crit ($\alpha=16$)	-30.62 (+28.20%)	16.70 (+0.38%)	+47.32	-358.52 (+70.03%)	0.00	+358.52
	SFTray ($\alpha=67$)	-30.35 (+27.07%)	16.32 (-1.86%)	+46.67	-391.47 (+85.66%)	0.00	+391.47
	MCF-Crit ($\alpha=67$)	-30.98 (+29.69%)	10.32 (-37.98%)	+41.29	-366.38 (+73.76%)	0.00	+366.38
	MCF+LP-Crit ($\alpha=67$)	-30.62 (+28.20%)	17.23 (+3.60%)	+47.85	-363.23 (+72.26%)	0.00	+363.23
	SFTray ($\alpha=130$)	-30.35 (+27.07%)	16.32 (-1.86%)	+46.67	-391.47 (+85.66%)	0.00	+391.47
	MCF-Crit ($\alpha=130$)	-30.98 (+29.69%)	10.32 (-37.98%)	+41.29	-366.38 (+73.76%)	0.00	+366.38
	MCF+LP-Crit ($\alpha=130$)	-30.62 (+28.20%)	17.23 (+3.60%)	+47.85	-363.23 (+72.26%)	0.00	+363.23
	SFTray ($\alpha=193$)	-30.35 (+27.07%)	16.32 (-1.86%)	+46.67	-391.47 (+85.66%)	0.00	+391.47
	MCF-Crit ($\alpha=193$)	-30.98 (+29.69%)	10.32 (-37.98%)	+41.29	-366.38 (+73.76%)	0.00	+366.38
	MCF+LP-Crit ($\alpha=193$)	-30.62 (+28.20%)	17.23 (+3.60%)	+47.85	-363.23 (+72.26%)	0.00	+363.23
	SFTray ($\alpha=256$)	-30.35 (+27.07%)	16.32 (-1.86%)	+46.67	-391.47 (+85.66%)	0.00	+391.47
	MCF-Crit ($\alpha=256$)	-30.98 (+29.69%)	10.32 (-37.98%)	+41.29	-366.38 (+73.76%)	0.00	+366.38
	MCF+LP-Crit ($\alpha=256$)	-30.62 (+28.20%)	17.23 (+3.60%)	+47.85	-363.23 (+72.26%)	0.00	+363.23

**APPENDIX B – COMPARISON OF MBFF CLUSTERING STRATEGIES FOR
VARYING α VALUES WITH $\beta = 0.1$. FF DISTRIBUTION, POWER AND AREA
RELATED ANALYSIS.**

Table 11 – Comparison of MBFF clustering strategies for varying α values with $\beta = 0.1$ (Part 1). NC rows present absolute values, while SFTray, MCF-Crit, and MCF+LP-Crit show absolute values followed by percentage variation relative to NC in parentheses (except for #1b, #2b and #4b, which are the number of 1-bit, 2-bit and 4-bit flip-flops, respectively). All metrics were extracted after detailed routing.

Design	Version	#1b	#2b	#4b	FF %	Clk. P (mW)	Swit. P (mW)	Total P (mW)	Area (μm^2)
aes	NC (abs.)	562	0	0	3.3%	2.500	98.881	166.145	1838.640
	SFTray ($\alpha=2$)	464	23	13	2.9%	2.310 (-7.60%)	98.737 (-0.15%)	165.802 (-0.21%)	1834.240 (-0.24%)
	MCF-Crit ($\alpha=2$)	480	21	10	3.0%	2.230 (-10.80%)	99.512 (+0.64%)	167.757 (+0.97%)	1855.730 (+0.93%)
	MCF+LP-Crit ($\alpha=2$)	436	27	18	2.8%	2.100 (-16.00%)	99.256 (+0.38%)	167.117 (+0.59%)	1852.490 (+0.75%)
	SFTray ($\alpha=4$)	222	20	75	1.9%	1.590 (-36.40%)	99.201 (+0.32%)	166.621 (+0.29%)	1852.870 (+0.77%)
	MCF-Crit ($\alpha=4$)	214	14	80	1.8%	1.520 (-39.20%)	99.355 (+0.48%)	167.045 (+0.54%)	1855.890 (+0.94%)
	MCF+LP-Crit ($\alpha=4$)	226	28	70	1.9%	1.690 (-32.40%)	99.501 (+0.63%)	167.495 (+0.81%)	1858.470 (+1.08%)
	SFTray ($\alpha=8$)	106	12	108	1.3%	1.280 (-48.80%)	99.148 (+0.27%)	166.447 (+0.18%)	1853.060 (+0.78%)
	MCF-Crit ($\alpha=8$)	94	12	111	1.3%	1.130 (-54.80%)	99.147 (+0.27%)	166.340 (+0.12%)	1852.560 (+0.76%)
	MCF+LP-Crit ($\alpha=8$)	94	12	111	1.3%	1.130 (-54.80%)	99.147 (+0.27%)	166.340 (+0.12%)	1852.560 (+0.76%)
	SFTray ($\alpha=16$)	36	5	129	1.0%	1.040 (-58.40%)	99.142 (+0.26%)	166.106 (-0.02%)	1850.170 (+0.63%)
	MCF-Crit ($\alpha=16$)	38	10	126	1.0%	1.100 (-56.00%)	99.499 (+0.63%)	167.131 (+0.59%)	1862.000 (+1.27%)
	MCF+LP-Crit ($\alpha=16$)	32	1	132	1.0%	1.060 (-57.60%)	99.059 (+0.18%)	166.515 (+0.22%)	1856.620 (+0.98%)
	SFTray ($\alpha=67$)	10	2	137	0.9%	1.160 (-53.60%)	99.372 (+0.50%)	167.002 (+0.52%)	1867.570 (+1.57%)
	MCF-Crit ($\alpha=67$)	30	6	130	1.0%	1.040 (-58.40%)	99.260 (+0.38%)	166.570 (+0.26%)	1857.450 (+1.02%)
	MCF+LP-Crit ($\alpha=67$)	10	0	138	0.9%	1.100 (-56.00%)	99.440 (+0.57%)	167.370 (+0.74%)	1862.350 (+1.29%)
	SFTray ($\alpha=130$)	10	2	137	0.9%	1.160 (-53.60%)	99.372 (+0.50%)	167.002 (+0.52%)	1867.570 (+1.57%)
	MCF-Crit ($\alpha=130$)	30	6	130	1.0%	1.040 (-58.40%)	99.260 (+0.38%)	166.570 (+0.26%)	1857.450 (+1.02%)
	MCF+LP-Crit ($\alpha=130$)	10	0	138	0.9%	1.100 (-56.00%)	99.440 (+0.57%)	167.370 (+0.74%)	1862.350 (+1.29%)
	SFTray ($\alpha=193$)	10	2	137	0.9%	1.160 (-53.60%)	99.372 (+0.50%)	167.002 (+0.52%)	1867.570 (+1.57%)
	MCF-Crit ($\alpha=193$)	30	6	130	1.0%	1.040 (-58.40%)	99.260 (+0.38%)	166.570 (+0.26%)	1857.450 (+1.02%)
	MCF+LP-Crit ($\alpha=193$)	10	0	138	0.9%	1.100 (-56.00%)	99.440 (+0.57%)	167.370 (+0.74%)	1862.350 (+1.29%)
	SFTray ($\alpha=256$)	10	2	137	0.9%	1.160 (-53.60%)	99.372 (+0.50%)	167.002 (+0.52%)	1867.570 (+1.57%)
	MCF-Crit ($\alpha=256$)	30	6	130	1.0%	1.040 (-58.40%)	99.260 (+0.38%)	166.570 (+0.26%)	1857.450 (+1.02%)
	MCF+LP-Crit ($\alpha=256$)	10	0	138	0.9%	1.100 (-56.00%)	99.440 (+0.57%)	167.370 (+0.74%)	1862.350 (+1.29%)
cva6	NC (abs.)	8465	0	0	7.2%	17.500	46.961	132.122	17586.400
	SFTray ($\alpha=2$)	6715	740	385	6.7%	17.300 (-1.14%)	48.095 (+2.41%)	133.153 (+0.78%)	17953.200 (+2.09%)
	MCF-Crit ($\alpha=2$)	6713	768	396	6.7%	17.200 (-1.71%)	48.760 (+3.83%)	133.763 (+1.24%)	17977.400 (+2.22%)
	MCF+LP-Crit ($\alpha=2$)	6719	761	392	6.7%	16.800 (-4.00%)	45.721 (-2.64%)	130.309 (-1.37%)	17958.100 (+2.11%)
	SFTray ($\alpha=4$)	6694	723	382	6.7%	17.100 (-2.29%)	47.782 (+1.75%)	132.504 (+0.29%)	17928.700 (+1.95%)
	MCF-Crit ($\alpha=4$)	6690	749	385	6.7%	17.300 (-1.14%)	48.614 (+3.52%)	133.640 (+1.15%)	17949.300 (+2.06%)
	MCF+LP-Crit ($\alpha=4$)	6691	733	385	6.7%	16.800 (-4.00%)	45.584 (-2.93%)	130.000 (-1.61%)	17925.600 (+1.93%)
	SFTray ($\alpha=8$)	6686	668	374	6.6%	17.200 (-1.71%)	47.191 (+0.49%)	131.893 (-0.17%)	17886.800 (+1.71%)
	MCF-Crit ($\alpha=8$)	6681	706	380	6.6%	17.100 (-2.29%)	47.846 (+1.88%)	132.515 (+0.30%)	17915.700 (+1.87%)
	MCF+LP-Crit ($\alpha=8$)	6683	687	378	6.6%	16.700 (-4.57%)	45.403 (-3.32%)	129.574 (-1.93%)	17888.500 (+1.72%)
	SFTray ($\alpha=16$)	6680	604	364	6.5%	17.000 (-2.86%)	46.729 (-0.50%)	131.079 (-0.79%)	17832.700 (+1.40%)
	MCF-Crit ($\alpha=16$)	6679	634	359	6.6%	17.000 (-2.86%)	46.963 (+0.00%)	131.346 (-0.59%)	17845.900 (+1.48%)
	MCF+LP-Crit ($\alpha=16$)	6682	595	367	6.5%	16.700 (-4.57%)	45.109 (-3.94%)	129.086 (-2.30%)	17820.300 (+1.33%)
	SFTray ($\alpha=67$)	6678	309	365	6.3%	16.400 (-6.29%)	46.316 (-1.37%)	129.453 (-2.02%)	17654.600 (+0.39%)
	MCF-Crit ($\alpha=67$)	6678	348	352	6.3%	16.500 (-5.71%)	46.312 (-1.38%)	129.618 (-1.90%)	17664.200 (+0.44%)
	MCF+LP-Crit ($\alpha=67$)	6678	334	362	6.3%	16.600 (-5.14%)	44.840 (-4.52%)	128.189 (-2.98%)	17663.600 (+0.44%)
	SFTray ($\alpha=130$)	6678	119	415	6.2%	16.000 (-8.57%)	46.133 (-1.76%)	128.630 (-2.64%)	17594.800 (+0.05%)
	MCF-Crit ($\alpha=130$)	6678	99	421	6.2%	15.800 (-9.71%)	46.210 (-1.60%)	128.614 (-2.66%)	17587.600 (+0.01%)
	MCF+LP-Crit ($\alpha=130$)	6678	129	410	6.2%	16.500 (-5.71%)	44.752 (-4.70%)	127.739 (-3.32%)	17598.700 (+0.07%)
	SFTray ($\alpha=193$)	6678	68	427	6.2%	15.900 (-9.14%)	46.113 (-1.81%)	128.471 (-2.76%)	17577.700 (-0.05%)
	MCF-Crit ($\alpha=193$)	6678	52	433	6.2%	15.800 (-9.71%)	46.190 (-1.64%)	128.614 (-2.66%)	17574.900 (-0.07%)
	MCF+LP-Crit ($\alpha=193$)	6678	62	427	6.2%	16.500 (-5.71%)	44.613 (-5.00%)	127.466 (-3.52%)	17577.900 (-0.05%)
	SFTray ($\alpha=256$)	6678	49	433	6.2%	15.700 (-10.29%)	46.052 (-1.94%)	128.284 (-2.90%)	17570.900 (-0.09%)
	MCF-Crit ($\alpha=256$)	6678	40	437	6.2%	15.700 (-10.29%)	46.129 (-1.77%)	128.409 (-2.81%)	17568.300 (-0.10%)
	MCF+LP-Crit ($\alpha=256$)	6678	39	435	6.2%	16.400 (-6.29%)	44.575 (-5.08%)	127.346 (-3.61%)	17572.100 (-0.08%)

Table 12 – Comparison of MBFF clustering strategies for varying α values with $\beta = 0.1$ (Part 2). NC rows present absolute values, while SFTray, MCF-Crit, and MCF+LP-Crit show absolute values followed by percentage variation relative to NC in parentheses (except for #1b, #2b and #4b, which are the number of 1-bit, 2-bit and 4-bit flip-flops, respectively). All metrics were extracted after detailed routing.

Design	Version	#1b	#2b	#4b	FF %	Clk. P (mW)	Swit. P (mW)	Total P (mW)	Area (μm^2)
ethmac	NC (abs.)	10546	0	0	17.3%	51.000	24.878	111.321	7935.720
	SFTray ($\alpha=2$)	8106	364	428	15.0%	44.400 (-12.94%)	22.357 (-10.13%)	101.096 (-9.19%)	7894.810 (-0.52%)
	MCF-Crit ($\alpha=2$)	8064	371	435	15.0%	44.900 (-11.96%)	22.404 (-9.94%)	101.514 (-8.81%)	7899.090 (-0.46%)
	MCF+LP-Crit ($\alpha=2$)	8064	371	435	15.0%	44.900 (-11.96%)	22.404 (-9.94%)	101.514 (-8.81%)	7899.090 (-0.46%)
	SFTray ($\alpha=4$)	2970	214	1787	9.1%	26.700 (-47.65%)	15.776 (-36.59%)	75.562 (-32.12%)	7781.430 (-1.94%)
	MCF-Crit ($\alpha=4$)	3030	246	1756	9.2%	26.800 (-47.45%)	15.932 (-35.96%)	75.711 (-31.99%)	7781.050 (-1.95%)
	MCF+LP-Crit ($\alpha=4$)	3030	246	1756	9.2%	26.800 (-47.45%)	15.932 (-35.96%)	75.711 (-31.99%)	7781.050 (-1.95%)
	SFTray ($\alpha=8$)	1396	31	2272	6.9%	21.500 (-57.84%)	13.523 (-45.64%)	67.949 (-38.96%)	7748.320 (-2.36%)
	MCF-Crit ($\alpha=8$)	1380	35	2274	6.9%	21.300 (-58.24%)	13.437 (-45.99%)	67.652 (-39.23%)	7745.510 (-2.40%)
	MCF+LP-Crit ($\alpha=8$)	1380	35	2274	6.9%	21.300 (-58.24%)	13.437 (-45.99%)	67.652 (-39.23%)	7745.510 (-2.40%)
	SFTray ($\alpha=16$)	1340	19	2292	6.8%	21.400 (-58.04%)	13.422 (-46.05%)	67.718 (-39.17%)	7747.510 (-2.37%)
	MCF-Crit ($\alpha=16$)	1330	26	2291	6.8%	20.900 (-59.02%)	13.376 (-46.23%)	67.239 (-39.60%)	7743.930 (-2.42%)
	MCF+LP-Crit ($\alpha=16$)	1330	26	2291	6.8%	20.900 (-59.02%)	13.376 (-46.23%)	67.239 (-39.60%)	7743.930 (-2.42%)
	SFTray ($\alpha=67$)	1312	11	2303	6.8%	20.900 (-59.02%)	13.321 (-46.46%)	67.198 (-39.64%)	7743.500 (-2.42%)
	MCF-Crit ($\alpha=67$)	1300	19	2302	6.8%	21.100 (-58.63%)	13.311 (-46.49%)	67.354 (-39.50%)	7745.520 (-2.40%)
	MCF+LP-Crit ($\alpha=67$)	1300	19	2302	6.8%	21.100 (-58.63%)	13.311 (-46.49%)	67.354 (-39.50%)	7745.520 (-2.40%)
	SFTray ($\alpha=130$)	1312	11	2303	6.8%	20.900 (-59.02%)	13.321 (-46.46%)	67.198 (-39.64%)	7743.500 (-2.42%)
	MCF-Crit ($\alpha=130$)	1300	19	2302	6.8%	21.100 (-58.63%)	13.311 (-46.49%)	67.354 (-39.50%)	7745.520 (-2.40%)
	MCF+LP-Crit ($\alpha=130$)	1300	19	2302	6.8%	21.100 (-58.63%)	13.311 (-46.49%)	67.354 (-39.50%)	7745.520 (-2.40%)
	SFTray ($\alpha=193$)	1312	11	2303	6.8%	20.900 (-59.02%)	13.321 (-46.46%)	67.198 (-39.64%)	7743.500 (-2.42%)
	MCF-Crit ($\alpha=193$)	1298	20	2302	6.8%	21.000 (-58.82%)	13.364 (-46.28%)	67.223 (-39.61%)	7744.850 (-2.41%)
	MCF+LP-Crit ($\alpha=193$)	1298	20	2302	6.8%	21.000 (-58.82%)	13.364 (-46.28%)	67.223 (-39.61%)	7744.850 (-2.41%)
	SFTray ($\alpha=256$)	1312	11	2303	6.8%	20.900 (-59.02%)	13.321 (-46.46%)	67.198 (-39.64%)	7743.500 (-2.42%)
	MCF-Crit ($\alpha=256$)	1298	20	2302	6.8%	21.000 (-58.82%)	13.364 (-46.28%)	67.223 (-39.61%)	7744.850 (-2.41%)
	MCF+LP-Crit ($\alpha=256$)	1298	20	2302	6.8%	21.000 (-58.82%)	13.364 (-46.28%)	67.223 (-39.61%)	7744.850 (-2.41%)
gcd	NC (abs.)	35	0	0	7.5%	0.099	0.323	0.918	48.347
	SFTray ($\alpha=2$)	31	2	0	7.0%	0.115 (+16.40%)	0.325 (+0.53%)	0.941 (+2.48%)	49.149 (+1.66%)
	MCF-Crit ($\alpha=2$)	33	1	0	7.3%	0.084 (-14.57%)	0.315 (-2.46%)	0.885 (-3.67%)	47.852 (-1.03%)
	MCF+LP-Crit ($\alpha=2$)	31	2	0	7.0%	0.115 (+16.40%)	0.325 (+0.53%)	0.941 (+2.48%)	49.149 (+1.66%)
	SFTray ($\alpha=4$)	13	3	4	4.4%	0.049 (-50.10%)	0.293 (-9.29%)	0.831 (-9.51%)	47.822 (-1.09%)
	MCF-Crit ($\alpha=4$)	21	3	2	5.8%	0.057 (-42.11%)	0.298 (-7.66%)	0.835 (-9.13%)	47.239 (-2.29%)
	MCF+LP-Crit ($\alpha=4$)	13	3	4	4.4%	0.051 (-48.28%)	0.295 (-8.86%)	0.831 (-9.57%)	47.954 (-0.81%)
	SFTray ($\alpha=8$)	5	3	6	3.1%	0.043 (-56.98%)	0.287 (-11.15%)	0.818 (-10.89%)	48.172 (-0.36%)
	MCF-Crit ($\alpha=8$)	11	2	5	4.0%	0.050 (-48.99%)	0.295 (-8.86%)	0.836 (-8.94%)	48.129 (-0.45%)
	MCF+LP-Crit ($\alpha=8$)	3	0	8	2.4%	0.057 (-42.11%)	0.293 (-9.44%)	0.852 (-7.27%)	49.251 (+1.87%)
	SFTray ($\alpha=16$)	3	0	8	2.5%	0.057 (-42.11%)	0.285 (-11.97%)	0.825 (-10.22%)	48.158 (-0.39%)
	MCF-Crit ($\alpha=16$)	9	3	5	3.8%	0.064 (-34.92%)	0.297 (-8.25%)	0.850 (-7.44%)	48.245 (-0.21%)
	MCF+LP-Crit ($\alpha=16$)	3	0	8	2.4%	0.057 (-42.11%)	0.293 (-9.44%)	0.852 (-7.27%)	49.251 (+1.87%)
	SFTray ($\alpha=67$)	3	0	8	2.5%	0.057 (-42.11%)	0.285 (-11.97%)	0.825 (-10.22%)	48.158 (-0.39%)
	MCF-Crit ($\alpha=67$)	9	3	5	3.8%	0.064 (-34.92%)	0.297 (-8.25%)	0.850 (-7.44%)	48.245 (-0.21%)
	MCF+LP-Crit ($\alpha=67$)	3	0	8	2.4%	0.057 (-42.11%)	0.293 (-9.44%)	0.852 (-7.27%)	49.251 (+1.87%)
	SFTray ($\alpha=130$)	3	0	8	2.5%	0.057 (-42.11%)	0.285 (-11.97%)	0.825 (-10.22%)	48.158 (-0.39%)
	MCF-Crit ($\alpha=130$)	9	3	5	3.8%	0.064 (-34.92%)	0.297 (-8.25%)	0.850 (-7.44%)	48.245 (-0.21%)
	MCF+LP-Crit ($\alpha=130$)	3	0	8	2.4%	0.057 (-42.11%)	0.293 (-9.44%)	0.852 (-7.27%)	49.251 (+1.87%)
	SFTray ($\alpha=193$)	3	0	8	2.5%	0.057 (-42.11%)	0.285 (-11.97%)	0.825 (-10.22%)	48.158 (-0.39%)
	MCF-Crit ($\alpha=193$)	9	3	5	3.8%	0.064 (-34.92%)	0.297 (-8.25%)	0.850 (-7.44%)	48.245 (-0.21%)
	MCF+LP-Crit ($\alpha=193$)	3	0	8	2.4%	0.057 (-42.11%)	0.293 (-9.44%)	0.852 (-7.27%)	49.251 (+1.87%)
	SFTray ($\alpha=256$)	3	0	8	2.5%	0.057 (-42.11%)	0.285 (-11.97%)	0.825 (-10.22%)	48.158 (-0.39%)
	MCF-Crit ($\alpha=256$)	9	3	5	3.8%	0.064 (-34.92%)	0.297 (-8.25%)	0.850 (-7.44%)	48.245 (-0.21%)
	MCF+LP-Crit ($\alpha=256$)	3	0	8	2.4%	0.057 (-42.11%)	0.293 (-9.44%)	0.852 (-7.27%)	49.251 (+1.87%)

Table 13 – Comparison of MBFF clustering strategies for varying α values with $\beta = 0.1$ (Part 3). NC rows present absolute values, while SFTray, MCF-Crit, and MCF+LP-Crit show absolute values followed by percentage variation relative to NC in parentheses (except for #1b, #2b and #4b, which are the number of 1-bit, 2-bit and 4-bit flip-flops, respectively). All metrics were extracted after detailed routing.

Design	Version	#1b	#2b	#4b	FF %	Clk. P (mW)	Swit. P (mW)	Total P (mW)	Area (μm^2)
ibex	NC (abs.)	1939	0	0	8.9%	2.920	12.254	25.188	2541.780
	SFTray ($\alpha=2$)	1929	1	2	8.9%	2.940 (+0.68%)	12.300 (+0.37%)	25.400 (+0.84%)	2555.950 (+0.56%)
	MCF-Crit ($\alpha=2$)	1931	0	2	8.9%	2.950 (+1.03%)	12.301 (+0.38%)	25.280 (+0.37%)	2553.370 (+0.46%)
	MCF+LP-Crit ($\alpha=2$)	1929	1	2	8.9%	2.940 (+0.68%)	12.269 (+0.12%)	25.241 (+0.21%)	2550.300 (+0.34%)
	SFTray ($\alpha=4$)	1799	8	31	8.5%	2.850 (-2.40%)	12.168 (-0.71%)	24.907 (-1.11%)	2534.600 (-0.28%)
	MCF-Crit ($\alpha=4$)	1801	5	32	8.5%	2.840 (-2.74%)	12.126 (-1.04%)	24.712 (-1.89%)	2527.150 (-0.58%)
	MCF+LP-Crit ($\alpha=4$)	1799	4	33	8.5%	2.850 (-2.40%)	12.184 (-0.57%)	24.951 (-0.94%)	2532.940 (-0.35%)
	SFTray ($\alpha=8$)	1691	2	61	8.2%	2.720 (-6.85%)	12.087 (-1.36%)	24.613 (-2.28%)	2529.090 (-0.50%)
	MCF-Crit ($\alpha=8$)	1693	1	61	8.1%	2.710 (-7.19%)	12.313 (+0.48%)	25.255 (+0.27%)	2555.220 (+0.53%)
	MCF+LP-Crit ($\alpha=8$)	1697	1	60	8.2%	2.730 (-6.51%)	12.086 (-1.38%)	24.621 (-2.25%)	2531.020 (-0.42%)
	SFTray ($\alpha=16$)	1675	2	65	8.1%	2.720 (-6.85%)	12.116 (-1.13%)	24.693 (-1.96%)	2532.180 (-0.38%)
	MCF-Crit ($\alpha=16$)	1677	1	65	8.1%	2.710 (-7.19%)	12.073 (-1.48%)	24.494 (-2.76%)	2528.900 (-0.51%)
	MCF+LP-Crit ($\alpha=16$)	1673	3	65	8.1%	2.710 (-7.19%)	12.073 (-1.48%)	24.494 (-2.76%)	2528.900 (-0.51%)
	SFTray ($\alpha=67$)	1673	1	66	8.0%	2.720 (-6.85%)	12.249 (-0.04%)	25.109 (-0.31%)	2560.040 (+0.72%)
	MCF-Crit ($\alpha=67$)	1673	1	66	8.1%	2.730 (-6.51%)	12.174 (-0.66%)	24.976 (-0.84%)	2546.350 (+0.18%)
	MCF+LP-Crit ($\alpha=67$)	1671	2	66	8.1%	2.710 (-7.19%)	12.125 (-1.05%)	24.638 (-2.18%)	2537.180 (-0.18%)
	SFTray ($\alpha=130$)	1673	1	66	8.0%	2.720 (-6.85%)	12.249 (-0.04%)	25.109 (-0.31%)	2560.040 (+0.72%)
	MCF-Crit ($\alpha=130$)	1673	1	66	8.1%	2.730 (-6.51%)	12.174 (-0.66%)	24.976 (-0.84%)	2546.350 (+0.18%)
	MCF+LP-Crit ($\alpha=130$)	1671	2	66	8.1%	2.710 (-7.19%)	12.125 (-1.05%)	24.638 (-2.18%)	2537.180 (-0.18%)
	SFTray ($\alpha=193$)	1673	1	66	8.0%	2.720 (-6.85%)	12.249 (-0.04%)	25.109 (-0.31%)	2560.040 (+0.72%)
	MCF-Crit ($\alpha=193$)	1673	1	66	8.1%	2.730 (-6.51%)	12.174 (-0.66%)	24.976 (-0.84%)	2546.350 (+0.18%)
	MCF+LP-Crit ($\alpha=193$)	1671	2	66	8.1%	2.710 (-7.19%)	12.125 (-1.05%)	24.638 (-2.18%)	2537.180 (-0.18%)
	SFTray ($\alpha=256$)	1673	1	66	8.0%	2.720 (-6.85%)	12.249 (-0.04%)	25.109 (-0.31%)	2560.040 (+0.72%)
	MCF-Crit ($\alpha=256$)	1673	1	66	8.1%	2.730 (-6.51%)	12.174 (-0.66%)	24.976 (-0.84%)	2546.350 (+0.18%)
	MCF+LP-Crit ($\alpha=256$)	1671	2	66	8.1%	2.710 (-7.19%)	12.125 (-1.05%)	24.638 (-2.18%)	2537.180 (-0.18%)
jpeg	NC (abs.)	4383	0	0	7.6%	9.800	56.825	119.187	6436.980
	SFTray ($\alpha=2$)	3505	111	164	6.7%	9.120 (-6.94%)	56.378 (-0.79%)	117.528 (-1.39%)	6432.130 (-0.08%)
	MCF-Crit ($\alpha=2$)	3457	129	167	6.6%	9.140 (-6.73%)	56.365 (-0.81%)	117.757 (-1.20%)	6435.380 (-0.02%)
	MCF+LP-Crit ($\alpha=2$)	3425	127	176	6.6%	8.950 (-8.67%)	56.330 (-0.87%)	117.541 (-1.38%)	6429.950 (-0.11%)
	SFTray ($\alpha=4$)	1229	113	732	3.8%	5.340 (-45.51%)	54.832 (-3.51%)	111.367 (-6.56%)	6377.800 (-0.92%)
	MCF-Crit ($\alpha=4$)	1215	114	735	3.8%	5.360 (-45.31%)	54.804 (-3.56%)	111.486 (-6.46%)	6379.390 (-0.89%)
	MCF+LP-Crit ($\alpha=4$)	1279	112	720	3.8%	5.280 (-46.12%)	54.825 (-3.52%)	111.339 (-6.58%)	6375.000 (-0.96%)
	SFTray ($\alpha=8$)	365	49	980	2.6%	4.000 (-59.18%)	54.254 (-4.53%)	109.118 (-8.45%)	6357.010 (-1.24%)
	MCF-Crit ($\alpha=8$)	301	41	1000	2.5%	3.980 (-59.39%)	54.200 (-4.62%)	109.032 (-8.52%)	6358.380 (-1.22%)
	MCF+LP-Crit ($\alpha=8$)	307	38	1000	2.5%	3.960 (-59.59%)	54.166 (-4.68%)	108.804 (-8.71%)	6354.260 (-1.29%)
	SFTray ($\alpha=16$)	135	16	1054	2.2%	3.680 (-62.45%)	54.116 (-4.77%)	108.452 (-9.01%)	6351.030 (-1.34%)
	MCF-Crit ($\alpha=16$)	109	11	1063	2.2%	3.750 (-61.73%)	54.032 (-4.92%)	108.387 (-9.06%)	6351.790 (-1.32%)
	MCF+LP-Crit ($\alpha=16$)	117	7	1063	2.2%	3.700 (-62.24%)	54.009 (-4.96%)	108.340 (-9.10%)	6351.750 (-1.32%)
	SFTray ($\alpha=67$)	85	7	1071	2.2%	3.540 (-63.88%)	54.136 (-4.73%)	108.443 (-9.01%)	6350.860 (-1.34%)
	MCF-Crit ($\alpha=67$)	87	6	1071	2.2%	3.760 (-61.63%)	54.082 (-4.83%)	108.556 (-8.92%)	6355.570 (-1.26%)
	MCF+LP-Crit ($\alpha=67$)	87	4	1072	2.2%	3.560 (-63.67%)	54.019 (-4.94%)	108.244 (-9.18%)	6349.150 (-1.36%)
	SFTray ($\alpha=130$)	85	7	1071	2.2%	3.540 (-63.88%)	54.136 (-4.73%)	108.443 (-9.01%)	6350.860 (-1.34%)
	MCF-Crit ($\alpha=130$)	87	6	1071	2.2%	3.760 (-61.63%)	54.082 (-4.83%)	108.556 (-8.92%)	6355.570 (-1.26%)
	MCF+LP-Crit ($\alpha=130$)	87	4	1072	2.2%	3.560 (-63.67%)	54.019 (-4.94%)	108.244 (-9.18%)	6349.150 (-1.36%)
	SFTray ($\alpha=193$)	85	7	1071	2.2%	3.540 (-63.88%)	54.136 (-4.73%)	108.443 (-9.01%)	6350.860 (-1.34%)
	MCF-Crit ($\alpha=193$)	87	6	1071	2.2%	3.760 (-61.63%)	54.082 (-4.83%)	108.556 (-8.92%)	6355.570 (-1.26%)
	MCF+LP-Crit ($\alpha=193$)	87	4	1072	2.2%	3.560 (-63.67%)	54.019 (-4.94%)	108.244 (-9.18%)	6349.150 (-1.36%)
	SFTray ($\alpha=256$)	85	7	1071	2.2%	3.540 (-63.88%)	54.136 (-4.73%)	108.443 (-9.01%)	6350.860 (-1.34%)
	MCF-Crit ($\alpha=256$)	87	6	1071	2.2%	3.760 (-61.63%)	54.082 (-4.83%)	108.556 (-8.92%)	6355.570 (-1.26%)
	MCF+LP-Crit ($\alpha=256$)	87	4	1072	2.2%	3.560 (-63.67%)	54.019 (-4.94%)	108.244 (-9.18%)	6349.150 (-1.36%)

Table 14 – Comparison of MBFF clustering strategies for varying α values with $\beta = 0.1$ (Part 4). NC rows present absolute values, while SFTray, MCF-Crit, and MCF+LP-Crit show absolute values followed by percentage variation relative to NC in parentheses (except for #1b, #2b and #4b, which are the number of 1-bit, 2-bit and 4-bit flip-flops, respectively). All metrics were extracted after detailed routing.

Design	Version	#1b	#2b	#4b	FF %	Clk. P (mW)	Swit. P (mW)	Total P (mW)	Area (μm^2)
mock-alu	NC (abs.)	1413	0	0	10.5%	6.680	27.989	58.655	1632.380
	SFTray ($\alpha=2$)	1143	69	33	9.4%	6.020 (-9.88%)	27.723 (-0.95%)	57.520 (-1.94%)	1627.860 (-0.28%)
	MCF-Crit ($\alpha=2$)	1107	73	40	9.2%	5.700 (-14.67%)	27.636 (-1.26%)	57.069 (-2.70%)	1628.370 (-0.25%)
	MCF+LP-Crit ($\alpha=2$)	1109	72	40	9.2%	6.000 (-10.18%)	27.738 (-0.90%)	57.465 (-2.03%)	1632.680 (+0.02%)
	SFTray ($\alpha=4$)	403	51	227	5.4%	3.700 (-44.61%)	26.804 (-4.24%)	53.798 (-8.28%)	1618.410 (-0.86%)
	MCF-Crit ($\alpha=4$)	387	43	235	5.3%	3.580 (-46.41%)	26.688 (-4.65%)	53.586 (-8.64%)	1606.380 (-1.59%)
	MCF+LP-Crit ($\alpha=4$)	397	40	234	5.3%	3.600 (-46.11%)	26.678 (-4.68%)	53.628 (-8.57%)	1613.540 (-1.15%)
	SFTray ($\alpha=8$)	59	9	334	3.2%	2.430 (-63.62%)	26.323 (-5.95%)	52.034 (-11.29%)	1613.040 (-1.18%)
	MCF-Crit ($\alpha=8$)	63	11	332	3.3%	2.410 (-63.92%)	26.201 (-6.39%)	51.808 (-11.67%)	1602.060 (-1.86%)
	MCF+LP-Crit ($\alpha=8$)	49	8	337	3.2%	2.410 (-63.92%)	26.146 (-6.58%)	51.728 (-11.81%)	1601.260 (-1.91%)
	SFTray ($\alpha=16$)	21	4	346	3.0%	2.270 (-66.02%)	26.240 (-6.25%)	51.772 (-11.73%)	1613.040 (-1.18%)
	MCF-Crit ($\alpha=16$)	31	5	343	3.1%	2.390 (-64.22%)	26.154 (-6.56%)	51.758 (-11.76%)	1600.050 (-1.98%)
	MCF+LP-Crit ($\alpha=16$)	19	3	347	3.0%	2.240 (-66.47%)	26.116 (-6.69%)	51.532 (-12.14%)	1598.260 (-2.09%)
	SFTray ($\alpha=67$)	13	0	350	2.9%	2.320 (-65.27%)	26.224 (-6.31%)	51.751 (-11.77%)	1602.590 (-1.82%)
	MCF-Crit ($\alpha=67$)	17	4	347	3.0%	2.240 (-66.47%)	26.171 (-6.50%)	51.542 (-12.13%)	1598.320 (-2.09%)
	MCF+LP-Crit ($\alpha=67$)	11	3	349	2.9%	2.250 (-66.32%)	26.165 (-6.52%)	51.608 (-12.01%)	1609.120 (-1.42%)
	SFTray ($\alpha=130$)	13	0	350	2.9%	2.320 (-65.27%)	26.224 (-6.31%)	51.751 (-11.77%)	1602.590 (-1.82%)
	MCF-Crit ($\alpha=130$)	23	5	345	3.0%	2.360 (-64.67%)	26.154 (-6.56%)	51.646 (-11.95%)	1598.380 (-2.08%)
	MCF+LP-Crit ($\alpha=130$)	11	3	349	2.9%	2.250 (-66.32%)	26.165 (-6.52%)	51.608 (-12.01%)	1609.120 (-1.42%)
	SFTray ($\alpha=193$)	13	0	350	2.9%	2.320 (-65.27%)	26.224 (-6.31%)	51.751 (-11.77%)	1602.590 (-1.82%)
	MCF-Crit ($\alpha=193$)	23	1	347	3.0%	2.290 (-65.72%)	26.216 (-6.33%)	51.687 (-11.88%)	1607.500 (-1.52%)
	MCF+LP-Crit ($\alpha=193$)	11	3	349	2.9%	2.250 (-66.32%)	26.165 (-6.52%)	51.608 (-12.01%)	1609.120 (-1.42%)
riscv32i	SFTray ($\alpha=256$)	13	0	350	2.9%	2.320 (-65.27%)	26.224 (-6.31%)	51.751 (-11.77%)	1602.590 (-1.82%)
	MCF-Crit ($\alpha=256$)	17	4	347	3.0%	2.240 (-66.47%)	26.171 (-6.50%)	51.542 (-12.13%)	1598.320 (-2.09%)
	MCF+LP-Crit ($\alpha=256$)	11	3	349	2.9%	2.250 (-66.32%)	26.165 (-6.52%)	51.608 (-12.01%)	1609.120 (-1.42%)
	NC (abs.)	1056	0	0	9.7%	2.560	8.428	34.073	2691.150
	SFTray ($\alpha=2$)	920	8	30	8.9%	2.410 (-5.86%)	8.319 (-1.29%)	33.764 (-0.91%)	2693.700 (+0.09%)
	MCF-Crit ($\alpha=2$)	896	16	32	8.7%	2.410 (-5.86%)	8.375 (-0.62%)	33.878 (-0.57%)	2693.690 (+0.09%)
	MCF+LP-Crit ($\alpha=2$)	896	16	32	8.7%	2.410 (-5.86%)	8.375 (-0.62%)	33.878 (-0.57%)	2693.690 (+0.09%)
	SFTray ($\alpha=4$)	324	18	174	5.0%	1.430 (-44.14%)	7.952 (-5.65%)	32.259 (-5.32%)	2671.860 (-0.72%)
	MCF-Crit ($\alpha=4$)	312	26	173	4.9%	1.420 (-44.53%)	7.945 (-5.73%)	32.191 (-5.52%)	2671.390 (-0.73%)
	MCF+LP-Crit ($\alpha=4$)	312	26	173	4.9%	1.420 (-44.53%)	7.945 (-5.73%)	32.191 (-5.52%)	2671.390 (-0.73%)
	SFTray ($\alpha=8$)	56	4	248	3.1%	0.958 (-62.58%)	7.738 (-8.19%)	31.411 (-7.81%)	2659.700 (-1.17%)
	MCF-Crit ($\alpha=8$)	56	4	248	3.1%	0.936 (-63.44%)	7.657 (-9.14%)	31.210 (-8.40%)	2657.030 (-1.27%)
	MCF+LP-Crit ($\alpha=8$)	56	4	248	3.1%	0.936 (-63.44%)	7.657 (-9.14%)	31.210 (-8.40%)	2657.030 (-1.27%)
	SFTray ($\alpha=16$)	42	1	253	3.0%	0.871 (-65.98%)	7.643 (-9.31%)	31.113 (-8.69%)	2652.760 (-1.43%)
	MCF-Crit ($\alpha=16$)	44	2	252	3.0%	0.920 (-64.06%)	7.581 (-10.04%)	30.994 (-9.04%)	2653.080 (-1.41%)
	MCF+LP-Crit ($\alpha=16$)	44	2	252	3.0%	0.920 (-64.06%)	7.581 (-10.04%)	30.994 (-9.04%)	2653.080 (-1.41%)
	SFTray ($\alpha=67$)	40	2	253	2.9%	0.878 (-65.70%)	7.660 (-9.11%)	31.149 (-8.58%)	2652.320 (-1.44%)
	MCF-Crit ($\alpha=67$)	38	3	253	2.9%	0.878 (-65.70%)	7.634 (-9.42%)	31.130 (-8.64%)	2656.410 (-1.29%)
	MCF+LP-Crit ($\alpha=67$)	38	3	253	2.9%	0.878 (-65.70%)	7.634 (-9.42%)	31.130 (-8.64%)	2656.410 (-1.29%)
	SFTray ($\alpha=130$)	40	2	253	2.9%	0.878 (-65.70%)	7.660 (-9.11%)	31.149 (-8.58%)	2652.320 (-1.44%)
	MCF-Crit ($\alpha=130$)	38	3	253	2.9%	0.878 (-65.70%)	7.634 (-9.42%)	31.130 (-8.64%)	2656.410 (-1.29%)
	MCF+LP-Crit ($\alpha=130$)	38	3	253	2.9%	0.878 (-65.70%)	7.634 (-9.42%)	31.130 (-8.64%)	2656.410 (-1.29%)
	SFTray ($\alpha=193$)	40	2	253	2.9%	0.878 (-65.70%)	7.660 (-9.11%)	31.149 (-8.58%)	2652.320 (-1.44%)
	MCF-Crit ($\alpha=193$)	38	3	253	2.9%	0.878 (-65.70%)	7.634 (-9.42%)	31.130 (-8.64%)	2656.410 (-1.29%)
	MCF+LP-Crit ($\alpha=193$)	38	3	253	2.9%	0.878 (-65.70%)	7.634 (-9.42%)	31.130 (-8.64%)	2656.410 (-1.29%)
	SFTray ($\alpha=256$)	40	2	253	2.9%	0.878 (-65.70%)	7.660 (-9.11%)	31.149 (-8.58%)	2652.320 (-1.44%)
	MCF-Crit ($\alpha=256$)	38	3	253	2.9%	0.878 (-65.70%)	7.634 (-9.42%)	31.130 (-8.64%)	2656.410 (-1.29%)
	MCF+LP-Crit ($\alpha=256$)	38	3	253	2.9%	0.878 (-65.70%)	7.634 (-9.42%)	31.130 (-8.64%)	2656.410 (-1.29%)

Table 15 – Comparison of MBFF clustering strategies for varying α values with $\beta = 0.1$ (Part 5). NC rows present absolute values, while SFTray, MCF-Crit, and MCF+LP-Crit show absolute values followed by percentage variation relative to NC in parentheses (except for #1b, #2b and #4b, which are the number of 1-bit, 2-bit and 4-bit flip-flops, respectively). All metrics were extracted after detailed routing.

Design	Version	#1b	#2b	#4b	FF %	Clk. P (mW)	Swit. P (mW)	Total P (mW)	Area (μm^2)
uart	NC (abs.)	79	0	0	9.3%	0.257	0.806	2.330	95.207
	SFTray ($\alpha=2$)	14	17	8	4.9%	0.093 (-63.81%)	0.718 (-10.90%)	2.136 (-8.34%)	93.123 (-2.19%)
	MCF-Crit ($\alpha=2$)	13	23	5	5.1%	0.120 (-53.31%)	0.734 (-8.95%)	2.146 (-7.90%)	92.933 (-2.39%)
	MCF+LP-Crit ($\alpha=2$)	13	23	5	5.1%	0.120 (-53.31%)	0.734 (-8.95%)	2.146 (-7.90%)	92.933 (-2.39%)
	SFTray ($\alpha=4$)	5	12	13	3.8%	0.092 (-64.01%)	0.706 (-12.45%)	2.098 (-9.98%)	93.735 (-1.55%)
	MCF-Crit ($\alpha=4$)	4	8	15	3.5%	0.102 (-60.31%)	0.701 (-13.12%)	2.095 (-10.11%)	92.466 (-2.88%)
	MCF+LP-Crit ($\alpha=4$)	4	8	15	3.5%	0.102 (-60.31%)	0.701 (-13.12%)	2.095 (-10.11%)	92.466 (-2.88%)
	SFTray ($\alpha=8$)	0	0	20	2.6%	0.052 (-79.77%)	0.706 (-12.46%)	2.082 (-10.67%)	92.977 (-2.34%)
	MCF-Crit ($\alpha=8$)	1	1	19	2.7%	0.071 (-72.26%)	0.703 (-12.79%)	2.079 (-10.78%)	92.058 (-3.31%)
	MCF+LP-Crit ($\alpha=8$)	1	1	19	2.7%	0.071 (-72.26%)	0.703 (-12.79%)	2.079 (-10.78%)	92.058 (-3.31%)
	SFTray ($\alpha=16$)	0	0	20	2.6%	0.052 (-79.77%)	0.706 (-12.46%)	2.082 (-10.67%)	92.977 (-2.34%)
	MCF-Crit ($\alpha=16$)	1	1	19	2.7%	0.071 (-72.26%)	0.703 (-12.79%)	2.079 (-10.78%)	92.058 (-3.31%)
	MCF+LP-Crit ($\alpha=16$)	1	1	19	2.7%	0.071 (-72.26%)	0.703 (-12.79%)	2.079 (-10.78%)	92.058 (-3.31%)
	SFTray ($\alpha=67$)	0	0	20	2.6%	0.052 (-79.77%)	0.706 (-12.46%)	2.082 (-10.67%)	92.977 (-2.34%)
	MCF-Crit ($\alpha=67$)	0	0	20	2.6%	0.061 (-76.38%)	0.706 (-12.40%)	2.095 (-10.11%)	93.210 (-2.10%)
	MCF+LP-Crit ($\alpha=67$)	0	0	20	2.6%	0.061 (-76.38%)	0.706 (-12.40%)	2.095 (-10.11%)	93.210 (-2.10%)
	SFTray ($\alpha=130$)	0	0	20	2.6%	0.052 (-79.77%)	0.706 (-12.46%)	2.082 (-10.67%)	92.977 (-2.34%)
	MCF-Crit ($\alpha=130$)	0	0	20	2.6%	0.061 (-76.38%)	0.706 (-12.40%)	2.095 (-10.11%)	93.210 (-2.10%)
	MCF+LP-Crit ($\alpha=130$)	0	0	20	2.6%	0.061 (-76.38%)	0.706 (-12.40%)	2.095 (-10.11%)	93.210 (-2.10%)
	SFTray ($\alpha=193$)	0	0	20	2.6%	0.052 (-79.77%)	0.706 (-12.46%)	2.082 (-10.67%)	92.977 (-2.34%)
	MCF-Crit ($\alpha=193$)	0	0	20	2.6%	0.061 (-76.38%)	0.706 (-12.40%)	2.095 (-10.11%)	93.210 (-2.10%)
	MCF+LP-Crit ($\alpha=193$)	0	0	20	2.6%	0.061 (-76.38%)	0.706 (-12.40%)	2.095 (-10.11%)	93.210 (-2.10%)
	SFTray ($\alpha=256$)	0	0	20	2.6%	0.052 (-79.77%)	0.706 (-12.46%)	2.082 (-10.67%)	92.977 (-2.34%)
	MCF-Crit ($\alpha=256$)	0	0	20	2.6%	0.061 (-76.38%)	0.706 (-12.40%)	2.095 (-10.11%)	93.210 (-2.10%)
	MCF+LP-Crit ($\alpha=256$)	0	0	20	2.6%	0.061 (-76.38%)	0.706 (-12.40%)	2.095 (-10.11%)	93.210 (-2.10%)

APPENDIX C – SOURCE CODE

The proposed code enhancements implemented are available in the following open repository: <https://github.com/lucasltinoco/OR-TCC>. More specifically, branches *tcc-mcf* and *tcc-mcf-lp*, contain MCF-Crit and MCF+LP-Crit changes, respectively. In addition, the repository containing the flow scripts used is <https://github.com/lucasltinoco/OR-flow-TCC>.

APPENDIX D – WORK PAPER IN SBC FORMAT

The following pages contain a version of this work formatted as an SBC (Brazilian Computer Society) paper.

Timing-Aware Enhancements to MBFF Clustering in OpenROAD

Lucas de Lacerda Tinoco¹, José Luís Almada Güntzel¹

¹Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brazil

lucas.tinoco@grad.ufsc.br, guntzel@inf.ufsc.br

Abstract. Clock networks represent a major source of power consumption in modern Very-Large-Scale Integration (VLSI) designs. Multi-Bit Flip-Flops (MBFFs) can reduce clock-network power by merging groups of Single-Bit Flip-Flops (SBFFs) into shared structures with common clock drivers. However, determining how to cluster SBFFs while preserving timing quality remains a challenging optimization problem. Among the state-of-the-art MBFF clustering techniques, the Scalable Divide-and-Conquer Flop-Tray (SFTray) algorithm available in the OpenROAD framework performs a Minimum-Cost Flow (MCF)-based assignment driven exclusively by Manhattan displacement, while timing information is considered only in later optimization stages. This work investigates the incorporation of continuous timing-criticality information into the SFTray clustering flow. A slack-derived criticality metric is introduced and combined with normalized displacement to define a timing-aware assignment cost for the MCF formulation. Additionally, timing criticality is incorporated into the Linear Programming (LP) tray-refinement stage to further improve timing awareness during clustering. The proposed modifications were implemented within OpenROAD and evaluated using nine benchmark circuits synthesized and physically implemented with the ASAP7 7 nm technology enablement. Experimental results demonstrate that the proposed formulations are able to better explore the optimization space in designs with high sequential-cell density, achieving improved power and timing trade-offs while preserving the area and clock-network benefits of MBFF clustering. These results indicate that timing criticality can be effectively exploited throughout multiple stages of the clustering process, leading to higher-quality MBFF solutions in open-source physical-design flows.

Resumo. As redes de distribuição de clock representam uma das principais fontes de consumo de potência em projetos modernos de circuitos integrados VLSI. Os Flip-Flops Multi-Bit (MBFFs) reduzem esse consumo ao agrupar múltiplos Flip-Flops Single-Bit (SBFFs) em estruturas compartilhadas com buffers de clock comuns. Entretanto, determinar como realizar esse agrupamento preservando a qualidade temporal do circuito permanece um problema de otimização desafiador. Entre as técnicas de agrupamento MBFF de estado da arte, o algoritmo Scalable Divide-and-Conquer Flop-Tray (SFTray), disponível na plataforma OpenROAD, realiza a atribuição de flip-flops por meio de uma formulação de Fluxo de Custo Mínimo (MCF) baseada exclusivamente no deslocamento

Manhattan, incorporando informações temporais apenas em etapas posteriores do processo de otimização. Este trabalho investiga a incorporação de informações contínuas de criticidade temporal ao fluxo de agrupamento do SF-Tray. Para isso, é introduzida uma métrica de criticidade derivada do slack, combinada ao deslocamento normalizado para compor uma função de custo temporalmente consciente na etapa de atribuição baseada em MCF. Além disso, a criticidade temporal é incorporada à etapa de refinamento baseada em Programação Linear (LP), ampliando a influência das informações temporais ao longo do processo de agrupamento. As modificações propostas foram implementadas diretamente no OpenROAD e avaliadas em nove circuitos benchmark sintetizados e implementados fisicamente utilizando a tecnologia ASAP7 de 7 nm. Os resultados experimentais demonstram que as formulações propostas exploram de forma mais eficiente o espaço de otimização em projetos com elevada densidade de elementos sequenciais, obtendo melhores compromissos entre potência e temporização sem comprometer os benefícios de área e redução da rede de clock proporcionados pelos MBFFs. Os resultados indicam que a criticidade temporal pode ser explorada de maneira eficaz em múltiplas etapas do processo de agrupamento, contribuindo para soluções MBFF de maior qualidade em fluxos de projeto físico de código aberto.

1. Introduction

Clock networks are among the largest contributors to power consumption in modern VLSI circuits, often accounting for up to 30% to 50% of the total dynamic power budget [Wu et al. 2016]. Dynamic power consists mostly of the switching power, described as

$$P_{\text{switching}} = \alpha C V_{DD}^2 f, \quad (1)$$

where α is the net switching activity, C is the switched capacitance, V_{DD} is the supply voltage, and f is the clock frequency [Weste and Harris 2011]. Unlike ordinary signal nets, the clock network switches every clock cycle, making its switching activity $\alpha = 1$ (or 2 depending on whether single or double edge triggering is considered). Also, its capacitive load is large, contributing to a high dynamic power.

To mitigate this issue, contemporary standard-cell libraries provide Multi-Bit Flip-Flops (MBFFs), which integrate several Single-Bit Flip-Flops (SBFFs) into a shared physical structure with common clock drivers and routing resources. By reducing the number of clock sinks and shortening clock-tree wiring, MBFF clustering can substantially decrease switching power while also improving routability and area utilization [Kahng et al. 2016, Cai et al. 2025].

Despite its widespread adoption in commercial Electronic Design Automation (EDA) flows, MBFF clustering remains a challenging optimization problem. The clustering process seeks to reduce power and preserve timing, while minimizing displacement and satisfying the legalization constraints. Furthermore, questions persist regarding the optimal stage in the design flow – synthesis or placement – to perform this clustering for the best Quality of Results (QoR). As a consequence, a significant body of research has been devoted to developing efficient clustering strategies, including K-Means-based approaches [Wu et al. 2016], Mean Shift [Chang et al. 2019], mixed-driving MBFF generation [Liu et al. 2022], and multi-objective optimization techniques [Li et al. 2025]. This

continued academic interest is further evidenced by recent competitions, such as the IC-CAD CAD Contest 2024 [Yang et al. 2025b] and ICCAD CAD Contest 2025 [Yang et al. 2025a].

Among the state-of-the-art approaches, the FTray technique [Kahng et al. 2016] and its scalable divide-and-conquer extension, SFTray [Kahng et al. 2024], stand out due to their integration of capacitated clustering with timing-aware optimization. SFTray formulates the flip-flop assignment problem using a Minimum-Cost Flow (MCF) model combined with Linear Programming (LP) and Integer Linear Programming (ILP) refinement stages. Furthermore, its divide-and-conquer strategy enables the efficient application of MBFF clustering to large-scale designs, leading to its integration into the open-source physical design framework OpenROAD [Kahng 2022].

However, although timing information is considered during the final stages of the original SFTray optimization flow, its treatment remains relatively limited. The MCF assignment stage relies exclusively on geometric displacement, measured through Manhattan distance, while timing information is only incorporated later through timing-related penalties in the ILP formulation. As a result, timing relationships between flip-flops do not directly influence the early clustering decisions that shape the subsequent optimization stages.

This work investigates the incorporation of timing criticality information into earlier stages of the SFTray flow. More specifically, a continuous timing-criticality model derived from path slack information is introduced and integrated into the MCF assignment formulation through a cost function that combines normalized displacement and normalized timing criticality. Additionally, an extended version of the approach incorporates timing criticality information into the LP refinement stage, allowing timing awareness to influence both assignment and tray-position optimization.

The proposed modifications were implemented directly in the SFTray infrastructure available in OpenROAD, preserving compatibility with a complete open-source physical design flow. Experimental evaluations were conducted using the ASAP7 technology enablement and a set of benchmark circuits. The results compare the original SFTray implementation and the proposed timing-aware formulations in terms of power consumption, timing quality, area utilization, and clock-network characteristics.

2. The Scalable Divide-and-Conquer FTray Approach

SFTray [Kahng et al. 2024] extends the original FTray technique [Kahng et al. 2016] by introducing a scalable divide-and-conquer strategy for MBFF clustering. Starting from a globally placed design containing only SBFFs, the method recursively partitions the design into localized subproblems and, within each partition, iteratively performs two optimization stages:

1. MCF-based assignment of flip-flops to MBFF tray slots;
2. LP-based refinement of tray locations.

A tray is a candidate placement structure representing a potential MBFF instance and its available bit positions (slots). During optimization, flip-flops are assigned to tray slots, and selected trays are later mapped to MBFF cells in the final design.

At the core of each iteration, SFTray constructs an assignment graph. The graph consists of a set F of flip-flop nodes, a set S of candidate tray-slot nodes, a supersource

s , and a supersink t . Edges from s to flip-flops and from tray slots to t have unit capacity and zero cost. Between the two sides, each feasible flip-flop-to-slot pair $(i, j) \in F \times S$ is connected by an edge with capacity one and an associated assignment cost. In the original SFTray formulation, this cost corresponds solely to the Manhattan distance between the current position of flip-flop i and candidate tray slot j :

$$C(i, j) = d(i, j) \quad (2)$$

where $d(i, j)$ denotes the Manhattan distance between the two locations. Consequently, the MCF stage determines a capacity-feasible assignment that minimizes the total displacement required to cluster SBFFs into candidate MBFF trays.

Given the assignment produced by the MCF stage, SFTray applies an LP refinement step to determine improved tray locations. While the MCF formulation determines which flip-flops are assigned to each tray slot, the tray positions themselves remain adjustable. The objective of the LP stage is therefore to relocate the trays so that the total displacement of the assigned flip-flops is minimized. Following Kahng et al. [Kahng et al. 2024], the LP formulation can be expressed as:

$$\min \sum_l d_l \quad (3)$$

subject to:

$$d_l = |X_i + x'_{ij} - x_l| + |Y_i + y'_{ij} - y_l| \quad \forall h_l \quad (4)$$

where (X_i, Y_i) denotes the center location of tray i , (x'_{ij}, y'_{ij}) represents the relative position of slot j within the tray, and (x_l, y_l) corresponds to the location of flip-flop h_l . Variable d_l represents the Manhattan displacement between a flip-flop and its assigned tray slot.

The MCF assignment and LP refinement stages are executed iteratively. After the LP solver updates tray positions, a new MCF assignment problem is constructed using the refined locations, potentially producing improved clustering solutions. The process continues until no further displacement reduction is achieved. After all candidate MBFF sizes have been processed, an ILP formulation selects the final set of MBFF instances by minimizing:

$$\min (\alpha \cdot W + D + \beta \cdot R) \quad (5)$$

where W denotes MBFF total power consumption, $D = \sum_l d_l$ represents total displacement, and R corresponds to the relative displacement of timing-critical launch-capture flip-flop pairs. Parameters α and β control the trade-off between power minimization and timing preservation.

3. Proposed Timing-Aware Enhancements

We propose two timing-aware enhancements to the SFTray technique. The first targets the MCF assignment stage (MCF-Crit), while the second modifies the LP-based tray-refinement stage (MCF+LP-Crit). Both modifications leverage a flip-flop criticality metric derived from static timing analysis.

3.1. Flip-Flop Criticality Metric

The timing criticality assigned to a flip-flop is derived from the set of violating timing paths in which the flip-flop participates. For a given flip-flop i , we consider the set $P^-(i)$ of all violating timing paths for which i appears as either the launch or capture endpoint. The criticality of flip-flop i is defined as:

$$c(i) = \max_{p \in P^-(i)} \left(\frac{|\text{slack}(p)|}{|\text{WNS}_{\text{FF}}|} \right) \quad (6)$$

where $\text{slack}(p)$ denotes the slack of timing path p , and WNS_{FF} is the worst negative slack among all analyzed timing paths from launch to capture flip-flops. By construction, criticality values are normalized to the interval $[0, 1]$, where a value of one corresponds to participation in a path whose slack is equal to the global worst negative slack.

3.2. MCF-Crit: Adding Normalized Timing-Criticality to MCF

In the original SFTray formulation, the MCF assignment cost is equal to the Manhattan distance $d(i, j)$ between flip-flop i and tray slot j . We replace this cost with a normalized, criticality-augmented formulation:

$$C(i, j) = \hat{d}(i, j) + c(i) \quad (7)$$

where the normalized displacement is defined as:

$$\hat{d}(i, j) = \frac{d(i, j)}{d_{\max}} \quad (8)$$

and $d_{\max}$ is the maximum Manhattan distance observed across all flip-flop-to-slot candidates in the current partition. Normalization maps both terms to the interval $[0, 1]$, ensuring that displacement and timing criticality contribute at comparable magnitudes.

The proposed formulation increases the influence of timing-critical flip-flops on the global MCF optimization objective. Consequently, assignments involving highly critical flip-flops become more significant during the capacity-constrained optimization process.

3.3. MCF+LP-Crit: Adding Timing-Criticality to LP Refinement

The second enhancement targets the LP-based tray-placement refinement stage. Instead of treating all flip-flops equally, the criticality metric defined in Eq. (6) is incorporated directly into the LP objective function. The proposed formulation minimizes a criticality-weighted displacement:

$$\min D_{\text{crit}} \quad (9)$$

with:

$$D_{\text{crit}} = \sum_i (1 + c(i)) (d_x^i + d_y^i) \quad (10)$$

where $c(i)$ denotes the criticality associated with flip-flop i . The factor $(1 + c(i))$ assigns larger weights to highly critical flip-flops while preserving a non-zero contribution from non-critical ones. Since criticality values are normalized to the interval $[0, 1]$, each displacement term receives a weight between 1 and 2.

As a consequence, the LP solver prioritizes reducing the displacement of flip-flops involved in severe timing violations, preserving timing quality during tray relocation.

4. Experimental Results

The proposed timing-aware enhancements were implemented in C++ within the SFTray clustering engine in the OpenROAD framework. The experiments employ the ASAP7 7 nm predictive FinFET standard-cell library. The evaluation is conducted after detailed routing.

Four configurations were evaluated:

- **NC**: original non-clustered OpenROAD implementation;
- **SFTray**: baseline SFTray implementation available in OpenROAD;
- **MCF-Crit**: SFTray enhanced with the timing-aware MCF assignment cost;
- **MCF+LP-Crit**: SFTray enhanced with both the timing-aware MCF assignment cost and the timing-aware LP refinement.

The parameter α was set to 8 and β to 0.1, representing the best trade-off between power minimization and timing preservation.

The benchmark circuits used in this work are summarized in Table 1. They cover a diverse set of digital designs with sequential-cell density (FF %) ranging from 3.3% to 17.3%.

Table 1. Benchmark circuits used in the experimental evaluation and their sequential-cell composition before MBFF clustering.

Design	#cells	#FFs	FF %
ethmac	60948	10546	17.3%
mock-alu	13454	1413	10.5%
riscv32i	10879	1056	9.7%
uart	854	79	9.3%
ibex	21734	1939	8.9%
jpeg	57382	4383	7.6%
gcd	468	35	7.5%
cva6	117754	8465	7.2%
aes	17106	562	3.3%

Table 2 shows the Worst Negative Slack (WNS) and Total Negative Slack (TNS) results obtained after Global Placement (GP) and Detailed Routing (DR).

Comparing MCF-Crit with SFTray, the proposed enhancement consistently outperforms it for five designs (*ethmac*, *mock-alu*, *uart*, *ibex*, and *cva6*) after Detailed Routing (DR). For the *cva6* benchmark, MCF+LP-Crit completely eliminated negative slack (TNS = 0) and recovered timing to achieve a positive final WNS.

Table 3 shows the MBFF distribution, Clock Power, Switching Power, Total Power, and Area comparison. The MCF-Crit and MCF+LP-Crit formulations consistently reduced clock and switching power without incurring area penalties compared to the original SFTray baseline.

5. Conclusion

This work investigated the incorporation of timing-awareness into the SFTray MBFF clustering technique in OpenROAD. We proposed two timing-aware enhancements: MCF-

Crit, which incorporates timing criticality into the MCF assignment cost, and MCF+LP-Crit, which extends it to the LP refinement stage.

Experimental evaluations demonstrate that timing information can be successfully exploited to guide early clustering decisions without degrading the primary power and area benefits of MBFF clustering. MCF-Crit and MCF+LP-Crit consistently outperformed the original SFTray in terms of final post-routing timing (WNS and TNS) for several designs with high sequential-cell densities. These findings suggest that timing-aware clustering is a viable strategy to enhance physical-design Quality of Results in open-source EDA flows.

References

- Cai, Y., Zhu, L., and Guo, X. (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)*, page 7 pages, New York, NY, USA. ACM. ASPDAC '25, January 20–23, 2025, Tokyo, Japan.
- Chang, Y.-C., Lin, T.-W., Jiang, I. H.-R., and Nam, G.-J. (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)*, page 8 pages, New York, NY, USA. ACM. ISPD '19, April 14–17, 2019, San Francisco, CA, USA.
- Kahng, A. B. (2022). Leveling up: A trajectory of openroad, tilos and beyond. In *Proceedings of the 2022 International Symposium on Physical Design, ISPD '22*, page 73–79, New York, NY, USA. Association for Computing Machinery.
- Kahng, A. B., Kundu, S., and Thumathy, S. (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*, page 8 pages, Clearwater, FL, USA. ACM.
- Kahng, A. B., Li, J., and Wang, L. (2016). Improved flop tray-based design implementation for power reduction. In *Proceedings of the 35th International Conference on Computer-Aided Design, ICCAD '16*, New York, NY, USA. ACM.
- Li, C.-Y., Su, C.-C., Chen, Z.-W., Chen, S.-H., and Chang, Y.-W. (2025). Late Breaking Results: Multi-Objective Multi-Bit Flip-Flop Placement Considering Pre-Placed Cells. In *2025 62nd ACM/IEEE Design Automation Conference (DAC)*, New York, NY, USA. IEEE. Late Breaking Results.
- Liu, M.-Y., Lai, Y.-C., Mak, W.-K., and Wang, T.-C. (2022). Generation of Mixed-Driving Multi-Bit Flip-Flops for Power Optimization. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD '22)*, page 9 pages, New York, NY, USA. ACM. ICCAD '22, October 30–November 3, 2022, San Diego, CA, USA.
- Weste, N. H. and Harris, D. M. (2011). *CMOS VLSI Design a Circuit and Systems Perspective*. Addison-Wesley.
- Wu, G., Xu, Y., Wu, D., Ragupathy, M., Mo, Y.-y., and Chu, C. (2016). Flip-flop Clustering by Weighted K-means Algorithm. In *Proceedings of the 53rd Annual Design*

Automation Conference (DAC '16), pages Article 82, 6 pages, New York, NY, USA. ACM. DAC '16, June 05–09, 2016, Austin, TX, USA.

Yang, S.-W., Hsu, J.-W., Cheng, Y.-H., and Shen, C.-F. (2025a). Invited paper: 2025 iccad cad contest problem b: Power and timing optimization using multibit flip-flop. pages 1–6.

Yang, S.-W., Hsu, J.-W., Lee, T.-W., Chen, T.-H., and Shen, C.-F. C. (2025b). 2024 iccad cad contest problem b: Power and timing optimization using multibit flip-flop. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design, ICCAD '24*, New York, NY, USA. Association for Computing Machinery.

Table 2. Worst Negative Slack (WNS) and Total Negative Slack (TNS) comparison after Global Placement (GP) and Detailed Routing (DR) for $\alpha = 8$ and $\beta = 0.1$. All values are in picoseconds (ps). Bold values indicate that MCF-Crit or MCF+LP-Crit achieved a better WNS or TNS timing value than SFTray.

Design	Version	WNS (ps)			TNS (ps)		
		GP	DR	$\Delta(DR - GP)$	GP	DR	$\Delta(DR - GP)$
ethmac	NC	-86.88	-86.27	+0.62	-1492.25	-1464.06	+28.19
	SFTray	-86.88	-82.04	+4.84	-1492.25	-1397.42	+94.83
	MCF-Crit	-86.88	-81.38	+5.50	-1492.25	-1369.36	+122.89
	MCF+LP-Crit	-86.88	-81.38	+5.50	-1492.25	-1369.36	+122.89
mock-alu	NC	-315.57	-283.77	+31.81	-21899.50	-12987.00	+8912.50
	SFTray	-343.18	-302.11	+41.07	-35900.10	-16891.60	+19008.50
	MCF-Crit	-315.58	-281.32	+34.26	-35672.40	-15439.70	+20232.70
	MCF+LP-Crit	-342.13	-300.12	+42.01	-35997.10	-16783.30	+19213.80
uart	NC	-64.65	-28.18	+36.47	-1601.70	-636.38	+965.32
	SFTray	-65.08	-22.87	+42.21	-1796.85	-624.99	+1171.86
	MCF-Crit	-65.16	-19.69	+45.47	-1722.20	-525.07	+1197.13
	MCF+LP-Crit	-65.16	-19.69	+45.47	-1722.20	-525.07	+1197.13
ibex	NC	-29.78	-22.04	+7.74	-10320.60	-166.71	+10153.89
	SFTray	-142.44	-43.83	+98.61	-114973.00	-1005.98	+113967.02
	MCF-Crit	-142.33	-35.90	+106.43	-114834.00	-432.02	+114401.98
	MCF+LP-Crit	-142.33	-35.79	+106.54	-114806.00	-400.23	+114405.77
gcd	NC	5.94	-16.70	-22.64	0.00	-79.20	-79.20
	SFTray	-14.61	-29.24	-14.63	-88.69	-215.44	-126.75
	MCF-Crit	-17.74	-30.51	-12.77	-52.27	-226.54	-174.28
	MCF+LP-Crit	-17.74	-28.32	-10.58	-139.97	-171.45	-31.48
cva6	NC	-38.08	-21.31	+16.76	-105.85	-146.32	-40.47
	SFTray	-38.08	-22.99	+15.09	-105.85	-253.49	-147.64
	MCF-Crit	-38.08	-15.31	+22.76	-105.85	-101.33	+4.52
	MCF+LP-Crit	-64.80	4.75	+69.56	-154.34	0.00	+154.34
aes	NC	-27.10	-10.26	+16.84	-637.80	-103.07	+534.73
	SFTray	-40.56	-33.98	+6.58	-3648.14	-2052.00	+1596.14
	MCF-Crit	-40.34	-36.24	+4.10	-3793.26	-2017.41	+1775.85
	MCF+LP-Crit	-42.08	-32.88	+9.20	-4009.93	-2041.88	+1968.05
riscv32i	NC	-135.54	21.09	+156.63	-66001.90	0.00	+66001.90
	SFTray	-133.13	18.42	+151.55	-66935.80	0.00	+66935.80
	MCF-Crit	-131.54	17.89	+149.44	-66921.30	0.00	+66921.30
	MCF+LP-Crit	-131.54	17.89	+149.44	-66921.30	0.00	+66921.30
jpeg	NC	-23.88	16.63	+40.52	-210.86	0.00	+210.86
	SFTray	-30.35	20.99	+51.34	-361.36	0.00	+361.36
	MCF-Crit	-28.45	10.27	+38.73	-341.76	0.00	+341.76
	MCF+LP-Crit	-27.60	17.11	+44.72	-339.80	0.00	+339.80

Table 3. Comparison of MBFF distribution, Clock Power, Switching Power, Total Power and Area for $\alpha = 8$ and $\beta = 0.1$. Bold values indicate that MCF-Crit or MCF+LP-Crit achieved a better result than SFTray.

Design	Version	#1b	#2b	#4b	FF %	Clk. P (mW)	Swit. P (mW)	Total P (mW)	Area (μm^2)
ethmac	NC	10546	0	0	17.3%	51.000	24.878	111.321	7935.720
	SFTray	1396	31	2272	6.9%	21.500 (-57.84%)	13.523 (-45.64%)	67.949 (-38.96%)	7748.320 (-2.36%)
	MCF-Crit	1380	35	2274	6.9%	21.300 (-58.24%)	13.437 (-45.99%)	67.652 (-39.23%)	7745.510 (-2.40%)
	MCF+LP-Crit	1380	35	2274	6.9%	21.300 (-58.24%)	13.437 (-45.99%)	67.652 (-39.23%)	7745.510 (-2.40%)
mock-alu	NC	1413	0	0	10.5%	6.680	27.989	58.655	1632.380
	SFTray	59	9	334	3.2%	2.430 (-63.62%)	26.323 (-5.95%)	52.034 (-11.29%)	1613.040 (-1.18%)
	MCF-Crit	63	11	332	3.3%	2.410 (-63.92%)	26.201 (-6.39%)	51.808 (-11.67%)	1602.060 (-1.86%)
	MCF+LP-Crit	49	8	337	3.2%	2.410 (-63.92%)	26.146 (-6.58%)	51.728 (-11.81%)	1601.260 (-1.91%)
uart	NC	79	0	0	9.3%	0.257	0.806	2.330	95.207
	SFTray	0	0	20	2.6%	0.052 (-79.77%)	0.706 (-12.46%)	2.082 (-10.67%)	92.977 (-2.34%)
	MCF-Crit	1	1	19	2.7%	0.071 (-72.26%)	0.703 (-12.79%)	2.079 (-10.78%)	92.058 (-3.31%)
	MCF+LP-Crit	1	1	19	2.7%	0.071 (-72.26%)	0.703 (-12.79%)	2.079 (-10.78%)	92.058 (-3.31%)
ibex	NC	1939	0	0	8.9%	2.920	12.254	25.188	2541.780
	SFTray	1691	2	61	8.2%	2.720 (-6.85%)	12.087 (-1.36%)	24.613 (-2.28%)	2529.090 (-0.50%)
	MCF-Crit	1693	1	61	8.1%	2.710 (-7.19%)	12.313 (+0.48%)	25.255 (+0.27%)	2555.220 (+0.53%)
	MCF+LP-Crit	1697	1	60	8.2%	2.730 (-6.51%)	12.086 (-1.38%)	24.621 (-2.25%)	2531.020 (-0.42%)
gcd	NC	35	0	0	7.5%	0.099	0.323	0.918	48.347
	SFTray	5	3	6	3.1%	0.043 (-56.98%)	0.287 (-11.15%)	0.818 (-10.89%)	48.172 (-0.36%)
	MCF-Crit	11	2	5	4.0%	0.050 (-48.99%)	0.295 (-8.86%)	0.836 (-8.94%)	48.129 (-0.45%)
	MCF+LP-Crit	3	0	8	2.4%	0.057 (-42.11%)	0.293 (-9.44%)	0.852 (-7.27%)	49.251 (+1.87%)
cva6	NC	8465	0	0	7.2%	17.500	46.961	132.122	17586.400
	SFTray	6686	668	374	6.6%	17.200 (-1.71%)	47.191 (+0.49%)	131.893 (-0.17%)	17886.800 (+1.71%)
	MCF-Crit	6681	706	380	6.6%	17.100 (-2.29%)	47.846 (+1.88%)	132.515 (+0.30%)	17915.700 (+1.87%)
	MCF+LP-Crit	6683	687	378	6.6%	16.700 (-4.57%)	45.403 (-3.32%)	129.574 (-1.93%)	17888.500 (+1.72%)
aes	NC	562	0	0	3.3%	2.500	98.881	166.145	1838.640
	SFTray	106	12	108	1.3%	1.280 (-48.80%)	99.148 (+0.27%)	166.447 (+0.18%)	1853.060 (+0.78%)
	MCF-Crit	102	12	109	1.3%	1.160 (-53.60%)	99.244 (+0.37%)	166.735 (+0.36%)	1854.810 (+0.88%)
	MCF+LP-Crit	94	12	111	1.3%	1.130 (-54.80%)	99.147 (+0.27%)	166.340 (+0.12%)	1852.560 (+0.76%)
riscv32i	NC	1056	0	0	9.7%	2.560	8.428	34.073	2691.150
	SFTray	56	4	248	3.1%	0.958 (-62.58%)	7.738 (-8.19%)	31.411 (-7.81%)	2659.700 (-1.17%)
	MCF-Crit	56	4	248	3.1%	0.936 (-63.44%)	7.657 (-9.14%)	31.210 (-8.40%)	2657.030 (-1.27%)
	MCF+LP-Crit	56	4	248	3.1%	0.936 (-63.44%)	7.657 (-9.14%)	31.210 (-8.40%)	2657.030 (-1.27%)
jpeg	NC	4383	0	0	7.6%	9.800	56.825	119.187	6436.980
	SFTray	365	49	980	2.6%	4.000 (-59.18%)	54.254 (-4.53%)	109.118 (-8.45%)	6357.010 (-1.24%)
	MCF-Crit	301	41	1000	2.5%	3.980 (-59.39%)	54.200 (-4.62%)	109.032 (-8.52%)	6358.380 (-1.22%)
	MCF+LP-Crit	307	38	1000	2.5%	3.960 (-59.59%)	54.166 (-4.68%)	108.804 (-8.71%)	6354.260 (-1.29%)