Binding Multi-bit Flip-flop Cells through Design and Technology Co-optimization

Jooyeon Jeong
Seoul National University
Seoul, South Korea
jooyeon@snucad.snu.ac.kr

Taewhan Kim
Seoul National University
Seoul, South Korea
tkim@snucad.snu.ac.kr

ABSTRACT

Though using multi-bit flip-flop (MBFF) cells provide the benefit of saving dynamic power, its big cell size with many D/Q-pins inherently entails two critical limitations, which are (1) *the loss of full flexibility in optimizing the wires* connecting to the D/Q-pins in MBFFs and (2) *the loss of selectively resizing i.e., controlling the driving strength of internal flip-flops in MBFFs* to optimize timing. In this work, we propose a comprehensive solution to resolving those limitations through design and technology co-optimization (DTCO) in physical design flow. Specifically, to address limitation 1, given an input circuit with MBFF allocation and binding, at the post-placement stage we explore diverse layouts of MBFF cells with various D/Q-pin locations and *rebind MBFF instances in the circuit to the MBFF cells that are the most suitable for minimizing the wirelength connecting D/Q-pins*. Meanwhile, to address limitation 2, at the post-route stage we explore MBFF cell layouts of non-rectangle, precisely, L- and T-shape to control the driving strength of internal flip-flops selectively, by which we *rebind MBFF instances with negative slack to the area-minimal rectangle or non-rectangle MBFF cells to optimize timing* while increasing the power overhead minimally. Through experiments with benchmark circuits, it is shown that our DTCO driven MBFF rebinding method is able to produce the circuit implementations with 1.42% less wirelength in comparison with that produced by the state-of-the-art commercial EDA tool using MBFFs.

ACM Reference Format:
Jooyeon Jeong and Taewhan Kim. 2024. Binding Multi-bit Flip-flop Cells through Design and Technology Co-optimization. In *61st ACM/IEEE Design Automation Conference (DAC '24), June 23–27, 2024, San Francisco, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3649329.3658264

1 INTRODUCTION

In modern synchronous digital circuits, minimizing the amount of power consumption is one of the utmost concerns in the process of design implementation. Particularly for high-speed circuits, the dynamic power occupies a considerable portion of the total power dissipated in circuits [11].

Since flip-flops are essential components in digital circuits for storing binary data, reducing the dynamic power consumed at the flip-flop instances in circuits has become increasingly important. Numerous research works have shown that using multi-bit flip-flop cells as opposed to using single-bit flip-flop cells enables to save the dynamic power considerably on the reduced clock delivery network as well as on the internals in flip-flops [2, 4, 5, 8, 10]. Fig. 1 shows the internal cell structure of 2-bit flip-flop (MBFF) which is formed by grouping two single-bit flip-flops. Using MBFFs is able to reduce the cell area and dynamic/leakage power since their individual flip-flops can share the internal clock inverters.

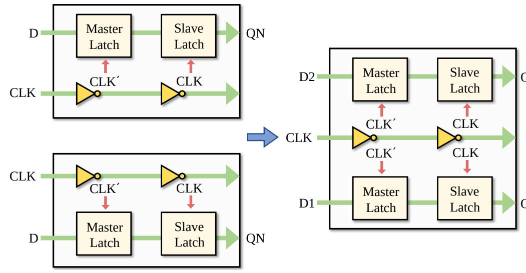
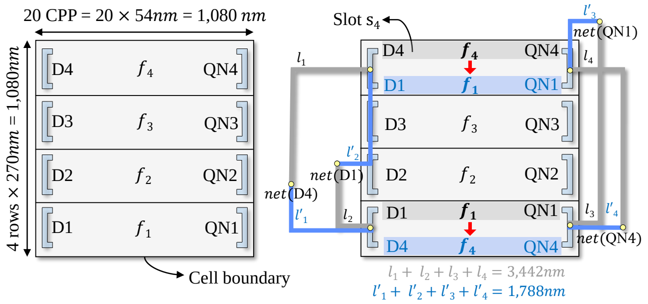


Figure 1: The internal structure of 2-bit flip-flop (right) that is formed by grouping two singe-bit flip-flops (left).



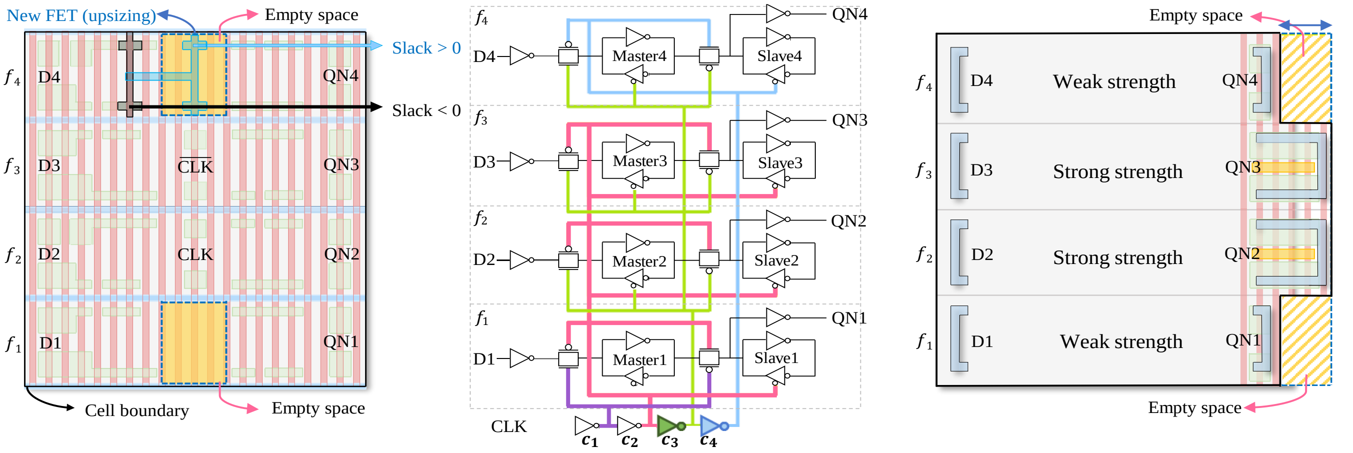
(a) Size of 4-bit MBFF according to PDK in ASAP7nm [7].

(b) Wirelength change by rebinding f_1 to s_4 and f_4 to s_1.

Figure 2: Typical shape of 4-bit MBFF and rebinding flip-flops.

Most of the existing MBFF-related works have focused on how the flip-flops in the initial circuit should be grouped to *allocate MBFF cells* considering the proximity of their location and the timing slack changes, but *no consideration of binding* i.e., how the grouped flip-flops being bound to the internal slots in MBFFs. For example, the work in [13] proposed a graph-based approach for grouping flip-flops, in which it formulated the k-bit MBFF allocation problem into the problem of finding k-cliques in a graph. On the other hand, the work in [6] applied the weighted K-means algorithm to the flip-flop grouping problem while considering the toggle similarity and

DAC '24, June 23–27, 2024, San Francisco, CA, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0601-1/24/06.
https://doi.org/10.1145/3649329.3658264



(a) Partially upsizing some driving transistor(s) [11].

(b) Generating additional clock skew(s) [12].

(c) Not-utilizing the empty space, increasing the resulting chip size to support MBFF cells with mixed driving strengths [13].

Figure 3: Utilization of wasted space in MBFF cells of rectangle shape.

physical proximity between flip-flops. In addition, [7] proposed an approach based on integer linear programming (ILP) to the problem of timing compatibility verification and identifying flip-flops to be grouped.

Recently, a few works have noticed that the big size of MBFF cells significantly influences the net route topology connecting D/Q-pins, as illustrated in Fig. 2(a), in which the height and width of 7nm 4-bit MBFF cell are longer than 1.0um, and the rectangle shape of MBFF cells invariably induces some empty space i.e., space waste in their layouts. In this regards, at the post-route stage, the work in [16] performed two tasks: (1) reducing y-directional wirelength (WL) connecting D/Q-pins by rebinding some internal flip-flops to another slots in MBFFs, as illustrated in Fig. 2(b), in which flip-flops f_1 and f_4 respectively bound to slots s_1 and s_4 are rebound to s_4 and s_1, resulting in reducing the WL connecting D/Q-pins, and (2) utilizing the MBFFs' empty space in upsizing the output driving transistor to optimize timing with no increase on the of MBFF cell size, as illustrated in Fig. 3(a). Similarly, the work in [17] exploited the MBFFs' empty space to include additional clock inverter(s), so that the inherent inflexibility of applying useful clock skew scheduling to MBFFs for timing optimization should be mitigated by providing more than one clock skew, as illustrated in Fig. 3(b). In addition, the work in [9] proposed to use MBFFs with non-uniform drive strengths of their internal flip-flops. However, the rectangle shape constraint of MBFF cells cannot but induce cells with considerable space waste, as illustrated in Fig. 3(c). which may increase the resulting chip size.

In this work, we overcome two critical limitations that occur in implementing circuits with MBFF cells, which are (*limitation 1*) the loss of full flexibility in optimizing the wires connecting to the D/Q-pins in MBFFs and (*limitation 2*) the loss of selectively resizing i.e., controlling the output driving strength of internal flip-flops in MBFFs to optimize timing. The main contributions of this work can be summarized as:

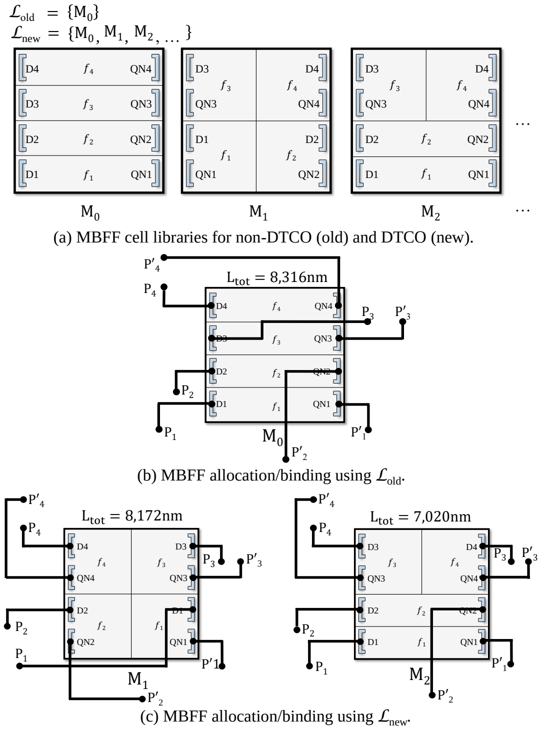
1. We propose a comprehensive solution to resolving the limitations through design and technology co-optimization (DTCO) in physical design flow. To this end, we develop a new MBFF cell library which includes MBFF cells of diverse shapes as well as diverse D/Q-pin locations.
2. To overcome *limitation 1*, given an input circuit with MBFF allocation and binding solution, at the post-placement stage we explore the diverse layouts of MBFF cells with various D/Q-pin locations and rebind every MBFF instance in the circuit to the MBFF cell layout that minimizes the WL connecting D/Q-pins.
3. To overcome *limitation 2*, at the post-route stage we explore MBFF cell layouts of non-rectangle shape in the new MBFF cell library to control the driving strength of internal flip-flops selectively, by which we rebind MBFF instances with negative slack to the area-minimal MBFF cells to optimize timing at the cost of minimal power increase.

2 MOTIVATIONS

Figs. 4 and 5 show how the MBFF allocation and the individual flip-flop binding inside MBFF cells utilizing diverse shapes of MBFF cells affect the design quality.

• *Impact of MBFF allocation and binding on WL*: Assume that the conventional cell library $\mathcal{L}_{old}$ has 4-bit MBFF cell M_0 of simple stacking structure, as shown in Fig. 4(a). Consequently, Fig. 4(b) shows the WL-minimal MBFF binding result. On the other hand, if we can use cell library $\mathcal{L}_{new}$ which includes two more new 4-bit MBFFs M_1 and M_2 shown in Fig. 4(a)[1], the resulting WL-minimal bindings for each of the MBFF cells are shown in Fig. 4(c), among which the allocation and binding to M_2 uses the minimal WL i.e., 7,020nm which is 15.6% reduction on that of the WL-minimal binding to M_0. This example clearly shows that due to the big size of MBFF cells, a comprehensive MBFF allocation and binding utilizing

[1] The layout structure of *two-row* flip-flops in M_1 and M_2 comes from the TSMC patent in [3].


$\mathcal{L}_{old}$ = {M$_0$}
$\mathcal{L}_{new}$ = {M$_0$, M$_1$, M$_2$, ... }
D4 f_4 QN4
D3 f_3 QN3
D2 f_2 QN2
D1 f_1 QN1
M$_0$
M$_1$
M$_2$
...
L$_{tot}$ = 8,316nm
P$_4$ P'$_4$ P$_3$ P'$_3$ P$_2$ P'$_2$ P$_1$ P'$_1$
L$_{tot}$ = 8,172nm
L$_{tot}$ = 7,020nm

(a) MBFF cell libraries for non-DTCO (old) and DTCO (new).

(b) MBFF allocation/binding using $\mathcal{L}_{old}$.

(c) MBFF allocation/binding using $\mathcal{L}_{new}$.

Figure 4: Impact of MBFF allocation and internal flip-flop binding on the D/Q-pin net WL.

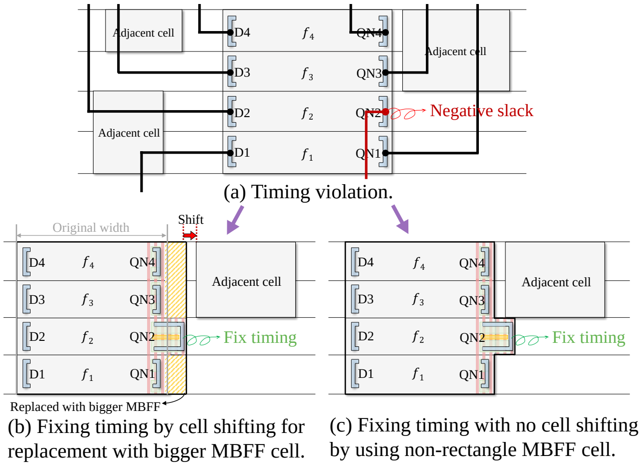
MBFF cells of various internal structures is able to significantly reduce the WL connecting to D/Q-pins.

● *Impact of MBFF allocation and binding on optimizing timing*: The conventional way of fixing timing violation is to replace the cell instance with the one with stronger output driving strength. However, for the MBFFs occupying large footprint, the method demands a considerable cell placement and net route perturbation to replace the MBFF instances with timing violation. Such perturbation requires additional burden on the routing and timing optimization. For example, Fig. 5(b) shows that the cell instance adjacent to the MBFF instance is shifted to make room to accommodate an MBFF cell with stronger driving strength to fix the timing violation, as shown in Fig. 5(a). However, if L- or T-shaped MBFF cells are available, their use can avoid the perturbation on the adjacent cells and connected routes.

3 MBFF REBINDING METHODOLOGY UTILIZING DESIGN AND TECHNOLOGY CO-OPTIMIZATION

Fig. 6 depicts the design flow of our proposed MBFF binding methodology. We synthesize and optimize the input circuit, followed by performing placement by using existing logic synthesis and placement tools, and target cell library with conventional MBFF cells.

In Step 1, by exploring our newly designed MBFF cells with various D/Q-pin locations, a simultaneous rebinding of the MBFF cells and the flip-flops within MBFF cells with the objective of minimizing total WL connecting D/Q-pins. We formulate this problem into an instance of the weighted maximal matching problem and solve it optimally in polynomial time. Then, we conduct net routing by using any existing router. In Step 2, at the post-route stage we perform, by exploiting our newly devised non-rectangle MBFF cells, rebinding MBFF cells to optimize timing while minimizing power overhead as well as minimally perturbing the placement and route that have already been optimized. We formulate this problem into an instance of ILP problem.



(a) Timing violation.

(b) Fixing timing by cell shifting for replacement with bigger MBFF cell.

(c) Fixing timing with no cell shifting by using non-rectangle MBFF cell.

Figure 5: Impact of MBFFs of non-rectangle shapes on cell placement perturbation for optimizing timing.

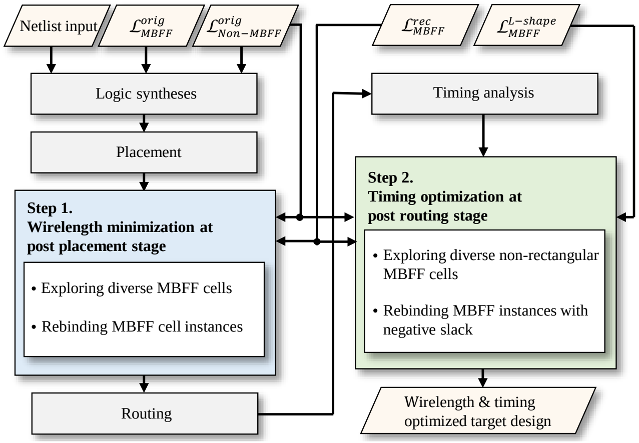

Netlist input
$\mathcal{L}^{orig}_{MBFF}$
$\mathcal{L}^{orig}_{Non-MBFF}$
$\mathcal{L}^{rec}_{MBFF}$
$\mathcal{L}^{L-shape}_{MBFF}$
Logic syntheses
Placement
Step 1.
Wirelength minimization at post placement stage
• Exploring diverse MBFF cells
• Rebinding MBFF cell instances
Routing
Timing analysis
Step 2.
Timing optimization at post routing stage
• Exploring diverse non-rectangular MBFF cells
• Rebinding MBFF instances with negative slack
Wirelength & timing optimized target design

Figure 6: The flow of our proposed MBFF rebinding methodology utilizing design and technology co-optimization.

3.1 MBFF Rebinding for Wirelength Minimization

Let $\mathcal{I} = \{I_1, I_2, \cdots\}$ be the set of K-bit MBFF instances in the circuit after placement and $\mathcal{M} = \{M_1, M_2, \cdots\}$ be the set of K-bit MBFF cells with diverse D/Q-pin locations available to use for rebinding. Our MBFF rebinding problem is that for each $I_i \in \mathcal{I}$, we want to find an MBFF cell $M_j \in \mathcal{M}$ such that the replacement of I_i with M_j leads to the smallest total wirelength connecting to the D/Q-pins.

Table 1: Notations used in our ILP formulation

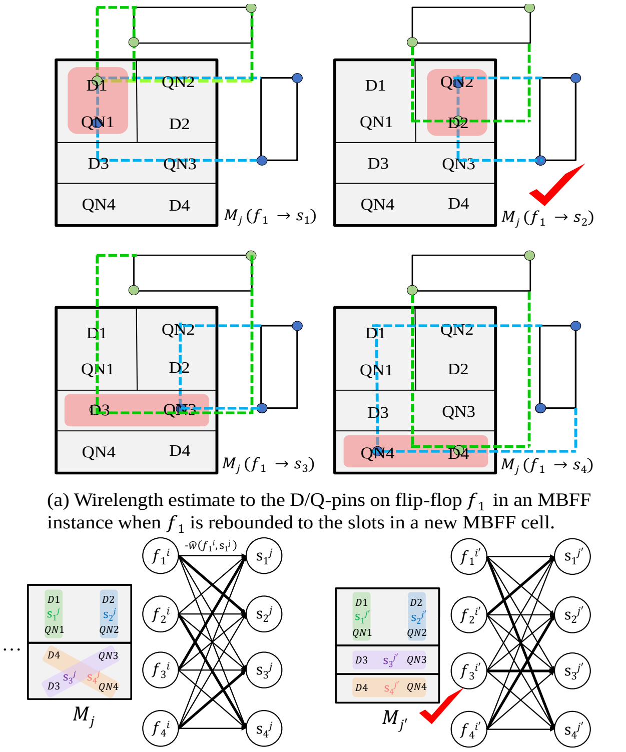
	Description
$\mathcal{M}$	Set of the MBFF instances in the target circuit
$\mathcal{M}_{L/T}$	Set of MBFF cells with transistor upsizing
id_M	Index of an internal flip-flop in $M \in \mathcal{M}$
$id0_{i,k}$	id_M if kth internal flip-flop is the launch register of M_i; -1, else
$id1_{i,k}$	id_M if kth internal flip-flop is the capture register of M_i; -1, else
P_i	Set of the timing paths on $M_i \in \mathcal{M}$. ($\|P_i\| = 2K$ for a K-bit MBFF instance since pins D and Q each has one timing path.)
$p_{i,k}$	$p_{i,k} \in P_i, k = 1, \cdots, \|P_i\|$
PWR_j	Power consumed by $M_j \in \mathcal{M}_{L/T}$
TNS_j	Total negative slack consumed by $M_j \in \mathcal{M}_{L/T}$
x_i^j	1 if $M_i \in \mathcal{M}$ is replaced by $M_j \in \mathcal{M}_{L/T}$; 0, else.
$req_{i,k}$	Required time on path $p_{i,k}$ before replacement
$arr_{i,k}$	Arrival time on path $p_{i,k}$ before replacement
$init\_setup_{i,k}$	Setup time on D of path $p_{i,k}$ before replacement
$init\_c2q_{i,k}$	Clock-to-Q delay on Q of path $p_{i,k}$ before replacement
$slack\_hold_{i,k}$	Hold time slack on path $p_{i,k}$
$slack\_setup_{i,k}$	Setup time slack on path $p_{i,k}$
$re\_setup_{i,k}$	Setup time on D of path $p_{i,k}$ after replacement
$re\_c2q_{i,k}$	Clock-to-Q delay on Q of path $p_{i,k}$ after replacement
$setup_{i,k}^j$	Setup time on pin D of path $p_{i,k}$ when $M_i \in \mathcal{M}$ is replaced with $M_j \in \mathcal{M}_{L/T}$
$c2q_{i,k}^j$	Clock-to-Q delay on pin Q of path $p_{i,k}$ when $M_i \in \mathcal{M}$ is replaced with $M_j \in \mathcal{M}_{L/T}$
$dir_{i,k}$	Direction of the pin. 1 if the pin is on the right side of M_i; 0, else
$e0_{i,k}$	Represents the leftover space on the internal flip-flop's left side.
$e1_{i,k}$	Represents the leftover space on the internal flip-flop's right side.
$u_{j,k}$	Additional space required if the k-th pin of M_j is used.

To this end, we estimate the WL required to connect the D/Q-pins on the k^{th} flip-flop $f_k^i \in I_i$ when f_k^i is bound to the l^{th} slot $s_l^j \in M_j$ by computing:

$$\hat{w}(f_k^i, s_l^j) = \sum_{\rho \in \{D,Q\}} WL(net(f_k^i, \rho), loc(s_l^j, \rho)),\ k, s \in \{1, \cdots, K\} \quad (1)$$

where $net(f_k^i, \rho)$ indicates the net with ρ-pin on f_k^i as its terminal, $loc(s_l^j, \rho)$ denotes the location at which ρ-pin on s_l^j is positioned, and $WL(net(\cdot), loc(\cdot))$ indicates the half-parameter WL of the minimal-size bounding box for $net(\cdot)$ with its ρ-pin located at $loc(\cdot)$. For example, in Fig. 7(a), bounding boxes illustrate nets with D/Q-pins of f_1 in an MBFF instance when f_1 is assigned to s_1, s_2, s_3, and s_4 in a new MBFF cell.

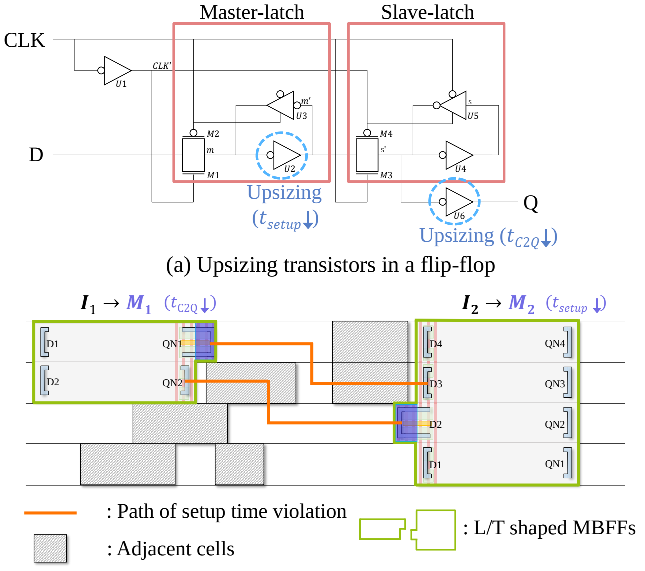
With the $\hat{w}(f_k^i, s_l^j)$ values in *Eq.1*, we minimize the total WL to their D/Q-pins by reassigning flip-flops in I_i to the slots in M_j. This is achieved by finding a weighted maximal matching in a bipartite graph $G(V_1, V_2, W)$ where V_1 and V_2 are respectively the set of K nodes corresponding to f_k^i, $k = 1, \cdots, K$ and s_l^j, $l = 1, \cdots, K$, and the value of $-\hat{w}(f_k^i, s_l^j) \in W$ is assigned to the edge in G corresponding to (f_k^i, s_l^j). For example, Fig. 7(b) shows $G(V_1, V_2, W)$ for the MBFF instance in Fig. 7(a) to be rebound to M_1 and M_2, where the heavy lines show the weighted maximal matching. Then, among all rebinding instances, we select the one with the largest value (i.e., the smallest total WL) of weighted maximal matching.



(a) Wirelength estimate to the D/Q-pins on flip-flop f_1 in an MBFF instance when f_1 is rebounded to the slots in a new MBFF cell.

(b) Graph model for wirelength-minimal flip-flop rebinding in an MBFF.

Figure 7: Illustration of MBFF reallocation and flip-flop rebinding.



(a) Upsizing transistors in a flip-flop

(b) Optimizing timing by replacing MBFFs

Figure 8: Generation of L/T-shaped MBFF cells and their use in timing optimization with no perturbation on cell placement and net route.

3.2 MBFF Rebinding for Timing Optimization

At the post-route, we optimize timing on MBFF instances by upsizing the internal flip-flops selectively while the already optimized placement and route should be intact. To this end, we prepare the set, $\mathcal{M}_{L/T}$, of MBFF cells with the diverse L- and T-shapes with selectively upsizing transistors in flip-flops as illustrated in Fig. 8(a). Then, we attempt to replace the MBFF instances involved in timing violation with MBFF cells in $\mathcal{M}_{L/T}$ to fix the timing or maximally reduce the negative time slack at the cost of minimal increase of power overhead, as illustrated in Fig. 8(b). We formulate this MBFF reallocation using $\mathcal{M}_{L/T}$ for the setup time constraint into an ILP (integer linear programming) as follows:

$$\textbf{max. :} \quad \sum_{i=1}^{|\mathcal{M}|} \sum_{j=1}^{|\mathcal{M}_{L/T}|} ((1-\lambda) \cdot (-PWR_j) + \lambda \cdot TNS_j) \cdot x_i^j \tag{2}$$

subject to:

for all $i = 1, \cdots, |\mathcal{M}|,\ j = 1, \cdots, |\mathcal{M}_{L/T}|,\ k = 1, \cdots, |P_i|$,

$$slack\_setup_{i,k} = (req_{i,k} + init\_setup_{i,k} - re\_setup_{i,k}) - (arr_{i,k} - init\_c2q_{i,k} + re\_c2q_{i,k}) \tag{3}$$

$$re\_setup_{i,k} = \begin{cases} init\_setup_{i,k} & \text{if } id1_{i,k} = -1 \\ setup_{i,k}^{j} & \text{if } x^{j}_{id1_{i,k}} = 1 \end{cases} \tag{4}$$

$$re\_c2q_{i,k} = \begin{cases} init\_c2q_{i,k} & \text{if } id0_{i,k} = -1 \\ c2q_{i,k}^{j} & \text{if } x^{j}_{id0_{i,k}} = 1 \end{cases} \tag{5}$$

$$slack\_hold_{i,k} = (arr_{i,k} - init\_c2q_{i,k} + re\_c2q_{i,k}) - (req_{i,k} - init\_hold_{i,k} + re\_hold_{i,k}) \tag{6}$$

$$TNS_j = min(slack\_setup_{i,k}, slack\_hold_{i,k}, 0) \tag{7}$$

$$(d_i^k \cdot e1_{i,k}) + ((1 - d_i^k) \cdot (e0_{i,k})) \geq u_j^k \tag{8}$$

$$\sum_{i=1}^{|\mathcal{M}|} \sum_{j=1}^{|\mathcal{M}_{L/T}|} x_i^j = 1 \tag{9}$$

The objective function in *Eq.2* is to minimize the weighted sum of the power consumption and total negative slacks across all MBFF instances in the target circuit. The hyperparameter λ in between 0 and 1 is used to adjust the extent to which power and negative slack are minimized. *Eq.3* ensures that all paths within the internal flip-flops of the MBFFs in circuit meet the setup timing constraints.[2] *Eq.4* and *Eq.5* evaluate MBFF cells within $\mathcal{M}_{L/T}$ to identify an MBFF cell.

Eq.7 is included for explanatory purposes. To convert the `min` operations into an ILP form, we employed a trick by adding the following constraints, which effectively guide TNS_j to select the minimum value:

$$slack\_setup_{i,k} = slack\_setup^{+}_{i,k} + slack\_setup^{-}_{i,k} \tag{9}$$

$$slack\_hold_{i,k} = slack\_hold^{+}_{i,k} + slack\_hold^{-}_{i,k} \tag{10}$$

$$slack\_setup^{+}_{i,k} \geq 0, \quad slack\_setup^{-}_{i,k} \leq 0 \tag{11}$$

$$slack\_hold^{+}_{i,k} \geq 0, \quad slack\_hold^{-}_{i,k} \leq 0 \tag{12}$$

This allows us to express TNS as:

$$TNS_j = \sum_{k \in |P_i|} (slack\_setup^{-}_{i,k} + slack\_hold^{-}_{i,k}). \tag{13}$$

Adopting L- or T-shaped cell instances introduces a new challenge of potential overlap with adjacent cells, not seen in the conventional cell layouts. To prevent the overlap, *Eq.8* is utilized, ensuring pins are correctly positioned relative to the cell's sides. This approach permits the use of L- or T-shaped cells only when adequate space is available, preventing overlap with neighboring cells. And *Eq.9* guarantees that each MBFF should be replaced with exactly one MBFF cell.

4 EXPERIMENTAL RESULTS

We implemented our DTCO driven MBFF binding methodolgy on a Linux machine equipped with an AMD Ryzen 3970X CPU operating at 2.2GHz and 128GB of memory with Python and the Gurobi optimizer[15] for solving ILP models. Logic synthesis was performed using Synopsys Design Compiler [20], and placement and routing were conducted with Cadence Innovus [1]. We created various layouts of MBFF cells using the ASAP 7nm standard cells and design rules in [12]. Since the open cell library does not provide MBFF cells, we used Cadence Virtuoso for manually creating only rectanglar-shaped MBFF cells. To generate non-rectangular-shaped MBFFs, we added space by SKILL language for upsizing transistor to the left or right boundary of the rectanglar-shaped MBFF cells. And we verified the cells with Mentor Calibre [14] and characterized them with Synopsys Primelib [19].

The following experiments compare: (1) *Conventional flow using rectangular-shaped MBFF cells only*, (2) *Conventional flow including our non-rectangular-shaped MBFF cells*, (3) *Our proposed flow with step 1 only*, and (4) *Our proposed flow with steps 1 and 2.* We used IWLS 2005 OpenCores benchmark circuits [18] for the experiments. Table 2 shows the number of cells (#cells), nets (#nets), and MBFFs (#MBFF instances) used in each benchmark with conventional flow 1, which is produced by applying Cadence Innovus. #Rep.MBFFs indicates the number of MBFF cell instances replaced in steps 1 and 2. We set chip utilization to 0.8.

Table 2: Totals for cells, nets, MBFFs in conventional flow, and our MBFF replacements in our approach.

	Conventional flow 1			**Our DTCO-flow**			
				Step 1		**Step 2**	
Circuit	#Cells	#Nets	#MBFFs	#Rep.MBFFs	Ratio (%)	#Rep.MBFFs	Ratio (%)
DES3	1759	2071	24	22	91.66	23	95.83
WB_DMA	2420	2962	83	34	40.96	80	96.38
AC97_CTRL	7277	8798	458	217	47.38	382	83.40
WB_CON_TOP	20458	21612	10	1	10	9	90
ETH_TOP	29205	36287	2356	1360	57.72	1877	79.66
Average	.	.	.	.	49.54	.	89.05

[2] The hold time constraints regarding $slack\_hold_{i,k}$ in *Eq.6* can be formulated similarly. Related parameters and constraints have been omitted in the paper for the space limitation.

Table 3: Power, performance and area (PPA) comparison for the designs produced by the conventional flows and our flow.

		Conventional flow			Our DTCO-flow			
Circuit		**Flow 1**	**Flow 2 (vs Flow 1)**		**Step 1 (vs Flow 1)**		**Step 1 + Step 2 (vs Flow 1)**	
DES3	Area	4761.71	4921.27	3.35%	4622.44	-2.92%	4637.6	-2.6%
	WL	21145.97	21489.07	1.62%	21569.22	2%	21658.48	2.42%
	#DRVs	0	4	<400	1	<400	4	<400
	TNS	-72.93	-80.97	11.02%	-79.24	8.65%	-79.23	8.63%
	WNS	-1	-0.99	-1%	-1.08	8%	-1.08	8%
	Power	3.57	3.7	3.64%	3.43	-3.92%	3.47	-2.8%
WB_DMA	Area	8796.75	8868.37	0.81%	7837.5	-10.9%	7888.36	-10.32%
	WL	33148.9	39706.18	19.78%	32096.71	-3.17%	32253.62	-2.7%
	#DRVs	0	5	<400	0	<400	7	<400
	TNS	-15.86	-21.76	37.2%	-42.23	166.27%	-42.19	166.01%
	WNS	-0.15	-0.21	40%	-0.23	53.33%	-0.24	60%
	Power	2.72	2.8	2.94%	2.2	-19.12%	2.23	-18.01%
AC97_CTRL	Area	27234.97	27256.43	0.08%	24664.69	-9.44%	24934.6	-8.44%
	WL	100649.6	118090.7	17.33%	99741.29	-0.9%	100042.92	-0.6%
	#DRVs	1	77	<400	0	<400	60	<400
	TNS	-418.13	-412.18	-1.42%	-489.65	17.1%	-488.96	16.94%
	WNS	-0.3	-0.32	6.67%	-0.35	16.67%	-0.35	16.66%
	Power	42.31	43.11	1.89%	36.26	-14.3%	36.88	-12.83%
WB_CON_TOP	Area	45454.6	44868.84	-1.29%	41122.13	-9.53%	41126.79	-9.52%
	WL	409470.5	397951.8	-2.81%	403576.69	-1.44%	403794.47	-1.38%
	#DRVs	50	63	<400	45	<400	65	<400
	TNS	-113.16	-100.88	-10.85%	-232.9	105.81%	-232.49	105.45%
	WNS	-0.52	-0.35	-32.69%	-0.84	61.54%	-0.84	61.53%
	Power	8.58	8.39	-2.21%	7.15	-16.67%	7.16	-16.55%
ETH_TOP	Area	113707	118588.8	4.29%	103133.55	-9.3%	104464.93	-8.12%
	WL	662278.3	746432.8	12.71%	652263.95	-1.51%	656291.44	-0.9%
	#DRVs	4	268	<400	12	<400	362	<400
	TNS	-711.07	-445.93	-37.29%	-2282.53	221%	-2276.56	220.16%
	WNS	-0.82	-0.98	19.51%	-0.62	-24.39%	-0.65	-20.73%
	Power	34.5	36.87	6.87%	29.46	-14.61%	30.04	-12.92%
Average	Area	39991	40900.75	2.27%	36276.06	-9.29%	36610.45	-8.45%
	WL	245338.7	264734.1	7.91%	241849.57	-1.42%	242808.18	-1.03%
	#DRVs	11	83.4	<400	11.6	<400	99.6	<400
	TNS	-266.23	-212.34	-20.24%	-625.31	134.88%	-623.88	134.34%
	WNS	-0.55	-0.57	2.15%	-0.62	11.83%	-0.63	13.26%
	Power	18.33	18.97	3.48%	15.7	-14.38%	15.95	-12.98%

Table 3 compares PPA between conventional and our flows, covering die area (Area in μm^2), wirelength (WL in μm), design rule violations (#DRVs), worst negative slack (WNS in ps), total negative slack (TNS in ps), and power (Power in mW). It is seen that our flow made a meaningful improvement in timing and wirelength with a slight power increase. The rise in #DRVs with L/T-shaped MBFFs indicates commercial tools struggle with non-rectangular cells. This finding underscores the importance of analyzing signal connection patterns beyond just designing L/T-shaped cells, a useful insight confirmed by our experiments.

5 CONCLUSION

Though MBFFs are very attractive for implementing low-power and high-speed chip, their use invariably induces two limitations: (1) less flexibility in optimizing wires connecting D/Q-pins and (2) more space waste in cell layout to support mixed-driving strengths for optimizing timing. This work overcame these two limitations through design and technology co-optimization (DTCO). We resolved the first limitation, at the post-placement, by exploring our newly designed MBFFs of various D/Q-pin locations and formulating the MBFF allocation and (internal flip-flop) binding problem into an instance of maximal weighted matching problem while we resolved the second problem, at the post-route, by exploring our non-rectangle-shaped MBFFs and formulating the MBFF rebinding problem into an instance of ILP problem.

6 ACKNOWLEDGEMENT

This work was supported by Samsung Electronics Company, Ltd. (IO201216-08205-01, and IO221227-04376-01), Samsung Advanced Institute of Technology (IO230223-05124-01), National Research Foundation grant by Korea Government (2021- R1A2C2008864), Institute of Information and communications Technology Planning and Evaluation (IITP) grant by Korea government (2021-0- 00754), Software Systems for AI Semiconductor Design and Artificial Intelligence Semiconductor Support Program to nurture the best talents (IITP-2023-RS-2023- 00256081) grant by Korea government, National R&D Program through NRF by Ministry of Science and ICT (2020M3H2A1078119), and BK21 Four Program of Education and Research Program for Future ICT Pioneers, Seoul National University. The EDA tool was supported by IC Design Education.

REFERENCES

[1] Cadence. 2016. Innovus user guide. https://www.cadence.com

[2] V. Dhivya and P. Prakasam. 2014. Effective and efficient technique for power reduction by multi-bit flip-flops. In *IEEE International Conference on Electronics Circuits and Systems.*

[3] C. Liu et al. 2017. Flip-flop with delineated layout for reduced footprint. Patent No. 9641161, 2017.

[4] C. Santos et al. 2012. Multi-bit flip-flop usage impact on physical synthesis. In *IEEE Symposium on Integrated Circuits and Systems Design.*

[5] G. Prakash et al. 2013. Achieveing reduced area by multi-bit flip flop design. In *IEEE International Conference on Computer Communication and Informatics.*

[6] G. Wu et al. 2016. Flip-flop clustering by weighted k-means algorithm. In *ACM/EDAC/IEEE Design Automation Conference.*

[7] I. Seitanidis et al. 2019. Timing-driven and placement-aware multibit register composition. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2019).

[8] M. Lin et al. 2011. Recent research in clock power saving with multi-bit flip-flops. In *IEEE International Midwest Symposium on Circuits and Systems.*

[9] M. Liu et al. 2022. Generation of mixed-driving multi-bit flip-flops for power optimization. In *IEEE/ACM International Conference on Computer-Aided Design.*

[10] V. Ashna et al. 2013. Design of low power clocking system using merged flip-flop technique. In *IEEE International Conference on Computer Communication and Informatics.*

[11] V. G. Oklobdzija et al. 2005. *Digital system clocking: high-performance and low-power aspects.* John Wiley & Sons.

[12] V. Vashishtha et al. 2017. ASAP7 predictive design kit development and cell design technology co-optimization. In *IEEE/ACM International Conference on Computer-Aided Design.*

[13] Y. Chang et al. 2010. Post-placement power optimization with multi-bit flip-flops. In *IEEE/ACM International Conference on Computer-Aided Design.*

[14] Mentor Graphics. 2008. Graphics mentor, calibre verification user's manual.

[15] Gurobi. 2023. Gurobi optimizer reference manual. https://www.gurobi.com

[16] S. Kim and T. Kim. 2022. Design and technology co-optimization utilizing multi-bit flip-flop cells. In *IEEE/ACM International Conference on Computer-Aided Design.*

[17] S. Kim and T. Kim. 2023. Design and technology co-optimization for useful skew scheduling on multi-bit flip-flops. In *IEEE/ACM International Conference on Computer-Aided Design.*

[18] Opencores. 2005. Open-source IP-cores, www.opencores.org.

[19] Synopsys. [n. d.]. PrimeLib, https://www.synopsys.com/implementation-and-signoff/signoff/primelib.html.

[20] Synopsys. 2016. Design compiler user guide N-2016.12-SP5.