Standard Cell Layout Generator Amenable to Design Technology Co-Optimization in Advanced Process Nodes

Handong Cho*, Hyunbae Seo*, Sehyeon Chung*, Kyu-Myung Choi†, and Taewhan Kim*
*†Department of Electrical and Computer Engineering, Seoul National University
*{chhd25, hyunbaeseo, shchung, tkim}@snucad.snu.ac.kr, †kmchoi@snu.ac.kr

***Abstract*—To generate standard cell (SC) layouts of competitive quality, pin accessibility and in-cell routing congestion should be thoroughly taken into account. In this work, we develop a new tool to address this issue. Precisely, we (1) develop a technology compilation module that can convert diverse cell architectures and design rules into grid based design parameters and layer configuration, (2) generate optimal FET placement using metrics that can accurately and efficiently predict intra-cell pin accessibility and in-cell routing congestion, and (3) introduce the concept of ghost-via and ghost-metal, and formulate in-cell routing using satisfiability modulo theory for pin separation and extension. Experimental results show that our system is able to synthesize SC layouts with a routing completion rate of 95∼98%, which is far better than the previous SC layout generator, and produce layouts comparable to the ARM's hand-crafted layouts. In addition, the design implementations produced by using our 2-layer 1D SC library exhibit on average 76.6% fewer design rule violations (DRVs) with similar or better quality of timing and area, while in comparison with that produced by using the library of hand-crafted ARM SCs, the implementations produced by using our 1-layer 2D SC library exhibit on average 11.7% smaller area with comparable timing and DRV count.**

I. Introduction

The concept of design and technology co-optimization (DTCO) [1] is to evaluate FET architecture, design rules, cell architecture, and design architectures (e.g., power delivery schemes and metal stacks) using design-level PPA metrics analyses. One of the most important prerequisites is an efficient DTCO framework, which can analyze the variations of the items mentioned above in a short time. The automatic standard cell (SC) layout generator, which enables generating SC layout for technology independence and generating SCs with competitive quality in a reasonable time, is the most important engine for the DTCO framework. Therefore, firstly, the SC layout generator must generate diverse cell architectures *quickly*. Secondly, to generate SC layouts of competitive quality in advanced process nodes, we should carefully consider *pin accessibility* and *in-cell routing congestion*. These three considerations are the most important items to be taken into the SC layout generator for the DTCO framework in advanced process nodes.

To our knowledge, many works have addressed SC layout generators, partially considering technology independence. The work in [2] proposed a method to find the FET placement with a minimum number of contacted poly pitches (CPPs) together with FET folding using a search tree based algorithm for the netlists and design rules (DRs) in ASAP7 PDK [3]. The work in [4] proposed an approach that combined dynamic programming and integer linear programming (ILP) for SC layout generation. This approach considered M0 routing (i.e., routing on the metal-0 layer) as well as the additional requirement of the drain-to-drain abutment. On the other side, the work in [5] used the satisfiability modulo theory (SMT) formulation to generate SC layouts for the netlists and DRs in ASAP7 PDK with the modification of using a 2-layer 1D routing scheme. The work in [6] proposed a DTCO framework that automatically evaluated the design rule violations and SC area as the design rules changed, and claimed that tight integration of automatic SC layout generation made the DTCO process more efficient. However, the applicability of this work was confined to a 1-layer 2D routing scheme. Thus, the framework shall require a non-trivial engineering effort to deal with variations in cell architecture.

With the increasing significance of pin accessibility, there are a number of works that have considered pin accessibility in physical design and SC generation. The work in [7] proposed a metric called remaining pin accessibility (RPA) to evaluate the remaining pin access points at the post-route stage. On the other hand, the work in [8] reflected the RPA metric in the SMT formulation of generating SC layouts. However, their methodologies still lack fully expressing the pin accessibility.

To sum up, unlike the prior SC layout generators, which are short of a full consideration of simultaneously supporting (1) *being free of technology dependence* (e.g., 2-layer 1D or 1-layer 2D) and (2) *being aware of net congestion* and *pin accessibility* in the course of FET placement and in-cell routing, our SC generation framework takes those factors fully and effectively into account. The distinct and main features of this work are listed in the following.

1. We develop a technology compilation module that translates various cell architectures and design rules into a set of grid based design parameters and layer configurations. We also develop an in-cell router that supports both 2-layer 1D and 1-layer 2D in-cell routing schemes.
2. We propose a set of novel metrics, which are carefully comprehending intra-cell pin accessibility and in-cell routing congestion, and more powerful in expressing pin accessibility than ever existing metrics.

979-8-3503-4859-0/DATE24/© 2024 EDAA

3. To boost the assurance in pin accessibility in our SC layout generation, we introduce a new concept called *ghost-via* and *ghost-metal*, based on which we develop our pin accessibility-aware in-cell router.

II. SC LAYOUT GENERATOR FOR DTCO FRAMEWORK OF ADVANCED PROCESS NODES

A. The Proposed SC Layout Generation System

Fig. 1 shows our proposed SC layout generation system supporting variations in cell architecture, design rules, and in-cell routing schemes for advanced process nodes. The main flow consists of three steps: (1) performing a technology compilation step that receives cell architecture and design rules as input, and generates grid based design parameters and layer configurations. Table I shows a set of possible variations (i.e., tuning options) in cell architecture and design rules supported by technology compilation; (2) sorting the FET placement order for optimal in-cell routing completion and pin accessibility from diverse FET placements generated by CSyn-fp [2], and inserting dummy polys around pins with the worst intra-cell connection pin accessibility to enhance in-cell routability of FET placement; (3) performing in-cell routing considering pin accessibility based on in-cell routing schemes. If in-cell routing is incomplete, we insert dummy polys around unrouted pins to make them routed. Once the SC layouts are synthesized, the SCs are characterized by extracting their delay and power to build a cell library, $\mathcal{L}$.

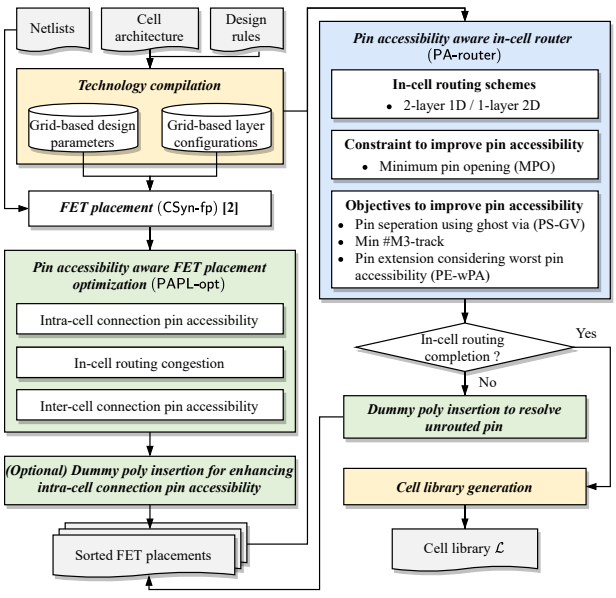

Fig. 1. Our SC layout generation system aims to improve in-cell routing completion and pin accessibility.

B. Pin Accessibility-aware FET Placement Optimization

CSyn-fp [2] not only generates the entire solution to the FET placement but also provides the solution expected to have better routability. However, the routability predicted by CSyn-fp is not so accurate since the estimation relies on a simplified SC routing model, which in turn causes to waste lots of iterations of in-cell routing to find an optimal FET placement. Thus, an in-depth investigation for better in-cell routing completion and pin access is necessary. To this end, we propose the following three new metrics: (1) intra-cell connection pin accessibility, (2) in-cell routing congestion to increase the in-cell routing completion, and (3) inter-cell connection pin accessibility to ensure pin access when connecting to other cells. We use those metrics to select an optimal FET placement by minimizing the objectives in lexicographical order ((a) #CPP, (b) intra-cell connection pin accessibility cost, (c) in-cell routing congestion cost, (d) inter-cell connection pin accessibility cost).

TABLE I
VARIATIONS OF CELL ARCHITECTURE (CA) AND DESIGN RULES (DR), THE TECHNOLOGY COMPILATION MODULE WILL TRANSLATE.

Type		Meaning	Option
CA	CH	Cell height	6, 7.5, 9
	DB	Diffusion break	single DB, double DB
	FIN	The number of fins for FET	2, 3, 4
	RD	Routing direction	1-layer 2D in-cell routing, 2-layer 1D in-cell routing
	RT	The number of routing tracks	5, 5.5, 6, 9
	MPO	Minimum pin opening	1, 2, 3
	COAG	Allowing contact over active gates	true, false
DR	MAR	Minimum area rule	any value
	EOL	End-of-line spacing rules (side-to-side (S2S), side-to-tip (S2T), tip-to-tip (T2T), corner-to-corner (C2C))	any value
	VR	Via spacing rule	any value

We assume a pin set $\mathcal{P}$ is composed of pin p_i, each corresponding to the single source/drain/gate in the FET. $p_{i,j}$ is a point of pin p_i, the intersection point of the vertical and horizontal grid inside the pin p_i. $|p_i|$ denotes the number of points inside pin p_i. We decompose the multi-pin net into two-pin nets. Let $\mathcal{N}$ be the set of two-pin net n_k. Our prediction assumes that net n_k in $\mathcal{N}$ would be routed within the bounding box BB_{n_k} of $M \times N$ dimensions.

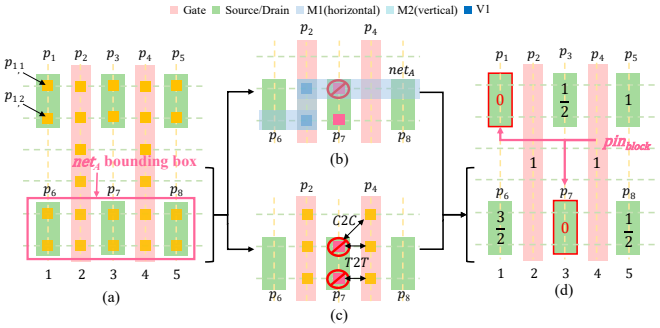

Fig. 2. Example of predicting intra-cell connection pin accessibility. (a) Each point of pin $p_{i,j}$ is marked with a yellow dot. (b) Two points at p_7 within the bounding box of net_A are each blocked by a routing segment of net_A with probability of $\frac{1}{2}$. The disturbing net probability of p_7, $DNP(p_7)$, is $\frac{1}{2} + \frac{1}{2} = 1$. (c) The points on p_7 may be blocked if each of the three points on adjacent pins p_2 and p_4 is accessed. The disturbing pin probability of p_7, $DPP(p_7)$, is $\frac{1}{|p_2|} \cdot 3 + \frac{1}{|p_4|} \cdot 3 = \frac{1}{2} + \frac{1}{2} = 1$. (d) The value on each pin p_i represents $RIPA(p_i)$ in *Eq.1* where p_1 and p_7 are inaccessible since $RIPA(p_1) = RIPA(p_7) = 0$. Thus, $Cost_{intra}$ in *Eq.2* is 0 + 2 = 2.

1) Intra-cell Connection Pin Accessibility: To predict the intra-cell connection pin accessibility, we compute the poten-

tial disturbance by routing segments and pins associated with other nets. We first consider the disturbing net probability (DNP), which means the potential disturbance by the net segments, depicted in Fig. 2(b). We define the probability of a routing segment of net n_k traversing other net's pin point $p_{i,j}$ to be $BB_{n_k}(p_{i,j}) = \frac{1}{N}$. Then, we define a function, $DNP(p_i)$, as the sum of the probability of all nets $n_k \in \mathcal{N}$ traversing pin points in p_i. Second, we consider the disturbing pin probability (DPP), which means the potential disturbance by pins of other nets, depicted in Fig. 2(c). We assume the use of a single pin point $p_{i,j}$ for each pin p_i and assign a probability of pin presence to each pin point $p_{i,j}$ as $\frac{1}{|p_i|}$. We define DRV_{p_i} to be the set of pin points closer to p_i than the spacing distance e.g., T2T and C2C rules in DRs.[1] Then, we define a function, $DPP(p_i)$, as the sum of the probability of $p_{i',j'} \in DRV_{p_i}$. Upon acquiring $DNP(p_i)$ and $DPP(p_i)$ through the above procedure, we compute *the remaining intra-cell connection pin access* ($RIPA(p_i)$) for each pin p_i:

$$RIPA(p_i) = |p_i| - (DNP(p_i) + DPP(p_i)). \quad (1)$$

Then, we define the set of blocked pins, $p_{block} = \{p_i | RIPA(p_i) \leq 0\}$, which is the collection of pins with no remaining space to access, from which we compute *the intra-cell connection pin accessibility cost*, $Cost_{intra}$ (See Fig. 2(d) for an example.):

$$Cost_{intra} = -min(RIPA(p_i)) + |p_{block}|. \quad (2)$$

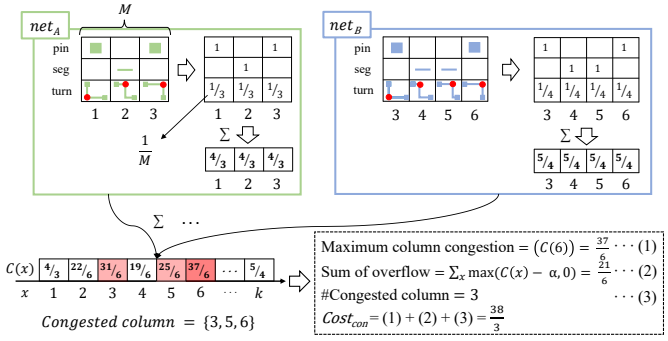

Fig. 3. Illustration of calculating column congestion. The first step is estimating the column congestion of net_A, whose pins are located in columns 1 and 3. The probability of column occupation by pin and net segments is 1 for each column, while turn segments are located in columns 1, 2, and 3, with a probability of $\frac{1}{3}$. Accumulating such three components for the respective column, we evaluate the column congestion by net_A. The next step is summing all congestion values of the column on the same x-coordinates in all nets. We can calculate the final congestion value as depicted in the figure.

2) In-cell Routing Congestion: To predict the in-cell routing congestion, we propose column congestion, which measures the congestion level at each column. We break down each net n_k into three components: 'pin', 'seg', and 'turn'. 'pin' and 'seg' respectively indicate a pin and a routing segment. If two pins on a net are located at different x and y positions, a 'turn' is needed to connect them. This increases congestion since a 'turn' requires an additional routing segment. The probability of a 'turn' occurring in a column is $\frac{1}{M}$. Aggregating 'pin', 'seg', and 'turn' across all nets within each column yields the corresponding column's congestion value, $C(x)$. We define overflow as $max(C(x) - \alpha, 0)$.[2] This value is subsequently employed to determine the maximum column congestion, the sum of overflow, and the number of congested columns. We finally calculate the congestion cost as a sum of these three factors.

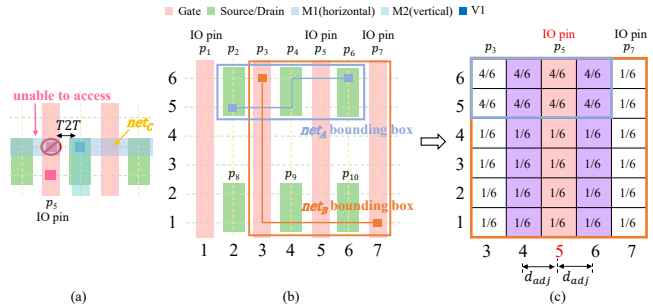

Fig. 4. Illustration of calculating inter-cell connection pin accessibility. (a) If net_C's segment is located adjacent to IO pin p_5, access to p_5 is interrupted. (b) The bounding boxes of net_A and net_B are blue and orange boxes, respectively, and pins p_1, p_5, and p_7 are IO pins. (c) The congestion map of FET placement can be obtained by summing congestion of net_A and net_B. The adjacent congestion, ADJ_{p_5}, can be calculated by summarizing purple bins closer to p_5 than d_{adj}, i.e., $ADJ_{p_5} = \frac{4}{6} \times 4 + \frac{1}{6} \times 8 = 4$. Then, the inter-cell connection pin accessibility cost $Cost_{inter} = ADJ_{p_1} + ADJ_{p_5} + ADJ_{p_7}$.

3) Inter-cell Connection Pin Accessibility: We propose IO pin congestion, which measures the congestion around the IO pin. The net segments adjacent to the IO pins can disturb access to those pins, depicted as Fig. 4(a). However, it is hard to find the exact locations where net segments will occupy before routing. Therefore, we use a bounding box to predict the congestion of net components and evaluate the FET placement based on the assumption that lower congestion around IO pins will make IO pin access easier. We define the probability of a routing segment of net n_k traversing the bin at coordinates (x, y) to be $BB_{n_k}(x, y) = \frac{1}{N}$. Aggregating these bin values across the entire grid and for all nets yields a congestion map, as shown in Fig. 4(c). d_{adj}[3] represents the distance where DRV occurs due to the short distance between segments. We can calculate the adjacent congestion, ADJ_{p_i}, by summing the congestion of adjacent bins closer to p_i than d_{adj}. The cost can be calculated by summing all IO pin congestion values of IO pins $p_i \in \mathcal{P}_{\mathcal{IO}}$.

C. Pin Accessibility-aware In-cell Router

We propose SMT formulation for in-cell routing focused on constraint and objectives to improve pin accessibility. It is based on the newly devised concept of ghost-via and ghost-metal. Also, our SMT formulation is constructed not only for 2-layer 1D but also for 1-layer 2D in-cell routing. The values

[1]T2T (tip-to-tip) and C2C (corner-to-corner) spacing values on metal 1 (M1) layer are computed by our technology compilation module where each is set to 1.

[2]We set α to #horizontal-track - 2. For 2-layer 1D in-cell routing, $\alpha = 4$.

[3]d_{adj} is the same as T2T spacing and set to 1.

calculated in the technology compilation module are used to formulate grid based design rules.

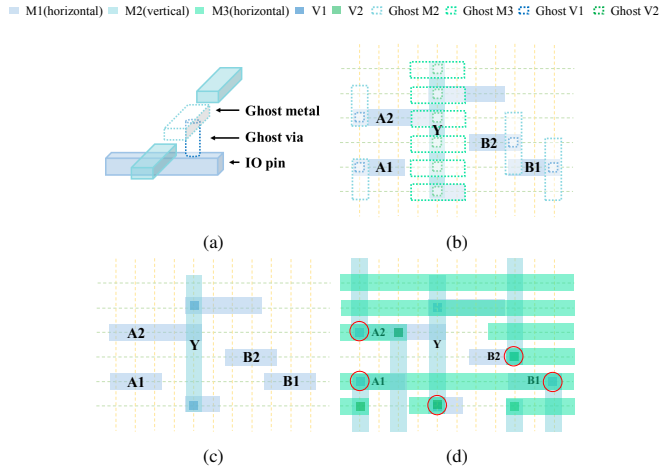

Fig. 5. (a) and (b) Ghost-via/metal. (c) After in-cell routing is completed, the ghost-via/metals are removed. (d) In block-level routing, the IO pin is connected to the blue circle point at which a ghost-via had occupied.

1) Ghost-via and Ghost-metal: We introduce ghost-via/metal in in-cell routing, as illustrated in Fig. 5. They are used to reserve, during in-cell routing, a pair of M1-M2 via and M2 metal contact segment over an access point in IO pin to acquire a pin access route in chip implementation. Ghost-vias are used to connect IO pins to M2 while ghost-metals in M2 are used to contact ghost-vias to meet the minimum area rule (MAR). Ghost-via/metals are then removed from the SC layout after in-cell routing.

2) Constraint-based Pin Accessibility Improvement: We formulate a stronger MPO (minimum pin opening) constraint such that each IO pin should acquire as many ghost-via/metals as the ordinary MPO value. Definitely, the ghost-via/metal should meet all DRs on metals used in in-cell routing. Thus, the new MPO constraint with ghost-via/metal ensures a minimum number of unblocked access points through M2, as illustrated in Fig. 6.

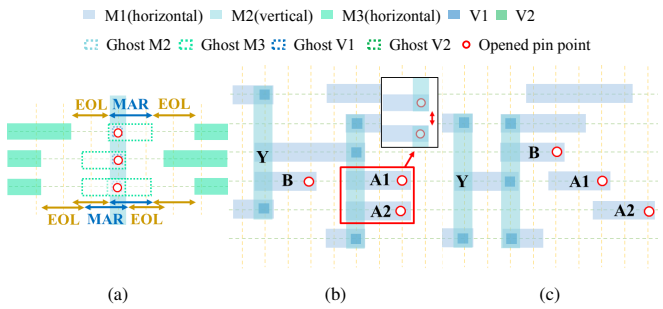

Fig. 6. Example of MPO constraint. (a) MPO constraint using ghost-via/metal ensures a minimum number of unblocked access points by the metal on the upper layer. (b) If only the upper metal is considered as [8], the access point may be obstructed by the metal accessing other IO pins. (c) MPO constraint using ghost-via/metal ensures the access point without obstruction because ghost-via/metals of other IO pins are considered.

3) Objective-based Pin Accessibility Improvement: We propose objectives to improve pin accessibility: pin separation using ghost-via (PS-GV) and pin extension considering the worst pin accessibility (PE-wPA).

To increase the number of accessible points on IO pin p, we use PE-wPA represented by *Eqs.3* and *4*, as illustrated in Fig. 7. By *Eq.3*, we extend the (worst) pin on which the number of ghost-vias is not more than the MPO. Then, by *Eq.4*, we extend the rest of the pins while keeping the already improved pin accessibility.

$$maximize \sum_{p \in IO} \min(\#Ghost\text{-}via(p) - (MPO + 1),\ 0) \quad (3)$$

$$maximize \sum_{p \in IO} \#Ghost\text{-}via(p) \quad (4)$$

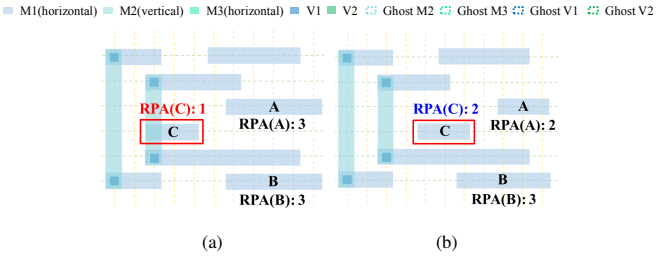

Fig. 7. Example of pin extension. (a) Pin extension without considering the worst accessible pins. RPA [7] of the worst accessible pin, C, is 1. (b) PE-wPA increases the RPA of pin C to 2 and improves the worst RPA to 2.

We optimize the objectives in lexicographical order ((a) PS-GV, (b) min #M3-track, (c) PE-wPA, (d) min #Via, (e) min #Metal).

III. Experimental Results

We implemented the proposed SC layout generator using Python and C++ on a Linux machine with an AMD Ryzen 3970X processor running at 2.2GHz and 128GB of memory. Our formulation was executed using SMT solver Z3 (ver.4.8.11) [9]. We defined two cell architectures: 1-layer 2D and 2-layer 1D in-cell routings[4] to assess the effectiveness of our proposed SC layout generator supporting all in-cell routing schemes. We synthesized 172 SC layouts in ASAP7 PDK. For each SC layout, we validated the DRC, LVS, and PEX rules using Mentor Calibre and characterized the delay and power of the SCs using Synopsys PrimeLib. We used OpenCores circuits [11] in the experiments. We used Synopsys Design Compiler for logic synthesis and Cadence Innovus for placement and routing.

A. 2-layer 1D SC Layout Competitiveness

Table II shows the comparison of SC layouts generated by CSyn-fp + UCSD[5], CSyn-fp + PA-router, and PAPL-opt + PA-router for 2-layer 1D in-cell routing. The average

[4]1-layer 2D in-cell routing is based on ASAP7 PDK, with M1 using 5.5-track 2D routing and M2 using 6-track 1D horizontal routing. 2-layer 1D in-cell routing is based on modified ASAP7 PDK to support COAGs [10], with M1/M3 using 6-track 1D horizontal and M2 using 1D vertical routing.

[5]To make a fair comparison, we apply the ordinary MPO and EB-PS in [8] instead of our proposed stronger MPO, PS-GV, and PE-wPA. We set both of the ordinary and our MPO values to 1.

TABLE II

COMPARISON OF SC LAYOUTS USING 2-LAYER 1D IN-CELL ROUTING FROM CSyn-fp + UCSD, CSyn-fp + PA-router, AND PAPL-opt + PA-router LIBRARIES. #RCS, #CPPS, $aRPA$, $wRPA$, IOC, AND M3 INDICATE THE NUMBER OF CELLS WITH IN-CELL ROUTING COMPLETED, CELL WIDTH, AVERAGE RPA, WORST RPA, INACCESSIBILITY OF CELL, AND M3 USAGE (%), RESPECTIVELY. RPA AND IOC ARE METRICS DEFINED IN [7].

Cell		#Cell	CSyn-fp [2] + UCSD [8]						CSyn-fp [2] + PA-router						PAPL-opt + PA-router					
			#RC	#CPP	aRPA	wRPA	IOC	M3	#RC	#CPP	aRPA	wRPA	IOC	M3	#RC	#CPP	aRPA	wRPA	IOC	M3
Comb.	AND/OR	20	20	8.9	2.67	1.15	0.00	0.00	20	8.9	3.25	2.03	0.00	0.42	20	8.9	3.25	2.03	0.00	0.00
	AO/OA	32	28	10.8	2.30	0.92	-0.11	3.61	28	10.8	2.63	1.23	-0.08	4.91	31	11.0	2.67	1.27	-0.05	4.93
	AOAI/OAOI	4	4	7.5	2.30	1.00	0.00	0.00	4	7.5	2.58	1.13	0.00	0.00	4	8.0	2.58	1.00	0.00	0.00
	AOI/OAI	40	35	8.0	2.24	0.70	-0.48	0.59	35	8.0	2.52	0.97	-0.20	0.82	40	8.0	2.53	1.16	-0.13	1.44
	BUF/HB/INV	27	27	9.4	4.15	2.15	0.00	0.00	27	9.4	4.76	3.11	0.00	0.00	27	9.0	4.76	3.11	0.00	0.00
	FA/HA	2	1	11.5	3.50	2.00	0.00	0.00	1	11.5	4.55	3.00	0.00	0.00	1	12.0	4.15	3.30	0.00	0.00
	MAJ	3	0	8.7	2.42	0.00	-1.00	0.00	0	8.7	2.83	0.00	-1.00	0.00	3	10.0	3.33	2.00	0.00	0.00
	NAND/NOR	23	20	9.0	3.24	1.63	0.00	0.56	20	9.0	3.60	2.48	0.00	0.49	22	9.0	3.65	2.55	0.00	0.26
	XOR/XNOR	6	6	10.7	3.67	2.33	0.00	0.95	6	10.7	4.67	2.67	0.00	0.00	6	11.0	4.89	3.00	0.00	0.00
Seq.	DHL/DLL	6	6	14.0	2.78	1.17	0.00	0.00	6	14.0	4.00	2.50	0.00	0.00	6	14.0	4.28	2.83	0.00	0.00
	ASYNC/DFF	9	9	20.6	2.94	1.33	0.00	4.00	9	20.6	4.19	2.78	0.00	4.22	9	21.0	4.24	2.67	0.00	2.91
Sum / Avg.		172	156	9.97	2.86	1.26	-0.15	1.13	156	9.97	3.37	1.90	-0.08	1.44	169	10.02	3.41	2.01	-0.04	1.44
Impr.									-	-	17.7%↑	50.4%↑	0.07↑	0.31↑	13↑	0.6%↑	19.2%↑	59.4%↑	0.11↑	0.31↑

runtime per cell is less than 14.96 seconds. Compared to CSyn-fp + UCSD, CSyn-fp + PA-router achieves significant improvements i.e., 17.7% on average RPA, 50.4% on worst RPA, and 0.07 in IOC. These enhancements are attributed to the utilization of PA-router's MPO constraints and PS-GV/PE-wPA objectives, effectively enhancing pin accessibility metrics. Compared to CSyn-fp + PA-router, PAPL-opt + PA-router creates 13 more SCs. PAPL-opt demonstrates its ability to improve pin accessibility and in-cell routing congestion in SC layouts. The increase in #CPP and M3 usage is due to the dummy poly insertion and net detouring to improve in-cell routing congestion and pin accessibility. Using PAPL-opt improves average RPA by 1.19%, worst RPA by 5.79%, and IOC by 0.04.

TABLE III

COMPARISON OF THE CHIP IMPLEMENTATIONS PRODUCED BY USING CSyn-fp + UCSD, CSyn-fp + PA-router, AND PAPL-opt + PA-router LIBRARIES IN TERMS OF THE NUMBER OF DRVS, WNS (WORST NEGATIVE SLACK IN ps), TOTAL CELL AREA (um^2), TOTAL WL (um), AND THE TOTAL AMOUNT OF POWER CONSUMPTION (mW).

Design		CSyn-fp [2] + UCSD [8]	CSyn-fp [2] + PA-router		PAPL-opt + PA-router	
AES_CIPHER	#DRV	88	26	**62**↓	13	**75**↓
	WNS	0.03	0.01	0.02↓	0.01	0.02↓
	Area	15336	15546	1.4%↑	15791	3.0%↑
	WL	171645	175147	2.0%↑	172862	0.7%↑
	Power	2.35	2.34	0.4%↓	2.31	1.7%↓
WB_CONMAX	#DRV	787	225	**562**↓	156	**631**↓
	WNS	3.78	1.77	2.01↓	2.46	1.32↓
	Area	25629	25498	0.5%↓	25590	0.2%↓
	WL	474120	467322	1.4%↓	465188	1.9%↓
	Power	2.99	2.93	1.8%↓	3.01	0.6%↑
ETHERNET	#DRV	326	9	**317**↓	0	**326**↓
	WNS	0.71	0.53	0.18↓	0.04	0.67↓
	Area	77688	77591	0.1%↓	77470	0.3%↓
	WL	785545	810653	3.2%↑	808273	2.9%↑
	Power	16.52	16.58	0.3%↑	16.60	0.4%↑
AES_128	#DRV	233	491	**258**↑	166	**67**↓
	WNS	0.00	0.03	0.03↑	0.06	0.06↑
	Area	171768	171152	0.4%↓	171888	0.1%↑
	WL	1756981	1721202	2.0%↓	1716168	2.3%↓
	Power	36.20	36.29	0.3%↑	37.08	2.4%↑

To evaluate the effectiveness of PAPL-opt and PA-router, we adjust the clock period and chip utilization of each design so that the worst negative slack outcomes converge to $-0ps$. Table III summarizes the PPA results of the chip implementations produced by three SC libraries using 2-layer 1D in-cell routing. Compared to using CSyn-fp + UCSD library, we reduce the number of DRVs, on average, by 47.6% and 76.6% by using CSyn-fp + PA-router and PARL-opt + PA-router, respectively, while WNS is maintained as similar positive slacks. Also, the other PPA metrics of area, total wirelength, and power consumption are similar, i.e., 12 increases and 12 decreases within the $\pm$ 3% range.

B. 1-layer 2D SC Layout Competitiveness

Table IV shows the comparison of SC layouts generated by ARM (hand-crafted) [3], ARM FET placement + PA-router, and PAPL-opt + PA-router for 1-layer 2D in-cell routing. The average runtime per cell is less than 18.85 seconds. ARM's hand-crafted layouts use off-grid in-cell routing and extend the IO pins as long as possible, making it difficult to compare one-to-one with our PA-router that uses on-grid in-cell routing. Compared to ARM FET placement + PA-router, PAPL-opt + PA-router creates 30 more cells. In particular, PAPL-opt reduces the #CPP of cells by 0.2% despite dummy poly insertion. PAPL-opt demonstrates the ability to improve pin accessibility and in-cell routing congestion in 2-layer 1D as well as 1-layer 2D in-cell routings. PAPL-opt achieves significant improvements i.e., 12.1% on average RPA, 44.9% on worst RPA, and 0.62 on IOC.

To evaluate the effectiveness of PAPL-opt and PA-router in 1-layer 2D in-cell routing, we implement the chip using the same clock period and chip utilization in Sec. III-A for each design. Table V summarizes the PPA results of the chips. Compared to using the ARM library, using PAPL-opt + PA-router increases #DRVs a little (by 3 and 50) in two designs and keeps the same in two designs. WNS maintains a similar positive slack. Also, the area decreases by 11.73% due to the reduction in #CPP. However, power increases by 11.94% on average. This may be due to M2 usage increase, especially in flip-flops. In summary, our SC layout generator system produces competitive results compared to ARM's hand-crafted layouts in a very short time whereas hand-crafted layouts need enormous cost (time $\times$ manpower). The average runtime by ours per cell is less than 20 seconds. Thus, the total runtime is 3,205 seconds, i.e., less than 1 hour.

TABLE IV

COMPARISON OF SC LAYOUTS USING 1-LAYER 2D IN-CELL ROUTING OF THE ARM (HAND-CRAFTED), ARM FET PLACEMENT + PA-router, AND PAPL-opt + PA-router LIBRARIES. THE MEANING OF EACH ABBREVIATION IS THE SAME AS THAT IN TABLE II. THE PERCENTAGE VALUES IN PAPL-opt + PA-router ARE COMPUTED WITH RESPECT TO ARM FET PLACEMENT + PA-router.

Cell		#Cell	ARM (hand-crafted) [3]					ARM FET placement [3] + PA-router						PAPL-opt + PA-router					
			#CPP	aRPA	wRPA	IOC	M2	#RC	#CPP	aRPA	wRPA	IOC	M2	#RC	#CPP	aRPA	wRPA	IOC	M2
Comb.	AND/OR	20	8.9	4.01	3.20	0.00	0.00	20	8.90	2.37	0.91	-0.19	2.80	20	8.85	2.80	1.25	0.00	3.40
	AO/OA	32	11.1	2.77	1.27	0.00	0.00	22	11.1	1.58	0.55	-1.30	3.06	29	11.2	2.26	1.16	-0.03	3.41
	AOAI/OAOI	4	7.3	3.02	1.88	0.00	0.00	4	7.3	1.69	0.64	-0.59	1.50	4	7.5	2.06	1.00	0.00	0.00
	AOI/OAI	40	8.2	2.77	1.44	0.00	0.00	23	8.2	1.62	0.37	-1.19	2.18	40	8.4	2.44	1.13	0.00	1.83
	BUF/HB/INV	27	9.4	5.69	5.37	0.00	0.00	26	9.4	4.57	2.53	-0.03	7.65	26	9.4	4.96	2.96	0.00	7.38
	FA/HA	2	11.5	6.01	1.00	0.00	21.76	0	11.5	4.05	0.00	-1.50	25.00	1	9.0	3.25	2.00	0.00	13.00
	MAJ	3	8.7	3.21	1.83	0.00	0.00	0	8.7	2.06	0.00	-1.08	7.67	3	9.7	2.42	1.00	0.00	6.67
	NAND/NOR	23	8.6	4.37	2.71	0.00	0.87	21	8.6	3.54	0.97	-0.19	5.22	22	9.1	3.22	1.52	0.00	3.48
	XOR/XNOR	6	10.7	5.72	4.00	0.00	5.28	4	10.7	4.77	1.83	0.00	17.50	4	10.7	3.47	2.33	0.00	11.83
Seq.	DHL/DLL	6	16.0	5.33	4.00	0.00	17.63	6	16.0	3.11	2.50	0.00	8.33	6	14.2	2.72	1.67	0.00	18.83
	ASYNC/DFF	9	22.4	5.50	4.00	0.00	16.84	7	22.4	2.97	1.56	-0.11	11.44	8	21.3	2.84	1.00	0.00	27.25
Sum / Avg.		172	10.20	3.97	2.72	0.00	2.05	133	10.20	2.69	1.07	-0.63	5.26	163	10.18	3.02	1.55	-0.01	5.84
Impr.														30↑	0.2%↓	12.1%↑	44.9%↑	0.62↑	0.58↑

TABLE V

COMPARISON OF THE CHIP IMPLEMENTATIONS PRODUCED BY USING ARM (HAND-CRAFTED), ARM FET PLACEMENT + PA-router, AND PAPL-opt + PA-router LIBRARIES. THE MEANING OF EACH ABBREVIATION IS THE SAME AS THAT IN TABLE III.

Design		ARM [3]	ARM FET placement [3] + PA-router		PAPL-opt + PA-router	
AES_CIPHER	#DRV	0	0	-	0	-
	WNS	0.01	0.01	-	0.07	0.06↑
	Area	16392	16326	0.4%↓	15447	5.8%↓
	WL	171645	175147	2.0%↑	172862	0.7%↑
	Power	2.27	2.90	27.8%↑	2.70	18.9%↑
WB_CONMAX	#DRV	0	0	-	0	-
	WNS	0.24	2.18	1.94↑	7.40	7.16↑
	Area	26730	25886	3.2%↓	25763	3.6%↓
	WL	395422	397818	0.6%↑	400693	1.3%↑
	Power	2.88	2.95	2.4%↑	2.96	2.8%↑
EHTERNET	#DRV	0	0	-	50	**50**↑
	WNS	3.43	0.22	3.21↓	0.04	3.39↓
	Area	83427	82237	1.4%↓	80152	3.9%↓
	WL	726182	732722	0.9%↑	731091	0.7%↑
	Power	17.73	23.53	32.7%↑	23.42	32.1%↑
AES_128	#DRV	0	0	-	3	**3**↑
	WNS	4.62	0.20	4.42↓	0.07	4.55↓
	Area	206192	173398	15.9%↓	172342	16.4%↓
	WL	1552056	1558710	0.4%↑	1492272	3.9%↓
	Power	33.72	34.42	2.1%↑	34.28	1.7%↑

IV. CONCLUSION

This paper proposed an SC layout generator targeting the DTCO framework. The key capabilities of our SC generator were technology independence, a high rate of layout completeness with optimal or near-optimal SC quality, and high pin accessibility in SC layouts. Through experiments, it was confirmed that our generator was able to synthesize SC layouts with a routing completion rate of 95∼98%, which was far better than that by the prior state-of-the-art SC generator. In addition, the chip implementations produced by using our 2-layer 1D SC library exhibited 76.6% fewer DRV counts with similar or better timing and area quality, while comparing those produced by using hand-crafted ARM layouts, the implementations produced by our 1-layer 2D SC library exhibited 11.7% reduced area with comparable timing and DRV count.

V. ACKNOWLEDGEMENT

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

REFERENCES

[1] A. P. Jacob *et al.*, "Scaling challenges for advanced cmos devices," 2017.
[2] K. Baek and T. Kim, "Simultaneous transistor folding and placement in standard cell layout synthesis," *Proc. ICCAD*, 2021.
[3] L. T. Clark *et al.*, "Asap7: A 7-nm finfet predictive process design kit," *Microelectronics Journal*, 2016.
[4] Y.-L. Li *et al.*, "Nctucell: A dda-aware cell library generator for finfet structure with implicitly adjustable grid map," *Proc. DAC*, 2019.
[5] D. Lee *et al.*, "Sp&r: Smt-based simultaneous place-and-route for standard cell synthesis of advanced nodes," *IEEE TCAD*, 2020.
[6] K. Jo *et al.*, "Design rule evaluation framework using automatic cell layout generator for design technology co-optimization," *IEEE TVLSI*, 2019.
[7] J. Seo *et al.*, "Pin accessibility-driven cell layout redesign and placement optimization," *Proc. DAC*, 2017.
[8] C.-K. Cheng *et al.*, "Complementary-fet (cfet) standard cell synthesis framework for design and system technology co-optimization using smt," *IEEE TVLSI*, 2021.
[9] L. De Moura and N. Bjørner, "Z3: An efficient smt solver," 2008.
[10] C. Auth *et al.*, "A 10nm high performance and low-power cmos technology featuring 3 rd generation finfet transistors, self-aligned quad patterning, contact over active gate and cobalt local interconnects," in *Proc. IEEE International Electron Devices Meeting (IEDM)*, 2017.
[11] OpenCores, "Open-Source IP-Cores." [Online]. Available: https://opencores.org