SP&R: SMT-Based Simultaneous Place-and-Route for Standard Cell Synthesis of Advanced Nodes

Daeyeal Lee, *Student Member, IEEE*, Dongwon Park, *Graduate Student Member, IEEE*, Chia-Tung Ho, *Student Member, IEEE*, Ilgweon Kang, *Member, IEEE*, Hayoung Kim, *Member, IEEE*, Sicun Gao, *Member, IEEE*, Bill Lin, *Member, IEEE*, and Chung-Kuan Cheng, *Life Fellow, IEEE*

***Abstract*— In this article, we propose an automated standard cell synthesis framework, SP&R, which simultaneously solves P&R without deploying any sequential/separate operations, by a novel dynamic pin allocation scheme. The proposed SP&R utilizes the multiobjective optimization feature of satisfiability modulo theories (SMT) to obtain optimal cell layouts. To achieve practical scalability of the framework, we develop various search-space reduction techniques, including breaking symmetry, conditional assignment/localization, and cell/objective function partitioning. Compared to the previous work, SP&R achieves 20.8× to 131.7× runtime improvements on average across the design-rule sets. As a result, SP&R successfully produces cell layouts up to 36 field-effect transistors (FETs) and 27 nets within 1.75 h by orchestrating all innovative tactics together, resulting in the generation of a whole 7-nm standard cell library. Compared to the known layouts, our work improves cell size and #*M*2 tracks by 0.1 contacted poly pitch and 0.3 tracks, respectively.**

***Index Terms*—Automatic standard cell generation, conditional design rules, constraint satisfaction problem (CSP), optimization, pin accessibility, satisfiability modulo theory (SMT).**

I. Introduction

As DEVICE integration-process technologies are continuously shrinking, standard cell synthesis has raised critically challenging problems. In particular, the gap between device and metal pitches becomes much larger in the cutting-edge technology nodes, so the number of available routing tracks per each row (i.e., cell height) is much smaller [1]. Consequently, sets of conditional design rules are newly introduced and/or modified, ensuring manufacturable IC layouts on sophisticated multiple-patterning technologies, such as litho-etch-litho-etch (LELE), self-aligned double patterning (SADP), and self-aligned quadruple patterning (SAQP) [2]. As a result, highly increased layout-design complexity obstructs the prompt development of standard cell libraries for the efficient design technology co-optimization (DTCO) workflow [3]. To overcome the current limitation and improve performance, power, area, and cost (PPAC) tradeoff, the automation of standard cell-layout design takes essential roles for achieving seamless technology transition and design-based equivalent scaling through manufacturability-aware standard cell layout design [1], [4]–[6]. However, designing an optimal-layout standard cell is nontrivial and extremely laborious since it requires to explore enormously large search space combined with complicated constraints of transistor-level placement and in-cell routing. Due to these difficulties, most of the previous works focus on divide-and-conquer-style subproblems and/or heuristic approaches, sacrificing optimality.

Standard Cell Synthesis: For transistor-level placement problem, many approaches have been proposed to reduce the search space by adopting heuristic approaches, such as "Eulerian trail" [7], [8], "Branch and Bound" [9], "Transistor connection pruning" [10], etc. For in-cell routing problem, several approaches based on traditional "Maze Routing algorithm" [11], [12] are suggested but inapplicable to modern multiple-patterning technologies because of the complex design rules. An SADP-aware routing solution is presented [13], and several pin-accessibility optimization techniques have attracted considerable attention to improve the pin-accessibility of standard cells in sub-7-nm technology [13]–[16]. However, these approaches that rely on solving subproblems are hard to reach the optimal solution of standard cell layout because of the intractable search space partitioning and the intrinsic limitation of heuristic methodology. For the automation of standard cell layout design procedure, a few works [17], [18] co-optimizing transistor-level placement and in-cell routing are published, however, these works are not suitable for the multiple-patterning technologies

Manuscript received May 25, 2020; revised September 2, 2020; accepted October 26, 2020. Date of publication November 13, 2020; date of current version September 20, 2021. The work of Chung-Kuan Cheng was supported by NSF under Grant CCF-1564302. This article was recommended by Associate Editor D. Z. Pan. *(Daeyeal Lee and Dongwon Park contributed equally to this work.) (Corresponding author: Daeyeal Lee.)*

Daeyeal Lee, Dongwon Park, Chia-Tung Ho, Hayoung Kim, and Bill Lin are with the Department of Electrical and Computer Engineering, University of California at San Diego, San Diego, CA 92093 USA (e-mail: ldaeyeal@ucsd.edu; dwp003@ucsd.edu; c2ho@ucsd.edu; hak005@ucsd.edu; billlin@ucsd.edu).

Ilgweon Kang is with the Digital and Signoff Group, Cadence Design Systems Inc., San Jose, CA 95134 USA (e-mail: ikang@cadence.com).

Sicun Gao and Chung-Kuan Cheng are with the Department of Computer Science and Engineering, University of California at San Diego, San Diego, CA 92093 USA (e-mail: sicung@ucsd.edu; ckcheng@ucsd.edu).

Digital Object Identifier 10.1109/TCAD.2020.3037885

in sub-7 nm. Recently, sub-7 nm applicable automatic standard cell synthesis frameworks have been proposed [19]–[21], however, following sequential and heuristic approaches in place-and-route (P&R) phase.

Satisfiability Modulo Theories (SMTs): Compared to SAT (Boolean satisfiability), SMT is a more expressive language containing nonBoolean variables (e.g., integer, bit vector, etc.) and predicate symbols as described in [22]. Several SMTs solvers, including the optimization methodology [i.e., optimization modulo theories (OMT)] are recently released [23], [24]. By virtue of SAT's fast reasoning ability, SMT-based methodology enables us to represent the given standard cell layout design problem with much richer modeling language. Park *et al.* [25] proposed an SMT-based automation framework that simultaneously solves the P&R problems without deploying any sequential procedures (between place and route steps). However, even if the authors demonstrate the feasibility of the framework, there are still rooms to further improve, e.g., the scalability to deal with a whole set of practical standard cell library [26].

A. Our Contributions

In this article, we propose a novel SMT-based framework that simultaneously optimizes place-and-route (SP&R) of standard cell layout in the highlight of practical design features and the improved scalability, resulting in the generation of a whole set of a standard cell library. Our contributions are as follows.

1) We propose an automated standard cell synthesis framework, SP&R, which simultaneously solves P&R optimization problems. We devise an innovative dynamic pin allocation (DPA) to integrate placement and routing steps into a single optimization procedure.
2) We develop efficient search-space reduction techniques, such as breaking design symmetry, conditional assignment, and objective function partitioning, including heuristic methods, e.g., localization of the routing region and cell partitioning, to improve the scalability.
3) SP&R utilizes an SMT solver, capable of SAT-based fast reasoning with an OMT-featured multiobjective optimization.
4) SP&R covers a wide variety of conditional design rules for securing design for manufacturing (DFM), producing pin-accessibility-aware cell layouts.
5) SP&R provides practical cell-design features to further optimize cell sizes and secure the stable operation of timing-critical sequential logic cells. For example, the use of: a) single diffusion breaks (DBs) with a crossover and b) crosstalk mitigations (CMs) of timing critical nets.
6) SP&R achieves an average of 20.8× to 131.7× runtime improvements over that of reported in [25], by paying less than 0.2% degradation of the total metal length (ML).
7) We demonstrate that our framework SP&R successfully generates a whole set of 7-nm standard cell library [26] with layouts, improving cell size and #*M*2 tracks by 0.1 contacted poly pitch (CPP) and 0.3 tracks on average compared to the known layouts, respectively.

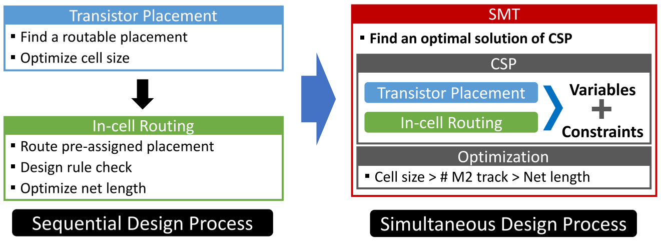

Transistor Placement
▪ Find a routable placement
▪ Optimize cell size
In-cell Routing
▪ Route pre-assigned placement
▪ Design rule check
▪ Optimize net length
Sequential Design Process
SMT
▪ Find an optimal solution of CSP
CSP
Transistor Placement
In-cell Routing
Variables
+
Constraints
Optimization
▪ Cell size > # M2 track > Net length
Simultaneous Design Process

Fig. 1. Sequential versus our proposed simultaneous cell design processes.

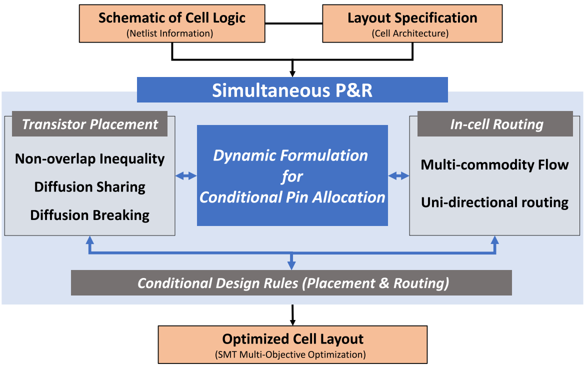


Fig. 2. Proposed simultaneous P&R framework.

The remainder of this article is organized as follows. Section II introduces our framework's configuration. Section III describes constraint formulation for the simultaneous P&R multiobjective problem. Section IV presents the scalability improvement techniques. Section V discusses our experimental setup/results. Section VI concludes this article.

II. Framework Preliminary

This section introduces an overview of the proposed SP&R framework, SMTs, multiobjective optimization, and target cell architecture.

A. Overview of SP&R Framework

We formulate a conventional (sequential) standard cell layout process as a constraint satisfaction problem (CSP) with variables and constraints to integrate P&R steps into a multiobjective optimization problem as shown in Fig. 1. We adopt the state-of-the-art *lazy*-approach SMT solver *Z3* [23], [27] to solve the given optimization problem. Fig. 2 illustrates an overview of our SP&R framework. Given netlist information and a cell architecture, our framework simultaneously obtains an optimal solution that strictly satisfies the constraints of transistor placement, in-cell routing, and conditional design rules. The individual placement and routing problems are combined by our novel dynamic formulation for conditional pin allocation (i.e., DPA). The notations are shown in Table I.

TABLE I
NOTATIONS FOR THE PROPOSED SP&R FRAMEWORK

Term	Description
X	The number of vertical tracks in the given cell bounding box
T	Set of FETs
t	t^{th} FET
ff_t	0-1 indicator if FET t is flipped
x_t	x-axis coordinate of lower-left corner of t
w_t (or h_t)	Width (or height) of FET t
P^t	Set of internal pins of FET t
p_i^t	i^{th} pin of FET t
$n(p)$	Net information of pin p
$G(V,E)$	Three-dimensional (3-D) routing graph
$V(V_i)$	Set of vertices in (i^{th} metal layer of) the routing graph G
v	A vertex with the coordinate (x_v, y_v, z_v)
v_d	A d-directional[1]adjacent vertex of v
$a(v)$	Set of adjacent vertices of v
$e_{v,u}$	An edge between v and u, $u \in a(v)$
$w_{v,u}$	Weighted cost for metal segment on $e_{v,u}$
N	Set of multi-pin nets in the given routing box
n	n^{th} multi-pin net
s^n	A source of n
D^n	Set of sinks of n
d_m^n	m^{th} sink of n
f_m^n	A two-pin subnet connecting s^n and d_m^n, i.e., a commodity
v^n	0-1 indicator if v is used for n
$e_{v,u}^n$	0-1 indicator if $e_{v,u}$ is used for n
$f_m^n(v,u)$	0-1 indicator if $e_{v,u}$ is used for commodity f_m^n
$m_{v,u}$	0-1 indicator if there is a metal segment on $e_{v,u}$
$C_m^n(v,u)$	Capacity variable for $e_{v,u}$ of commodity f_m^n
$g_{d,v}$	0-1 indicator if v forms d-side EOL of a metal segment

B. Satisfiability Modulo Theories

On top of efficient problem-solving ability of SAT, SMT provides the feature of OMT [23], [24] to obtain the optimal solution. Furthermore, SMT formulas support much richer modeling language (e.g., "if-then-else" for the "either-or" constraint, built-in Boolean cardinality functions, such as "at-most k," "at-least k," etc.) than is possible with SAT or integer linear program (ILP) formulas. These key features of SMT efficiently accomplish exhaustive searching for the optimal solution with the concise expressions of constraints. Sections III and IV, respectively, describe our methodology to develop SMT formulation of constraints for SP&R and our techniques to improve SP&R's scalability.

C. Multiobjective Optimization

SP&R has multiple objectives associated with placement and routing problems for standard cell layout design. The cell size is defined as the maximum occupation of vertical tracks by field-effect transistors (FETs) as shown in (1). The number of $M2$ (i.e., top-most metal layer) tracks is defined as the number of occupied $M2$ routing tracks in a generated cell (2). The total ML is the weighted sum of the routed metal segments as shown in (3). In practice, the cell size has the highest priority because it has a direct impact on the footprint area of the entire IC layout. The number of $M2$ tracks is regarded as a more important metric than Total ML: 1) to enhance the cell's PPAC tradeoff by suppressing the usage of higher metal layers and 2) to maximize the routability during detailed routing step by reserving upper routing resources. Therefore, SP&R simultaneously optimizes these multiple objectives in the light of "lexicographic" order with an optimization feature of OMT [23], [28]. The objectives can be ranked in the order of emphasized, as described in (4)

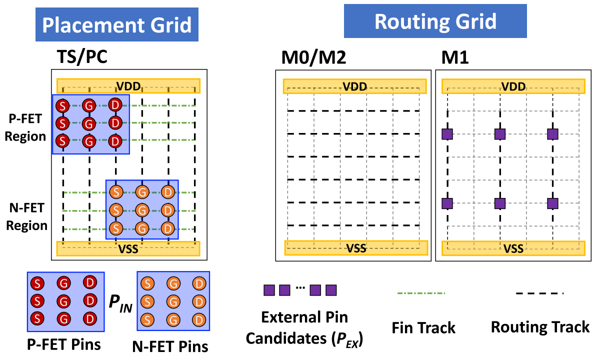


Fig. 3. Grid-based placement and 3-D routing graph.

$$\textit{Placement (Cell Size):}\ \max\left\{x_t + w_t \ \middle|\ t \in T\right\} \tag{1}$$

$$\textit{Routability (\#M2 Track):}\ \sum_{k=1}^{l} \bigvee_{e_{v,u} \in E_k} m_{v,u} \tag{2}$$

l=#Horizontal Tracks

E_k=Set of $M2$ Layer Edges in kth Track

$$\textit{Routing (Total ML):}\ \sum_{e_{v,u} \in E} \left(w_{v,u} \times m_{v,u}\right) \tag{3}$$

$$\textit{LexMin:}\ \text{(a) Cell Size, (b) \#}M2\text{ Track, (c) Total ML.} \tag{4}$$

D. Cell Architecture

In this ork, our framework considers 7-nm standard cell architecture (e.g., the layer/track information) of [19] and [26] as depicted in Fig. 3. Inspired by Kang *et al.* [29] and Park *et al.* [30], [31], we adopt supernodes to cover the multiple candidates for each pin, either the pin of FET (i.e., internal pin) or the I/O pin of a standard cell (i.e., external pin).

Layer Configuration: We define the grid-based placement and 3-D routing graph composed of four metal layers (i.e., TS/PC, $M0$, $M1$, and $M2$) as shown in Fig. 3. In practice, routing layers' multiple interchanges on timing critical paths are undesirable due to the severe performance loss caused by the high resistance of VIA elements. Therefore, we determine the weighted cost $w_{v,u}$ of VIA metal segments by four times higher than that of horizontal and vertical metal segments. In placement grid (i.e., TS/PC), there are three placement tracks (i.e., fin tracks) for an allocation of FETs in the corresponding P-FET/N-FET region. Due to the limited placement tracks, we only consider the single-stack placement of FETs in each region. The routing grid (i.e., $M0/1/2$) consists of six horizontal tracks.

On-Grid and Unidirectional Routing: We assume on-grid and unidirectional routing scheme for each layer due to the process resolution of sub-7-nm multipatterning technologies and IC practitioners' restriction of preferred routing direction per each layer [32]. The preferred directions of $M0/2$ and $M1$ layers are horizontal and vertical, respectively, as illustrated in Fig. 3. Fig. 4 shows a multilayered 3-D routing graph $G = (V, E)$ to represent the available routing resources

[1] The symbol d is L (Left), R (Right), F (Front), B (Back), U (Up), D (Down), or a combination of these directions, e.g., FL means FrontLeft.

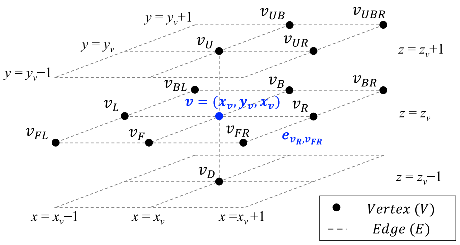


Fig. 4. Example of 3-D grid-based routing graph, $G(V, E)$.

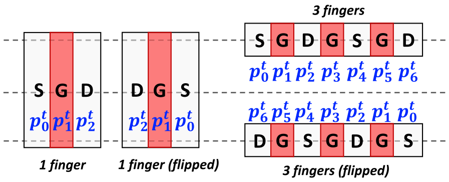


Fig. 5. Configuration of a FET with size of 3.

(e.g., horizontal and vertical tracks on each layer, interlayer VIAs) and routing paths between sources and sinks. Each vertex v is mapped to coordinates (x_v, y_v, z_v), where x, y, and z are induced from horizontal routing tracks, vertical routing tracks, and metal layers, respectively. Each edge $e_{v,u}$ between vertices v and u represents a flow with capacity of one, including interlayer VIAs. Source-sink network-flow connectivity is represented on the graph G.

Internal Pin (P_{IN}) for FET: P_{IN} refers to the source, drain, and gate of each FET and is defined in placement graph as depicted in Fig. 3. The location of each pin is dynamically determined by placement formulation (Section III-A) and is associated with the flow formulation for routing through our DPA scheme (Section III-C).

External Pin (P_{EX}) for I/O Pin Access: P_{EX} represents I/O pins of a standard cell. Vertices (depicted in purple squares in Fig. 3) interconnected to P_{EX} on $M1$ layer are defined as candidates for each I/O pin's access point. One of these candidates is assigned as an I/O pin access point by our flow formulation (Section III-B1). The routed metal segments on $M1$ and $M2$ layers, including the assigned vertices represent the I/O pin of a standard cell for the detailed routing phase.

III. Simultaneous Placement and Routing

In this section, we describe our SMT formulation of the constraints for the proposed SP&R framework. This section consists of: 1) transistor placement; 2) in-cell routing; and 3) DPA.

A. Transistor Placement

FET Configuration: Fig. 5 illustrates an example of variable types of an FET with size of 3. There are four possible FET types, such as "1 finger," "1 finger (flipped)," "3 fingers," and "3 fingers (flipped)." Since we only consider a single-stack placement in sub-7-nm technology nodes, SP&R selects FET types having the minimum number of fingers [i.e., 1 finger and 1 finger (flipped)] to minimize the cell size. SP&R defines the pin information based on the selected FET type (i.e., p_0^t, p_1^t, and p_2^t as shown in 1 finger cartoons of Fig. 5).

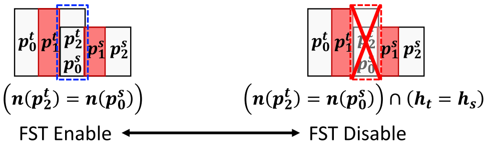

p_0^t p_1^t p_2^t p_0^s p_1^s p_2^s
$\left(n(p_2^t) = n(p_0^s)\right)$
FST Enable
p_0^t p_1^t p_2^t p_0^s p_1^s p_2^s
$\left(n(p_2^t) = n(p_0^s)\right) \cap (h_t = h_s)$
FST Disable

Fig. 6. DS with the FST option.

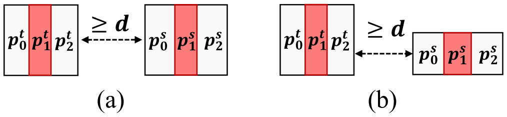


Fig. 7. DB. (a) Different net information. (b) Different diffusion heights with FST disable.

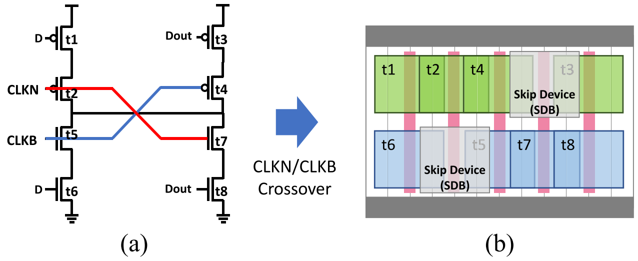


Fig. 8. Example of SDB with a crossover (a) Schematic with crossover signals (CLKN/CLKB). (b) Use of SDBs in a crossover region.

Diffusion Sharing (DS): DS is a common placement technique when the net information and the diffusion height (numbers of fins in P-FETs and N-FETs) are the same between pins of individual FETs. However, inevitably, standard cells must have different diffusion heights to enable flexible power-performance exploration. Also, the disparity in diffusion height brings harmful side effects, such as yield loss or neighbor diffusion effect [33] because of the distortion during diffusion process for adjacent FETs. In a conventional physical design flow, these size transition within a standard cell is captured by the library characterization as the diffusion shapes are predetermined. SP&R provides an optional FET size transition (FST) in DS between FETs with different diffusion heights, supporting the library characterization in various process architectures. Fig. 6 depicts the DS rule according to the FST option. When the FST is disabled, DS is not allowed between FETs with different diffusion heights.

DB: As shown in Fig. 7, DB refers to the minimum space d between distinct diffusion regions when they are not shared due to the different net information or different diffusion heights. SP&R supports single DB (SDB) and double DB (DDB). The minimum space d of SDB and DDB is 2 and 4, respectively.

SDB With a Crossover: A crossover of signals in a standard cell causes "skip device" (i.e., whitespace without FETs) due to the mismatches of gate signal connections. Fig. 8(b) shows the example of a cell placement when skip device occurred by the crossover of CLKN and CLKB signals as shown in Fig. 8(a). When DDB is a major DB, these skip devices significantly increase cell size. In practice, to minimize the cell-area loss, SDB is used in a specific crossover region. SP&R provides the use of SDB for the FETs that are specified as being in a crossover region when the major DB is DDB.

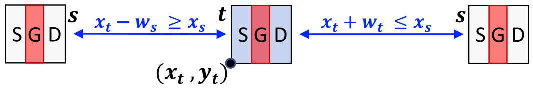


Fig. 9. Relative positions between two FETs.

Algorithm 1 SetRPCwithSDB (FETs t, s)

```
Input: t, s: a pair of FETs, d_s: distance of a single diffusion break
/* FST: 0-1 indicator if FET size transition is enabled */

// Set SMT Constraint
 1: if $x_t > x_s + w_s$ then
 2:     $x_t \geq x_s + w_s + d_s$;
 3: else if $x_t = x_s + w_s$ & $n(p_l^t) = n(p_r^s)$ & (FST | (!FST & $h_t = h_s$)) then
 4:     $x_t = x_s + w_s$;
 5: else if $x_t + w_t < x_s$ then
 6:     $x_t + w_t + d_s \leq x_s$;
 7: else if $x_t + w_t = x_s$ & $n(p_r^t) = n(p_l^s)$ & (FST | (!FST & $h_t = h_s$)) then
 8:     $x_t + w_t = x_s$;
 9: else
10:     Unsatisfiable Condition;
11: end if
```

Algorithm 2 SetRPCwithDDB (FETs t, s)

```
Input: t, s: a pair of FETs, d_d: distance of a double diffusion break
/* leftmost (resp. rightmost) pin of t and s: $p_l^t$ and $p_l^s$ (resp. $p_r^t$ and $p_r^s$) */
/* FST: 0-1 indicator if FET size transition is enabled */
/* $o_l^t$ (or $o_r^t$): 0-1 indicator if FET t shares diffusion on the left (or right) side */

// Set SMT Constraint
 1: if $x_t > x_s + w_s$ & $o_l^t$ then
 2:     $x_t \geq x_s + w_s + 2$;
 3: else if $x_t > x_s + w_s$ & $!o_l^t$ then
 4:     $x_t \geq x_s + w_s + d_d$;
 5: else if $x_t = x_s + w_s$ & $n(p_l^t) = n(p_r^s)$ & (FST | (!FST & $h_t = h_s$)) then
 6:     $x_t = x_s + w_s$;
 7: else if $x_t + w_t < x_s$ & $o_r^t$ then
 8:     $x_t + w_t + 2 \leq x_s$;
 9: else if $x_t + w_t < x_s$ & $!o_r^t$ then
10:     $x_t + w_t + d_d \leq x_s$;
11: else if $x_t + w_t = x_s$ & $n(p_r^t) = n(p_l^s)$ & (FST | (!FST & $h_t = h_s$)) then
12:     $x_t + w_t = x_s$;
13: else
14:     Unsatisfiable Condition;
15: end if
```

Algorithm 3 SetRPCwithMixedDB (FETs t, s)

```
Input: t, s: a pair of FETs, d_s (or d_d): distance of a single (or double) diffusion break
/* leftmost (resp. rightmost) pin of t and s: $p_l^t$ and $p_l^s$ (resp. $p_r^t$ and $p_r^s$) */
/* FST: 0-1 indicator if FET size transition is enabled */
/* $o_l^t$ (or $o_r^t$): 0-1 indicator if FET t shares diffusion on the left (or right) side */

// Set SMT Constraint
 1: if $x_t > x_s + w_s$ & $o_l^t$ then
 2:     $x_t \geq x_s + w_s + 2$;
 3: else if $x_t > x_s + w_s$ & $!o_l^t$ then
 4:     if $n(p_l^t)$ equals $n(p_r^s)$ then
 5:         $x_t \geq x_s + w_s + d_s$;
 6:     else
 7:         $x_t \geq x_s + w_s + d_d$;
 8:     end if
 9: else if $x_t = x_s + w_s$ & $n(p_l^t) = n(p_r^s)$ & (FST | (!FST & $h_t = h_s$)) then
10:     $x_t = x_s + w_s$;
11: else if $x_t + w_t < x_s$ & $o_r^t$ then
12:     $x_t + w_t + 2 \leq x_s$;
13: else if $x_t + w_t < x_s$ & $!o_r^t$ then
14:     if $n(p_r^t)$ equals $n(p_l^s)$ then
15:         $x_t + w_t + d_s \leq x_s$;
16:     else
17:         $x_t + w_t + d_d \leq x_s$;
18:     end if
19: else if $x_t + w_t = x_s$ & $n(p_r^t) = n(p_l^s)$ & (FST | (!FST & $h_t = h_s$)) then
20:     $x_t + w_t = x_s$;
21: else
22:     Unsatisfiable Condition;
23: end if
```

Relative Positioning Constraint (RPC): We utilize the conventional floorplanning design approach (i.e., RPC) for the transistor placement problem [34]. All transistor positions can be represented by two RPCs as shown in Fig. 9 because we only consider a single-stack placement. According to the input parameters that determine the type of DB, our SP&R calls subprocedures defined in Algorithms 1–3 to set corresponding RPC formulation with DS and DB. Algorithm 1 sets the RPC for SDB when SDB is a major DB. Each RPC on the left-/right-hand side is separated into two cases with/without DS. This geometric SMT constraint ensures that only one case is enabled at once and determines the position and the flip status of FETs. When DDB is a major DB, there exists a case that the other FETs are placed between two FETs of interest and share a diffusion region with one of the two FETs. This prohibits the consecutive DS occurrence of FETs because the RPC only considers the relative position between two FETs. To prevent this case, the RPC for DDB refers the DS indicators o_l^t and o_r^t as shown in Algorithm 2. When FET t is on the right-hand side of FET s, also when FET t is sharing a diffusion on the left-hand side (lines 1 and 2), the distance between FETs t and s is set to the minimum value 2. So the RPC does not restrict DS of FET s and the FET that is placed between FETs t and s. If there are no FETs between FETs t and s, the RPC sets the distance using d_d (lines 3 and 4). Algorithm 3 represents the RPC with the mixed SDB in a crossover. Since SP&R minimizes the cell size, all the pairs of FETs that have the same net information on their facing nodes must be placed with DS. However, the gate signals' mismatch in a crossover prohibits DS, resulting in the skip device. Therefore, we can detect the skip device by finding FET pairs that are not sharing diffusion regions (lines 3 and 13) even though they meet the sharing conditions (lines 4 and 14).

B. In-Cell Routing

We adopt conditional design rule-aware multicommodity network flow theory to formulate the in-cell routing problem as described in [25] and [29]–[31]. Specifically, the refined constraints for commodity flow conservation (CFC) and vertex exclusiveness (VE) in unidirectional edges [25], [31] are implemented in our framework to reduce the search space of the routing formulation. The routing formulation consists of two parts, *flow formulation* and *conditional design rules*, as shown in Fig. 10. The flow formulation secures the routing path between the source and the sink for each commodity without heuristic modeling. The conditional design rules work as constraints to route through design-rule violation-free paths. The built-in functions, such as at-most k (AMk) and at-least k (ALk) are used to formulate cardinality constraints.

1) Flow Formulation: SP&R implements flow formulations, such as *Commodity Flow Conservation, Vertex Exclusiveness, Edge Assignment,* and *Metal Segment*, by utilizing the same methodology of [30] and [31].

2) Conditional Design Rule: Previous works [29], [30], [31] mainly tackle three representative conditional design rules, e.g., minimum area (MAR), end-of-line spacing (EOL), and via rule (VR). In SP&R, MAR and EOL follow the

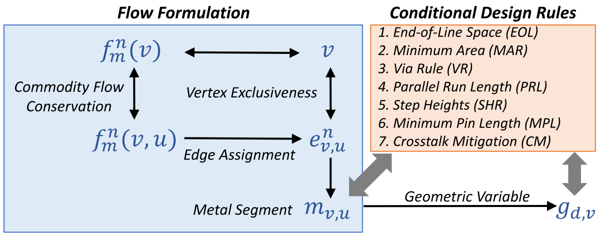


Fig. 10. Flow formulation with conditional design rules.

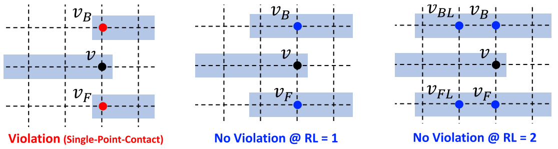


Fig. 11. Example of PRL rule.

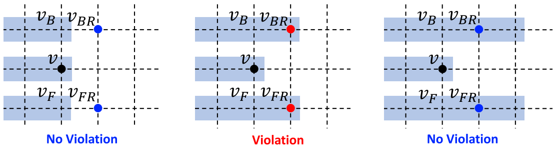


Fig. 12. Example of SHR.

same principle of [30] and [31]. Compared to [30] and [31], we adopt stack VR (stack-able) for VR. Furthermore, SP&R includes multipattern-aware design rules, such as parallel run length (PRL) and step heights rule (SHR) [35], [36]. PRL and SHR have essential roles for handling the complex line-end overlap rules of SADP/SAQP processes in advanced technology nodes. To ensure the pin accessibility, we consider minimum I/O pin length (MPL). CM contributes to maintain a stable operation of timing-critical sequential logic cells.

PRL: PRL rule is a design rule to avoid "single-point-contact" in manufacturing SADP mask [36]. Fig. 11 and constraint (5) represent an example of PRL rule and the corresponding formulation when the run length (RL) is 2

$$\mathbf{AM1}(g_{R,v}, g_{L,v_B}, g_{L,v_{BL}});\ \mathbf{AM1}(g_{R,v}, g_{L,v_F}, g_{L,v_{FL}})$$
$$\forall v \in V_0, V_2. \quad (5)$$

SHR: SHR is a design rule to avoid "the small step" in manufacturing SADP mask [36]. Fig. 12 and constraint (6) describe an example of SHR and the corresponding formulation when the step height is 2

$$\mathbf{AM1}(g_{R,v}, g_{R,v_{BR}});\ \mathbf{AM1}(g_{R,v}, g_{R,v_{FR}}) \quad \forall v \in V_0, V_2. \quad (6)$$

Minimum Pin Length (MPL): MPL rule ensures the minimum number of metal segments of the commodity heading to the external pin P_{EX} on $M1$ layer. At-least 1 metal segment on $M1$ layer must be assigned to the commodity whose sink is P_{EX} as expressed in constraint (7). Then, the metal segment on $M1$ layer is extended to have the minimum length defined by MAR as depicted in Fig. 13

$$\mathbf{AL1}(m_{v,v_F}, m_{v,v_B}), \quad \text{if } f_m^n(v, v_D) = 1\ ,\ f_m^n(v, v_U) = 1$$
$$\forall v \in V_1, d_m^n = P_{EX}. \quad (7)$$

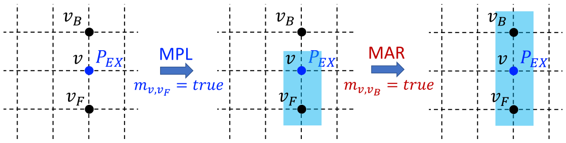


Fig. 13. Example of MPL.

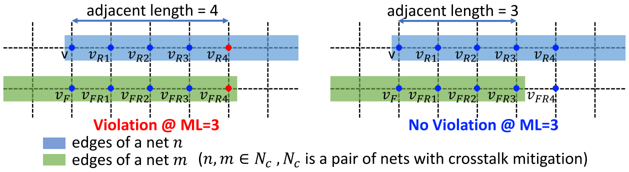


Fig. 14. Example of CM rule.

CM: The crosstalk between differential clock signals in the sequential logic cells, such as latches and flipflops, may cause severe timing violation, thus failure of timing closure due to the cross-coupling capacitance. When the switching windows of the clock and the inverted clock overlap and switch in opposite directions, the crosstalk will increase the delay of the clock nets, which may result in setup violations. More specifically, the strength of crosstalk is a function of the geometrical adjacent length (parallel running length) between adjacent nets [37]. Therefore, to mitigate the crosstalk effects for timing-critical cells, SP&R provides an optional design rule constraint to restrict the maximum adjacent length (ML) of a selected pair of nets. Fig. 14 and (8) represent the CM constraint between nets n and m that are in a pair of nets with CM N_c when ML = 3

$$\mathbf{AL1}\big((\neg e^n_{v,v_{R1}} \ \vee\ \neg e^m_{v_F,v_{FR1}}),\ (\neg e^n_{v_{R1},v_{R2}} \ \vee\ \neg e^m_{v_{FR1},v_{FR2}}),$$
$$(\neg e^n_{v_{R2},v_{R3}} \ \vee\ \neg e^m_{v_{FR2},v_{FR3}}),\ (\neg e^n_{v_{R3},v_{R4}} \ \vee\ \neg e^m_{v_{FR3},v_{FR4}})\big)$$
$$\forall n, m \in N_c. \quad (8)$$

C. Dynamic Pin Allocation

We devise a dynamical pin allocation (DPA) scheme between placement and routing grids. In the *TS/PC* layer, the placement tracks are not exactly aligned with the routing tracks. Therefore, we have to map the pins of each FET on the placement grid to the routing pins on the routing grid to utilize the grid-based routing formulation as shown in Fig. 15.

From Placement (Pin Allocation): Every pin in each FET has its own flow capacity variable $C_m^n(p, r)$ on their corresponding vertices of *TS/PC* routing grid as shown in Fig. 15(a). When locations of FETs are determined by the placement formulation, the flow capacity variables of each pin are conditionally assigned to the corresponding location of each pin. Algorithm 4 presents the flow capacity control constraint. For certain net n and commodity m, $C_m^n(p, r)$ is set as 0 if vertex r is not in the range of p. Fig. 15(b) shows the flow capacity variables assigned to 0 outside the corresponding column of a source pin on P-FET (depicted in red dashed box).

To Routing (Flow Capacity Connection): The flow variable $f_m^n(v, u)$ [routing formulation] is associated with the flow

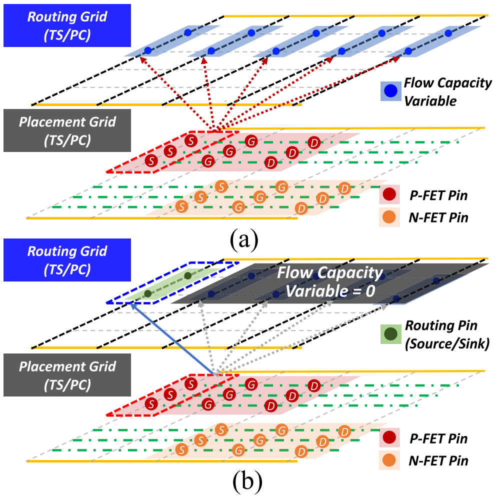


Fig. 15. DPA between placement and routing grids. (a) From placement (pin allocation). (b) To routing (flow capacity connection).

Algorithm 4 Flow Capacity Control Constraint ($C_m^n(p, r)$)

```
/* x coordinate (resp. y coordinate) of a routing grid r: $x_r$ (resp. $y_r$) */
/* Height and x coordinate (resp. y coordinate) of a pin p: $h_p$ and $x_p$ (resp. $y_p$) */
/* Single column pin only: Set of x is singleton */
/* p is either source or sink of a net n and commodity m */

// Set SMT Constraint
if $(x_r \neq x_p) \mid (y_r < y_p) \mid (y_r > y_p + h_p)$ then
    $C_m^n(p, r) = 0$;
else
    $C_m^n(p, r)$ is Determined by Routing Formulation;
end if
```

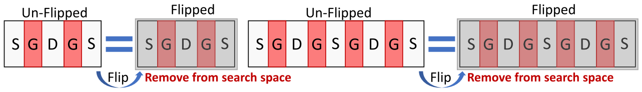


Fig. 16. Flipped case exclusion of even-numbered finger FETs.

Algorithm 5 Exclusion of Whole Cell Design Flipping Cases

```
Input: a set of FETs $T$, a set of P-FETs $T_p$, # of P-FETs $N_p$
Data: a set of FET groups $T_{\text{comb}}$, $k$th FET group in $T_{\text{comb}}$ $T_k^{\text{comb}}$
if $N_p$ is even then
    $T_{\text{comb}} \leftarrow GetCombination(T_p, N_p/2)$;
    while $T_k^{\text{comb}} \neq \emptyset$ do
        Set SMT Constraint $x_m < x_n$, $\forall m \in T_k^{\text{comb}}$, $\forall n \in \{T_p - T_k^{\text{comb}}\}$;
        $k \leftarrow k + 1$;
    end while
else
    for $i = 1$ to $N_p$ do
        $T_{\text{comb}} \leftarrow GetCombination(\{T_p - T_i^p\}, (N_p - 1)/2)$;
        while $T_k^{\text{comb}} \neq \emptyset$ do
            Set SMT Constraint $x_m < x_n$, $\forall m \in T_k^{\text{comb}}$, $n = T_i^p$;
            Set SMT Constraint $x_m < x_n$, $m = T_i^p$, $\forall n \in \{T_p - T_k^{\text{comb}}\}$;
            $k \leftarrow k + 1$;
        end while
    end for
end if
procedure $GetCombination(T, N)$
    Add All Possible N-FET Combinations from $T$ into $T_{tmp}$;
    while $T_k^{tmp} \neq \emptyset$ do
        if $T_0 \in T_k^{tmp}$ then
            $T_{\text{comb}} \leftarrow T_{\text{comb}} \cup T_k^{tmp}$;
        end if
        $k \leftarrow k + 1$;
    end while
    return $T_{\text{comb}}$;
end procedure
```

capacity variable $C_m^n(p, r)$ by the constraint described in (9). Each $f_m^n(v, u)$ is determined by the routing formulation when vertex v is the internal pin p, and the adjacent vertex u is the adjacent vertex r of p in TS/PC (i.e., V_0). Thus, routing formulation can recognize the feasible sets of r in V_0 layer as routing pins [depicted in blue dashed box of Fig. 15(b)]

$$f_m^n(v = p, u = r) \leq C_m^n(p, r) \quad \forall r \in a(p)\ \forall r \in V_0. \tag{9}$$

IV. Scalability Improvement

In this section, we propose search-space reduction methods to improve the scalability of the proposed SP&R framework. This section consists of: 1) breaking design symmetry; 2) conditional assignment; 3) localization of the routing region; 4) cell partitioning; and 5) objective partitioning.

A. Breaking Design Symmetry

The proposed SP&R reduces the search space by eliminating symmetries existing in standard cell layout design [38], [39].

1) Flipping of Even-Numbered Multifinger FETs: Since FETs with even-numbered fingers have the same source/drain node on the leftmost/rightmost nodes, flipped FETs are the same with un-flipped FETs as shown in Fig. 16. Therefore, for every FET t with even-numbered fingers, ff_t is set to 0 to remove the flipped FETs from the search space.

2) Flipping of Whole Cell Design: In SP&R, every generated layout solution has a pair of dual solutions that are equivalent to their horizontal-flipped shapes as shown in Fig. 17. The pair of dual solutions have the identical key metrics. Therefore, excluding the exploration of the dual solutions effectively cuts the search space in half. Furthermore, since SP&R combines the placement of P-FETs and N-FETs which are mutually dependent of each other, the dual solutions can be removed from the search space by simply setting the relative positions of P-FETs in the way of preventing the opposite order. Algorithm 5 presents the exclusion of whole cell design flipping cases. The function $GetCombination(T, N)$ returns a set of N FET combinations that always includes the first FET element in a set of FETs T. For example, with $T = \{t_1, t_2, t_3, t_4\}$ and $N = 2$, $GetCombination(T, N)$ returns a set $T_{\text{comb}} = \{\{t_1, t_2\}, \{t_1, t_3\}, \{t_1, t_4\}\}$. When N_p is even, T_{comb} indicates a set of FET groups that should be placed on the left-hand side in an original solution. Therefore, by setting the relative position of FETs in kth FET group T_k^{comb} to be placed on the left side of the other FETs that are not in T_k^{comb}, the dual solution of each original solution is removed from the search space. When N_p is odd, while selecting each of FETs in T_p as a center position (T_{center}), the relative constraints are set in the order of $T_k^{\text{comb}} < T_{\text{center}} < \{T_p - T_{\text{center}} - T_k^{\text{comb}}\}$.

B. Conditional Assignment

The conditional assignment dynamically cuts the search space by assigning *true*/*false* to the variables according to the intermediate conditions satisfied during the problem solving. Some routing variables depend on the assignments of other variables as shown in Fig. 18. When a source (s^n) and a

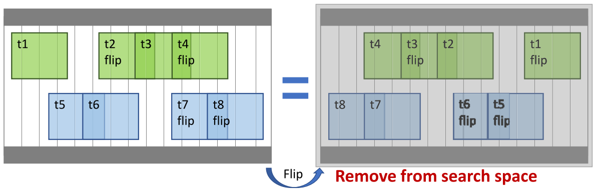
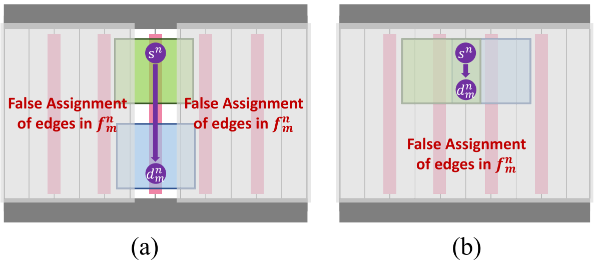
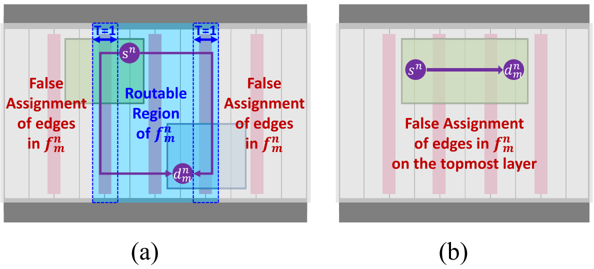


Fig. 17. Flipped case exclusion of whole cell design from search space.


s^n
d^n_m
False Assignment of edges in f^n_m
False Assignment of edges in f^n_m
s^n
d^n_m
False Assignment of edges in f^n_m

(a) (b)

Fig. 18. Conditional assignment. (a) Commodity flow through the same gate column. (b) Commodity flow through the DS.


T=1
T=1
s^n
d^n_m
False Assignment of edges in f^n_m
Routable Region of f^n_m
False Assignment of edges in f^n_m
s^n
d^n_m
False Assignment of edges in f^n_m on the topmost layer

(a) (b)

Fig. 19. Conditional localization. (a) Localization of commodity flows with a tolerance $T = 1$. (b) Localization of a commodity flow within the same FET.

sink (d^n_m) nodes of mth commodity in net n on the gate of each P-FET and N-FET are connected through PC (i.e., the x-coordinates of the source and sink are the same), the other edge variables in f^n_m outside this column will be set as 0 by the flow formulation [Fig. 18(a)]. Fig. 18(b) shows a commodity flow through TS at the same column by DS. Since the source and sink can be enabled on the same vertex, all edge variables in f^n_m are conditionally set to 0.

C. Localization of the Routing Region

The range of potential routing region for each commodity covers the entire bounding box of the cell because the location of each source/sink node is dynamically determined in SP&R. Therefore, a proper localization of routing regions reduces the complexity of SP&R.

1) Conditional Localization: Fig. 19(a) shows the example of the conditional localization. When intermediate locations of source s^n and sink d^n_m of commodity f^n_m are determined, we restrict the path connecting s^n and d^n_m in the minimum bounding box that covers both s^n and d^n_m (depicted in blue rectangle). The offset with tolerance T gives a margin to prevent from overcutting.

2) Localization of Intra-FET Routing: Achieving the minimum wire length without using the topmost layer $M2$ is highly preferred for connecting nodes within the same FET. Therefore, the edge variables on the topmost layer of the commodities whose source/sink nodes are in the same FET are set to *false* as shown in Fig. 19(b).

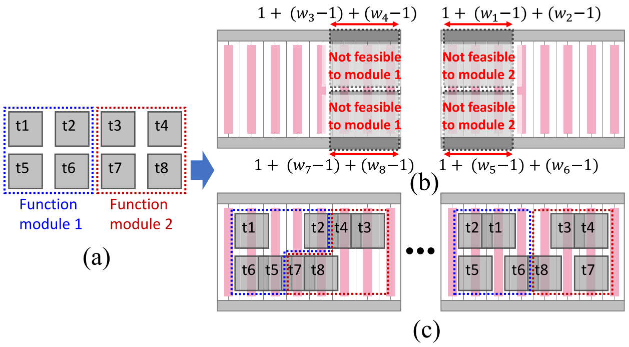


Fig. 20. Cell partitioning. (a) Functional module partitioning. (b) Localization of the placement area. (c) Examples of SP&R with cell partitioning.

D. Cell Partitioning

Designing sequential logic cells requires special attention to timing-critical paths. Functional modules are strictly ordered by the sequential datapath to optimize the cells' PPAC, e.g., flipflop's functional modules, i.e., Clk, Din, Dout, Master/Slave latches should follow the order of Din-Master-Slave-Dout (or Dout-Slave-Master-Din) to optimize the setup time (i.e., t_{setup}) and the delay of the flipflop (i.e., $t_{clk-to-q}$). Also, a datapath-aware placement of functional modules reduces the probability of path-level timing violations due to the twisted routing paths. Clk module is usually placed inside the cell to prevent noises from adjacent cells. To fulfill this timing-design requirement, SP&R performs a functional module-based cell partitioning as shown in Fig. 20. With the predefined FET groups by the functionality [Fig. 20(a)], SP&R honors the order of FETs among functional-module groups [Fig. 20(c)]. The freedom of the FET placement in each group and the DS between groups is not restricted by the DPA. Besides, ordering FET groups significantly reduces the search space by setting the relative position between FETs as well as the upper bound and lower bound of each FET according to the order of groups as shown in Fig. 20(b). The minimum-achievable track occupation of each FET group is calculated by assuming that all FETs in each group share their diffusions side by side.

E. Objective Function Partitioning

SP&R co-optimizes multiple objectives at once by using the lexicographic method [23], [28]. The lexicographic method consists of solving a sequence of single-objective optimization problems under the constraining condition that optimizes higher priority objectives. This results in gradual reductions of the search space by virtue of the implicitly added constraints. Therefore, partitioning an objective function with a proper priority helps to improve the scalability. The total ML objective [described in (3)] is defined as the weighted sum of metal segments (i.e., CA,[2] VIA, and metal). The weight of the VIA is set higher than the metal to minimize the use of upper-layer metals as well as the use of more resistive VIA elements. Thus, this weight can be used to separate and optimize the total ML objective with the priority as shown in (10). Among the three layers, we assign the higher priority for the lower layers to prevent redundant routing detours, causing the increment on the total ML because the lower layers have more elements than the upper layers in our framework[3]

$$\textit{LexMin: } \text{(a) } \#CA,\ \text{(b) } \#VIA01,\ \text{(c) } \#VIA12,\ \text{(d) } \#M0,\ \text{(e) } \#M1,\ \text{(f) } \#M2. \tag{10}$$

V. Experiments

We have implemented the proposed SP&R framework in *Perl/SMT-LIB* 2.0 standard-based formula and validated on a Linux desktop with 3.6 GHz AMD Ryzen5 3600 CPU and 32-GB memory. The SMT Solver *Z3* (ver. 4.8.5) [27] is used to produce the optimized solution through the proposed SP&R formulation. SP&R generates the "design layout" file with the information of FETs, nets (i.e., target nets for in-cell routing), and I/O pins (i.e., P_{EX}) from netlist of standard cell libraries. We choose 37 out of 69 cells in NanGate's 15-nm open cell library [40] and 85 out of 183 cells in the ASAP7 library [26] to show the improvement of the scalability.[4] Then, we generate a whole set of ASAP7 library and compare the results to the previous works [19], [26]. The 15-nm library is converted to the 7-nm cell-equivalent architecture (six horizontal routing tracks and three fins) of ASAP7 library for having the same number of routing tracks and fins. We tightly specify design parameters (MAR/EOL/VR/SHR/PRL/FST/T = 2/2/1.5/2/1/disable/1 or 2) for NanGate library to demonstrate innovative features of SP&R while the parameters of the ASAP7 library are specified to have the most similar routing result with the original library (MAR/EOL/VR/SHR/PRL/FST/T = 1/1/1.5/1/1/enable/1). The major DB of NanGate and ASAP7 libraries are SDB and DDB, respectively.

A. Sequential Versus Simultaneous P&R

We compare our simultaneous P&R approach to the conventional sequential approach [19]. To simulate the sequential P&R of [19], we first find the optimal placement solution satisfying all of the design-rule constraints with the minimum cell size and the total node-to-node distances between pins. Then, we find the optimal routing solution under the same multiple objectives described in Section II-C for fair comparison.

TABLE II
Experimental Results Presenting the Comparison of Key Metrics Between Sequential-P&R and Simultaneous-P&R. Impr. = Improvement(in %) Ratio (Reference = Sequential P&R), ML = Total Metal Length, M_2 = the Number of Used $M2$ Tracks, and R = Total Resistance(in Ω) of Metals/Vias

Library	Cell	Sequential P&R				Simultaneous P&R				Impr.	
		Size	ML	M_2	R	Size	ML	M_2	R	M_2	R
NanGate	AOI21_X2	11	222	1	875.2	11	191	0	757.2	100%	13%
	OAI21_X2	11	217	1	856.8	11	191	0	757.2	100%	12%
	DFFSNQ_X1	23	662	4	2566.0	23	612	3	2370.4	25%	8%
	HA_X1	10	241	2	936.8	10	230	1	892.4	50%	5%
	SDFFSNQ_X1	28	751	5	2905.2	28	702	3	2720.8	40%	6%
ASAP7	AND3x4	14	221	0	870.4	14	192	0	757.6	-	13%
	AOI222xp33	10	207	1	812.4	10	138	0	539.2	100%	34%
	HAxp5	9	166	1	651.2	9	124	0	482.8	100%	26%
	OAI222xp33	10	162	0	632.8	10	138	0	539.2	-	15%
	XNOR2x2	11	196	0	768.4	11	176	0	685.2	-	11%
	XOR2x2	11	214	0	840.4	11	176	0	685.2	-	18%
	SDFLx1	25	603	3	2356.0	25	633	2	2469.2	33%	-5%
	SDFLx2	26	609	3	2380.4	26	638	2	2490.0	33%	-5%
	SDFLx3	27	629	3	2460.8	27	658	2	2570.4	33%	-4%
	SDFLx4	28	635	3	2485.2	28	664	2	2594.8	33%	-4%
Average		13.45	296.27	1.36	1155.96	13.45	260.91	0.64	1017.02	53%	12%

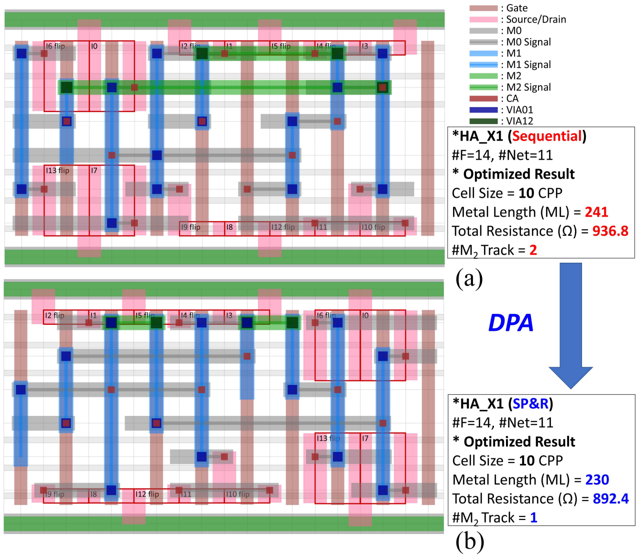


Fig. 21. Comparison between (a) sequential (#$M2$ = 2, ML = 241, Total Resistance = 936.8 Ω) and (b) simultaneous P&R (the proposed SP&R) (#$M2$ = 1, ML = 230, Total Resistance = 892.4 Ω) using HA_X1.

Table II presents the comparison of key metrics for 15 out of 122 cells that have more than 10% of improvement in #$M2$ tracks or total parasitic resistance[5] between sequential and simultaneous P&R approaches. SP&R, respectively, shows 53% and 12% of improvements on average in #$M2$ tracks and the total resistance. The ML improvement is directly related to the total parasitic resistance reduction of wire/via, resulting in each cell's better PPAC achievements. Besides, the smaller number of $M2$ tracks substantially improves the routability in detailed routing phase. Though SDFLx1–x4 show 4%–5% increment of resistance, #$M2$ tracks have been reduced 33% under the priority of our key objectives.

Fig. 21 shows the difference in terms of the key metrics between SP&R and the sequential approach for a HA_X1 cell. With the same cell size of ten CPP, SP&R [Fig. 21(b)], respectively, reduces the #$M2$ tracks and the total resistance by 50% (2 → 1) and 5% (936.8 → 892.4) by virtue of the DPA scheme [Fig. 21(a)].

[2] CA refers to the VIA element which connects gates and source/drain contacts with $M0$.

[3] In our experimental data, the lower layer, respectively, shows 94%, 126% more *VIA* and metal elements compared to the upper layer on average due to the 1:1 gear ratio (poly pitch/metal pitch) of our framework.

[4] In this experiment, we select representative, typical types of standard cells carrying various structures of combinational and sequential logic cells by reflecting field engineers' opinions.

[5] The total parasitic resistance of wires is calculated using sheet resistance information of ASAP7 library published in [41].

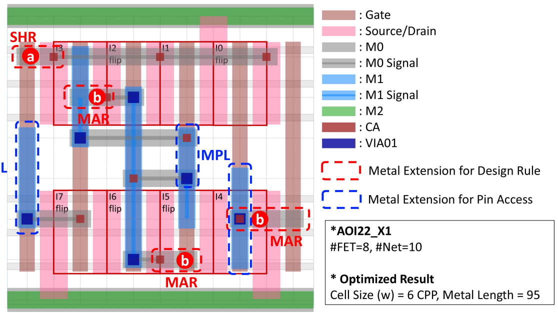


Fig. 22. Optimized cell layout for DFM and I/O pin accessibility.

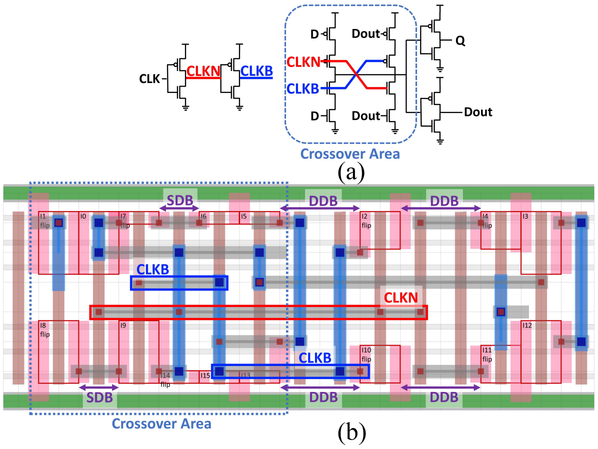


Fig. 23. Example of SDBs in a crossover area. (a) Schematic of DHLx1 and (b) DHLx1 layout generation.

B. Optimization for DFM and I/O Pin Accessibility

Fig. 22 shows an example of a generated AOI22_X1 cell layout satisfying all prediscussed design-rule constraints.[6] The metal segments (a) and (b) (red dashed region) are extended to satisfy SHR and MAR, respectively. The blue dashed region shows that the metal segment is extended to satisfy MPL design rule for I/O pin accessibility.

C. Single Diffusion Break in Crossover

Fig. 23(a) shows a schematic of DHLx1 with a crossover of signals *CLKN* and *CLKB* (depicted in blue dashed region). With the additional input of the specified crossover area, SP&R selects SDBs for the skip device regions formed by the signal crossover instead of the major DDB in the crossover area [depicted in blue dashed rectangle in Fig. 23(b)].

D. Crosstalk Mitigation

Fig. 24(a) shows a layout of a DFFHQNx1 from ASAP7 library, displayed by a commercial tool [42]. The clock signals (*CLKN* and *CLKB*) with the opposite directions are routed in the adjacent *M*2 tracks with a PRL over seven CPPs. This may cause a substantial crosstalk between the clock signals that increases the delay and the power consumption. In constrast, the result of SP&R with a CM parameter ML = 4 between the clock signals successfully prevents the crosstalk by restricting the PRL of those signals less than two CPPs as shown in Fig. 24(b).

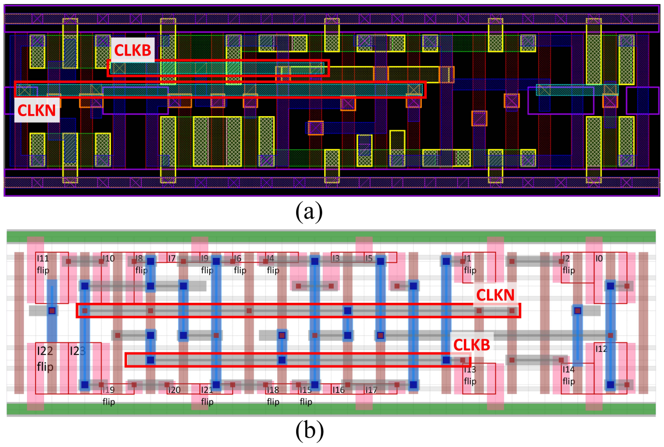


Fig. 24. Layout of DFFHQNx1. (a) Layout from the standard cell library [26], displayed by a commercial tool [42]. (b) SP&R's layout generation.

TABLE III
SCALABILITY IMPROVEMENT STAGES

Stage	Description
Phase I	The framework of [25]
Phase II	Phase I + pre-processing + breaking design symmetry + conditional assignment
Phase III	Phase II + localization of the routing region
Phase IV	Phase III + objective function partitioning

TABLE IV
EXPERIMENTAL RESULTS PRESENTING THE TRADEOFF BETWEEN SCALABILITY AND KEY METRICS IN SP&R: ALL VALUES ARE ON AVERAGE. $T1/T2/T3$ = TOLERANCE 1/2/3 OF THE LOCALIZATION, #VAR/#CON = THE NUMBER OF VARIABLES/CONSTRAINTS, INC./IMPR. = INCREMENT/IMPROVEMENT(IN MAGNIFICATION) RATIO (REFERENCE = PHASE I), AND M_2 = THE NUMBER OF USED $M2$ TRACKS

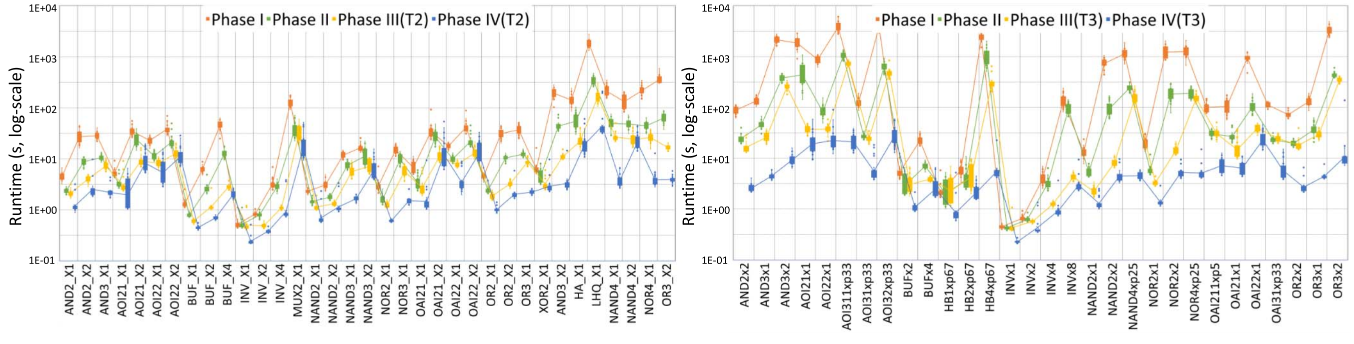
Library	Stage	SMT Formulation		Key Metrics							Runtime	
		#Var	#Con	Size	Metal Length				M_2			
					Total	inc.	VIA	Metal	Avg.	inc.	Avg.(s)	impr.
NanGate	**Phase I**	**14016.2**	**40000.7**	**6.91**	**117.89**	**0.00%**	**74.40**	**43.49**	**0.20**	**0%**	**75.14**	**1.0×**
	Phase II	13439.5	38077.6								18.18	4.1×
	Phase III ($T1$)	**13083.4**	**36856.7**		118.09	0.17%		43.69			7.03	10.7×
	Phase III ($T2$)				**117.89**	**0.00%**		**43.49**			**8.94**	**8.4×**
	Phase IV ($T1$)				118.11	0.19%		43.71			3.61	20.8×
	Phase IV ($T2$)				**117.94**	**0.05%**	**74.29**	**43.66**			**3.80**	**19.8×**
ASAP7	**Phase I**	**14127.3**	**39626.7**	**6.97**	**99.20**	**0.00%**	**66.67**	**32.53**	**0.00**	**0%**	**557.34**	**1.0×**
	Phase II	13493.7	37520.9								114.45	4.9×
	Phase III ($T1$)	**12890.6**	**35347.7**		99.27	0.07%		32.60			34.23	16.3×
	Phase III ($T3$)				**99.20**	**0.00%**		**32.53**			**70.93**	**7.9×**
	Phase IV ($T1$)				99.27	0.07%		32.60			4.23	131.7×
	Phase IV ($T3$)				**99.20**	**0.00%**		**32.53**			**4.42**	**126.1×**

E. Scalability Improvement

Table III represents the scalability improvement stages of SP&R framework. Phase I refers to the base framework of [25]. Phase II includes a simple preprocessing based on the Boolean constraint propagation (BCP), breaking design symmetry, and conditional assignment based on Phase I. Phases III and IV perform localization of the routing region and objective function partitioning based on Phases II and III, respectively.

1) Tradeoff Between Scalability and Key Metrics: Table IV presents the experimental results showing the tradeoff between scalability and key metrics of each improvement stage for 35 NandGate [40] and 30 ASAP7 [26] combinational logic cells that can be generated within an hour in Phase I with different

[6] In this research, SP&R considers several representative design rules. By the nature of on-grid routing, the authors firmly believe that all other conditional design rules can be properly formulated and integrated.



(a) (b)

Fig. 25. Contributions of each scalability improvement phase for runtime reduction with statistical runtime visualization (21 random seeds). (a) 35 combinational logic cells of NanGate library. (b) 30 combinational logic cells of ASAP7 library.

design rule sets. Phases III and IV have split cases according to tolerance T of the conditional localization.

For NanGate cells, the average runtime has been improved up to 20.8× [75.14 s@Phase I → 3.61 s@Phase IV ($T1$)]. Compared to Phase I (the simple preprocessing with BCP), the average numbers of variables (#Var)/constraints (#Con) have been reduced by 4.1%/4.8% and 6.7%/7.9% for Phases II and III, respectively. In Phase III, the conditional localization with tolerance $T = 1$ ($T1$) results in 0.17% increment of the total ML compared to Phase I while all key metrics are the same with tolerance $T = 2$ ($T2$). In Phase IV, the total MLs are, respectively, increased by 0.02% and 0.05% with $T1$ and $T2$ while the length of VIA is smaller than or equal to that of Phase I.

For ASAP7 cells, the average runtime has been improved up to 131.7× [557.34 s@Phase I → 4.23 s@Phase IV ($T1$)]. In Phase III, the conditional localization with tolerance $T = 1$ ($T1$) results in 0.07% increment of the total ML while all key metrics are the same with tolerance $T = 3$ ($T3$). Phase IV shows the same results with Phase I, except for the increment of the total ML caused by the localization.

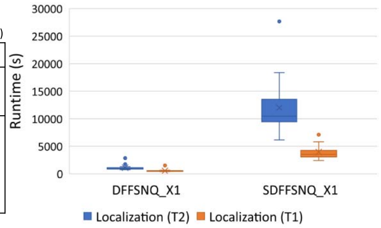
The average runtime of NanGate cells with tighter design rules tends to be shorter in each stage because of the smaller feasible search space induced by stricter conditional design-rule constraints. Though the conditional localization may hurt the optimality of the solutions, our results show that the conditional localization with $T = 2$ or 3 is not critically harmful to the key metrics across the design rule sets. Our proposed objective function partitioning provides the equivalent optimization results to the original objective function with a maximum gap of 0.05% under the proper priority.[7]

2) Combinational Logic Cells: Fig. 25 visualizes the runtime reduction through the scalability improvement. The time intervals depicted in boxplots are derived from 21 random seeds. The design symmetry breaking (Phase II) method brings significant improvement by cutting the search space for the most of combinational logic cells except the inverter-type cells (i.e., INV_X1–X4 and INVx1–x8). Meanwhile, the runtime depends on the random seed due to the heuristic aspects of SMT. Therefore, we select the best-achieved results as the runtime from the multiple random seeds for each cell. The average gap between the first quartile and the third quartile of runtime boxplots tends to decrease by adding improvement stages [Phase I → Phase IV : 28.0 → 7.6 → 3.6 → 2.2 in Fig. 25(a), 222.3 → 52.0 → 16.5 → 2.6 in Fig. 25(b)]. This demonstrates that the search space is effectively reduced with the scalability improvement constraints.

(T : tolerance of localization)

	T	Key Metrics				
		Size	Metal Length			M_2
			Total	VIA	Metal	
DFFSNQ_X1	T1	23	612	340	272	3
	T2		606	336	270	
SDFFSNQ_X1	T1	28	702	388	314	
	T2	28	697	384	313	

(a)



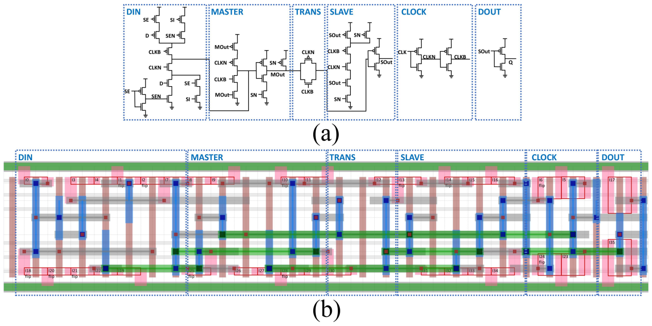
(b)

Fig. 26. Sequential logic cells. (a) Key metrics statistics. (b) Runtime variation depicted in boxplots (21 random seeds).

3) Sequential Logic Cells: Fig. 26(a) shows the key metrics of sequential logic cells in NanGate library [40]. The number of FETs/nets in DFFSNQ_X1 and SDFFSNQ_X1 is 28/19 and 36/27, respectively. Due to the high complexity, the cell-partitioning constraints with six functional modules are applied to the improvement features of Phase IV. Since the cell partitioning itself implies the breaking of design symmetry, the breaking design symmetry constraint is excluded. With the increased tolerance (1 → 2) of the conditional localization and the objective function partitioning, the total ML as well as VIA length is decreased similar to Phase IV cases in Table IV. The decreased runtime deviation in Fig. 26(b) with the decrease of the tolerance shows the reduction of the search space in the conditional localization. The cell is partitioned into and ordered with six functional modules as follows: DIN-MASTER-TRANS-SLAVE-CLOCK-DOUT. Examples of the cell partitioning based on the functionalities and the generated layout of SDFFSNQ_X1 cell are shown in Fig. 27(a) and (b), respectively.

4) Scalability of SP&R Framework: The placement permutation is expressed in (11). The number of clauses for routing is derived as (12) [31]. In order to predict the runtime of

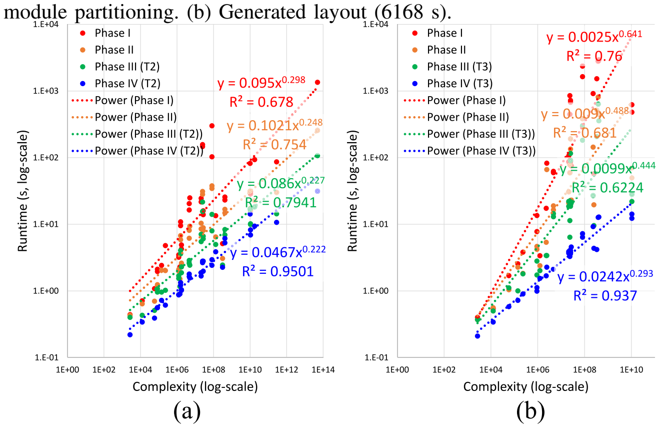
[7] In this work, the amount of VIA in Phase IV of NanGate cells is less than Phase I or Phase II, indicating that the same amount of VIA in Phases I and II can be achievable with the higher weight (i.e., > 4) of VIA elements.



(a)

(b)

Fig. 27. SDFFSNQ_X1. The largest cell in this work (28 CPPs). (a) Function module partitioning. (b) Generated layout (6168 s).



(a)

(b)

Fig. 28. Scalability of SP&R for combinational logic cells (in log-scale). (a) NanGate 35 combinational logic cells. (b) ASAP7 30 combinational logic cells.

SP&R, we test various structures combining (11) and (12) to maximize the correlation R^2. Using (13), we achieve $R^2 = 0.9501$ for Phase IV on NanGate 35 combinational logic cells [Fig. 28(a)] and $R^2 = 0.937$ for Phase IV on ASAP7 on 30 combinational logic cells [Fig. 28(b)]

$$\textit{Placement permutation: } O\left(\left(\frac{(X/2)!}{(X/2-N/2)!}\right)^2\right) \quad (11)$$

$$\textit{Routing clauses: } O(X \cdot P) \quad (12)$$

$$\textit{SP\&R runtime prediction base:}$$

$$O\left(\frac{(X/2)!}{(X/2-N/2)!} \cdot X^2 \cdot P^2\right). \quad (13)$$

F. Experimental Statistics of Practical 7 nm Cell Library

We show that the proposed SP&R satisfies both practicality and scalability through comparing key metrics and runtime with ASAP7 library [26] and the previous work [19], respectively. Though the apple-to-apple comparison may not be possible due to the different netlists and cell architectures, cell size and number of occupied $M2$ tracks can be compared because: 1) the important routing resources, such as the numbers of horizontal routing tracks and fins are the same and 2) the design rule parameters in SP&R are carefully tuned to match the routability. Also, we have compared the runtime between the cells generated under the equivalent complexity (i.e., FETs, NETs, and size). All results are generated with the proposed features of Phase IV ($T1$). For latch/flipflop cells, cell partitioning, CM, and SDB in a crossover constraint are applied.

TABLE V

SP&R RESULTS OF 142 COMBINATIONAL LOGIC CELLS FROM ASAP7 LIBRARY WITHOUT FET SEPARATION: #CELL = THE NUMBER OF VARIANTS OF EACH CELLTYPE IN COLUMN 1, #FET/#NET = THE MINIMUM/MAXIMUM NUMBER OF FETS/NETS, M_2 = THE NUMBER OF USED $M2$ TRACKS, SIZE/M_2/RUNTIME ARE ON AVERAGE VALUE

CellType	#Cell	#FET		#NET		Size		ASAP7 [26]		SP&R		Runtime(s)	
		Min.	Max.	Min.	Max.	Min.	Max.	Size	M_2	Size	M_2	SP&R	[19]
INV/BUF	25	2	10	4	11	3	30	9.6	0.0	9.6	0.0	8.1	17.0
AND/OR	16	6	12	7	13	6	14	7.8	0.0	7.8	0.0	36.3	-
NAND/NOR	21	4	10	5	12	4	14	7.4	0.0	7.4	0.0	14.7	84.5
AOI/OAI	34	6	18	8	20	5	13	7.9	0.0	7.9	0.0	160.4	116.5
AO/OA	32	8	20	9	21	6	16	11.1	0.0	11.0	0.0	579.8	-
AOAI/OAOI	3	8	10	10	12	6	9	7.0	0.0	7.0	0.0	13.7	-
MAJ	3	10	12	10	11	7	10	8.7	0.0	8.7	0.0	14.2	-
XNOR/XOR	4	10	12	9	10	9	11	10.0	0.0	10.0	0.0	684.5	-
TIE	2	2	2	4	4	4	4	3.0	0.0	4.0	0.0	0.4	-
HA/FA	2	10	24	9	17	9	14	11.5	2.0	11.5	0.0	126.8	-
Total	**142**	**Average**						**8.84**	**0.03**	**8.83**	**0.00**	**203.4**	**-**

TABLE VI

SP&R RESULTS OF 15 COMBINATIONAL/3 SEQUENTIAL LOGIC CELLS FROM ASAP7 LIBRARY WITH FET SEPARATION

CellType	#Cell	#FET		#NET		Size		ASAP7 [26]		SP&R		
		Min.	Max.	Min.	Max.	Min.	Max.	Size	M_2	Size	M_2	Runtime(s)
AND/OR	4	12	19	7	13	10	14	13.5	0.0	12.0	0.5	15783.8
NAND/NOR	4	6	9	6	8	8	20	14.0	0.5	14.0	0.0	22205.8
AOI/OAI	4	12	16	8	10	8	10	9.0	0.0	9.0	0.0	44.2
OAOI	1	10	10	10	10	8	8	8.0	0.0	8.0	0.0	13.1
XNOR/XOR	2	16	16	9	9	10	10	12.0	2.0	10.0	0.0	62.3
ICG	3	26	30	18	18	18	20	19.0	2.0	19.0	0.0	250.2
Total	**18**	**Average**						**13.1**	**0.67**	**12.5**	**0.11**	**8501.3**

1) Combinational Logic Cells: Table V presents the results of 142 combinational logic cells in ASAP7 library. The #Cell refers to the number of variants of each cell type. The number of FETs in each cell ranges from 2 to 24 and the cell size is in the range of 3–30 CPPs. Compared to the known layouts of ASAP7 library, SP&R has similar or better results in terms of cell size and number of used $M2$ tracks. Two TIE cells are the two exceptions that require a gate cut structure (i.e., connecting different gate signals on the same gate column), which our framework does not support. Among 142 cells, one cell has reduced the number of $M2$ tracks by 4, and two cells have reduced cell size by one to two CPPs. Compared to the previous sequential approach [19], the average runtime of SP&R for the same cell types with equivalent complexity shows the reasonable overhead. The average runtime per cell is about 4 min.

Table VI presents the results of 18 combinational and sequential logic cells with separated FETs. In ASAP7 library, the size of some cells is further reduced by separating multi-fingered FETs into several unit-finger FETs. SP&R supports user-friendly interface, providing these features by modifying the input "design layout" file. SP&R shows further improvements both in the size (13.1 → 12.5) and #$M2$ tracks (0.67 → 0.11) on average than ASAP7 library. Fig. 29 illustrates an example. By separating FETs, the cell size is reduced by four CPPs [Fig. 29(b) and (c)]. Furthermore, the result of SP&R has a smaller cell size and the less number of $M2$ tracks compared to the known layout [Fig. 29(a)]. The increment on #$M2$ at the "AND/OR" cell type is caused by the "AND5x2" cell whose cell size is reduced by six CPPs compared to the known layout at the cost of 2 more #$M2$ tracks. The average

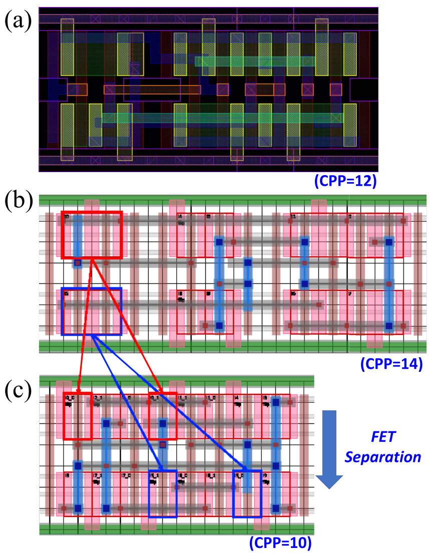


Fig. 29. Layout of XOR2x1. (a) Layout from ASAP7 standard cell library [26]. SP&R layout (b) without and (c) with FET separation.

TABLE VII
SP&R RESULTS OF 23 SEQUENTIAL LOGIC CELLS FROM ASAP7 LIBRARY WITHOUT FET SEPARATION

CellType	#FET	#NET	ASAP7 [26]		SP&R			[19]				
			Size	M2	Size	M2	Runtime	CellType	#FET	#NET	Size	Runtime
DHLx1	16	13	15	2	15	0	95.3	ELATN X1	12	11	12	1272
DHLx2	16	13	16	2	16	0	95.8	ELATS X1	10	11	10	1048
DHLx3	16	13	17	2	17	0	124.3	ELAT X1	12	11	12	1220
DLLx1	16	13	15	2	15	0	93.1	ELAT X3	12	11	16	2740
DLLx2	16	13	16	2	16	0	90.0	INV ELAT X1	14	12	16	3657
DLLx3	16	13	17	2	17	0	126.1	INV ELAT X3	14	12	20	654
DFFHQNx1	24	17	20	2	20	0	170.7	DFFQ X1	28	21	20	4351
DFFHQNx2	24	17	21	2	21	0	174.6	ESLATS X1	26	25	32	4217
DFFHQNx3	24	17	22	2	22	0	212.5	L1LATF X1	26	21	22	4155
DFFHQx4	26	18	25	2	25	0	342.5					
DFFLQNx1	24	17	20	2	20	0	173.2					
DFFLQNx2	24	17	21	2	21	0	197.5					
DFFLQNx3	24	17	22	2	22	0	210.8					
DFFLQx4	26	18	25	2	25	1	519.0					
SDFHx1	32	23	25	5	25	3	1880.4	ESLATN X1	32	25	36	6729
SDFHx2	32	23	26	4	26	3	2327.2	ESLAT X1	32	25	36	5763
SDFHx3	32	23	27	4	27	3	2466.5	ESLAT X3	32	25	36	4250
SDFHx4	32	23	31	3	28	3	2184.1	SDFFQS X1	32	27	24	31630
SDFLx1	32	23	25	4	25	2	1511.3					
SDFLx2	32	23	26	4	26	2	1479.5					
SDFLx3	32	23	27	4	27	2	1632.0					
SDFLx4	32	23	31	3	28	2	1824.8					
ASYNC_DFFHx1	32	23	26	6	26	2	2848.0					
average	**25.2**	**18.4**	**22.4**	**2.8**	**22.2**	**1.0**	**903.4**					

runtimes of AND/OR and NAND/NOR types of cells are much higher than the other cells in Table V due to the increased complexity and the larger cell size.

2) Sequential Logic Cells: Table VII presents the results of 23 sequential logic cells in ASAP7 library. For all sequential logic cells, SP&R obtains superior solutions that are smaller than or equal to the known layouts from ASAP7 library in terms of the cell size and the number of used $M2$ tracks. Fig. 24(b) displays a layout of DFFHQNx1 generated by SP&R. With the same cell size, the result of SP&R requires less $M2$ routing tracks than the known layout of Fig. 24(a). All sequential logic cells are generated within 42 min and the average runtime per cell is less than 16 min. Compared to the previous work [19], SP&R's cell generation shows the smaller runtime for the cells with the equivalent complexity.

Overall, SP&R reduces the cell size and #$M2$ tracks by 0.1 CPP and 0.3 tracks on average for all 183 ASAP7 cells compared to the known layouts, respectively.

VI. CONCLUSION

We have described a new SMT-based standard cell-layout design framework that satisfies both practicality and scalability. Our framework provides fully automated procedures generating the optimal cell layouts that combine the P&R in search space. We have improved the scalability of our framework by introducing several search-space reduction techniques, resulting in the generation of a whole standard cell library[8] with layouts that provide improvement of cell size and #$M2$ tracks by 0.1 CPP and 0.3 tracks on average compared to the known layouts, respectively. We show that our framework successfully produces DRC-clean layouts with substantial design features. SP&R achieves an average of 20.8× to 131.7× runtime improvement over the previous work [25] by exchanging less than 0.2% of the total ML. We demonstrate that our framework successfully accomplishes a wide variety of cell-layout designs, up to 28 CPPs, 36 FETs, 27 nets, and 92 commodities, within 1.75 h for the largest cell (SDFFSNQ_X1, Fig. 27).

REFERENCES

[1] S. M. Y. Sherazi *et al.*, "Standard-cell design architecture options below 5nm node: The ultimate scaling of finfet and nanosheet," in *Proc. Design Process Technol. Co-Optim. Manuf. XIII*, vol. 10962, 2019, Art. no. 1096202.

[2] A. P. Jacob, R. Xie, M. G. Sung, L. Liebmann, R. T. Lee, and B. Taylor, "Scaling challenges for advanced CMOS devices," *Int. J. High Speed Electron. Syst.*, vol. 26, no. 01n02, 2017, Art. no. 1740001.

[3] L. Liebmann, J. Zeng, X. Zhu, L. Yuan, G. Bouche, and J. Kye, "Overcoming scaling barriers through design technology cooptimization," in *Proc. Symp. VLSI Technol.*, Honolulu, HI, USA, 2016, pp. 1–2.

[4] C.-K. Cheng, D. Lee, and D. Park, "Standard-cell scaling framework with guaranteed pin-accessibility," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Sevilla, Spain, 2020, pp. 1–5.

[5] A. B. Kahng, H. Lee, and J. Li, "Measuring progress and value of IC implementation technology," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Austin, TX, USA, 2016, pp. 1–8.

[6] K. Jo, S. Ahn, J. Do, T. Song, T. Kim, and K. Choi, "Design rule evaluation framework using automatic cell layout generator for design technology co-optimization," *IEEE Trans. Very Large Scale (VLSI) Integr. Syst.*, vol. 27, no. 8, pp. 1933–1946, Aug. 2019.

[7] C. Lazzari, C. Santos, and R. Reis, "A new transistor-level layout generation strategy for static CMOS circuits," in *Proc. 13th IEEE Int. Conf. Electron. Circuits Syst.*, Nice, France, 2006, pp. 660–663.

[8] X. Xu, N. Shah, A. Evans, S. Sinha, B. Cline, and G. Yeric, "Standard cell library design and optimization methodology for ASAP7 PDK," in *Proc. 36th Int. Conf. Comput.-Aided Design (ICCAD)*, 2017, pp. 999–1004.

[9] B. Taylor and L. Pileggi, "Exact combinatorial optimization methods for physical design of regular logic bricks," in *Proc. 44th Annu. Design Autom. Conf. (DAC)*, 2007, pp. 344–349.

[10] P.-H. Wu *et al.*, "1-D cell generation with printability enhancement," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 3, pp. 419–432, Mar. 2013.

[11] N. Ryzhenko and S. Burns, "Standard cell routing via Boolean satisfiability," in *Proc. Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 2012, pp. 603–612.

[12] C. J. Poirier, "Excellerator: Custom CMOS leaf cell layout generator," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 8, no. 7, pp. 744–755, Jul. 1989.

[13] W. Ye, B. Yu, D. Z. Pan, Y.-C. Ban, and L. Liebmann, "Standard cell layout regularity and pin access optimization considering middle-of-line," in *Proc. 25th Ed. Great Lakes Symp. VLSI (GLSVLSI)*, 2015, pp. 289–294.

[8] TBUF_X16 cell in NanGate library can not be generated within a few days because it requires a gate cut structure to reduce the excessive cell size (CPP = 46). By applying more separated objective functions, SP&R generates a suboptimal solution (in terms of the total ML) within an hour.

[14] J. Seo, J. Jung, S. Kim, and Y. Shin, "Pin accessibility-driven cell layout redesign and placement optimization," in *Proc. Design. Autom. Conf. (DAC)*, Austin, TX, USA, 2017, pp. 1–6.

[15] X. Xu, B. Yu, J.-R. Gao, C.-L. Hsu, and D. Z. Pan, "PARR: Pin-access planning and regular routing for self-aligned double patterning," *ACM Trans. Design Autom. Electron. Syst.*, vol. 21, no. 3, p. 42, 2016.

[16] N. Ryzhenko, S. Burns, A. Sorokin, and M. Talalay, "Pin access-driven design rule clean and DFM optimized routing of standard cells under Boolean constraints," in *Proc. Int. Symp. Phys. Design (ISPD)*, 2019, pp. 41–47.

[17] M. Guruswamy *et al.*, "CELLERITY: A fully automatic layout synthesis system for standard cell libraries," in *Proc. 34th Design Autom. Conf. (DAC)*, Anaheim, CA, USA, 1997, pp. 327–332.

[18] A. M. Ziesemer and R. A. da Luz Reis, "Simultaneous two-dimensional cell layout compaction using MILP with ASTRAN," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Tampa, FL, USA, 2014, pp. 350–355.

[19] P. Van Cleeff, S. Hougardy, J. Silvanus, and T. Werner, "BonnCell: Automatic cell layout in the 7-nm era," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 10, pp. 2872–2885, Oct. 2020.

[20] Y.-L. Li *et al.*, "NCTUcell: A DDA-aware cell library generator for FinFet structure with implicitly adjustable grid map," in *Proc. Design Autom. Conf. (DAC)*, Las Vegas, NV, USA, 2019, p. 120.

[21] A. Sorokin and N. Ryzhenko, "Sat-based placement adjustment of FinFets inside unroutable standard cells targeting feasible DRC-clean routing," in *Proc. Great Lakes Symp. VLSI (GLSVLSI)*, 2019, pp. 159–164.

[22] C. Barrett and C. Tinelli, "Satisfiability modulo theories," in *Handbook of Model Checking*. Cham, Switzerland: Springer, 2018, pp. 305–343.

[23] N. Bjørner, A.-D. Phan, and L. Fleckenstein, "νz-an optimizing SMT solver," in *Proc. Int. Conf. Tools Algorithms Construct. Anal. Syst. (TACAS)*, 2015, pp. 194–199.

[24] R. Sebastiani and P. Trentin, "OptimathSAT: A tool for optimization modulo theories," in *Proc. Int. Conf. Comput.-Aided Verif. (CAV)*, 2015, pp. 447–454.

[25] D. Park, D. Lee, I. Kang, S. Gao, B. Lin, and C.-K. Cheng, "SP&R: Simultaneous placement and routing framework for standard cell synthesis in sub-7 nm," in *Proc. 25th Asia South Pac. Design Autom. Conf. (ASP-DAC)*, Beijing, China, 2020, pp. 345–350.

[26] V. Vashishtha, M. Vangala, and L. T. Clark, "ASAP7 predictive design kit development and cell design technology co-optimization," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Irvine, CA, USA, 2017, pp. 992–998.

[27] L. De Moura and N. Bjørner, "Z3: An efficient smt solver," in *Proc. Int. Conf. Tools Algorithms Construct. Anal. Syst. (TACAS)*, 2008, pp. 337–340.

[28] M. Rentmeesters, W. Tsai, and K.-J. Lin, "A theory of lexicographic multi-criteria optimization," in *Proc. 2nd IEEE Int. Conf. Eng. Complex Comput. Syst. Held Jointly 6th CSESAW 4th IEEE RTAW*, Montreal, QC, Canada, 1996, pp. 76–79.

[29] I. Kang, D. Park, C. Han, and C.-K. Cheng, "Fast and precise routability analysis with conditional design rules," in *Proc. ACM/IEEE Int. Workshop Syst. Level Interconnect Predict. (SLIP)*, San Francisco, CA, USA, 2018, p. 4.

[30] D. Park, I. Kang, Y. Kim, S. Gao, B. Lin, and C.-K. Cheng, "ROAD: Routability analysis and diagnosis framework based on SAT techniques," in *Proc. Int. Symp. Phys. Design (ISPD)*, 2019, pp. 65–72.

[31] D. Park *et al.*, "Grid-based framework for routability analysis and diagnosis with conditional design rules," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 12, pp. 5097–5110, Dec. 2020.

[32] Sr Software Engineering Group Director, *Personal Communication*, Cadence Design Syst., Inc., San Jose, CA, USA, 2019.

[33] C. Han, K. Han, A. B. Kahng, H. Lee, L. Wang, and B. Xu, "Optimal multi-row detailed placement for yield and model-hardware correlation improvements in sub-10nm VLSI," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Irvine, CA, USA, 2017, pp. 667–674.

[34] S. Sutanthavibul, E. Shragowitz, and J. B. Rosen, "An analytical approach to floorplan design and optimization," in *Proc. Design Autom. Conf. (DAC)*, 1991, pp. 187–192.

[35] X. Xu, B. Cline, G. Yeric, B. Yu, and D. Z. Pan, "Self-aligned double patterning aware pin access and standard cell layout co-optimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 5, pp. 699–712, May 2015.

[36] Y. Ma, J. Sweis, H. Yoshida, Y. Wang, J. Kye, and H. J. Levinson, "Self-aligned double patterning (SADP) compliant design flow," in *Proc. SPIE Design Manuf. Through Design Process Integr.*, vol. 8327, 2012, Art. no. 832706.

[37] T. Stöhr, M. Alt, A. Hetzel, and J. Koehl, "Analysis, reduction and avoidance of crosstalk on VLSI chips," in *Proc. Int. Symp. Phys. Design (ISPD)*, 1998, pp. 211–218.

[38] D. Déharbe, P. Fontaine, S. Merz, and B. W. Paleo, "Exploiting symmetry in SMT problems," in *Proc. Int. Conf. Autom. Deduction (CADE)*, 2011, pp. 222–236.

[39] J. P. Galeotti, N. S. Rosner, C. G. L. Pombo, and M. F. Frias, "TACO: Efficient SAT-based bounded verification using symmetry breaking and tight bounds," *IEEE Trans. Softw. Eng.*, vol. 39, no. 9, pp. 1283–1307, Sep. 2013.

[40] M. Martins *et al.*, "Open cell library in 15 nm freePDK technology," in *Proc. Symp. Int. Symp. Phys. Design (ISPD)*, 2015, pp. 171–178.

[41] L. T. Clark, V. Vashishtha, D. M. Harris, S. Dietrich, and Z. Wang, "Design flows and collateral for the ASAP7 7nm FinFet predictive process design kit," in *Proc. IEEE Int. Conf. Microelectron. Syst. Educ. (MSE)*, Lake Louise, AB, Canada, 2017, pp. 1–4.

[42] *Cadence Innovus User Guide*. Accessed: 2020. [Online]. Available: http:/www.cadence.com

Daeyeal Lee (Student Member, IEEE) received the B.S. and M.S. degrees from Yonsei University, Seoul, South Korea, in 2006 and 2008, respectively. He is currently pursuing the Ph.D. degree with the University of California at San Diego, San Diego, CA, USA.

Since then, he has been with Samsung Electronics, Hwaseong, South Korea, where he is currently a Principal Engineer with the FLASH Memory Product Engineering Department. His research interests include the automation of VLSI design and statistical learning for IC design, manufacturing, and VLSI testing.

Dongwon Park (Graduate Student Member, IEEE) received the B.S. and M.S. degrees from Seoul National University, Seoul, South Korea, in 2003 and 2009, respectively. He is currently pursuing the Ph.D. degree with the University of California at San Diego, San Diego, CA, USA.

Since then, he has been with Samsung Display, Giheung, South Korea, where he is currently the Staff Engineer with the Display Electronic Development Department. His research interests include high speed off/on-chip interconnect and design automation/optimization for VLSI design.

Chia-Tung Ho (Student Member, IEEE) received the B.S. and M.S. degrees from National Chiao Tung University, Hsinchu, Taiwan, in 2011 and 2013, respectively. He is currently pursuing the Ph.D. degree with the University of California at San Diego, San Diego, CA, USA.

He worked with Macronix, Hsinchu, and Novatek, Tarko-Sale, Russia, as a CAD and R&D Engineer. He joined Mentor Graphic for developing fastSPICE and Synopsys, Mountain View, CA, USA, as a Senior R&D Engineer. His research interests include DTCO pathfinding, PDN optimization, Power Grid simulation, and machine learning in VLSI.

Ilgweon Kang (Member, IEEE), photograph and biography not available at the time of publication.

Hayoung Kim (Member, IEEE), photograph and biography not available at the time of publication.

Sicun Gao (Member, IEEE), photograph and biography not available at the time of publication.

Bill Lin (Member, IEEE), photograph and biography not available at the time of publication.

Chung-Kuan Cheng (Life Fellow, IEEE), photograph and biography not available at the time of publication.