Title

Logical Reasoning Techniques for VLSI Applications

Author

Lee, Daeyeal

Publication Date

2022

UNIVERSITY OF CALIFORNIA SAN DIEGO

Logical Reasoning Techniques for VLSI Applications

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Computer Engineering)

by

Daeyeal Lee

Committee in charge:

Professor Chung-Kuan Cheng, Chair
Professor Bill Lin, Co-Chair
Professor Sicun Gao
Professor Ryan Kastner
Professor Hanh-Phuc Le

2022

The dissertation of Daeyeal Lee is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

DEDICATION

To my wife, Jisu,

my son, Jihu,

my daughters, Yeonhu, Seohu,

and my family

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my advisor, Prof. Chung-Kuan Cheng, for his constant guidance and support. A special thanks to Prof. Bill Lin for his valuable inputs and suggestions throughout my Ph.D. research. I would also like to thank my committee members for their feedbacks and contributions to this work: Prof. Hanh-Phuc Le, Prof. Ryan Kastner, and Prof. Sicun Gao. I also express my deepest gratitude to my supervisors, Kihwan Choi and Jaewoo Lim, for their guidance and encouragement to start pursuing a Ph.D. degree in Samsung Electronics. I acknowledge the generous financial support from my company, Samsung Electronics.

I gratefully recognize the help of my great lab members, Chester Holtz, Chia-Tung Ho, Uday Mallappa. I had the pleasure of working with them. I also would like to give special thanks to Ilgweon Kang and Dongwon Park.

Most importantly, I am grateful for my family's unconditional, unequivocal, and loving support.

Chapter 2 contains materials from "SMT-Based Contention-Free Task Mapping and Scheduling on 2D/3D SMART NoC with Mixed Dimension-Order Routing", by Daeyeal Lee, Bill Lin and Chung-Kuan Cheng, which appears in ACM Transactions on Architecture and Code Optimization, March 2022. The dissertation author was the primary investigator and author of this paper.

Chapter 3 contains materials from "SP&R: SMT-based Simultaneous Place-&-Route for Standard Cell Synthesis of Advanced Nodes", by Daeyeal Lee, Dongwon Park, Chia-Tung Ho, Ilgweon Kang, Hayoung Kim, Sicun Gao, Bill Lin and Chung-Kuan Cheng, which appears in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, November 2020. The dissertation author was the primary investigator and author of this paper.

Chapter 4 contains materials from "Many-Tier Vertical Gate-All-Around Nanowire FET Standard Cell Synthesis for Advanced Technology Nodes", by Daeyeal Lee, Chia-Tung Ho, Ilgweon Kang, Sicun Gao, Bill Lin and Chung-Kuan Cheng, which appears in IEEE Journal on

Exploratory Solid-State Computational Devices and Circuits, June 2021. The dissertation author was the primary investigator and author of this paper.

Chapter 5 contains materials from “CoRe-ECO: Concurrent Refinement of Detailed Place-and-Route for an Efficient ECO Automation”, by Chung-Kuan Cheng, Andrew B. Kahng, Ilgweon Kang, Minsoo Kim, Daeyeal Lee, Bill Lin, Dongwon Park, Mingyu Woo, which appears in International Conference on Computer Design, December 2021. The dissertation author was the primary investigator and author of this paper.

VITA

2001-2004	Software Engineer, Interhouse Co., Korea
2006	B. S. in Electrical and Electronic Engineering, Yonsei University, Korea
2008	M. S. in Electrical and Electronic Engineering, Yonsei University, Korea
2008-2011	Engineer, Samsung Electronics Co., Ltd, Korea
2011-2018	Staff Engineer, Samsung Electronics Co., Ltd, Korea
2018-Present	Principal Engineer, Samsung Electronics Co., Ltd, Korea
2022	Ph. D. in Electrical Engineering (Computer Engineering), University of California San Diego, US

PUBLICATIONS

Daeyeal Lee, Bill Lin, and Chung-Kuan Cheng, “SMT-Based Contention-Free Task Mapping and Scheduling on 2D/3D SMART NoC with Mixed Dimension-Order Routing”, *ACM Transactions on Architecture and Code Optimization (TACO)*, March 2022

Chung-Kuan Cheng, Andrew B. Kahng, Ilgweon Kang, Minsoo Kim, Daeyeal Lee, Bill Lin, Dongwon Park, and Mingyu Woo, “CoRe-ECO: Concurrent Refinement of Detailed Place-and-Route for an Efficient ECO Automation”, *International Conference on Computer Design (ICCD)*, December 2021 (Alphabetical Order)

Daeyeal Lee, Bill Lin, and Chung-Kuan Cheng, “SMT-based Contention-Free Task Mapping and Scheduling on SMART NoC”, *IEEE Embedded Systems Letters*, December 2021

Chung-Kuan Cheng, Andrew B. Kahng, Ilgweon Kang, Minsoo Kim, Daeyeal Lee, Dongwon Park, and Mingyu Woo, “PROBE2.0: A Systematic Framework for Routability Assessment from Technology to Design in Advanced Nodes”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, June 2021 (early access available, Alphabetical Order).

Chung-Kuan Cheng, Chia-Tung Ho, Daeyeal Lee, and Bill Lin, “Multirow Complementary-FET (CFET) Standard Cell Synthesis Framework Using Satisfiability Modulo Theories (SMTs)”, *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits (JXCDC)*, June 2021

Daeyeal Lee, Chia-Tung Ho, Ilgweon Kang, Sicun Gao, Bill Lin, and Chung-Kuan Cheng, “Many-Tier Vertical Gate-All-Around Nanowire FET Standard Cell Synthesis for Advanced Technology Nodes”, *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits (JXCDC)*, June 2021

Chung-Kuan Cheng, Chia-Tung Ho, Daeyeal Lee, Bill Lin, and Dongwon Park, “Complementary-FET (CFET) Standard Cell Synthesis Framework for Design and System Technology Co-Optimization Using SMT”, *IEEE Transactions on Very Large Scale Integration (TVLSI)*, June 2021

Daeyeal Lee*, Dongwon Park*, Chia-Tung Ho, Ilgweon Kang, Hayoung Kim, Sicun Gao, Bill Lin, and Chung-Kuan Cheng, “SP&R: SMT-based Simultaneous Place-&-Route for Standard Cell Synthesis of Advanced Nodes”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, November 2020 *Contributed Equally

Chung-Kuan Cheng, Chiatung Ho, Daeyeal Lee, and Dongwon Park, “A Routability-Driven Complimentary-FET (CFET) Standard Cell Synthesis Framework using SMT”, *2020 International Conference on Computer-Aided Design (ICCAD)*, November 2020 (Alphabetical Order)

Chung-Kuan Cheng, Daeyeal Lee, and Dongwon Park, “Standard-Cell Scaling Framework with Guaranteed Pin-Accessibility”, *IEEE International Symposium on circuits and systems (ISCAS)*, October 2020 (Alphabetical Order)

Dongwon Park, Daeyeal Lee, Ilgweon Kang, Chester Holtz, Sicun Gao, Bill Lin, and Chung-Kuan Cheng, “Grid-based framework for routability analysis and diagnosis with conditional design rules”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, February 2020

Dongwon Park*, Daeyeal Lee*, Ilgweon Kang, Sicun Gao, Bill Lin, and Chung-Kuan Cheng, “SP&R: Simultaneous Placement and Routing framework for standard cell synthesis in sub-7nm”, *IEEE/ACM Asia and South Pacific Design Automation Conference (ASP-DAC)*, January 2020 *Contributed Equally

Daeyeal Lee et al., “A 64Gb 533Mb/s DDR interface MLC NAND Flash in sub-20nm technology”, *IEEE International Solid-State Circuits Conference (ISSCC)*, April 2012

Daeyeal Lee, Myunghoon Yang, Yongjoon Kim, Youngkyu Park, Hyunjun Yoon, and Sungho Kang, “A Low Power ECC H-matrix Optimization Method using an Ant Colony Optimization”, *The Institute of Electronics Engineers of Korea Journal, SD*, January 2008

Daeyeal Lee, Yongjoon Kim, and Sungho Kang, “A New Hardware Efficient Interconnect BIST”, *International SoC Design Conference (ISOCC)*, October 2005

ABSTRACT OF THE DISSERTATION

Logical Reasoning Techniques for VLSI Applications

by

Daeyeal Lee

Doctor of Philosophy in Electrical Engineering (Computer Engineering)

University of California San Diego, 2022

Professor Chung-Kuan Cheng, Chair
Professor Bill Lin, Co-Chair

With the relentless scaling of technology nodes, VLSI design engineers encounter non-trivial challenges, particularly in the physical layout and multiprocessor system-on-chips design. Thus, a holistic exploration, co-optimizing the design/process side of the technology/system architecture, becomes an essential approach to maintaining the power, performance, area, and cost (PPAC) gain. Those co-optimizations require fidelity in terms of the optimality of solutions. However, most of the VLSI application problems are NP-complete so the complexity of derivation is too huge to find an optimal solution. Therefore, many conventional works focus on divide-and-conquer-style and/or heuristic approaches due to the intrinsic scalability limitation of the problem. As a result, outcomes of these approaches are hard to reach the optimal solution due to the intractable search space partitioning and heuristic manner. In this dissertation, we propose logical reasoning techniques and automated frameworks to tackle those challenges for several VLSI applications in the physical design (PD) and network-on-chip (NoC) design. We mainly focus on finding an optimal solution by exact solving of the integrated constraint satisfaction problem (CSP) which enables simultaneous optimization with one-time execution without any iteration. To alleviate the huge complexity of the VLSI application problems, (i) we utilize a matured fast reasoning method: satisfiability modulo theories (SMT) and formulate all our variables/constraints in a Boolean manner, (ii) we simplify our placement-and-routing (P&R) and task mapping/scheduling graphs and apply encoding techniques for further refinement, and (iii) we identify and implement practical constraints to reduce the search space.

In the physical design applications, we propose automated frameworks, which simultaneously solve place and route without deploying any sequential/separate operations for the conventional FinFET and many-tier Vertical Gate-All-Around-FET (VFET) standard cell synthesis and concurrent refinement in the engineering change order (ECO) stage. The proposed standard cell synthesis frameworks utilize the multi-objective optimization feature of SMT to obtain optimal layout results. To achieve practical scalability of the framework, we develop various search-space reduction techniques. Through orchestrating all innovative tactics together, our framework successfully generates a whole 7*nm* FinFET standard cell library and one to four tiers VFETs. Our ECO automation framework efficiently resolves pin accessibility-induced design rule violations (DRVs) by simultaneously performing detailed placement, detailed routing, and cell replacement. In addition to perturbation-minimized solutions, our proposed SMT-based optimization framework also suggests the adoption of alternative master cells to better achieve DRV-clean layouts. We demonstrate that our framework successfully resolves 58.6% of remaining DRVs on average, across a range of benchmark circuits with various cell architectures.

In NoC application, we propose an SMT-based framework to find optimal contention-free task mappings with minimum application schedule lengths on 2D/3D SMART NoCs with mixed dimension-order routing. We develop efficient search-space reduction techniques to achieve practical scalability. Experiments demonstrate that our SMT framework achieves 10× higher scalability than ILP (Integer Linear Programming) for finding optimum solutions on 2D and 3D SMART NoCs and our 2D and 3D extensions of the SMT framework with mixed dimension-order routing also maintain the improved scalability with the extended and diversified routing paths, resulting in reduced application schedule lengths throughout various application benchmarks.

Chapter 1

Introduction and Preliminaries

Since the publication of Moore's law in 1965 [11], we have observed prominent efforts to push for and/or facilitate the scaling, including the Dennard's scaling prediction [12], the "More than Moore" roadmap [13], and a recent new metric proposal [14]. These self-fulfilling prophecies are essential for the continued growth of the market, expansion of the industry, and demand for research and development. Figure 1.1 illustrates the scaling roadmap of the technology nodes released by the IMEC team [1, 15]. Up to the year 2013, the scaling trend has sustained Moore's law mainly with pitch shrinkage. However, starting from 2012, the industry has been co-optimizing the design or process side of the technology (i.e., design-technology co-optimization (DTCO)) as shown in Fig. 1.2(a) because geometric reduction alone was no longer sufficient for the desired scaling. Furthermore, after 2022, standard cell device scaling starts to saturate due to yield and cost [16]. One way to continue the reduction of the device footprint is by expanding in the third dimension, e.g. using stacked gate-all-around (GAA) devices, complementary FETs, vertical FETs, and 3D logic (Fig. 1.2(b)). This system-technology co-optimization (STCO) approach changes the VLSI application problems from a conventional planar device placement problem to a three-dimensional spatial arrangement problem.

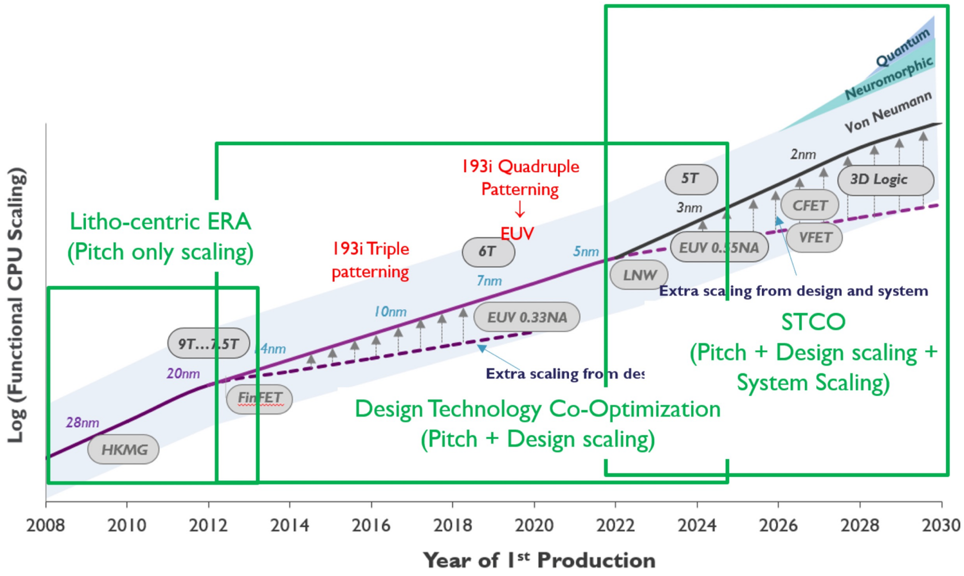


Figure 1.1: Scaling roadmap [1].

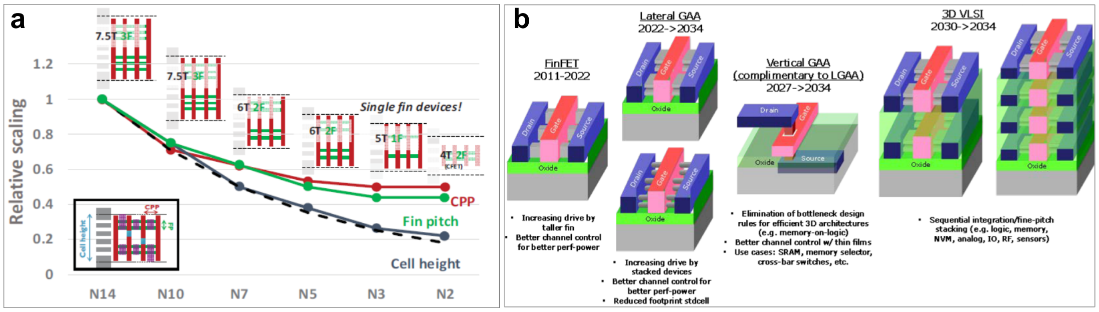


Figure 1.2: Scaling with design and system technology co-optimization. (a) Design technology co-optimization [2, 3]. (b) System technology co-optimization [4].

The biggest challenge for DTCO and STCO is that the most fundamental decisions are made with the least amount of data at the beginning of the technology development cycle because of the enormous cost and time required to maintain multiple options (e.g., the exploration of various cell architectures and design-rule sensitivity to the power, performance, area, and cost (PPAC)) for mitigating the risk of making a wrong decision. Also, the exploration of multiple possible options for technology development in DTCO and STCO requires fidelity in terms of the optimality of solutions and the precision of sensitivity derivation. However, most of the VLSI application problems are NP-complete so the complexity of derivation is too huge to find an optimal solution. Therefore, many conventional works focus on divide-and-conquer-style and/or heuristic approaches due to the intrinsic scalability limitation of the problem. As a result, outcomes of these approaches are hard to reach the optimal solution due to the intractable search space partitioning and heuristic manner.

In this dissertation, we propose logical reasoning techniques and automated frameworks to tackle those challenges for several VLSI applications in the physical design (PD) and network-on-chip (NoC) design. We mainly focus on finding an optimal solution by exact solving of the integrated constraint satisfaction problem (CSP) which enables simultaneous optimization with one-time execution without any iteration. To alleviate the huge complexity of the VLSI application problems, (i) we utilize a matured fast reasoning method: satisfiability modulo theories (SMT) and formulate all our variables/constraints in a Boolean manner, (ii) we simplify our placement-

and-routing (P&R) and task mapping/scheduling graphs and apply encoding techniques for further refinement, and (iii) we identify and implement practical constraints to reduce the search space. In the sequel, we first introduce SMT and the preliminaries of our proposed frameworks in the remaining Chapter 1. Then, we present automated frameworks utilizing logical reasoning techniques in three topics of VLSI applications; (i) NoC task mapping and scheduling (Chapter 2), (ii) standard cell synthesis (Chapter 3), and (iii) engineering change order (ECO) (Chapter 5).

The rest of this chapter is organized as follows. Section 1.1 introduces SMT and its multi-objective optimization feature. Section 1.2 describes the preliminaries of our physical design automation framework. Section 1.3 introduces preliminaries of our NoC task mapping and scheduling framework.

1.1 Satisfiability modulo theories (SMT)

Satisfiability modulo theories (SMT) generalizes the Boolean satisfiability problem (SAT) to more complex formulas. A proposition logic formula of SAT consists of variables, AND (i.e., conjunction), OR (i.e., disjunction), and NOT (i.e, negation). Given a proposition logic formula, the SAT problem is to find a variable assignment to make the formula evaluates to true (i.e., Satisfiable), or prove that no such assignment exists (i.e., Unsatisfiable). Compared to SAT, SMT is a more expressive language containing non-Boolean variables (e.g., real, integer, etc.), various data structures (e.g., lists, arrays, bit-vectors, and strings), and predicate symbols as described in [17]. Furthermore, SMT provides the feature of OMT (Optimization Modulo Theories) [18][19] to obtain the optimal solution.

1.1.1 Expressiveness of SMT for Conditional Constraints

SMT provides much more expressive modeling language (e.g., “if-then-else” for the “Either-Or” constraint, built-in Boolean cardinality functions such as “at-most k” and “at-least k”, etc.) than is possible with SAT or integer linear programming (ILP) formulas. For example, to formulate a simple conditional constraint, "$if\ a \leq b\ then\ c = 1\ else\ c = 0$", SMT only requires a single-statement as:

$$(assert\ (ite\ (<= a\ b)\ (= c\ 1)\ (= c\ 0))) \tag{1.1}$$

while ILP needs 6 constraints with one auxiliary variable as:

$$\begin{aligned} a + (z-1)M \leq b; \quad a + zM \geq b + \varepsilon \\ 1 - M(1-z) \leq c; \quad c \leq 1 + M(1-z) \\ -zM \leq c; \quad c \leq zM \end{aligned} \tag{1.2}$$

where M is a large constant chosen as an upper bound on $a-b$, ε is a small number stating the tolerance for when a is considered to exceed b, and z is an auxiliary Boolean (0,1) variable which indicates if $a = b$. Therefore, as the conditional constraints become more complicated, more auxiliary variables and the corresponding constraints are necessary for the ILP formulation. Since our problem contains several complex conditional constraints as well as Boolean decision variables that have a significant impact on exploring the feasible solutions, SMT can solve our problem more efficiently than ILP under (i) the powerful expressiveness (i.e., the lower formulation complexity) and (ii) the faster reasoning ability of SAT solver.

1.1.2 Multi-Objective Optimization

Physical design problems in this dissertation have multiple objectives associated with placement and routing problems for standard cell layout and refinement of block-level layout results. Several SMT solvers including the optimization methodology (i.e., OMT) are recently released [18][19]. We adopt the state-of-the-art *lazy*-approach SMT solver *Z3* [18][20] to solve the given multi-objective optimization problem. *Z3* simultaneously optimizes multiple objectives in the light of "lexicographic" order. For each given objective, this effectively induces a single-objective optimization problem under the constraining condition that optimizes the higher-priority objectives.

1.2 Preliminaries of Physical Design Automation Framework Configurations

We assume on-grid and uni-directional routing graph for each layer due to the process resolution of sub-7*nm* multi-patterning technologies such as LELE (litho-etch-litho-etch), SADP and SAQP (self-aligned double and quadruple patterning) [21, 22, 23] and IC practitioners' restriction of preferred routing direction per each layer [24]. For the routing, we adopt conditional design rule-aware multi-commodity network flow theory inspired by [25, 26]. Table 1.1 presents the basic notations for the physical design automation framework.

1.2.1 On-grid Unidirectional Routing Graph: G(V, E)

Fig. 1.3 shows a multi-layered 3-D routing graph $G = (V,E)$ to represent the available routing resources (e.g., horizontal and vertical tracks on each layer, inter-layer 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 inter-layer VIAs. Source-sink network-flow connectivity is represented on the graph G.

[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.

Table 1.1: Basic notations for the physical design automation framework.

Term	Description
$G(V,E)$	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, $u \in a(v)$
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^n_m	m^{th} sink of n
f^n_m	A two-pin subnet connecting s^n and d^n_m, i.e., a commodity
v^n	0-1 indicator if v is used for n
$e^n_{v,u}$	0-1 indicator if $e_{v,u}$ is used for n
$f^n_m(v,u)$	0-1 indicator if $e_{v,u}$ is used for commodity f^n_m
$m_{v,u}$	0-1 indicator if there is a metal segment on $e_{v,u}$
$g_{d,v}$	0-1 indicator if v forms d-side EOL of a metal segment

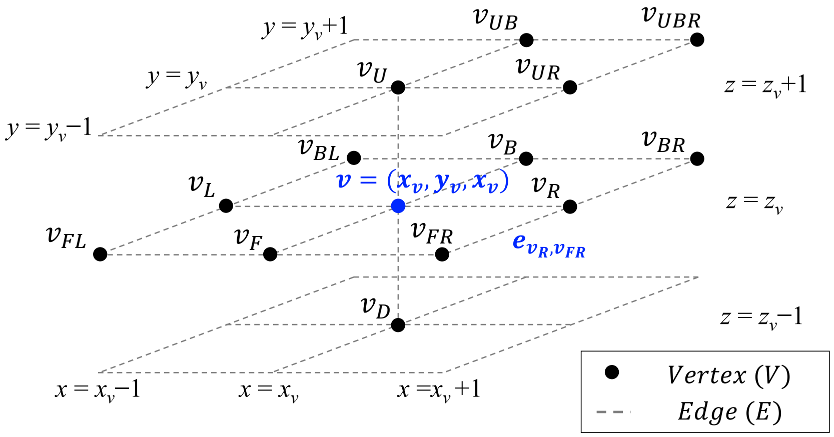


Figure 1.3: Example of 3-D grid-based routing graph, $G(V,E)$.

1.2.2 Multi-Commodity Network Flow Theory

We adopt a multi-commodity network flow theory to represent routing flows for a given layout. A flow network is a directed graph where each edge has a capacity and each edge receives a flow. The amount of flow on an edge cannot exceed the capacity of the edge. Also, the amount of incoming flow into a node must be the same as the amount of outgoing flow unless the node is a source, which has only outgoing flow, or sink, which has only incoming flow. Inspired by [26, 27], we define the flows as an undirected graph to reduce the solution space as shown in Expression (1.3). Expression (1.4) is the modified flow conservation constraint without flow direction. In case of source s^n or sink d^n_m, the summation of commodity flow indicator $f^n_m(v,u)$ between a certain vertex v and its adjacent vertices $a(v)$ is set as 1, 0 or 2 in the other cases.

$$f^n_m(v,u) = f^n_m(u,v), \quad \mathit{No\ Direction} \tag{1.3}$$

$$\sum_{u \in a(v)} f^n_m(v,u) = \begin{cases} 1, & \text{if } v = s^n, d^n_m \\ 2p, p = \{0,1\}, & \text{otherwise} \end{cases} \tag{1.4}$$

$$\forall v \in V,\ \forall n \in N,\ \forall d^n_m \in D^n$$

1.2.3 Implementation of Conditional Design Rules

Conditional design rules are implemented as constraints using geometry variables $g_{d,v}$ that are defined as end-of-line indicators of each vertex as shown in Expression (1.5). For example, the PRL rule is a design rule to avoid “single-point-contact” in manufacturing SADP mask [23]. Fig. 1.4 and Expression (1.6) represent an example of PRL rule and the corresponding formulation when the run-length (RL) is 2. The built-in functions of $Z3$ SMT solver such as at-most k (AMk) and at-least k (ALk) are used to formulate cardinality constraints in the SMT formulation.

$$\begin{aligned} g_{L,v} &= \neg m_{v_L,v} \wedge m_{v,v_R}, \\ g_{R,v} &= m_{v_L,v} \wedge \neg m_{v,v_R}, \end{aligned} \qquad \forall v \in V_2 \tag{1.5}$$

$$\mathbf{AM1}(g_{R,v}, g_{L,v_B}, g_{L,v_{BL}}); \mathbf{AM1}(g_{R,v}, g_{L,v_F}, g_{L,v_{FL}}), \forall v \in V_0, V_2 \tag{1.6}$$

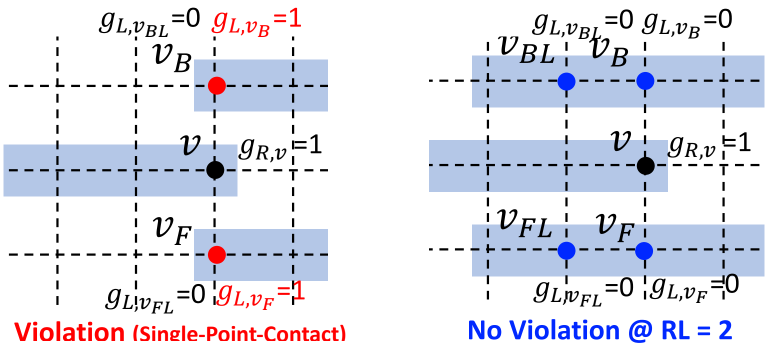


Figure 1.4: Example of PRL (Parallel Run-Length) rule.

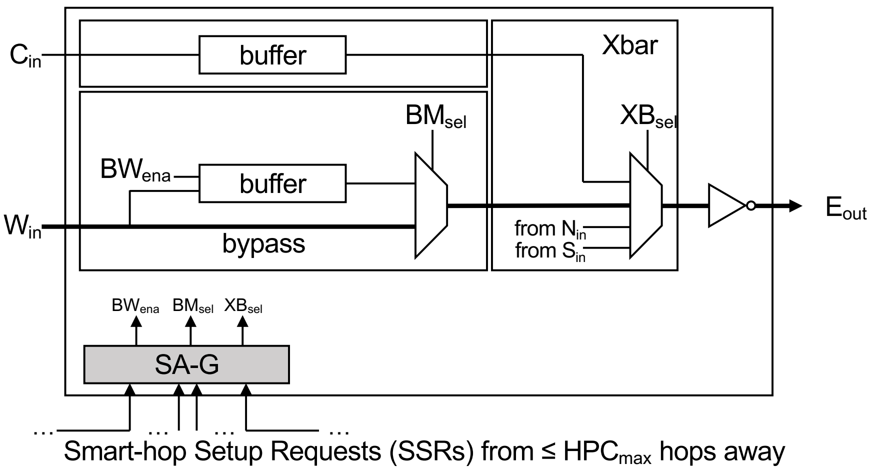


Figure 1.5: SMART router microarchitecture.

1.3 Preliminaries of NoC-based MPSoC

Network-on-Chip (NoC) is a widely used interconnection fabric that provides a highly scalable low-latency on-chip communication solution for multiprocessor system-on-chips (MPSoCs) [28]. Recently, an NoC design called SMART (Single-cycle Multi-hop Asynchronous Repeated Traversal) NoC [29, 30] has been proposed that enables a flit to traverse many router hops within a single clock cycle, potentially from the source all the way to the destination. This section describes the basic concept and communication latency model of SMART NoC.

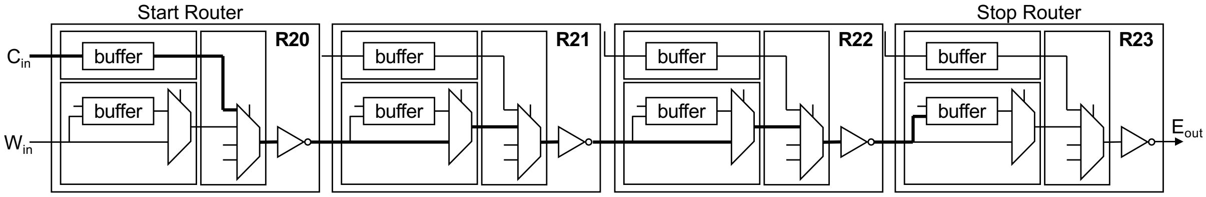


Figure 1.6: Multi-hop Traversal over a SMART-hop path.

1.3.1 Communication latency in SMART NoC

Figure 1.5 depicts the microarchitecture of a 5-port SMART router for a mesh network. For simplicity, only the $Core_{in}$ (C_{in}), $West_{in}$ (W_{in}), and $East_{out}$ (E_{out}) ports are shown in detail[2]. All other input ports are identical to W_{in}, and all other output ports are identical to E_{out}.

In a SMART NoC, asynchronous repeaters, replaced with conventional clocked link drivers at every hop allow a flit to traverse multiple hops in a single clock cycle. An alternative data-path in each router allows a flit to bypass the entire router pipeline and go directly to the next router. The bold line going from W_{in} to E_{out} in Figure 1.5 illustrates this bypass operation. An example of a multi-hop traversal (i.e., SMART-hop) is depicted in Figure 1.6. A flit travels three hops from router R20 to R23 within a single-cycle via a SMART-hop path created by appropriately controlled BW_{ena}, BM_{sel}, and XB_{sel} at intermediate routers.

To setup SMART-hops, SMART performs two-stage switch allocation: local switch allocation (SA-L) and global switch allocation (SA-G). In the SA-L stage, buffered flits at a start router arbitrate among themselves to gain access to the output ports. For each winning flit, the start router broadcasts a SMART-hop Setup Request (SSR), which carries the information about the route. Each SSR is sent through dedicated multi-drop wires that are repeated from the start router to all intermediate routers up to HPC_{max} hops away. HPC_{max} refers to the maximum number of hops that a flit can traverse in a single cycle. Upon receiving the SSRs, the recipient routers set up the control signals (i.e., BW_{ena}, BM_{sel}, and XB_{sel}) to operate in bypass or stop mode

[2] A SMART router consists of five input/output ports (i.e., *West*, *East*, *North*, *South*, and *Core*) for connections on a mesh topology.

in SA-G stage. When multiple SSRs from multiple start routers arrive at the same time (i.e., contention on the bypass path), only a winning flit determined by priority policies at the recipient router can proceed and bypass the intermediate routers. Thus, the transmission latency of the SMART-hop path can be formulated as inspired by [31]:

$$L = (t_r + t_w) \cdot (N_c + 1) + t_w \cdot (M^e - 1) + L_b \tag{1.7}$$

where t_r and t_w respectively denote the number of stages in a start router and the link latency between two routers; M^e refers to the amount of data units (i.e., flits); N_c denotes the number of links having flit contention on the bypass path; L_b refers to the delay due to the contention. In this paper, we employ contention-free task mapping and scheduling. Consequently, the communication latency L_{cf} without contention (i.e., N_c, L_b = 0) can be expressed as:

$$L_{cf} = (t_r + t_w) + t_w \cdot (M^e - 1) \tag{1.8}$$

Chapter 2

NoC Task Mapping and Scheduling: 2D/3D SMART NoC

2.1 Introduction

Network-on-Chip (NoC) is a widely used interconnection fabric that provides a highly scalable low-latency on-chip communication solution for multiprocessor system-on-chips (MPSoCs) [28]. However, since communications between cores in a regular NoC are achieved by routing messages hop-by-hop from the source to the destination, their effectiveness at non-local communications quickly diminishes due to long on-chip latencies, which degrades performance and limits the flexible usage of on-chip resources. Delays due to router pipelines, queuing, and serialization all contribute towards a much longer on-chip latency than an ideal point-to-point interconnect.

Recently, an NoC design called SMART (Single-cycle Multi-hop Asynchronous Repeated Traversal) NoC [29, 30] has been proposed that enables a flit to traverse many router hops within a single clock cycle, potentially from the source all the way to the destination. Such a multi-hop traversal is made possible by utilizing efficient router-bypass mechanisms and properly engineered wires with asynchronous repeaters. In particular, router data-paths can not only be dynamically but also statically [32] configured to enable multi-hop traversal so that flits can bypass the pipelines of intermediate routers entirely, resulting in ultra-low latency performance. Although SMART offers outstanding advantages, the performance benefits can only be fully realized if there is no contention among flows that share common links along their routing paths. When contention occurs, bypass paths must terminate early, and the corresponding flits must be stopped and buffered at intermediate routers for arbitration, degenerating in the worst-case to hop-by-hop communication. Without proper contention management, the benefits of SMART can easily vanish.

In this paper, we address the aforementioned contention problem by developing an SMT (Satisfiability Modulo Theories)-based contention-free task mapping and scheduling framework for the embedded computing application in which an application task graph can be statically compiled to a multi-core or parallel processing platform for non-preemptive execution based on 2D and 3D SMART NoC architectures. In particular, tasks are mapped to processors and scheduled to ensure contention-free routing of all messages over a SMART NoC from their source to their destination in a single cycle. Our SMT formulation can find theoretically optimal solutions that minimize the application schedule length. In contrast to prior ILP-based formulation for 2D SMART NoCs [31], our SMT formulation is substantially more compact, thanks in part to SMT's expressive power in capturing conditional constraints that constitute a large proportion of the task mapping and scheduling problem. Combined with efficient search-space reduction techniques, our SMT formulation is considerably more scalable, enabling our framework to find optimal solutions for far larger problem instances with dramatically faster runtimes.

In addition, we extend our SMT-based formulation to consider the 3D SMART NoC case. Three-dimensional (3D) IC integration is becoming increasingly important, and correspondingly, 3D NoCs are emerging as promising solutions that can deliver lower latency, higher throughput, and reduced energy consumption in comparison with their 2D counterparts [33]. In particular, recent SMART NoC advances [34, 35, 36] have substantially reduced the wiring and area overhead of SMART NoCs to enable 3D extensions. Moreover, the emergence of monolithic 3D (M3D) integration has opened up new possibilities for designing SMART 3D NoC architectures with monolithic inter-tier vias (MIVs) for vertical interconnections [37], which have much smaller dimensions than the more popular through silicon vias (TSVs) [38]. Further, a 3D SMART NoC can potentially operate at higher clock frequencies due to the utilization of vertical interconnects, which results in a reduction of the effective physical link distances [35]. In the context of our task mapping and scheduling problem, a 3D SMART NoC provides greater path diversity that makes it easier for our SMT-based formulation to find better optimal solutions with contention-free routing.

The main contributions of our work are as follows:

- We propose an SMT-based contention-free task mapping and scheduling framework for

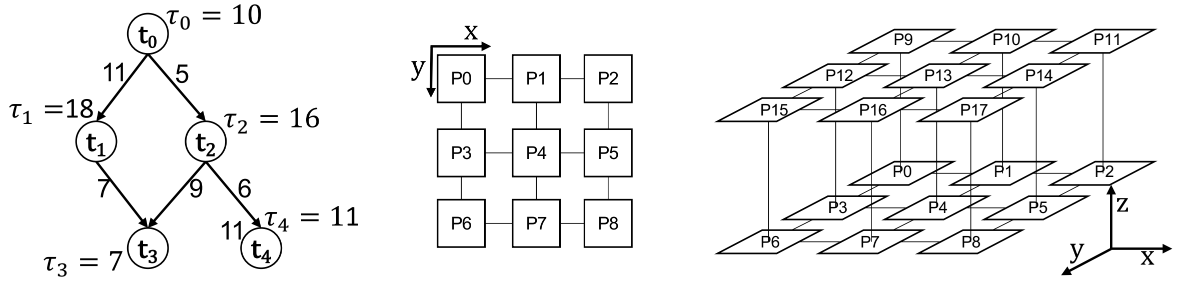
2D/3D SMART NoCs, including support for mixed dimension-order routing.

- We devise a concise model under SMT's support for expressive modeling, resulting in the fast reasoning of conditional constraints.
- We develop efficient search-space reduction techniques, e.g., adaptive boundary condition and breaking design symmetry to further improve scalability.
- We demonstrate that our framework achieves smaller formulation complexity with 16.6×/ 12.1× and 26.2×/18.7× reductions of variables and constraints on average and 10× higher scalability with 931.1× (ranges from 2.2× to 1532.1×) and 1237.1× (ranges from 4× to 4373.8×) faster average runtimes for finding optimum solutions on 2D and 3D SMART NoCs, respectively.
- Our experiments further demonstrate that the 3D extension with mixed dimension-order routing not only maintains the improved scalability but also helps to reduce the application schedule length by exploiting the greater path diversity.

The rest of this paper is organized as follows: Section 2.2 describes the problem definition. Section 2.3 presents our SMT formulation and search-space reduction techniques for improving scalability. Section 2.4 validates the proposed frameworks with extensive experimental results. Section 2.5 outlines additional related work. Section 2.6 concludes the paper.

2.2 Problem Definition

We represent an application as a task graph (TG) which is a directed acyclic graph (DAG), $TG = (T, E)$, where T is a set of all tasks $t_i \in T$ in the application and E is a set of edges $e_{u,v} \in E$, representing precedence relations between tasks t_u and t_v. In our problem, the target application domain is embedded computing in which an application task graph can be statically compiled to a

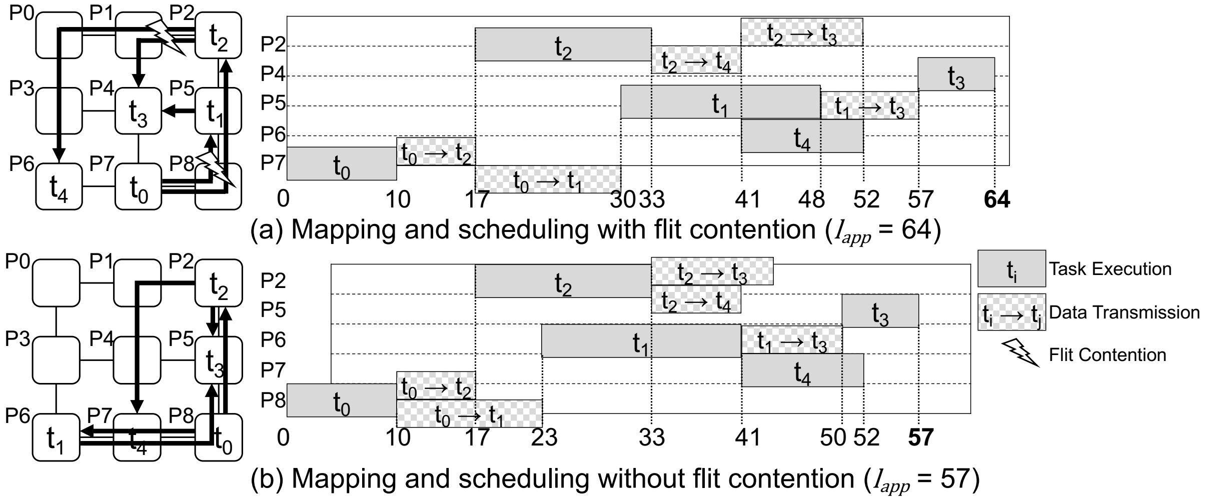


(a) Task graph

(b) 2D mesh topology graph (3X3)

(c) 3D mesh topology graph (3X3X2)

Figure 2.1: Example task graph (TG) and two-dimensional/three-dimensional mesh topology graph.



(a) Mapping and scheduling with flit contention (l_{app} = 64)

(b) Mapping and scheduling without flit contention (l_{app} = 57)

Figure 2.2: Example of task mapping and scheduling.

multi-core or parallel processing platform for non-preemptive execution, and tasks correspond to the blocks of significant computations, like signal processing tasks, not at the level of individual instructions. Each task and edge are respectively associated with the task execution time τ_i and the amount of data transferred between each pair of tasks. For example, Figure 2.1(a) shows an example task graph with five tasks and five edges including the task execution time and the amount of data.

The target architecture is SMART NoC-based homogeneous MPSoCs [29, 30] with 2D/3D mesh topology graph (MTG), $MTG = (P, L)$, where each node $p_i \in P$ represents a process element (PE) with a router and L is a set of edges representing bi-directional communication paths between adjacent PEs. Figure 2.1(b) and Figure 2.1(c) respectively show an example 3×3 2D mesh topology with 9 PEs and 3×3×2 3D mesh topology with 18 PEs. Within this topology, each PE can accommodate more than one task. The location of assigned PEs for each task $t_i \in T$ is determined by a pair of coordinates (x^{t_i}, y^{t_i}) and $(x^{t_i}, y^{t_i}, z^{t_i})$ for 2D and 3D. If two consequential tasks are mapped to the same PE, data transmission can be skipped without spending transmission latency.

The scheduling of tasks is performed by determining the release/completion time of task execution and data transmission in such a way to achieve the minimum application schedule length. Note that we assume static task execution time and a fixed amount of data for each transmission. Also, we assume that tasks are non-preemptive and no deadline requirements are enforced in our application. Therefore, for a pair of tasks u, v having precedence relation, the produced data from task u is transmitted to the subsequent task v after the completion of the task u.

We assume XY-routing in 2D topology and XYZ-routing in 3D topology as a default routing path. In this work, to achieve better latency through the extended path diversification, we explore the impact of mixed dimension-order routing on the application schedule length (l_{app}). For 2D and 3D topology, we employ all possible routing paths (i.e., XY/YX routing paths for 2D, also known as O1TURN [39], and XYZ/YXZ, ZXY/ZYX, and XZY/YZX routing paths for 3D). Then, in our formulation, we decide one of the dimension-order routing paths that can guarantee flit contention-free routing by detecting and avoiding temporal and spatial overlaps of SMART-hop paths. When we inject the traffic to the network in the scheduled time slot, we only activate the SSR signals along the path we already have decided.

Figure 2.2 illustrates the impact of the flit contention on application schedule lengths. For the same example TG in Figure 2.1(a), the mapping and scheduling solution in Figure 2.2(a) provides a larger l_{app} than that of Figure 2.2(b) due to the flit contention on its routing paths (i.e., $t_0 \rightarrow t_1/t_0 \rightarrow t_2$ and $t_2 \rightarrow t_3/t_2 \rightarrow t_4$). Based on the above definitions, our task mapping and scheduling problem can be defined as:

Given a TG and MTG, find a contention-free mapping and scheduling function from TG to MTG so that the overall end-to-end latency of the designed NoC application (i.e., application schedule length l_{app}) is minimized.

2.3 SMT Formulation for Joint Task Mapping, Scheduling, and SMART routing

In this section, we describe basic SMT formulation and scalability improvement constraints for 2D/3D SMART NoC. We formulate task mapping and scheduling of 2D/3D SMART NoC as a constraint satisfaction problem (CSP) with variables and constraints. Thus, the release time of task execution and a data transmission (i.e., scheduling), as well as the task assignment (i.e., task mapping), are both determined by our constraints. The formulations that can be adopted for both 2D and 3D mesh topologies by simple adjustments of the conditions related to the corresponding coordinate variables (i.e., x,y, and z) are expressed based on the 3D mesh topology. For the constraints which have to be carefully revised according to the dimension of topology and routing schemes, we provide separate expressions and algorithms. The notations are shown in Table 2.1.

2.3.1 Basic Formulation

Objective

As described in Section 2.2, our goal is to find a contention-free mapping and scheduling solution so that the overall end-to-end latency of the designed NoC application (i.e., application schedule length l_{app}) is minimized. Thus, our objective function minimizes the maximum completion time of tasks that have no out-going edges.

Table 2.1: Notations for the proposed SMT formulation.

Term	Description
T, E, P	Set of Tasks, Edges, and PEs (processing elements)
t	t^{th} task
e	e^{th} edge
p	p^{th} PE
x^t/y^t	*x*/*y*-coordinates of a processor on which a task t is mapped (for 2D)
$x^t/y^t/z^t$	*x*/*y*/*z*-coordinates of a processor on which a task t is mapped (for 3D)
$e_{u,v}$	A directed edge from task u to v, $\forall u, v \in T$
τ^t	The execution time of task t
M^e	The amount of data transferred on edge e
s^t/f^t	The release/completion time of task t
$s^e_{u,v}/f^e_{u,v}$	The release/completion time of data transmission on $e_{u,v}$
α_{e_i,e_j}	0-1 indicator if edges e_i, e_j have a transmission time overlap
β_{e_i,e_j}	0-1 indicator if edges e_i, e_j have shared routing paths
r^e	The type of dimension-order routing on edge e (for mixed routing, 0-1 for 2D/0-5 for 3D)

$$\begin{aligned} &Min\{l_{app}\} \\ l_{app} = &Max\{f^t | t \in T_{final}\} \\ &T_{final}\text{: a set of tasks } t \text{ without out-going edges}, \forall t \in T \end{aligned} \tag{2.1}$$

Boundary condition for processor coordinates

Given an $m \times n \times l$ PE tiles, the *x*/*y*/*z*-coordinates of a task t are bounded by *m*/*n*/*l*, respectively. Each task can only be mapped to one PE on (x^t, y^t, z^t), while each PE can accommodate multiple tasks without limitation on the number of assigned tasks.

$$0 \le x^t \le m-1;\ 0 \le y^t \le n-1;\ 0 \le z^t \le l-1,\ \forall t \in T \tag{2.2}$$

Scheduling constraints of tasks

Constraint (2.3) represents the quantitative timing relation of tasks. The source task u between two consequential tasks u, v have to be released prior to the destination task v.

$$f^u = s^u + \tau^u,\ \forall u \in T$$

$$s^u < s^v,\ \forall u, v \in T, \forall e_{u,v} \in E \tag{2.3}$$

Non-overlap of tasks

No pairs of tasks $u, v \in T$, mapped on the same PE, can overlap.

$$(s^u \geq f^v) \vee (f^u \leq s^v),\ \text{if } (x^u = x^v) \wedge (y^u = y^v) \wedge (z^u = z^v),\ \forall u, v \in T \tag{2.4}$$

Scheduling constraints of data transmissions

Constraint (2.5) represents the quantitative timing relation between tasks and data transmissions. We assume that the data transmission can be released and completed in any time slot between the completion of the precedent task and the release of the succeeding task. If the source and destination tasks are mapped on the same PE, the data can be directly transmitted without spending additional latency. Otherwise, the minimum data transmission latency is required as expressed in Constraint (2.6).

$$s^e_{u,v} \geq f^u; \quad s^v \geq f^e_{u,v},\ \forall u, v \in T, \forall e_{u,v} \in E \tag{2.5}$$

$$\begin{cases} f^e_{u,v} \geq s^e_{u,v}, & \text{if } (x^u = x^v) \wedge (y^u = y^v) \wedge (z^u = z^v) \\ f^e_{u,v} \geq s^e_{u,v} + t_r + t_w \cdot M^e, & \text{otherwise} \end{cases}$$

$$,\ \forall u, v \in T, \forall e_{u,v} \in E \tag{2.6}$$

Non-overlap of data transmission

No pairs of data transmissions on edges $e_i, e_j \in E$ can overlap in time and space at the same time.

$$(\mathbf{at} - \mathbf{most1})\left(\alpha_{e_i,e_j}, \beta_{e_i,e_j}\right),\ \forall e_i, e_j \in E \tag{2.7}$$

Overlap of data transmission in time

Constraint (2.8) determines the overlap of data transmissions on any pairs of edges $e_i, e_j \in E$ in time.

$$\begin{cases} \alpha_{e_i,e_j} = true, & \text{if } (s^{e_i}_{u_i,v_i} < f^{e_j}_{u_j,v_j}) \wedge (f^{e_i}_{u_i,v_i} > s^{e_j}_{u_j,v_j}) \\ \alpha_{e_i,e_j} = false, & \text{otherwise} \end{cases}$$

$$, \forall u, v \in T, \forall e_i, e_j \in E \qquad (2.8)$$

Overlap of data transmission in space

The overlap of data transmissions in space is determined by the shared routing paths between any pairs of edges $e_i, e_j \in E$. Since we assume bi-directional links, only the shared links in the same direction are detected as an overlap. Constraint (2.9) determines the horizontal and vertical sharing of 2D routing paths under XY-routing. Constraint (2.10) determines the sharing of 3D routing paths under XYZ-routing. For simplification, detailed constraints for detecting overlaps in the x and y directional links that are the same as the Constraint (2.9) are not expressed in Constraint (2.10).

$$
\begin{cases}
\beta_{e_i,e_j} = true, & \text{if } \big\{(y^{u_i} = y^{u_j}) \wedge \\
& \big([(x^{u_i} \leq x^{u_j}) \wedge (x^{v_i} > x^{u_j}) \wedge (x^{v_j} > x^{u_j})] \vee [(x^{u_i} \leq x^{u_j}) \wedge (x^{v_i} < x^{u_i}) \wedge (x^{v_j} < x^{u_i})] \vee \\
& [(x^{u_i} \geq x^{u_j}) \wedge (x^{v_i} > x^{u_i}) \wedge (x^{v_j} > x^{u_i})] \vee [(x^{u_i} \geq x^{u_j}) \wedge (x^{v_i} < x^{u_j}) \wedge (x^{v_j} < x^{u_j})]\big)\big\} \\
& \vee \big\{(x^{v_i} = x^{v_j}) \wedge \\
& \big([(y^{v_i} \leq y^{v_j}) \wedge (y^{u_i} > y^{v_j}) \wedge (y^{u_j} > y^{v_j})] \vee [(y^{v_i} \leq y^{v_j}) \wedge (y^{u_i} < y^{v_i}) \wedge (y^{u_j} < y^{v_i})] \vee \\
& [(y^{v_i} \geq y^{v_j}) \wedge (y^{u_i} > y^{v_i}) \wedge (y^{u_j} > y^{v_i})] \vee [(y^{v_i} \geq y^{v_j}) \wedge (y^{u_i} < y^{v_j}) \wedge (y^{u_j} < y^{v_j})]\big)\big\} \\
\beta_{e_i,e_j} = false, & \text{otherwise}
\end{cases}
$$

$$
, \forall u, v \in T, \forall e_i, e_j \in E \tag{2.9}
$$

$$
\begin{cases}
\beta_{e_i,e_j} = true, & \text{if } \big\{(y^{u_i} = y^{u_j}) \wedge (z^{u_i} = z^{u_j}) \wedge \big(\text{Constraints for detecting shared links in } x\text{-direction}\big)\big\} \\
& \vee \big\{(x^{v_i} = x^{v_j}) \wedge (z^{u_i} = z^{u_j}) \wedge \big(\text{Constraints for detecting shared links in } y\text{-direction}\big)\big\} \\
& \vee \big\{(x^{v_i} = x^{v_j}) \wedge (z^{v_i} = z^{v_j}) \wedge \\
& \big([(z^{v_i} \leq z^{v_j}) \wedge (z^{u_i} > z^{v_j}) \wedge (z^{u_j} > z^{v_j})] \vee [(z^{v_i} \leq z^{v_j}) \wedge (z^{u_i} < z^{v_i}) \wedge (z^{u_j} < z^{v_i})] \vee \\
& [(z^{v_i} \geq z^{v_j}) \wedge (z^{u_i} > z^{v_i}) \wedge (z^{u_j} > z^{v_i})] \vee [(z^{v_i} \geq z^{v_j}) \wedge (z^{u_i} < z^{v_j}) \wedge (z^{u_j} < y^{v_j})]\big)\big\} \\
\beta_{e_i,e_j} = false, & \text{otherwise}
\end{cases}
$$

$$
, \forall u, v \in T, \forall e_i, e_j \in E \tag{2.10}
$$

Non-overlap of data transmission on the same PE

No pairs of out-going data transmissions from the same PE with different source tasks on edges $e_i, e_j \in E$ can overlap.

$$(s^{e_i}_{u_i,v_i} \geq f^{e_j}_{u_j,v_j}) \vee (f^{e_i}_{u_i,v_i} \leq s^{e_j}_{u_j,v_j}),$$

$$\text{if } (x^{u_i} = x^{u_j}) \wedge (y^{u_i} = y^{u_j}) \wedge (z^{u_i} = z^{u_j}) \wedge (s^{e_i}_{u_i,v_i} \neq f^{e_i}_{u_i,v_i}) \wedge (s^{e_j}_{u_j,v_j} \neq f^{e_j}_{u_j,v_j})$$

$$, \quad u_i \neq u_j, \forall u, v \in T, \forall e_i, e_j \in E \tag{2.11}$$

Maximum number of hops (HPC_{max})

The number of hops between any source/destination pairs of tasks $u, v \in T$ have to be less than or equal to HPC_{max}. Manhattan distance is used to estimate the number of hops between two consequential tasks as expressed in Constraint (2.12).

$$|x^u - x^v| + |y^u - y^v| + |z^u - z^v| \leq HPC_{max},\ \forall u, v \in T, \forall e_{u,v} \in E \tag{2.12}$$

2.3.2 Scalability Improvements

Adaptive Boundary Condition

The adaptive boundary condition reduces the search-space by narrowing the feasible time-ranges of task release-time variables. For each task, we set the low-bound as the maximum sum of task execution times on the paths from any starting tasks (i.e., tasks with no incoming edges) to the target task. For example, in Figure 2.3, the low-bound of $T4$ is defined as the sum of $T1$ and $T3$'s execution times. To set the upper-bound, we first respectively define feasible solution boundaries f_{min} and f_{max} as the sum of task execution times on the longest path and the sum of f_{min} and the offset. The offset is empirically determined by examining the maximum gap between f_{min} and the estimated maximum upper-bound of each test case[1]. Note that the offset needs to be increased if there exists a specific test case that has the larger actual upper-bound than f_{min}+offset (i.e., infeasible condition). Since our goal is minimizing the application schedule

[1] In this work, we estimate the maximum upper-bound of each test case as the sum of all task execution time divided by the minimum number of PEs in our experiments (i.e., 16 for 4×4 mesh) assuming full PE resource utilization with direct data transmissions.

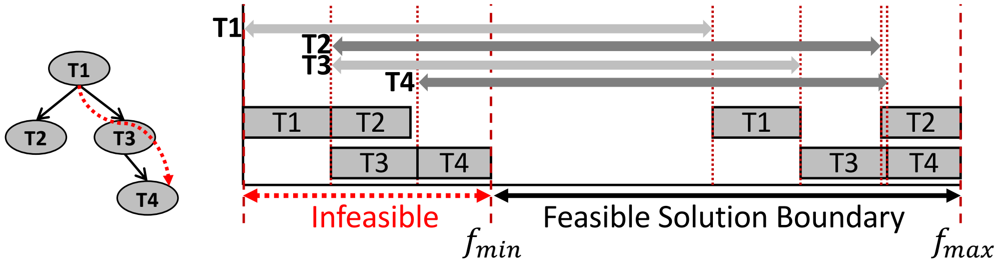


Figure 2.3: Example of adaptive boundary condition.

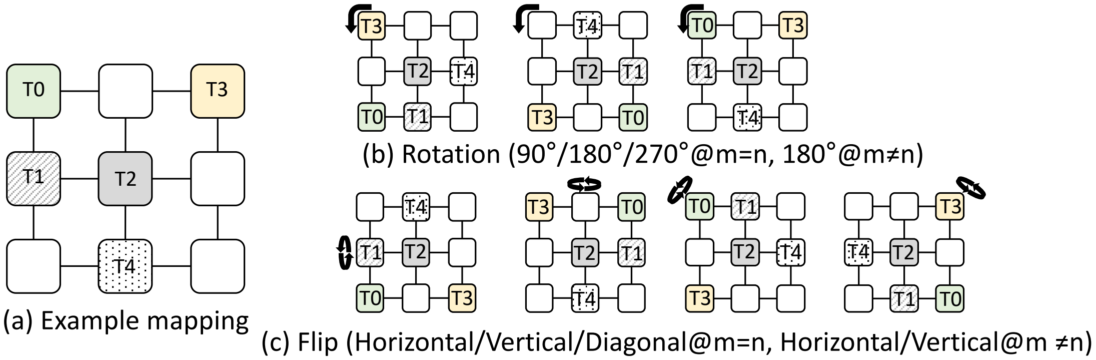
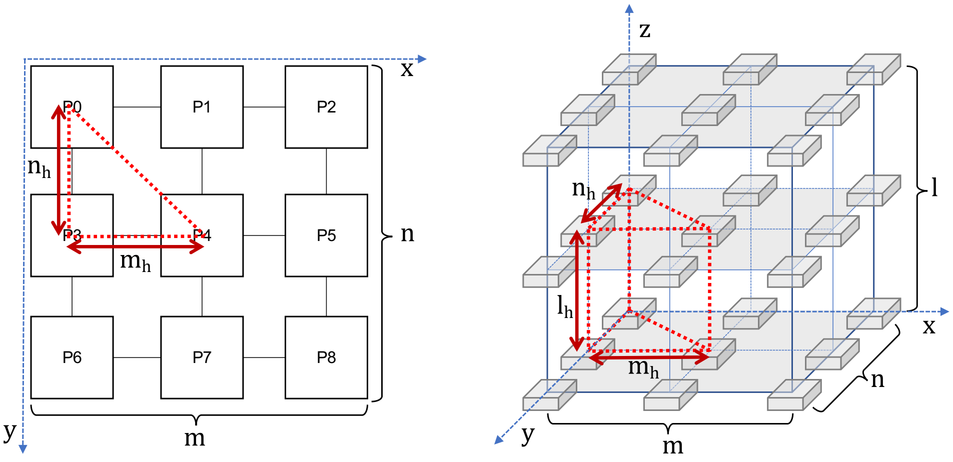

T0
T3
T1
T2
T4
(a) Example mapping
(b) Rotation (90°/180°/270°@m=n, 180°@m≠n)
(c) Flip (Horizontal/Vertical/Diagonal@m=n, Horizontal/Vertical@m ≠n)

Figure 2.4: Example of the symmetric task mappings on 2D $m \times n$ topology.

length, the offset does not affect the optimality of solutions if there exist any feasible solution with the given offset. In this work, we set the offset to 500 satisfying all the experimental cases. Once the f_{min} and the offset are determined, the upper-bound of each task is defined as a subtraction of the maximum sum of task execution times on the reversed paths from any finishing tasks (i.e., tasks with no outgoing edges) to the target task from f_{max}. For example, the upper-bound of $T1$ in Figure 2.3 is the sum of $T4$, $T3$, and $T1$'s execution times.

Breaking Design Symmetry

The breaking design symmetry constraint excludes redundant exploration of the symmetric solutions by restricting PE assignments of tasks to the specific region, resulting in the reduction of search-space. Figure 2.4 depicts examples of symmetric task mapping patterns in an $m \times n$ 2D mesh topology. The example mapping in Figure 2.4(a) has several symmetric task mappings that are equivalent to their rotated (i.e., Figure 2.4(b)) and flipped shapes (i.e., Figure 2.4(c)). In an



(a) 2D mesh topology graph (3X3)

(b) 3D mesh topology graph (3X3X3)

Figure 2.5: Restriction of PE assignments for excluding symmetric mapping patterns on 2D/3D topologies.

Algorithm 1 Exclusion of symmetric task mapping cases for 2D/3D mesh topology

Input: a task graph $TG = (T, E)$, # of tasks N_t, a SMART NoC with $m \times n$ / $m \times n \times l$ tiles for 2D / 3D
Data: a sorted queue of tasks in T as descending number of in/out-degree: T_s, a k_{th} task in T_s: T_k^s
/ (x, y) / (x, y, z) coordinate of the upper-left corner of 2D / 3D NoC tiles are* (0, 0) */* (0, 0, 0) **/*

```
$m_h \leftarrow \lceil \frac{m}{2} \rceil - 1$, $n_h \leftarrow \lceil \frac{n}{2} \rceil - 1$, $l_h \leftarrow \lceil \frac{l}{2} \rceil - 1$ (for 3D only)
for $k = 1$ to $N_t - 1$ do
    if $k$ = 1 then
        $\left(x^{T_1^s} \leq \left\lceil \frac{m_h}{n_h} \cdot y^{T_1^s} \right\rceil\right) \wedge (y^{T_1^s} \leq n_h)$;          ▷ Set Constraint for 2D only
        $\left(x^{T_1^s} \leq \left\lceil \frac{m_h}{n_h} \cdot y^{T_1^s} \right\rceil\right) \wedge (y^{T_1^s} \leq n_h) \wedge (z^{T_1^s} \leq l_h)$;          ▷ Set Constraint for 3D only
    else
        if $(m = n) \wedge \left(\bigwedge_{i=1}^{k-1} (x^{T_i^s} = y^{T_i^s}) = true\right)$ then
            $x^{T_k^s} \leq y^{T_k^s}$;          ▷ Set Conditional Constraint
        end if
        if $\left(\bigwedge_{i=1}^{k-1} (x^{T_i^s} = m_h) = true\right) \wedge (m = odd)$ then
            $x^{T_k^s} \leq m_h$;          ▷ Set Conditional Constraint
        end if
        if $\left(\bigwedge_{i=1}^{k-1} (y^{T_i^s} = n_h) = true\right) \wedge (n = odd)$ then
            $y^{T_k^s} \leq n_h$;          ▷ Set Conditional Constraint
        end if
        if $\left(\bigwedge_{i=1}^{k-1} (z^{T_i^s} = l_h) = true\right) \wedge (l = odd)$ then
            $z^{T_k^s} \leq l_h$;          ▷ Set Conditional Constraint for 3D only
        end if
    end if
end for
```

asymmetric topology (i.e., $m \neq n$), the symmetric cases are limited to the rotation in 180° and flip in horizontal/vertical direction.

To exclude symmetric mapping patterns, we first sort task elements by descending order of the number of incoming/outgoing edges of each task so that the following symmetry breaking constraint can reduce as many search-space of related tasks as possible. Then, we set the PE assignment boundary condition for the first task element to be assigned to the specific region of the topology as illustrated in Figure 2.5. For the remaining task elements, we recursively set the conditional constraints to keep excluding symmetric cases even if previously constrained tasks are mapped to the PEs on a line or plane that can cut the entire topology in half. Algorithm 1 describes conditional constraints excluding symmetric task mapping cases for 2D/3D mesh topology. Given a sorted queue T_s of tasks in T as descending number of in/out-degree, we first set the boundary condition for the PE assignment of the first task element (i.e., the task with the largest in/out-degree) T_1^s to the 4^{th} octant (Line 4-5). Then, we recursively set conditional constraints for the next task elements according to the intermediate PE assignments of previous task elements. If previous task elements are assigned on a diagonal line or plane in the topology with symmetric XY-plane (i.e., $m = n$), the location of the next task elements is restricted to the half area divided by the diagonal line or plane (Lines 7-9). When previous task elements are on one of horizontal line or plane (i.e., x,y, and z directions) dividing the entire topology in half, the location of the next task elements is restricted to the half area divided by the horizontal line or plane (Lines 10-18).

2.3.3 Mixed 2D/3D dimension-order routing

Overlap of data transmission in space for 2D mixed routing

For 2D NoC, we allow the mixed-use of XY/YX routing paths. Routing type indicator r^e is determined for each edge so that the application schedule length is minimized. r^e is set to 1 if the determined routing type is XY-routing. Otherwise (i.e., YX-routing), r^e is set to 0. Constraint (2.13) determines the horizontal and vertical sharing of routing paths under mixed XY/YX-routing.

$$\begin{cases} \beta_{e_i,e_j} = true, & \text{if } \big\{ \big([(r^{u_i}=0) \wedge (r^{u_j}=0) \wedge (y^{v_i}=y^{v_j})] \vee [(r^{u_i}=0) \wedge (r^{u_j}=1) \wedge (y^{v_i}=y^{u_j})] \vee \\ & [(r^{u_i}=1) \wedge (r^{u_j}=0) \wedge (y^{u_i}=y^{v_j})] \vee [(r^{u_i}=1) \wedge (r^{u_j}=1) \wedge (y^{u_i}=y^{u_j})] \big) \wedge \\ & \big(\text{Constraints for detecting shared links in } x\text{-direction in Constraint (2.9)}\big) \big\} \\ & \vee \big\{ \big([(r^{u_i}=0) \wedge (r^{u_j}=0) \wedge (x^{u_i}=x^{u_j})] \vee [(r^{u_i}=0) \wedge (r^{u_j}=1) \wedge (x^{u_i}=x^{v_j})] \vee \\ & [(r^{u_i}=1) \wedge (r^{u_j}=0) \wedge (x^{v_i}=x^{u_j})] \vee [(r^{u_i}=1) \wedge (r^{u_j}=1) \wedge (x^{v_i}=x^{v_j})] \big) \wedge \\ & \big(\text{Constraints for detecting shared links in } y\text{-direction in Constraint (2.9)}\big) \big\} \\ \beta_{e_i,e_j} = false, & \text{otherwise} \end{cases}$$

$$, \forall u, v \in T, \forall e_i, e_j \in E \tag{2.13}$$

Overlap of data transmission in space for 3D mixed routing

For 3D NoC, we allow the mixed-use of XYZ/YXZ, ZXY/ZYX, and XZY/YZX routing paths. The routing type indicator r^e is respectively set as 0 to 5 for XYZ, YXZ, ZXY, ZYX, XZY, and YZX routing paths. Constraint (2.14) determines the sharing of links under mixed routing paths. Detailed constraints in the y and z directional links that can be defined similar to the x-directional constraint are not described in the Constraint (2.14).

$$
\begin{cases}
\beta_{e_i,e_j} = true, \\
\text{if } \Big\{ \Big([(r^{u_i}=0) \wedge \{((r^{u_j}=0) \wedge (y^{u_i}=y^{u_j}) \wedge (z^{u_i}=z^{u_j})) \vee ((r^{u_j}=1) \wedge (y^{u_i}=y^{v_j}) \wedge (z^{u_i}=z^{u_j})) \vee \\
\qquad\qquad ((r^{u_j}=2) \wedge (y^{u_i}=y^{u_j}) \wedge (z^{u_i}=z^{v_j})) \vee ((r^{u_j}=3) \wedge (y^{u_i}=y^{u_j}) \wedge (z^{u_i}=z^{u_j})) \vee \\
\qquad\qquad ((r^{u_j}=4) \wedge (y^{u_i}=y^{v_j}) \wedge (z^{u_i}=z^{v_j})) \vee ((r^{u_j}=5) \wedge (y^{v_i}=y^{v_j}) \wedge (z^{u_i}=z^{u_j}))\}] \vee \\
\quad [(r^{u_i}=1) \wedge \{((r^{u_j}=0) \wedge (y^{v_i}=y^{u_j}) \wedge (z^{u_i}=z^{u_j})) \vee ((r^{u_j}=1) \wedge (y^{v_i}=y^{v_j}) \wedge (z^{u_i}=z^{u_j})) \vee \\
\qquad\qquad ((r^{u_j}=2) \wedge (y^{v_i}=y^{u_j}) \wedge (z^{u_i}=z^{v_j})) \vee ((r^{u_j}=3) \wedge (y^{v_i}=y^{v_j}) \wedge (z^{u_i}=z^{v_j})) \vee \\
\qquad\qquad ((r^{u_j}=4) \wedge (y^{v_i}=y^{u_j}) \wedge (z^{u_i}=z^{u_j})) \vee ((r^{u_j}=5) \wedge (y^{v_i}=y^{v_j}) \wedge (z^{u_i}=z^{v_j}))\}] \vee \\
\quad [(r^{u_i}=2) \wedge \{((r^{u_j}=0) \wedge (y^{u_i}=y^{u_j}) \wedge (z^{v_i}=z^{u_j})) \vee ((r^{u_j}=1) \wedge (y^{u_i}=y^{v_j}) \wedge (z^{v_i}=z^{u_j})) \vee \\
\qquad\qquad ((r^{u_j}=2) \wedge (y^{u_i}=y^{u_j}) \wedge (z^{v_i}=z^{v_j})) \vee ((r^{u_j}=3) \wedge (y^{u_i}=y^{v_j}) \wedge (z^{v_i}=z^{v_j})) \vee \\
\qquad\qquad ((r^{u_j}=4) \wedge (y^{u_i}=y^{u_j}) \wedge (z^{v_i}=z^{u_j})) \vee ((r^{u_j}=5) \wedge (y^{u_i}=y^{v_j}) \wedge (z^{v_i}=z^{v_j}))\}] \vee \\
\quad [(r^{u_i}=3) \wedge \{((r^{u_j}=0) \wedge (y^{v_i}=y^{u_j}) \wedge (z^{v_i}=z^{u_j})) \vee ((r^{u_j}=1) \wedge (y^{v_i}=y^{v_j}) \wedge (z^{v_i}=z^{u_j})) \vee \\
\qquad\qquad ((r^{u_j}=2) \wedge (y^{v_i}=y^{u_j}) \wedge (z^{v_i}=z^{v_j})) \vee ((r^{u_j}=3) \wedge (y^{v_i}=y^{v_j}) \wedge (z^{v_i}=z^{v_j})) \vee \\
\qquad\qquad ((r^{u_j}=4) \wedge (y^{v_i}=y^{u_j}) \wedge (z^{v_i}=z^{u_j})) \vee ((r^{u_j}=5) \wedge (y^{v_i}=y^{v_j}) \wedge (z^{v_i}=z^{v_j}))\}] \vee \\
\quad [(r^{u_i}=4) \wedge \{((r^{u_j}=0) \wedge (y^{u_i}=y^{u_j}) \wedge (z^{u_i}=z^{u_j})) \vee ((r^{u_j}=1) \wedge (y^{u_i}=y^{v_j}) \wedge (z^{u_i}=z^{u_j})) \vee \\
\qquad\qquad ((r^{u_j}=2) \wedge (y^{u_i}=y^{u_j}) \wedge (z^{u_i}=z^{v_j})) \vee ((r^{u_j}=3) \wedge (y^{u_i}=y^{v_j}) \wedge (z^{u_i}=z^{v_j})) \vee \\
\qquad\qquad ((r^{u_j}=4) \wedge (y^{u_i}=y^{u_j}) \wedge (z^{u_i}=z^{u_j})) \vee ((r^{u_j}=5) \wedge (y^{u_i}=y^{v_j}) \wedge (z^{u_i}=z^{v_j}))\}] \vee \\
\quad [(r^{u_i}=5) \wedge \{((r^{u_j}=0) \wedge (y^{v_i}=y^{u_j}) \wedge (z^{v_i}=z^{u_j})) \vee ((r^{u_j}=1) \wedge (y^{v_i}=y^{v_j}) \wedge (z^{v_i}=z^{u_j})) \vee \\
\qquad\qquad ((r^{u_j}=2) \wedge (y^{v_i}=y^{u_j}) \wedge (z^{v_i}=z^{v_j})) \vee ((r^{u_j}=3) \wedge (y^{v_i}=y^{v_j}) \wedge (z^{v_i}=z^{v_j})) \vee \\
\qquad\qquad ((r^{u_j}=4) \wedge (y^{v_i}=y^{u_j}) \wedge (z^{v_i}=z^{u_j})) \vee ((r^{u_j}=5) \wedge (y^{v_i}=y^{v_j}) \wedge (z^{v_i}=z^{v_j}))\}] \Big) \wedge \\
\Big(\text{Detecting shared links in } x\text{-direction}\Big) \Big\} \vee \Big\{\text{Detecting shared links in } y\text{-direction in mixed routing}\Big\} \\
\vee \Big\{\text{Detecting shared links in } z\text{-direction in mixed routing}\Big\} \\
\beta_{e_i,e_j} = false, \text{otherwise}
\end{cases}
$$

$$, \forall u, v \in T, \forall e_i, e_j \in E \qquad (2.14)$$

2.4 Experimental Results

2.4.1 Experimental Setup

We have implemented the proposed framework in both *ILP/SMT* formulas including (i) the basic formulations and (ii) the scalability improvement constraints for 2D topology. Then, we have extended SMT framework to 3D topology and implemented mixed dimension-order routing scheme for both 2D/3D frameworks. Our frameworks are validated on a workstation with Intel Xeon E5-2650L at 1.8GHz and 128GB memory. The *Gurobi* (version 9.0.2) [40] and *Z3* (version 4.8.5) [20] solvers are used to produce the optimized solutions for ILP and SMT, respectively.

We employ applications from (i) randomly generated cases by TGFF tool [5] and (ii) real benchmarks, including MWD (multi-window display), H263 encoder/MP3 decoder, H263 decoder/MP3 decoder, MP3 encoder/decoder, MMS(multi-media system) [41], Robot (Newton-Euler dynamic control calculation), Sparse (Random sparse matrix solver), and RS-32 encoder (Reed-Solomon code encoder) [42]. We consider 4×4, 6×6, 8×4, 8×8, and 16×16 2D Mesh and 4×4×4 and 8×8×4 3D Mesh for 2D/3D SMART NoC architecture and assume the homogeneous PEs with the same execution efficiency. The random applications are generated with the maximum in/out-degree of 2/2 or 2/3 (suffixed with "_a"). We use HPC$_{max}$ = 8 that is the best achievable for 2D configurations when energy is taken into consideration in [29]. HPC_{max} may affect the resource utilization if the number of outgoing edges from a task exceeds the number of available PEs within HPC_{max} (i.e., $4 \cdot \sum_{k=1}^{HPC_{max}}$), resulting in the diminished parallelism and performance. In our experiments, the maximum number of outgoing edges from one task is not restricted by HPC_{max} = 8 for all cases.

Table 2.2: Formulation complexity of ILP and SMT on 2D/3D SMART NoC. $\lvert T\rvert$ = #Tasks, $\lvert E\rvert$ = #Edges, #Var (Aux) = # of auxiliary variables.

Main Variables	#Variables (ILP/SMT 2D/3D)
Tasks	$4\cdot\lvert T\rvert$ (2D) / $5\cdot\lvert T\rvert$ (3D)
Data Transmissions	$2\cdot\lvert E\rvert$
Overlap Flags	$\lvert E\rvert^2-\lvert E\rvert$

Constraints	ILP 2D		ILP 3D		SMT 2D/3D	
	#Var (Aux)	#Constraints	#Var (Aux)	#Constraints	#Var (Aux)	#Constraints
Timing (tasks)	-	$\lvert E\rvert+\lvert T\rvert$	-	$\lvert E\rvert+\lvert T\rvert$	-	$\lvert E\rvert+\lvert T\rvert$
Non-overlap (tasks)	$\frac{7}{2}\cdot(\lvert T\rvert^2-\lvert T\rvert)$	$5\cdot(\lvert T\rvert^2-\lvert T\rvert)$	$5\cdot(\lvert T\rvert^2-\lvert T\rvert)$	$\frac{13}{2}\cdot(\lvert T\rvert^2-\lvert T\rvert)$		$\frac{1}{2}\cdot(\lvert T\rvert^2-\lvert T\rvert)$
Timing (data trans.)	$7\cdot\lvert E\rvert$	$16\cdot\lvert E\rvert$	$10\cdot\lvert E\rvert$	$19\cdot\lvert E\rvert$		$4\cdot\lvert E\rvert$
Non-overlap (data trans.)	-	$\frac{1}{2}\cdot(\lvert E\rvert^2-\lvert E\rvert)$	-	$\frac{1}{2}\cdot(\lvert E\rvert^2-\lvert E\rvert)$		$\frac{1}{2}\cdot(\lvert E\rvert^2-\lvert E\rvert)$
Overlap in time	$\lvert E\rvert^2-\lvert E\rvert$	$3\cdot(\lvert E\rvert^2-\lvert E\rvert)$	$\lvert E\rvert^2-\lvert E\rvert$	$3\cdot(\lvert E\rvert^2-\lvert E\rvert)$		$\frac{1}{2}\cdot(\lvert E\rvert^2-\lvert E\rvert)$
Overlap in space	$\frac{43}{2}\cdot(\lvert E\rvert^2-\lvert E\rvert)$	$\frac{81}{2}\cdot(\lvert E\rvert^2-\lvert E\rvert)$	$\frac{67}{2}\cdot(\lvert E\rvert^2-\lvert E\rvert)$	$\frac{123}{2}\cdot(\lvert E\rvert^2-\lvert E\rvert)$		$\frac{1}{2}\cdot(\lvert E\rvert^2-\lvert E\rvert)$
Non-overlap on same PE	$\frac{13}{2}\cdot(\lvert E\rvert^2-\lvert E\rvert)$	$\frac{21}{2}\cdot(\lvert E\rvert^2-\lvert E\rvert)$	$8\cdot(\lvert E\rvert^2-\lvert E\rvert)$	$12\cdot(\lvert E\rvert^2-\lvert E\rvert)$		$\frac{1}{2}\cdot(\lvert E\rvert^2-\lvert E\rvert)$
Breaking symmetry	$4\cdot\lvert T\rvert$	$6\cdot\lvert T\rvert$	$4\cdot\lvert T\rvert$	$6\cdot\lvert T\rvert$		$\lvert T\rvert$

Total	ILP 2D	ILP 3D	SMT
#Variables (Main + Aux)	$\frac{7}{2}\cdot\lvert T\rvert^2+30\cdot\lvert E\rvert^2+\frac{9}{2}\cdot\lvert T\rvert-21\cdot\lvert E\rvert$	$5\cdot\lvert T\rvert^2+45\cdot\lvert E\rvert^2+4\cdot\lvert T\rvert-30\cdot\lvert E\rvert$	$\lvert E\rvert^2+\lvert E\rvert+4\cdot\lvert T\rvert$
#Constraints	$5\cdot\lvert T\rvert^2+\frac{109}{2}\cdot\lvert E\rvert^2+2\cdot\lvert T\rvert-\frac{75}{2}\cdot\lvert E\rvert$	$\frac{13}{2}\cdot\lvert T\rvert^2+\frac{157}{2}\cdot\lvert E\rvert^2+\frac{1}{2}\cdot\lvert T\rvert-\frac{117}{2}\cdot\lvert E\rvert$	$\frac{1}{2}\cdot\lvert T\rvert^2+2\cdot\lvert E\rvert^2+\frac{3}{2}\cdot\lvert T\rvert+3\cdot\lvert E\rvert$

Table 2.3: Formulation complexity evaluation on 2D/3D SMART NoC. inc. = increment ratio (ref. = SMT)

TestCase	Tasks	Edges	2D							3D						
			#Variables				#Constraints			#Variables				#Constraints		
			ILP		SMT	**inc.**	ILP	SMT	**inc.**	ILP		SMT	**inc.**	ILP	SMT	**inc.**
			Total	Aux						Total	Aux					
tgff1	10	9	1,675	1,545	130	**12.9**×	2,776	256	**10.8**×	2,474	2,334	140	**17.7**×	3,962	256	**15.5**×
tgff2	22	24	10,345	9,657	688	**15.0**×	17,358	1,485	**11.7**×	15,714	15,004	710	**22.1**×	25,578	1,485	**17.2**×
tgff3	31	34	20,512	19,198	1,314	**15.6**×	34,406	2,918	**11.8**×	31,158	29,813	1,345	**23.2**×	50,672	2,918	**17.4**×
tgff4	41	43	33,826	31,770	2,056	**16.5**×	56,653	4,702	**12.0**×	51,952	49,855	2,097	**24.8**×	84,655	4,702	**18.0**×
tgff5	51	62	64,808	60,698	4,110	**15.8**×	109,401	9,210	**11.9**×	100,078	95,917	4,161	**24.1**×	164,093	9,210	**17.8**×
tgff20	201	245	1,013,036	951,962	61,074	**16.6**×	1,712,204	141,183	**12.1**×	1,592,759	1,531,484	61,275	**26.0**×	2,624,757	141,183	**18.6**×
tgff35	351	435	3,171,862	2,980,798	191,064	**16.6**×	5,363,446	441,700	**12.1**×	4,995,915	4,804,500	191,415	**26.1**×	8,237,709	441,700	**18.7**×
tgff50	501	606	6,224,082	5,854,236	369,846	**16.8**×	10,519,070	862,296	**12.2**×	9,823,519	9,453,172	370,347	**26.5**×	16,202,741	862,296	**18.8**×
tgff3_a	30	36	22,077	20,625	1,452	**15.2**×	37,295	3,182	**11.7**×	34,197	32,715	1,482	**23.1**×	56,169	3,180	**17.7**×
tgff4_a	40	47	38,049	35,633	2,416	**15.7**×	64,204	5,399	**11.9**×	59,298	56,842	2,456	**24.1**×	97,458	5,399	**18.1**×
tgff5_a	51	62	64,728	60,618	4,110	**15.7**×	109,552	9,225	**11.9**×	101,388	97,227	4,161	**24.4**×	167,174	9,225	**18.1**×
tgff10_a	102	118	239,515	225,065	14,450	**16.6**×	403,236	33,474	**12.0**×	369,642	355,090	14,552	**25.4**×	604,757	33,474	**18.1**×
MWD	12	12	2,683	2,479	204	**13.2**×	4,534	411	**11.0**×	4,112	3,896	216	**19.0**×	6,766	411	**16.5**×
H263encMP3dec	12	12	2,667	2,463	204	**13.1**×	4,483	410	**10.9**×	3,969	3,753	216	**18.4**×	6,452	412	**15.7**×
MP3encMP3dec	13	13	3,148	2,914	234	**13.5**×	5,323	477	**11.2**×	4,853	4,606	247	**19.6**×	8,000	479	**16.7**×
H263decMP3dec	14	14	3,637	3,371	266	**13.7**×	6,147	547	**11.2**×	5,553	5,273	280	**19.8**×	9,130	549	**16.6**×
MMS	40	48	38,809	36,297	2,512	**15.4**×	66,001	5,602	**11.8**×	61,450	58,898	2,552	**24.1**×	102,030	5,600	**18.2**×
Robot	88	131	263,483	245,839	17,644	**14.9**×	449,888	38,620	**11.6**×	421,139	403,407	17,732	**23.8**×	700,969	38,620	**18.2**×
Sparse	96	67	104,507	99,567	4,940	**21.2**×	164,117	13,907	**11.8**×	151,931	146,895	5,036	**30.2**×	229,848	13,927	**16.5**×
RS-32_28_8_enc	262	348	1,990,333	1,867,833	122,500	**16.2**×	3,354,633	277,795	**12.1**×	3,187,636	3,064,874	122,762	**26.0**×	5,254,697	277,795	**18.9**×
Average						**16.6**×			**12.1**×				**26.2**×			**18.7**×

2.4.2 ILP vs. SMT for 2D/3D SMART NoC

Formulation complexity analysis

Table 2.2 presents the formulation complexity of ILP/SMT frameworks for 2D/3D SMART NoCs. The number of variables and constraints is significantly related to the num-

Table 2.4: Minimum application schedule length and simulation runtime of ILP and SMT on 2D SMART NoC. Min. l_{app} = minimum application schedule length. Spd.Up = average runtime speed up ratio (ref. = ILP), t.o.=optimization not completed within 12 hours.

TestCase	Tasks	Edges	Min. l_{app}		Simulation Runtime(s)										
			ILP	**SMT**	ILP					SMT					**Spd.Up**
					4×4	6×6	8×4	8×8	**Avg.**	4×4	6×6	8×4	8×8	**Avg.**	
tgff1	10	9	**173**	**173**	0.37	0.73	0.16	0.21	**0.37**	0.06	0.07	0.07	0.07	**0.07**	**5.5**×
tgff2	22	24	**240**	**240**	53.21	163.94	96.11	42.84	**89.03**	1.46	1.77	1.27	1.66	**1.54**	**57.8**×
tgff3	31	34	**248**	**248**	746.28	524.37	1,332.02	2,221.27	**1,205.99**	4.03	4.62	4.21	4.38	**4.31**	**279.7**×
tgff4	41	43	**334**	**334**	10,400.94	3,396.40	12,181.23	10,608.73	**9,146.83**	9.75	10.09	10.10	10.04	**9.99**	**915.3**×
tgff5	51	62	**-**	**365**	t.o.	t.o.	t.o.	t.o.	-	18.29	18.79	20.53	18.35	**18.99**	**-**
tgff20	201	245	**-**	**764**						1,519.11	857.79	754.49	842.35	**993.44**	**-**
tgff35	351	435	**-**	**854**						t.o.	29,894.46	t.o.	6,209.88	**18,052.17**	**-**
tgff50	501	606	**-**	**903**							t.o.	t.o.	42,755.09	**42,755.09**	**-**
tgff3_a	30	36	**294**	**294**	9,241.54	3,119.83	7,450.77	6,977.98	**6,697.53**	5.63	6.21	6.37	6.27	**6.12**	**1094.1**×
tgff4_a	40	47	**366**	**366**	11,949.99	7,692.02	10,293.86	3,818.10	**8,438.49**	9.31	10.15	11.15	9.79	**10.10**	**835.6**×
tgff5_a	51	62	**449**	**449**	t.o.	t.o.	30,221.38	37,003.00	**33,612.19**	20.59	21.45	24.36	21.35	**21.94**	**1532.1**×
tgff10_a	102	118	**-**	**383**			t.o.	t.o.	-	414.41	125.62	71.44	145.88	**189.34**	**-**
MWD	12	12	**281**	**281**	9.34	11.09	19.93	16.69	**14.26**	0.84	0.77	0.85	0.79	**0.81**	**17.6**×
H263encMP3dec	12	12	**235**	**235**	0.76	0.94	0.62	0.62	**0.74**	0.31	0.39	0.30	0.35	**0.34**	**2.2**×
MP3encMP3dec	13	13	**251**	**251**	0.74	0.45	0.49	0.45	**0.53**	0.21	0.28	0.19	0.24	**0.23**	**2.3**×
H263decMP3dec	14	14	**219**	**219**	4.42	3.44	0.98	3.62	**3.12**	0.53	0.64	0.51	0.57	**0.56**	**5.5**×
MMS	40	48	**325**	**325**	417.33	254.15	281.47	487.79	**360.19**	7.44	7.83	9.09	7.48	**7.96**	**45.2**×
Robot	88	131	**-**	**617**	t.o.	t.o.	t.o.	t.o.	-	146.59	151.49	138.60	138.43	**143.78**	**-**
Sparse	96	67	**-**	**240**						10,366.52	16,721.70	15,150.02	38,713.89	**21,934.04**	**-**
RS-32_28_8_enc	262	348	**-**	**1704**						2,524.37	2,608.05	2,456.07	2,600.79	**2,547.32**	**-**
														Average	**931.1**×

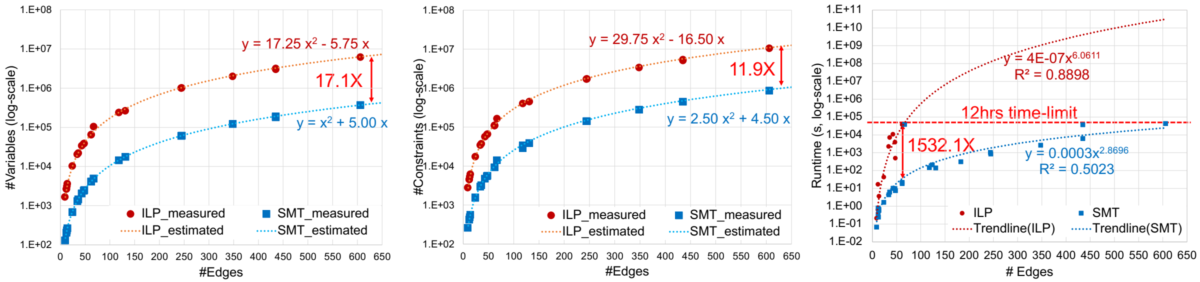
Table 2.5: Minimum application schedule length and simulation runtime of ILP and SMT on 3D SMART NoC. Min. l_{app} = minimum application schedule length. Spd.Up = average runtime speed up ratio (ref. = ILP), t.o.=optimization not completed within 12 hours.

TestCase	Tasks	Edges	Solution		Simulation Runtime (s)						
			ILP	SMT	ILP			SMT			**Spd.Up**
					4×4×4	8×8×4	**Avg.**	4×4×4	8×8×4	**Avg.**	
tgff1	10	9	173	173	0.16	1.50	**0.83**	0.10	0.11	**0.11**	**7.9**×
tgff2	22	24	240	240	150.03	76.24	**113.14**	1.23	1.85	**1.54**	**73.5**×
tgff3	31	34	248	248	5,874.83	2,260.49	**4,067.66**	4.18	4.78	**4.48**	**908**×
tgff4	41	43	334	334	12,673.58	11,948.77	**12,311.18**	10.49	11.22	**10.86**	**1134.1**×
tgff5	51	62	-	365	t.o.	t.o.	**-**	32.80	30.43	**31.62**	**-**
tgff20	201	245	-	764	t.o.	t.o.	**-**	1,077.30	1,060.85	**1,069.08**	**-**
tgff35	351	435	-	854	t.o.	t.o.	**-**	5,229.47	4,872.55	**5,051.01**	**-**
tgff50	501	606	-	903	t.o.	t.o.	**-**	36,580.00	20,005.61	**28,292.81**	**-**
tgff3_a	30	36	294	294	28,641.77	35,916.15	**32,278.96**	6.61	8.15	**7.38**	**4373.8**×
tgff4_a	40	47	366	366	6,493.93	9,944.42	**8,219.18**	11.01	12.26	**11.64**	**706.4**×
tgff5_a	51	62	-	449	t.o.	t.o.	**-**	32.20	33.41	**32.81**	**-**
tgff10_a	102	118	-	383	t.o.	t.o.	**-**	112.70	123.90	**118.30**	**-**
MWD	12	12	281	281	14.87	21.74	**18.31**	0.74	0.95	**0.85**	**21.7**×
H263encMP3dec	12	12	235	235	0.95	1.26	**1.11**	0.23	0.29	**0.26**	**4.3**×
MP3encMP3dec	13	13	251	251	1.49	0.96	**1.23**	0.22	0.24	**0.23**	**5.3**×
H263decMP3dec	14	14	219	219	1.28	4.42	**2.85**	0.58	0.84	**0.71**	**4**×
MMS	40	48	325	325	851.10	822.47	**836.79**	8.12	9.33	**8.73**	**95.9**×
Robot	88	131	-	617	t.o.	t.o.	**-**	167.08	163.21	**165.15**	**-**
Sparse	96	67	-	228	t.o.	t.o.	**-**	167.43	120.81	**144.12**	**-**
RS-32_28_8_enc	262	348	-	1703	t.o.	t.o.	**-**	2,510.11	3,396.16	**2,953.14**	**-**
Average											**1237.1**×

ber of tasks $|T|$ and edges $|E|$. The main variables of our framework consist of x/y-coordinates, release/completion time of each task/data transmission, and overlap indicators in time/space. The coordinate and time variables are proportional to $|T|$ and $|E|$ while the overlap indicators are related to the number of combinations between edges (i.e., $\frac{|E|\cdot(|E|-1)}{2}$). The conditional constraints which describe the overlap between pairs of tasks and edges are proportional to the number of combinations of tasks and edges. ILP requires auxiliary variables and the corresponding constraints for conditional constraints. In particular, as the dimension of the structure increases from 2D to 3D, the number of auxiliary variables and constraints in ILP also shows an increment because of the more complicated conditional constraints for determining the overlap in tasks, data transmissions, whereas the SMT keeps the same formulation complexity. The number of additional variables and constraints is represented as the multiplication of the number of corresponding SMT constraints. Note that the final estimated complexity of ILP is further reduced by around 50% across the benchmarks because we remove the duplicated literals in conditional constraints.

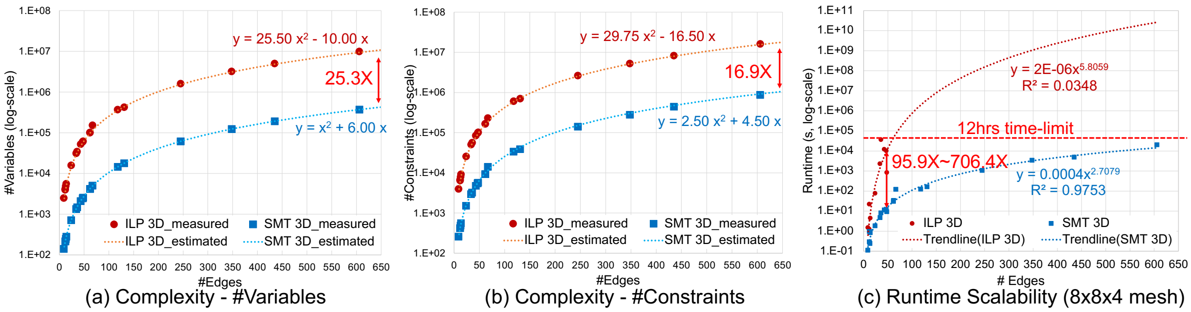
Evaluation - formulation complexity

Table 2.3 presents the comparison of formulation complexity between ILP and SMT for 2D and 3D structures. Compared to ILP, SMT has 16.6× and 12.1× smaller number of variables and constraints on average for 2D, respectively. For 3D, SMT respectively shows 26.2× and 18.7× smaller number of variables and constraints. The difference is mainly due to the auxiliary variables which occupy 92% to 95% of total variables and the corresponding constraints of ILP.



(a) Complexity - #Variables (b) Complexity - #Constraints (c) Runtime Scalability (8x8 mesh)

Figure 2.6: Comparison of the formulation complexity and runtime scalability (8×8 mesh) between ILP and SMT for 2D SMART NoC.



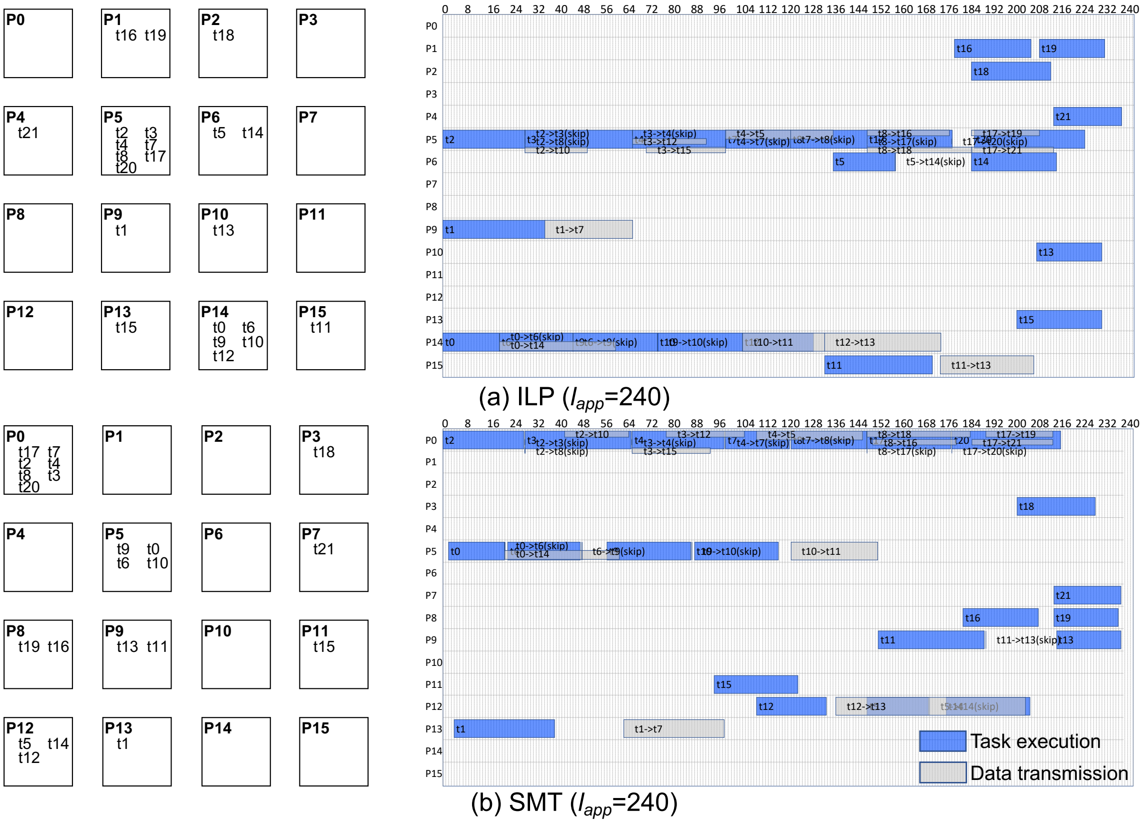
(a) Complexity - #Variables (b) Complexity - #Constraints (c) Runtime Scalability (8x8x4 mesh)

Figure 2.7: Comparison of the formulation complexity and runtime scalability (8×8×4 mesh) between ILP and SMT for 3D SMART NoC.

Figure 2.6(a), 2.6(b) and Figure 2.7(a), 2.7(b) visualize the estimated and measured complexity of ILP and SMT for $|E|$ in 2D and 3D structures, respectively. Note that we assume the same $|T|$ and $|E|$ for the estimation. As $|T|$ and $|E|$ increase, the estimated number of variables and constraints in ILP has respectively saturated to 17.1×/11.9× and 25.3×/16.9× larger values than those of SMT for 2D and 3D.

Evaluation - solutions and runtime

Table 2.4 and Table 2.5 present the comparison of the minimum application schedule length and runtime between ILP and SMT with 12 hours of the time-limit for 2D and 3D SMART NoC, respectively. We observe that the application schedule length of both ILP and SMT are the same (i.e., equal application performance) for all cases regardless of mesh sizes. Figure 2.8 illustrates examples of detailed task mapping and scheduling solutions provided by ILP and SMT. The detailed solution includes information on the PE assignment and a designated release time for each task. Though the detailed composition of task mapping and scheduling of these solutions can be different from each other due to the possible existence of several feasible solutions, both ILP and SMT provided the same minimum application schedule length of 240. The runtime trend tends to increase as the size of the mesh and the number of in/out-degree increases. For the cases that the ILP can provide an optimal solution, SMT solves problems with 931.1× (ranges from 2.2× to 1532.1×) and 1237.1× (ranges from 4× to 4373.8×) faster runtime on average than ILP



(a) ILP (l_{app}=240)

(b) SMT (l_{app}=240)

Figure 2.8: Example task mapping and scheduling of tgff2 case in Table 2.4 for 4×4 mesh. (a) SMT, (b) ILP

for 2D and 3D NoCs, respectively. Figure 2.6(c) and Figure 2.7(c), which respectively visualize the comparison of the scalability for 8×8 and 8×8×4 meshes, show that SMT achieves 10× higher scalability up to 500 tasks than that of ILP up to 50 tasks within 12 hours. For the largest case in ILP (i.e., tgff5_a with 8×8 mesh and tgff4_a, MMS with 8×8×4 mesh), SMT provides the solution up to 1532.1× faster than ILP.

2.4.3 2D vs. 3D SMART NoC

In Section 2.4.2, we have demonstrated the superior scalability of SMT over ILP for 2D and 3D SMART NoC structures. Therefore, we use only the SMT framework for implementing a mixed dimension-order routing in the following experiments because the scalability of ILP is

Table 2.6: Minimum application schedule length and simulation runtime on 2D/3D SMART NoC. t.o. = optimization not completed within 12 hours.

TestCase	#Tasks	#Edges	Minimum Application Schedule Length (l_{app})								Simulation Runtime (s)							
			2D (XY Only)		2D (mixed)		3D (XYZ Only)		3D (mixed)		2D (XY Only)		2D (mixed)		3D (XYZ Only)		3D (mixed)	
			8×8	16×16	8×8	16×16	4×4×4	8×8×4	4×4×4	8×8×4	8×8	16×16	8×8	16×16	4×4×4	8×8×4	4×4×4	8×8×4
tgff5	51	62	365	365	365	365	365	365	365	365	18.35	37.06	30.22	31.06	32.80	30.43	59.97	59.60
tgff10	101	124	664	664	664	664	664	664	664	664	206.32	261.02	212.92	246.56	225.21	208.07	444.76	523.50
tgff20	201	245	764	764	764	764	764	764	764	764	842.35	1,302.41	1,260.40	1,294.33	1,077.28	1,060.85	1,978.02	2,140.43
tgff30	301	369	770	770	770	770	770	770	770	770	7,339.49	7,106.80	6,910.83	7,057.29	6,212.25	6,819.45	7,625.03	7,345.88
tgff40	401	495	818	818	818	818	818	818	818	818	15,961.35	14,131.35	30,817.41	22,887.27	21,563.24	16,004.14	33,160.45	22,708.54
tgff50	501	606	903	903	-	903	903	903	-	903	42,755.09	24,422.54	t.o.	34,112.05	36,579.98	20,005.61	t.o.	23,687.59
tgff5_a	51	62	449	449	**440**	**440**	449	449	**440**	**440**	21.35	43.95	28.09	23.02	32.19	33.41	49.52	61.47
tgff10_a	102	118	383	383	383	383	383	383	383	383	145.88	159.12	129.38	137.02	112.74	123.90	195.40	224.38
tgff20_a	203	244	492	492	492	492	492	492	492	492	1,025.82	1,655.52	1,260.40	1,319.78	1,206.36	1,193.70	2,301.11	2,275.93
tgff30_a	301	359	528	528	-	528	528	528	528	528	35,734.00	9,799.09	t.o.	5,982.25	9,876.86	5,628.88	13,287.17	7,407.61
MWD	12	12	281	281	281	281	281	281	281	281	0.79	1.93	1.25	2.12	0.74	0.95	3.20	3.42
H263encMP3dec	12	12	235	235	235	235	235	235	235	235	0.35	0.55	0.70	0.58	0.23	0.29	0.83	0.96
MP3encMP3dec	13	13	251	251	251	251	251	251	251	251	0.24	0.58	0.33	0.33	0.22	0.24	0.46	0.90
H263decMP3dec	14	14	219	219	219	219	219	219	219	219	0.57	0.99	1.02	0.79	0.58	0.84	0.68	1.42
MMS	40	48	325	325	325	325	325	325	325	325	7.48	17.36	15.55	14.61	8.12	9.33	29.39	37.07
Robot	88	131	617	617	**615**	**615**	617	617	**615**	**615**	138.43	273.21	133.66	193.48	167.08	163.21	215.62	248.07
Sparse	96	67	240	-	-	-	**228**	**228**	**222**	**222**	38,713.89	t.o.	t.o.	t.o.	167.43	120.81	1,338.58	2,047.77
RS-32_28_8_enc	262	348	1704	1704	1704	1704	1704	1704	1704	1704	2,600.79	3,734.78	2,672.55	4,862.26	2,510.11	3,396.16	2,957.08	4,464.72
Average			**556.0**	**574.6**	**573.9**	**573.9**	**555.3**	**555.3**	**533.9**	**554.4**	**8,084.03**	**3,702.84**	**6,332.97**	**4,597.93**	**4,431.86**	**3,044.46**	**3,743.96**	**4,068.85**

expected to be decreased for the mixed routing which involves more complicated conditional constraints as described in Section 2.3.3. 8×8/16×16 and 4×4×4/8×8×4 mesh topologies are used for 2D and 3D structures, respectively.

Table 2.6 presents the comparison of minimum application schedule length and simulation runtime of 2D and 3D SMART NoCs. Unlike the hop-by-hop transmission-based regular NoCs, l_{app} of SMART NoC is not affected by the reduced average number of hops by the extension to the 3D routing structure. Therefore, most of cases in Table 2.6 have the same l_{app} for 2D XY-only and 3D XYZ-only routing NoCs except for the random sparse matrix solver (i.e., "Sparse") case. Figure 2.9 depicts a task mapping solution of "Sparse" case in 4×4×4 3D mesh. The arrow lines in the black and red colors illustrate incoming data transmissions from six PEs (i.e., P3, P17, P20, P22, P28, and P45) to P21 at a certain clock cycle in the detailed scheduling solution. The black and red arrow lines respectively indicate the transmissions on the same and different XY-planes. From this solution, we observe that the extended incoming paths up to six (i.e., four from the same and two from the different XY planes) on each PE by the extension of 2D to 3D routing paths enables further reduction of the minimum l_{app} from 240 to 228.

The simulation runtime of 3D XYZ-only routing shows a reduced trend compared to 2D

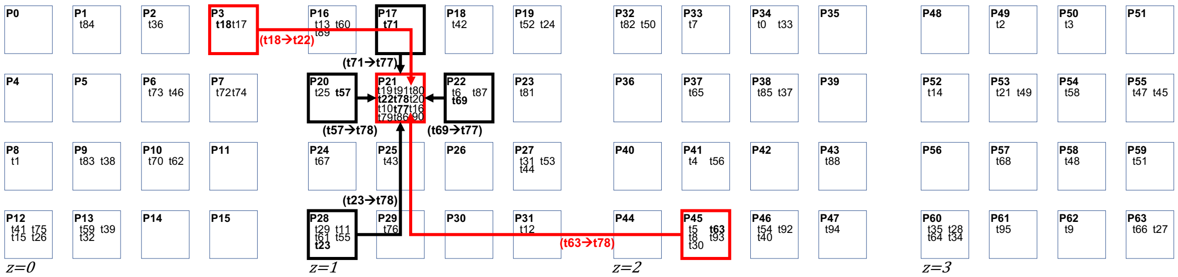
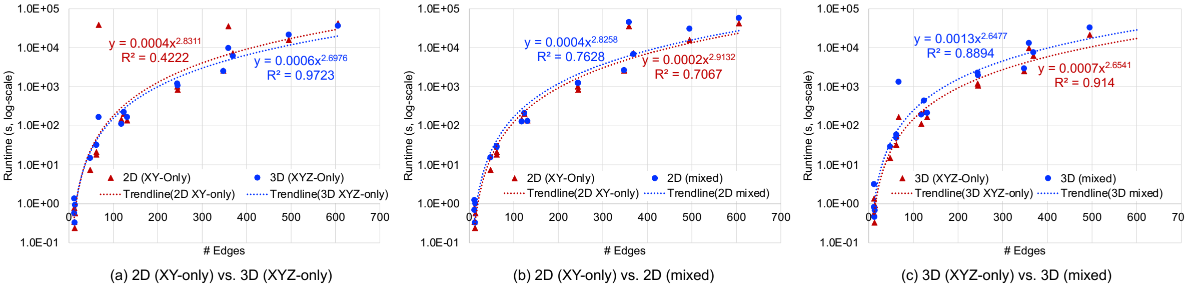


Figure 2.9: Example of the extension of routing paths from 2D to 3D routing (Sparse case in Table 2.4, l_{app} : $240@2D \rightarrow 228@3D$).



(a) 2D (XY-only) vs. 3D (XYZ-only)

(b) 2D (XY-only) vs. 2D (mixed)

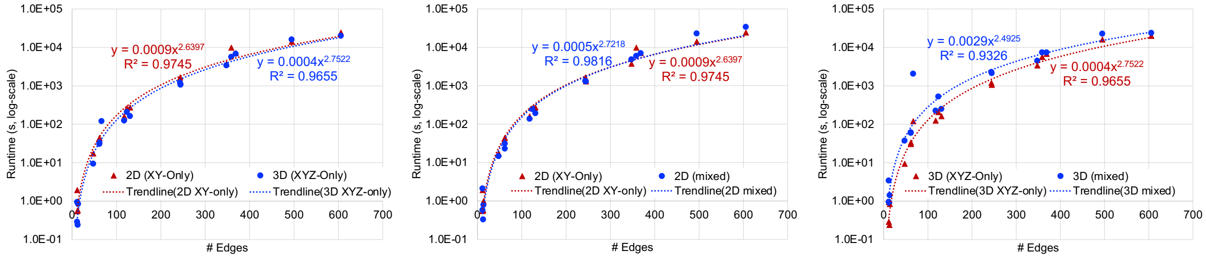
(c) 3D (XYZ-only) vs. 3D (mixed)

Figure 2.10: Runtime scalability (2D@8×8 mesh, 3D@4×4×4 mesh). (a) 2D (XY-only) vs. 3D (XYZ-only), (b) 3D (XYZ-only) vs 3D (mixed).

XY-only routing as depicted in Figure 2.10(a) and Figure 2.11(a). We observe that the "tgff30_a" and "Sparse" cases require extraordinary simulation runtime to find an optimal solution in 8×8 2D topology while the runtime of their 3D counterparts follows the normal trend. Note that the TGs of these two cases have many more parallel task routing paths compared to the other randomly generated or real application TGs as shown in Figure 2.12. This high-level task parallelism can cause a larger fluctuation of runtime due to the existence of multiple combinations of symmetric paths with the same application schedule length as well as the nature of the exact method to solve NP-hard problems [43].

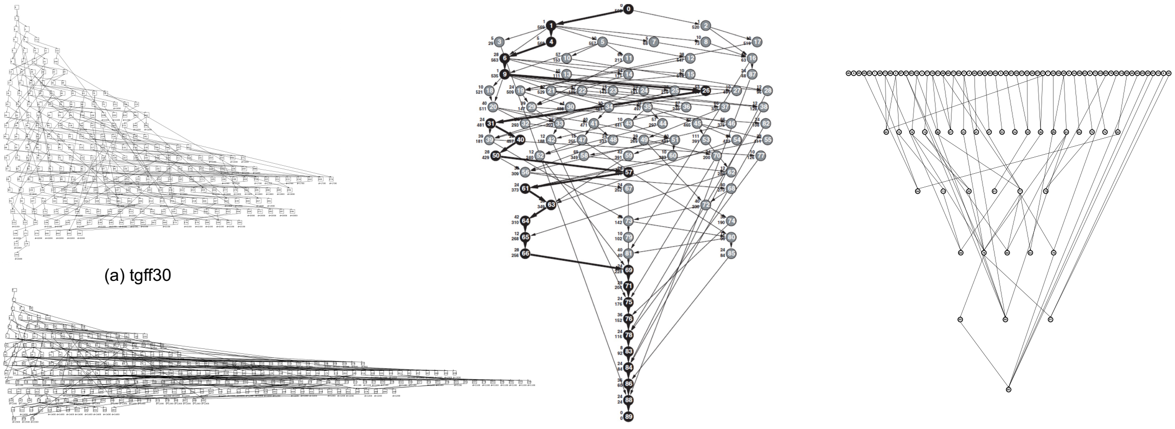
2.4.4 Mixed Dimension-Order Routing

In this section, we explore the impact of the mixed dimension-order routing in both 2D/3D topologies presented in Table 2.6. Three cases (i.e., "tgff5_a", "Robot", and "Sparse") show



(a) 2D (XY-only) vs. 3D (XYZ-only)   (b) 2D (XY-only) vs. 2D (mixed)   (c) 3D (XYZ-only) vs. 3D (mixed)

Figure 2.11: Runtime scalability (2D@16×16 mesh, 3D@8×8×4 mesh). (a) 2D (XY-only) vs. 3D (XYZ-only), (b) 3D (XYZ-only) vs 3D (mixed).



(a) tgff30

(b) tgff30_a   (c) Robot (Newton-Euler method for Stanford manipulator)   (d) Sparse (random sparse matrix solver)

Figure 2.12: Task graph of (a) tgff30, (b) tgff30_a, (c) Robot, and (d) Sparse cases in Table 2.4 [5, 6]

the reduction in the minimum application schedule length due to the diversified routing paths of mixed dimension-order routing. Figure 2.13 illustrates the example of path diversification in the "Sparse" case with 4×4×4 3D mesh. The arrow lines in the black and red colors illustrate incoming data transmissions from 5 PEs (i.e., P9, P20, P25, P33, and P55) to P21 at a certain clock cycle in the detailed scheduling solution. The red arrow lines indicate two data transmissions with different routing paths between the same source/destination PEs. From this solution, we observe that the diversified paths enable further reduction of the minimum l_{app} from 228 to 222.

The overall runtime scalabilities of the 2D/3D mixed routing show decreased trends compared to the 2D/3D single routing as depicted in Figure 2.10(b), (c) and Figure 2.11(b), (c). The additional variables for the routing path type indicator, the more complicated conditional

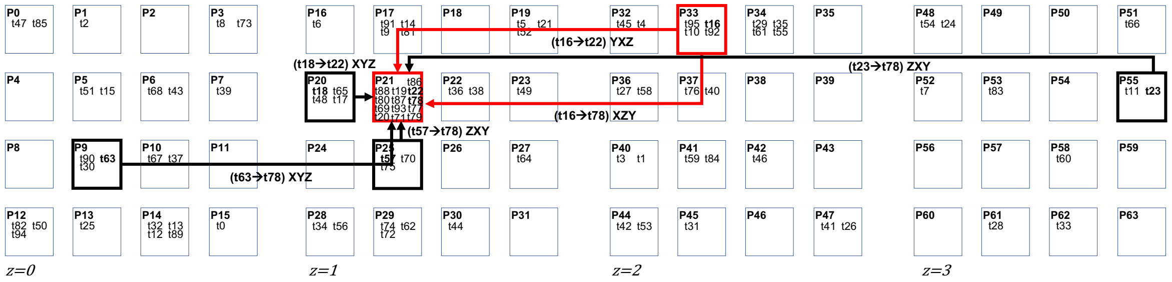


Figure 2.13: Example of path diversification from 3D XYZ-only to mixed routing (Sparse case in Table 2.4, $l_{app} : 228@XYZ \rightarrow 222@mixed$).

constraints for detecting the link overlap in space, and the increased search-space due to the diversified routing paths have contributed to the increment in the simulation runtime. However, despite the slightly diminished scalability, the results demonstrate that our framework still successfully finds optimal task mapping and scheduling solutions up to 500 tasks within 12 hours.

2.5 Related Work

The problem of task mapping and scheduling has been extensively studied in the literature [44, 45, 46, 47, 48, 49, 50]. Traditional approaches to task mapping [46, 47, 48, 49, 50] do not consider SMART NoCs where contention-free routing is required to fully realized the benefits of single-cycle multi-hop traversal. An optimal algorithm based on an integer linear programming (ILP) formulation was proposed in [31] for the 2D SMART NoC case. Although their ILP formulation can find optimal solutions, the runtimes are prohibitive for large problem instances. This is in part due to ILP's inability to express conditional constraints directly. On the other hand, our SMT-based formulation enables us to leverage SMT's ability to expresss conditional constraints succinctly, which enables us to derive a much more compact formulation and harness the logical reasoning power of SMT solvers. SMT's ability to capture conditional constraints also enables us to easily consider the 3D SMART NoC case and mixed dimension-order routing in our formulations, which were not considered in [31]. A polynomial-time heuristic algorithm was also proposed in [31] for 2D SMART NoCs. Although their heuristic algorithm often achieves good results, exploration of optimal solutions still plays an essential role in calibrating and evaluating heuristic approaches for more advanced and complicated system configurations. Recently, SMT-based scheduling optimization frameworks for 2D NoCs have been proposed [51, 52] to overcome the limited expressiveness of ILP for the conditional constraints that constitute a large proportion of the task mapping and scheduling problem, but these works did not consider mixed dimension-order routing or the 3D SMART NoC case.

2.6 Conclusion

In this chapter, we develop an SMT-based task mapping and scheduling framework that guarantees contention-free data transmissions to achieve the optimal latency for 2D/3D SMART NoCs. Also, we develop link overlap detection constraints for the mixed dimension-order routing. We have reduced the formulation complexity by utilizing SMT's efficient modeling capability for the conditional constraints and also improved the scalability by introducing efficient search-space reduction techniques. We demonstrated that our SMT framework achieves 10× higher scalability than ILP, solving the problem within 12 hours up to 500 tasks for 2D and the 3D extension. Also, the 2D and 3D extensions of our SMT framework with the mixed dimension-order routing maintain the improved scalability with the diversified routing paths, resulting in the reduced latency through various application benchmarks. Lastly, we find that there are still rooms to further improve, e.g., the static task execution and data transmission time calls future research topics to accommodate the variability of real systems.

This chapter contains materials from "SMT-Based Contention-Free Task Mapping and Scheduling on 2D/3D SMART NoC with Mixed Dimension-Order Routing", by Daeyeal Lee, Bill Lin, and Chung-Kuan Cheng, which appears in ACM Transactions on Architecture and Code Optimization, March 2022. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Standard Cell Synthesis: Simultaneous Placement & Routing

3.1 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 LELE (litho-etch-litho-etch), SADP (self-aligned double patterning), and SAQP (self-aligned quadruple patterning) [53]. As a result, highly increased layout-design complexity obstructs the prompt development of standard cell libraries for the efficient DTCO (design technology co-optimization) workflow [54]. To overcome the current limitation and improve PPAC (performance, power, area, and cost) trade-off, 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, 55, 56, 57]. 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 sub-problems 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” [58][59], “Branch and Bound” [60], “Transistor connection pruning” [61], etc. For in-cell routing problem, several approaches based on traditional “Maze Routing algorithm” [62][63] are suggested but inapplicable to modern multiple-patterning technologies because of the complex design rules. An SADP-aware routing solution is presented [64], and several pin-accessibility optimization techniques have attracted considerable attention to improve the pin-accessibility of standard cells in sub-7*nm* technology [64, 65, 66, 67]. However, these approaches which rely on solving sub-problems 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 [68][69] co-optimizing transistor-level placement and in-cell routing are published, however, these works are not suitable for the multiple-patterning technologies in sub-7*nm*. Recently, sub-7*nm* applicable automatic standard cell synthesis frameworks have been proposed [70, 71, 72], however, following sequential and heuristic approaches in place-and-route phase.

Satisfiability Modulo Theories (SMT)

Compared to SAT (Boolean satisfiability), SMT is a more expressive language containing non-Boolean variables (e.g., integer, bit-vector, etc.) and predicate symbols as described in [17]. Several SMT solvers including the optimization methodology (i.e., OMT) are recently released [18][19]. 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.* [73] propose an SMT-based automation framework that simultaneously solves the place-and-route 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 [7].

3.1.1 Our Contributions

In this paper, we propose a novel SMT-based framework that **S**imultaneously optimizes **P**lace-**&**-**R**oute (*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:

- We propose an automated standard cell synthesis framework, *SP&R*, which simultaneously solves place-and-route (P&R) optimization problems. We devise an innovative dynamic pin allocation (DPA) to integrate placement and routing steps into a single optimization procedure.

- 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.

- *SP&R* utilizes an SMT solver, capable of SAT-based fast reasoning with an OMT-featured (Optimization Modulo Theories) multi-objective optimization.

- *SP&R* covers a wide variety of conditional design rules for securing DFM (design for manufacturing), producing pin-accessibility-aware cell layouts.

- *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 (i) single diffusion breaks with a crossover and (ii) crosstalk mitigations of timing critical nets.

- *SP&R* achieves an average of 20.8× to 131.7× runtime improvements over that of reported in [73], by paying less than 0.2% degradation of the total metal length.

- We demonstrate that our framework *SP&R* successfully generates a whole set of 7*nm* standard cell library [7] with layouts, improving cell size and #*M*2 tracks by 0.1 CPP and 0.3 tracks on average compared to the known layouts, respectively.

The remaining sections are organized as follows. Section 3.2 introduces our framework's configuration. Section 3.3 describes constraint formulation for the simultaneous place-and-route multi-objective problem. Section 3.4 presents the scalability improvement techniques. Section 3.5 discusses our experimental setup/results. Section 3.6 concludes the paper.

3.2 Framework Preliminary

This section introduces an overview of the proposed *SP&R* framework, SMT (satisfiability modulo theories), multi-objective optimization, and target cell architecture.

3.2.1 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 place-and-route steps into a multi-objective optimization problem as shown in Fig. 3.1. We adopt the state-of-the-art *lazy*-approach SMT solver *Z3* [18][20] to solve the given optimization problem. Fig. 3.2 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 3.1.

3.2.2 SMT (Satisfiability Modulo Theories)

On top of efficient problem-solving ability of SAT, SMT provides the feature of OMT (Optimization Modulo Theories) [18][19] 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” and “at-least k”, etc.) than is possible

[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.

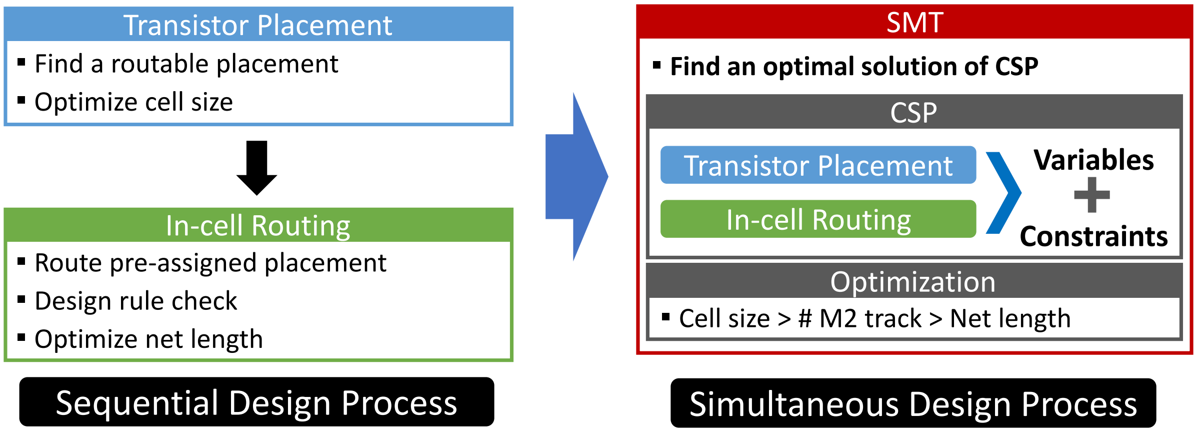

Transistor Placement
▪ Find a routable placement
▪ Optimize cell size
In-cell Routing
▪ Route pre-assigned placement
▪ Design rule check
▪ Optimize net length
Sequential Design Process
SMT
▪ Find an optimal solution of CSP
CSP
Transistor Placement
In-cell Routing
Variables
Constraints
Optimization
▪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

Figure 3.1: Sequential vs. our proposed simultaneous cell design processes.

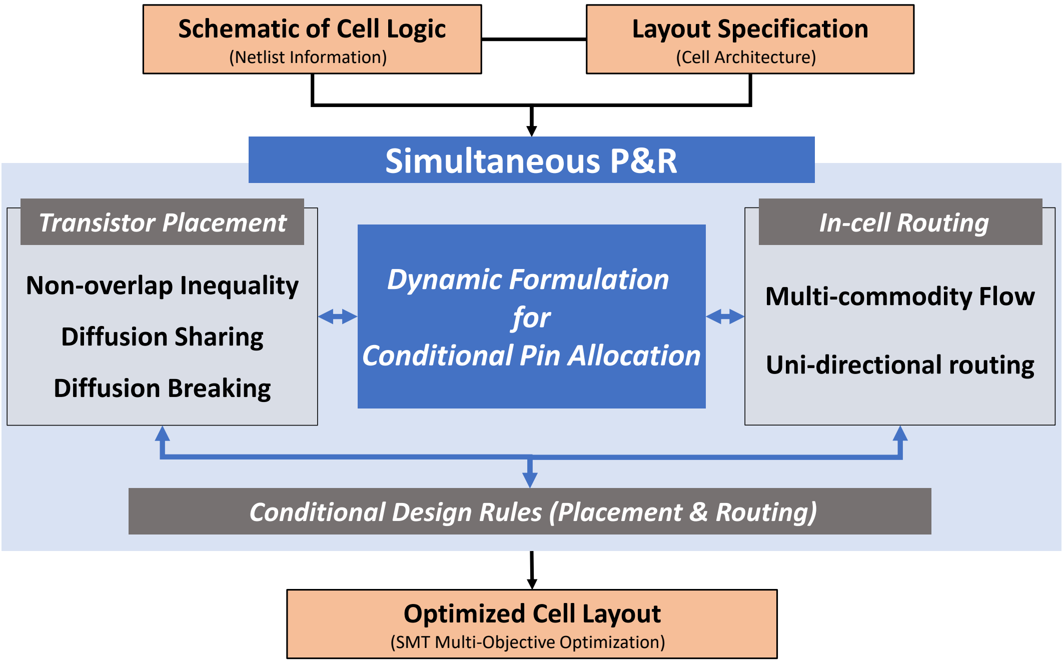


Figure 3.2: The proposed **S**imultaneous **P&R** framework.

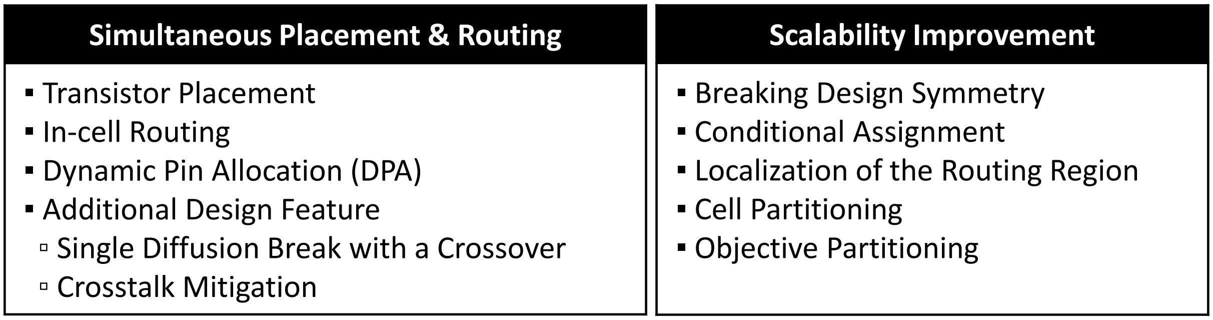
with SAT or ILP (Integer Linear Program) formulas. These key features of SMT efficiently accomplish exhaustive searching for the optimal solution with the concise expressions of constraints. Fig. 3.3 shows design constraints of our framework. Sections 3.3 and 3.4 respectively describe our methodology to develop SMT formulation of constraints for *SP&R* and our techniques to improve *SP&R*'s scalability.

Table 3.1: 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

3.2.3 Multi-Objective 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 FETs (field-effect transistors) as shown in Expression (3.1). 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 (Expression (3.2)). The total metal length (ML) is the weighted sum of the routed metal segments as shown in Expression (3.3). 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



Simultaneous Placement & Routing

- Transistor Placement
- In-cell Routing
- Dynamic Pin Allocation (DPA)
- Additional Design Feature
  - Single Diffusion Break with a Crossover
  - Crosstalk Mitigation

Scalability Improvement

- Breaking Design Symmetry
- Conditional Assignment
- Localization of the Routing Region
- Cell Partitioning
- Objective Partitioning

Figure 3.3: Design constraints of the proposed framework *SP&R*.

a more important metric than Total ML; (i) to enhance the cell's PPAC trade-off by suppressing the usage of higher metal layers and (ii) 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 [18][74]. The objectives can be ranked in the order of emphasized, as described in Expression (3.4).

$$\textbf{Placement (Cell Size)}: \max\left\{x_t + w_t \,\middle|\, t \in T\right\} \tag{3.1}$$

$$\textbf{Routability (\#}M2\textbf{ Track)}: \sum_{k=1}^{l} \bigvee_{e_{v,u}\in E_k} m_{v,u} \tag{3.2}$$

l=#Horizontal Tracks

E_k=Set of $M2$ Layer Edges in k_{th} Track

$$\textbf{Routing (Total ML)}: \sum_{e_{v,u}\in E} \left(w_{v,u} \times m_{v,u}\right) \tag{3.3}$$

$$\textbf{LexMin:}\ \text{(a) Cell Size, (b) \#}M2\text{ Track, (c) Total ML} \tag{3.4}$$

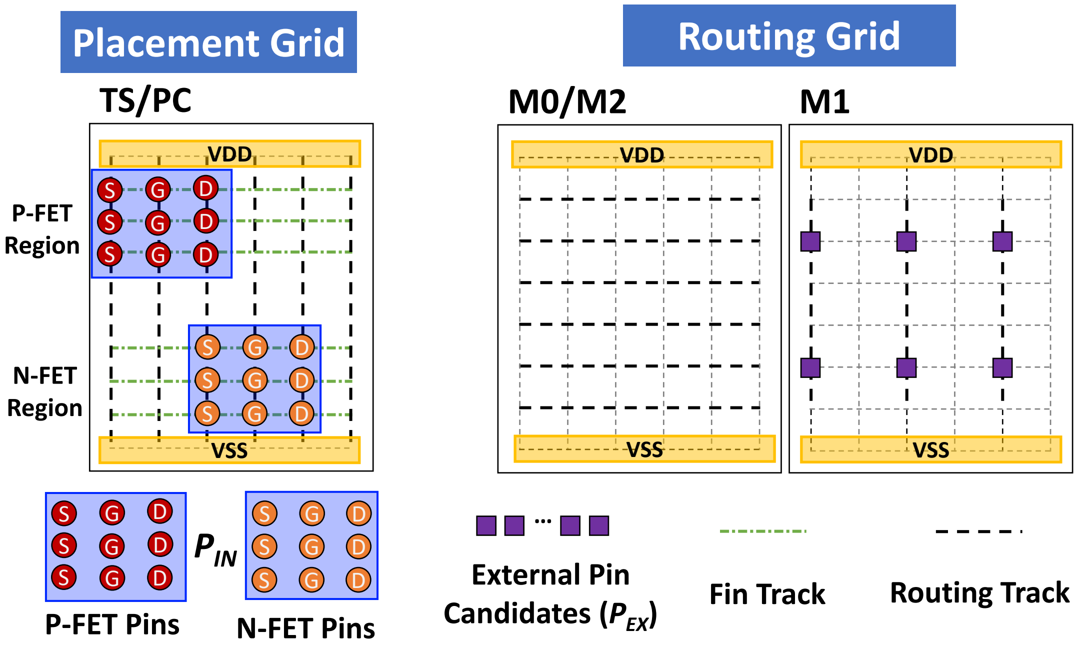


Figure 3.4: Grid-based placement & 3-D routing graph.

3.2.4 Cell Architecture

In this work, our framework considers 7*nm* standard cell architecture (e.g., the layer/track information) of [70][7] as depicted in Fig. 3.4. Inspired by [25][75][27], 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. 3.4. 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 Uni-Directional Routing

We assume on-grid and uni-directional routing scheme for each layer due to the process resolution of sub-7*nm* multi-patterning technologies and IC practitioners' restriction of preferred routing direction per each layer [24]. The preferred directions of $M0/2$ and $M1$ layers are horizontal and vertical, respectively, as illustrated in Fig. 3.4.

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. 3.4. The location of each pin is dynamically determined by placement formulation (Section 3.3.1) and is associated with the flow formulation for routing through our DPA (dynamic pin allocation) scheme (Section 3.3.3).

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. 3.4) 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 3.3.2). 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.

3.3 Simultaneous Placement & Routing

In this section, we describe our SMT formulation of the constraints for the proposed *SP&R* framework. This section consists of *(i) Transistor Placement*, *(ii) In-Cell Routing*, and *(iii)*

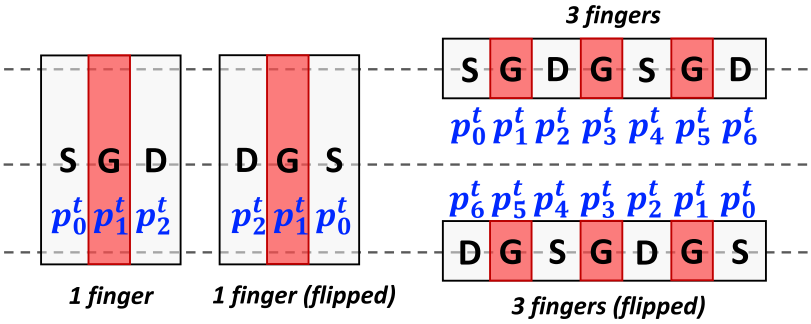


Figure 3.5: Configuration of a FET with size of 3.

Dynamic Pin Allocation (DPA).

3.3.1 Transistor Placement

FET Configuration

Fig. 3.5 illustrates an example of variable types of a 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. 3.5).

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 [76] because of the distortion during diffusion process for adjacent FETs. In a conventional physical design flow, these size transition within a standard

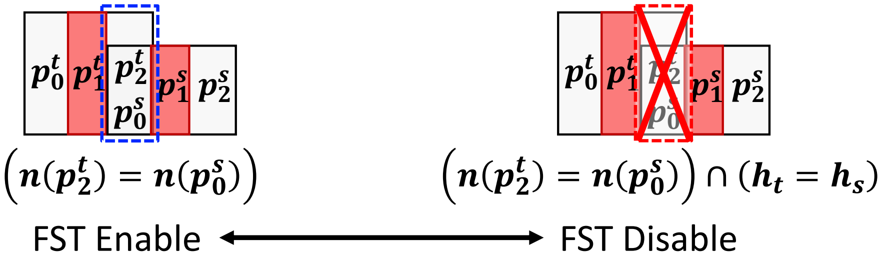

p_0^t
p_1^t
p_2^t
p_0^s
p_1^s
p_2^s
$\left(n(p_2^t) = n(p_0^s)\right)$
$\left(n(p_2^t) = n(p_0^s)\right) \cap (h_t = h_s)$
FST Enable
FST Disable

Figure 3.6: Diffusion sharing (DS) with the FET size transition (FST) option.

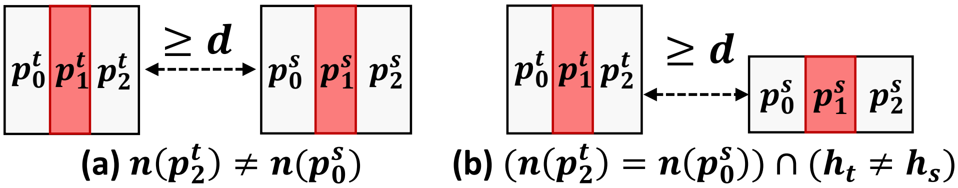

p_0^t
p_1^t
p_2^t
$\geq d$
p_0^s
p_1^s
p_2^s
(a) $n(p_2^t) \neq n(p_0^s)$
(b) $(n(p_2^t) = n(p_0^s)) \cap (h_t \neq h_s)$

Figure 3.7: Diffusion break (DB) (a) different net information, (b) different diffusion heights with FST disable.

cell is captured by the library characterization as the diffusion shapes are pre-determined. *SP&R* provides an optional FET size transition (FST) in diffusion sharing (DS) between FETs with different diffusion heights, supporting the library characterization in various process architectures. Fig. 3.6 depicts the DS rule according to the FST option. When the FST is disabled, DS is not allowed between FETs with different diffusion heights.

Diffusion Break (DB)

As shown in Fig. 3.7, 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 diffusion break (SDB) and double diffusion break (DDB). The minimum space *d* of SDB and DDB is 2 and 4, respectively.

Single Diffusion Break (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. 3.8(b) shows the example of a

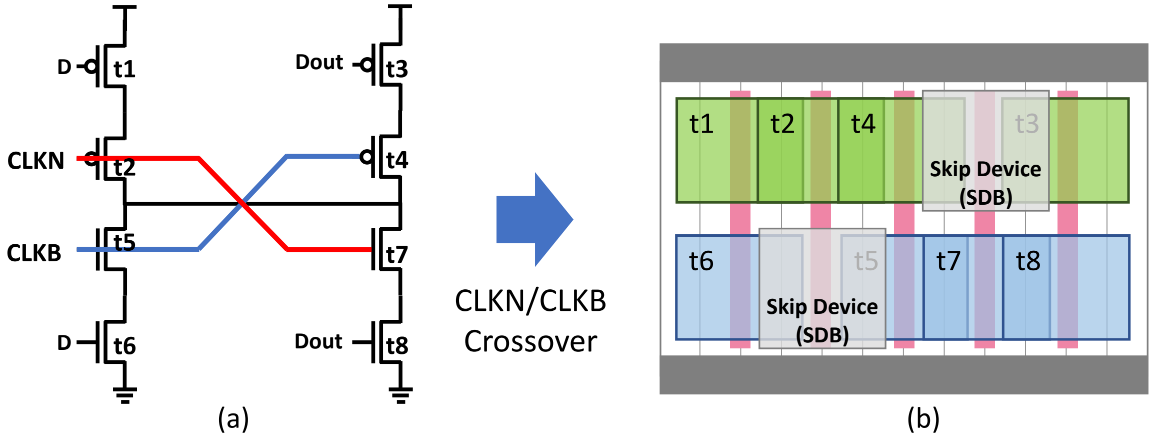


Figure 3.8: Example of SDB with a crossover (CLKN/CLKB).

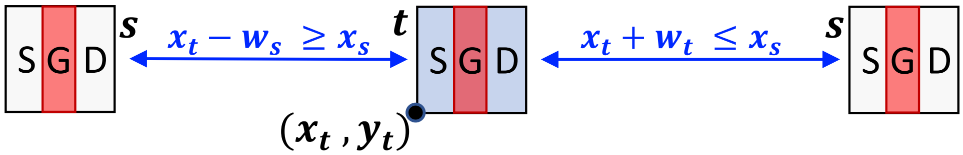


Figure 3.9: Relative positions between two FETs.

cell placement when “skip device” occurred by the crossover of CLKN and CLKB signals as shown in Fig. 3.8(a). When DDB (double diffusion break) is a major diffusion break, 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.

Relative Positioning Constraint (RPC)

We utilize the conventional floorplanning design approach (i.e., *Relative Positioning Constraint (RPC)*) for the transistor placement problem [77]. All transistor positions can be represented by two RPCs as shown in Fig. 3.9 because we only consider a single-stack placement. According to the input parameters which determine the type of DB, our *SP&R* calls sub-procedures defined in Algorithms 2-4 to set corresponding RPC formulation with DS and DB. Algorithm 2

Algorithm 2 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
```

sets the RPC for SDB when SDB is a major DB. Each RPC on the left/right 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 3. When FET t is on the right side of FET s ,also when FET t is sharing a diffusion on the left side (Lines 1-2), 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 3-4). Algorithm 4 represents the RPC with the mixed SDB in a crossover. Since *SP&R* minimizes the cell size, all the pairs of FETs which 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, 13) even though they meet the sharing conditions (Lines 4, 14).

Algorithm 3 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
if $x_t > x_s + w_s$ & $o_l^t$ then
    $x_t \geq x_s + w_s + 2$;
else if $x_t > x_s + w_s$ & $!o_l^t$ then
    $x_t \geq x_s + w_s + d_d$;
else if $x_t = x_s + w_s$ & $n(p_l^t) = n(p_r^s)$ & (FST | (!FST & $h_t$ = $h_s$)) then
    $x_t = x_s + w_s$;
else if $x_t + w_t < x_s$ & $o_r^t$ then
    $x_t + w_t + 2 \leq x_s$;
else if $x_t + w_t < x_s$ & $!o_r^t$ then
    $x_t + w_t + d_d \leq x_s$;
else if $x_t + w_t = x_s$ & $n(p_r^t) = n(p_l^s)$ & (FST | (!FST & $h_t$ = $h_s$)) then
    $x_t + w_t = x_s$;
else
    Unsatisfiable Condition;
end if
```

3.3.2 In-Cell Routing

We adopt conditional design rule-aware multi-commodity network flow theory to formulate the in-cell routing problem as described in [73][25, 75, 27]. Specifically, the refined constraints for *commodity flow conservation (CFC)* and *vertex exclusiveness (VE)* in uni-directional edges [73][27] 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. 3.10. 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.

```
Algorithm 4 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
if $x_t > x_s + w_s$ & $o_l^t$ then
    $x_t \geq x_s + w_s + 2$;
else if $x_t > x_s + w_s$ & $!o_l^t$ then
    if $n(p_l^t)$ equals $n(p_r^s)$ then
        $x_t \geq x_s + w_s + d_s$;
    else
        $x_t \geq x_s + w_s + d_d$;
    end if
else if $x_t = x_s + w_s$ & $n(p_l^t) = n(p_r^s)$ & (FST | (!FST & $h_t = h_s$)) then
    $x_t = x_s + w_s$;
else if $x_t + w_t < x_s$ & $o_r^t$ then
    $x_t + w_t + 2 \leq x_s$;
else if $x_t + w_t < x_s$ & $!o_r^t$ then
    if $n(p_r^t)$ equals $n(p_l^s)$ then
        $x_t + w_t + d_s \leq x_s$;
    else
        $x_t + w_t + d_d \leq x_s$;
    end if
else if $x_t + w_t = x_s$ & $n(p_r^t) = n(p_l^s)$ & (FST | (!FST & $h_t = h_s$)) then
    $x_t + w_t = x_s$;
else
    Unsatisfiable Condition;
end if
```

Flow Formulation

SP&R implements flow formulations such as *Edge Assignment* and *Metal Segment* by utilizing the same methodology of [75][27]. The refined SMT representations of *CFC* and *VE* are as follows.

Commodity Flow Conservation (CFC)

Expression (3.5) represents the CFC constraint. The number of activated commodity-flow indicators $f_m^n(v,u)$ between a certain vertex v and its adjacent vertices $a(v)$ is 1 (Exactly-1) in case

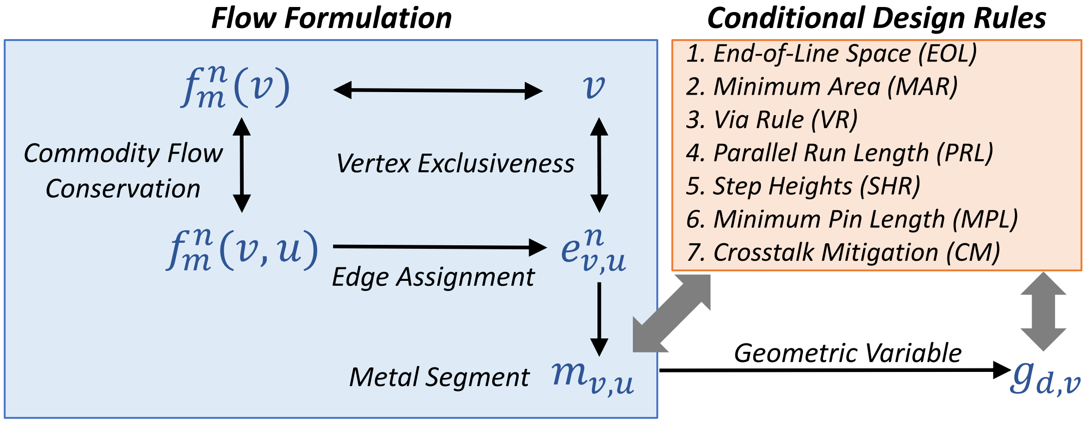


Figure 3.10: Flow formulation with conditional design rules.

of source s^n or sink d_m^n, and is 0 or 2 in the other cases. "Exactly-k" constraints are represented by combining "AMk" and "ALk".

$$\begin{cases} \mathbf{AL1}\big(F_m^n(v)\big) \wedge \mathbf{AM1}\big(F_m^n(v)\big), & \text{if } v = s^n, d_m^n \\ \mathbf{AM0}\big(F_m^n(v)\big) \vee \big\{\mathbf{AL2}\big(F_m^n(v)\big) \wedge \mathbf{AM2}\big(F_m^n(v)\big)\big\}, & \text{otherwise} \end{cases} \tag{3.5}$$

$$F_m^n(v) = \{f_m^n(v,u) \mid u \in a(v)\}, \forall v \in V,\ \forall n \in N,\ \forall d_m^n \in D^n$$

Vertex Exclusiveness (VE)

Expression (3.6) ensures that there are no intersecting nets on any vertices except P_{EX} (see Section 3.2.4). For P_{EX}, Exactly-k (E-k) constraint is set because the supernode of external pins should be shared as many as the number of P_{EX}. When v = P_{IN} or P_{EX}, only one edge indicator must be used. Otherwise, we allow multiple uses of edges against vertex v for a certain net.

$$\begin{cases} \mathbf{AL1}(E_{IN}(v)) \wedge \mathbf{AM1}(E_{IN}(v)), & \text{if } v = P_{IN} \\ \mathbf{ALk}(E_{EX}(v)) \wedge \mathbf{AMk}(E_{EX}(v)), k = |P_{EX}|, & \text{else if } v = P_{EX} \\ \mathbf{AM1}(\{\bigvee_{u \in a(v)} e^n_{v,u} \mid n \in N\}), & \text{otherwise} \end{cases}$$

$$E_{IN}(v) = \{e^n_{v,u} \mid u \in a(v)\}, E_{EX}(v) = \{e^n_{v,u} \mid n \in N, u \in a(v)\}, \quad \forall n \in N, \forall v \in V \qquad (3.6)$$

Conditional Design Rule

Previous works [25][75][27] mainly tackle three representative conditional design rules, e.g., *Minimum Area (MAR)*, *End-of-Line Spacing (EOL)*, and *Via Rule (VR)*. In *SP&R*, *Minimum Area* and *End-of-Line Spacing* follow the same principle of [75][27]. Compared to [75][27], we adopt stack via rule (stack-able) for *Via Rule*. Furthermore, *SP&R* includes multi-pattern-aware design rules such as *Parallel Run Length (PRL)* and *Step Heights Rule (SHR)* [78][23]. 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)*. *Crosstalk Mitigation (CM)* contributes to maintain a stable operation of timing-critical sequential logic cells. **Parallel Run-Length (PRL).** PRL rule is a design rule to avoid "single-point-contact" in manufacturing SADP mask [23]. Fig. 3.11 and Constraint (3.7) represent an example of PRL rule and the corresponding formulation when the run-length (RL) is 2.

$$\mathbf{AM1}(g_{R,v}, g_{L,v_B}, g_{L,v_{BL}}); \mathbf{AM1}(g_{R,v}, g_{L,v_F}, g_{L,v_{FL}}), \forall v \in V_0, V_2 \qquad (3.7)$$

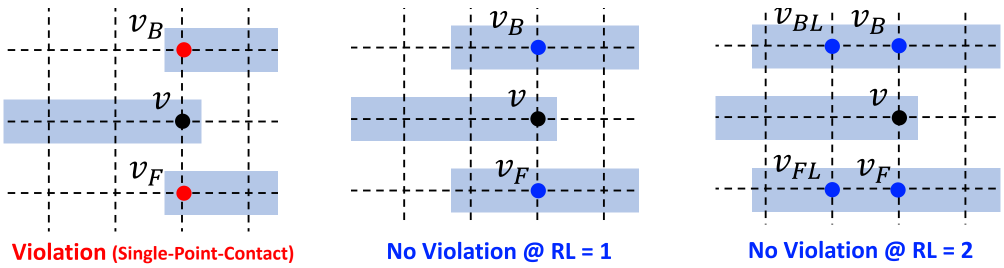


Figure 3.11: Example of PRL (Parallel Run-Length) rule.

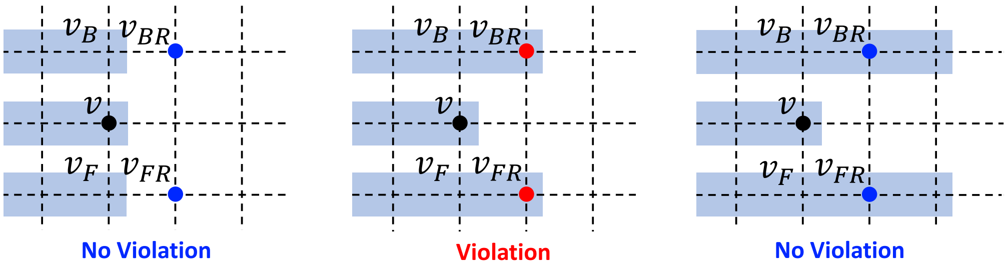


Figure 3.12: Example of SHR (Step Heights Rule).

Step Heights Rule (SHR)

SHR is a design rule to avoid "the small step" in manufacturing SADP mask [23]. Fig. 3.12 and Constraint (3.8) describe an example of SHR and the corresponding formulation when the step height is 2.

$$\mathbf{AM1}(g_{R,v}, g_{R,v_{BR}}); \mathbf{AM1}(g_{R,v}, g_{R,v_{FR}}), \quad \forall v \in V_0, V_2 \tag{3.8}$$

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 (3.9). Then, the metal segment on $M1$ layer is extended to have the minimum length defined by MAR as depicted in Fig. 3.13.

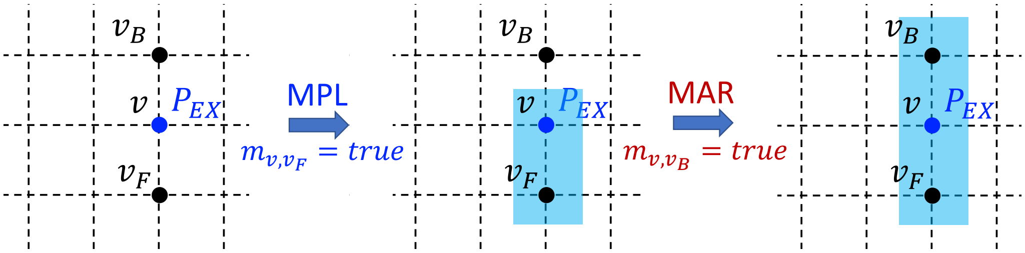


Figure 3.13: Example of MPL (Minimum Pin Length Rule).

$$\mathbf{AL1}(m_{v,v_F}, m_{v,v_B}), \quad \text{if } f_m^n(v, v_D) = 1\ ,\ f_m^n(v, v_U) = 1 \qquad \forall v \in V_1, d_m^n = P_{EX} \tag{3.9}$$

Crosstalk Mitigation (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 [79]. 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. 3.14 and Expression (3.10) represent the crosstalk mitigation constraint between nets *n* and *m* that are in a pair of nets with crosstalk mitigation N_c when $ML = 3$.

$$\begin{aligned}\mathbf{AL1}\big((\neg e^n_{v,v_{R1}} \vee \neg e^m_{v_F,v_{FR1}}),\ (\neg e^n_{v_{R1},v_{R2}} \vee \neg e^m_{v_{FR1},v_{FR2}}),\\ (\neg e^n_{v_{R2},v_{R3}} \vee \neg e^m_{v_{FR2},v_{FR3}}),\ (\neg e^n_{v_{R3},v_{R4}} \vee \neg e^m_{v_{FR3},v_{FR4}})\big)\\ \forall n, m \in N_c\end{aligned} \tag{3.10}$$

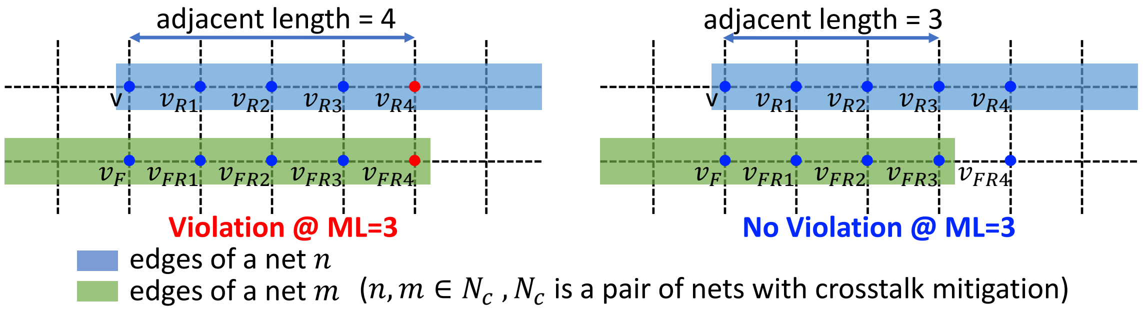


Figure 3.14: Example of CM (Crosstalk Mitigation) rule.

3.3.3 Dynamic Pin Allocation (DPA)

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. 3.15.

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. 3.15(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 5 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. 3.15(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 $f_m^n(v,u)$ (in Expression (3.5) of routing formulation) is associated with the flow capacity variable $C_m^n(p,r)$ by the constraint described in Expression (3.11). Each $f_m^n(v,u)$ is determined by the routing formulation when vertex v is the internal pin p, and the adjacent

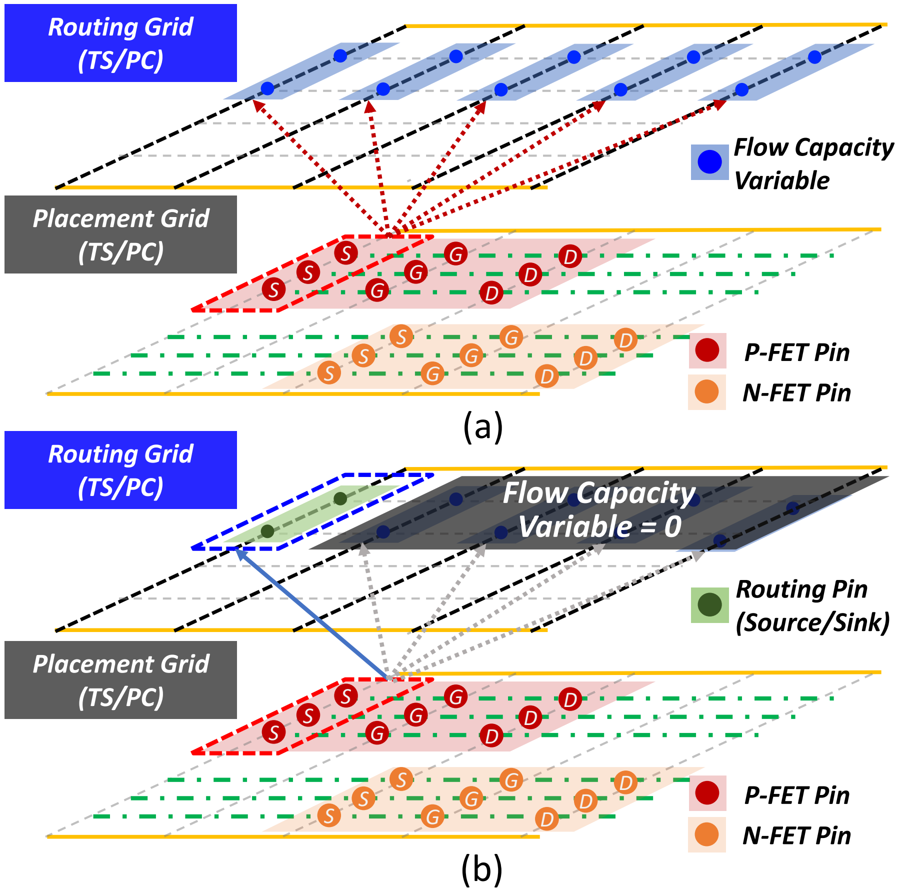


Figure 3.15: Dynamic pin allocation (DPA) between placement and routing grids. In (b), the flow capacities are assigned to 0, a source pin's outside column.

Algorithm 5 Flow Capacity Control Constraint ($C^n_m(p,r)$)

/ 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
    $C^n_m(p,r) = 0$;
else
    $C^n_m(p,r)$ is Determined by Routing Formulation;
end if
```

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. 3.15(b)).

$$f_m^n(v = p, u = r) \leq C_m^n(p, r), \quad \forall r \in a(p), \forall r \in V_0 \tag{3.11}$$

3.4 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 *(i) Breaking Design Symmetry*, *(ii) Conditional Assignment*, *(iii) Localization of the Routing Region*, *(iv) Cell Partitioning*, and *(v) Objective Partitioning*.

3.4.1 Breaking Design Symmetry

The proposed *SP&R* reduces the search space by eliminating symmetries existing in standard cell layout design [80][81].

Flipping of Even-Numbered Multi-Finger 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. 3.16. Therefore, for every FET t with even-numbered fingers, ff_t is set to 0 to remove the flipped FETs from the search space.

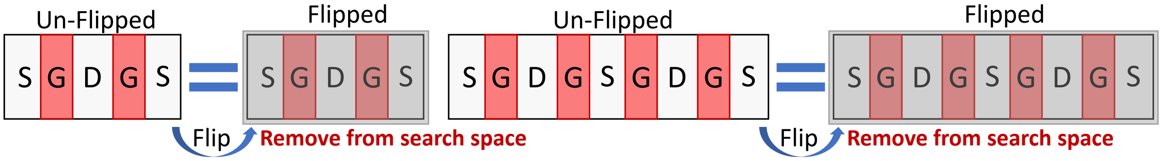


Figure 3.16: Flipped case exclusion of even-numbered finger FETs.

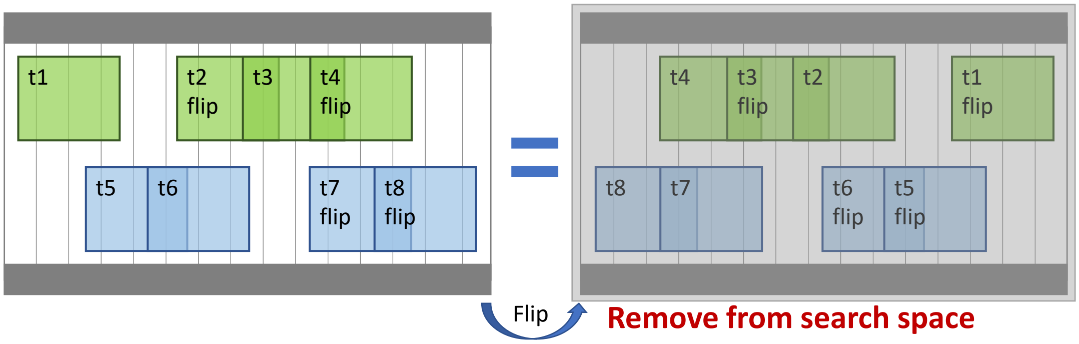


Figure 3.17: Flipped case exclusion of whole cell design from search space.

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. 3.17. 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 6 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 $T_{comb} = \{\{t_1,t_2\},\{t_1,t_3\},\{t_1,t_4\}\}$. When N_p is even, T_{comb} indicates a set of FET groups that should be placed on the left side in an original solution. Therefore, by setting the relative position of FETs in k_{th} 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 $T_k^{comb} < T_{center} < \{T_p - T_{center} - T_k^{comb}\}$.

Algorithm 6 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 $T_{comb}$, $k_{th}$ FET group in $T_{comb}$ $T_k^{comb}$
if $N_p$ is even then
    $T_{comb} \leftarrow GetCombination(T_p, N_p/2)$;
    while $T_k^{comb} \neq \emptyset$ do
        Set SMT Constraint $x_m < x_n$, $\forall m \in T_k^{comb}$, $\forall n \in \{T_p - T_k^{comb}\}$;
        $k \leftarrow k+1$;
    end while
else
    for $i = 1$ to $N_p$ do
        $T_{comb} \leftarrow GetCombination(\{T_p - T_i^p\}, (N_p - 1)/2)$;
        while $T_k^{comb} \neq \emptyset$ do
            Set SMT Constraint $x_m < x_n$, $\forall m \in T_k^{comb}$, $n = T_i^p$;
            Set SMT Constraint $x_m < x_n$, $m = T_i^p$, $\forall n \in \{T_p - T_k^{comb}\}$;
            $k \leftarrow k+1$;
        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
            $T_{comb} \leftarrow T_{comb} \cup T_k^{tmp}$;
        end if
        $k \leftarrow k+1$;
    end while
    return $T_{comb}$;
end procedure
```

3.4.2 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. 3.18. When a source (s^n) and a sink (d_m^n) nodes of m_{th} 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_m^n outside this column will be set as 0 by the flow formulation (Fig. 3.18(a)). Fig. 3.18(b) shows a commodity flow through TS at the same column by DS. Since the source

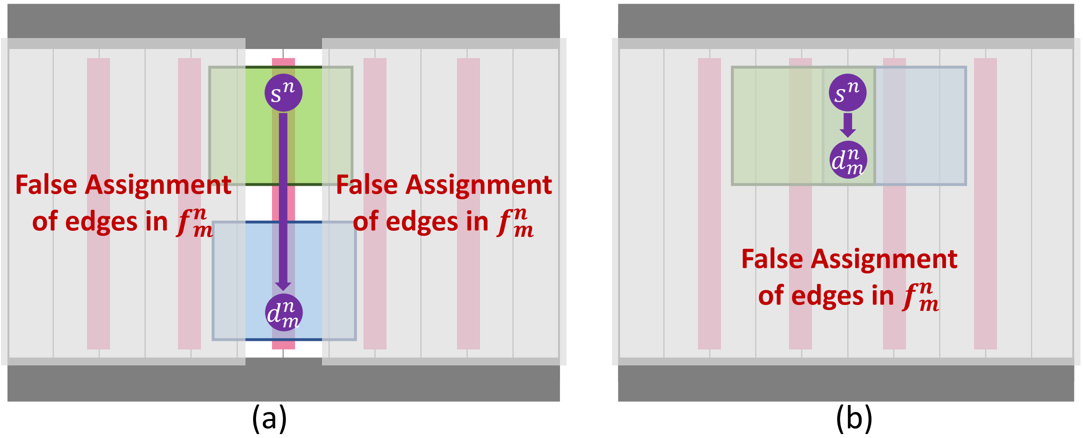

s^n
d_m^n
False Assignment of edges in f_m^n
False Assignment of edges in f_m^n
(a)
s^n
d_m^n
False Assignment of edges in f_m^n
(b)

Figure 3.18: Conditional assignment. (a) A commodity flow through the same gate column, and (b) a commodity flow through the DS.

Algorithm 7 Conditional Localization

```
Input: tolerance of localization $T$
/* x coordinates of routing grids r and p: $x_r$ and $x_p$ */
/* x coordinate of a source (resp. sink) pin s (resp. d): $x_s$ (resp. $x_d$) */
/* Single column pin only: Set of x is singleton */
/* s is a source and d is a sink of net n and commodity m */

if $(x_s \geq x_d)$ & $\big((x_r < x_d - T) \mid (x_r > x_s + T)\big)$ then
    $f_m^n(v = q, u = r) = 0, \quad \forall q \in a(r)$;
else if $(x_s < x_d)$ & $\big((x_r < x_s - T) \mid (x_r > x_d + T)\big)$ then
    $f_m^n(v = q, u = r) = 0, \quad \forall q \in a(r)$;
else
    $f_m^n(v = q, u = r), \quad \forall q \in a(r)$ Determined by the Routing Formulation;
end if
```

and sink can be enabled on the same vertex, all edge variables in f_m^n are conditionally set to 0.

3.4.3 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*.

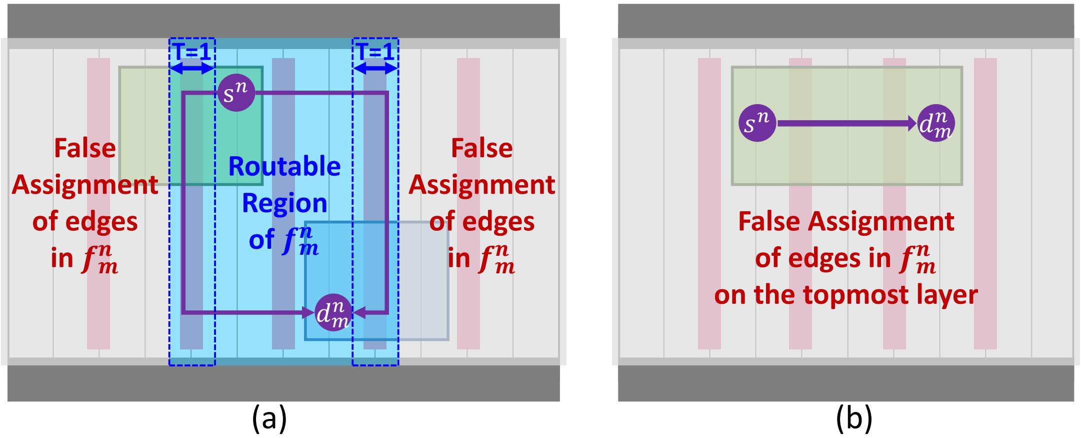

T=1
T=1
s^n
d_m^n
False Assignment of edges in f_m^n
Routable Region of f_m^n
False Assignment of edges in f_m^n
(a)
s^n
d_m^n
False Assignment of edges in f_m^n on the topmost layer
(b)

Figure 3.19: Conditional Localization. (a) Localization of commodity flows with a tolerance T=1. (b) Localization of a commodity flow within the same FET.

Conditional Localization

Fig. 3.19(a) shows the example of the conditional localization. When intermediate locations of source s^n and sink d_m^n of commodity f_m^n are determined, we restrict the path connecting s^n and d_m^n in the minimum bounding box that covers both s^n and d_m^n (depicted in blue rectangle). In Algorithm 7, the offset with tolerance T gives a margin to prevent from over-cutting.

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. 3.19(b).

3.4.4 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

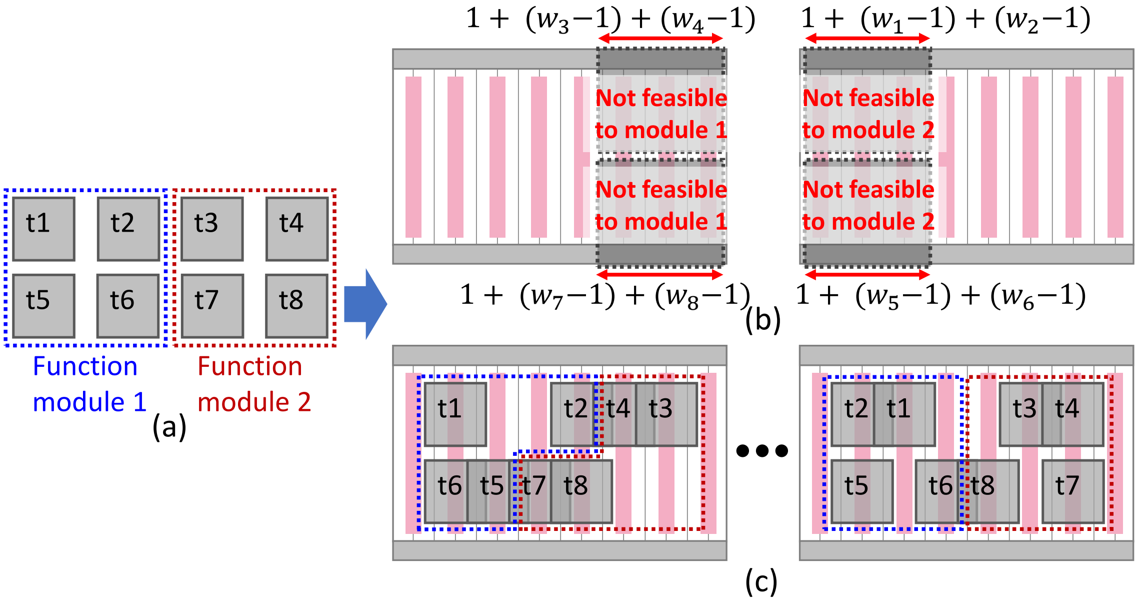


Figure 3.20: Cell Partitioning. (a) Functional module partitioning. (b) Localization of the placement area. (c) Examples of *SP&R* with cell partitioning.

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. 3.20. With the pre-defined FET groups by the functionality (Fig. 3.20(a)), *SP&R* honors the order of FETs among functional-module groups (Fig. 3.20(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. 3.20(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.

3.4.5 Objective Function Partitioning

SP&R co-optimizes multiple objectives at once by using the lexicographic method [18][74]. 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 metal length objective (described in Expression (3.3)) is defined as the weighted sum of metal segments (i.e., CA[2], VIA, 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 metal length objective with the priority as shown in Expression (3.12). Among the three layers, we assign the higher priority for the lower layers to prevent redundant routing detours, causing the increment on the total metal length because the lower layers have more elements than the upper layers in our framework[3].

$$\begin{aligned} \textbf{LexMin: } & \text{(a) } \#CA, \text{ (b) } \#VIA01, \text{ (c) } \#VIA12, \\ & \text{(d) } \#M0, \text{ (e) } \#M1, \text{ (f) } \#M2 \end{aligned} \tag{3.12}$$

3.5 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 3.6GHz AMD Ryzen5 3600 CPU and 32GB memory. The SMT Solver *Z3* (ver. 4.8.5) [20] 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

[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.

cell libraries. We choose 37 out of 69 cells in NanGate's 15*nm* open cell library [82] and 85 out of 183 cells in the ASAP7 library [7] 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 [7, 70]. The 15*nm* library is converted to the 7*nm* cell-equivalent architecture (6 horizontal routing tracks and 3 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 is 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.

Table 3.2: 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, 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%

[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.

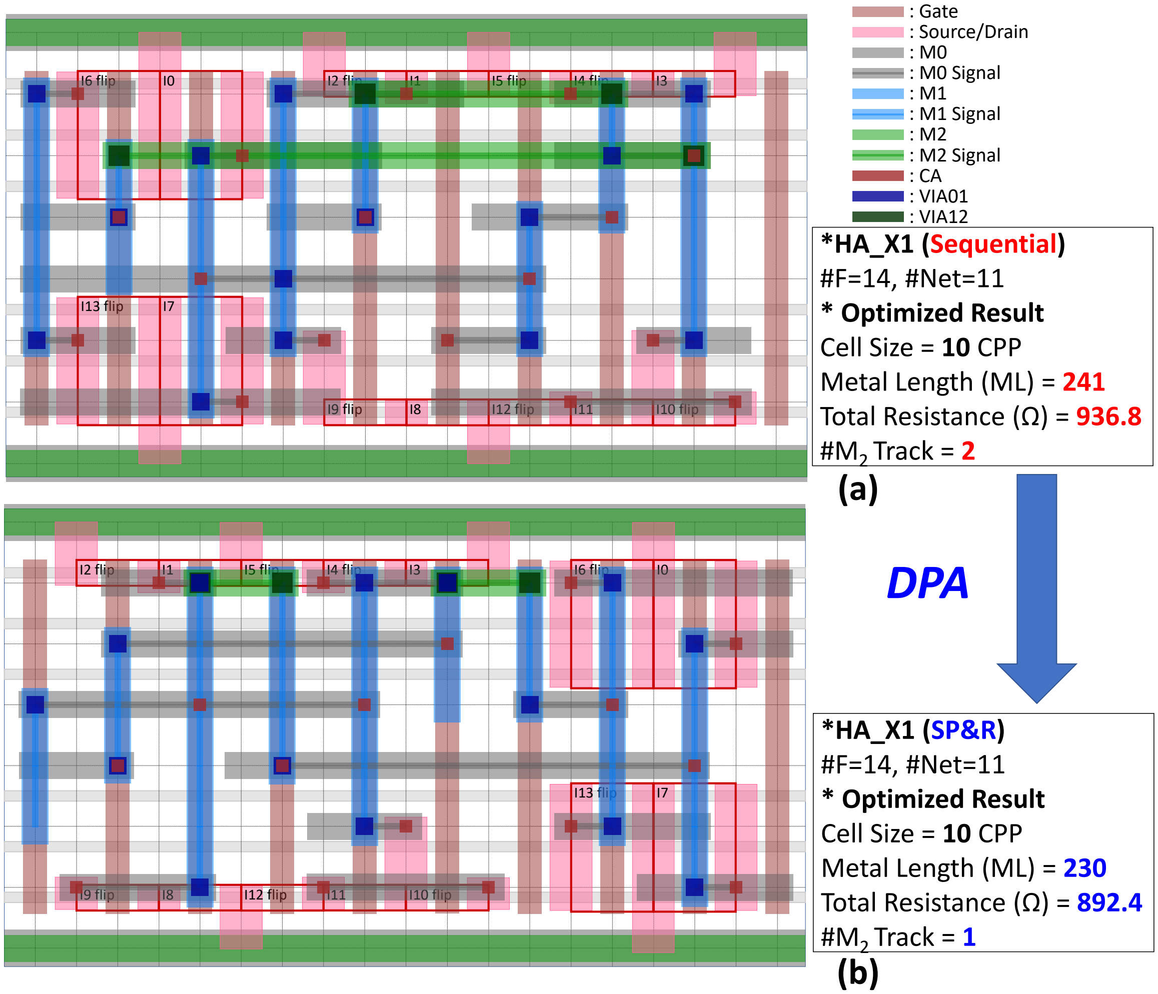


Figure 3.21: The comparison between (a) sequential (#$M2$=2, ML=241, Total Resistance=936.8Ω) and (b) simultaneous P&R (the proposed *SP&R*) (#$M2$=1, ML=230, Total Resistance=892.4Ω) using HA_X1.

3.5.1 Sequential vs. Simultaneous P&R

We compare our simultaneous P&R approach to the conventional sequential approach [70]. To simulate the sequential P&R of [70], 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 3.2.3 for fair comparison.

Table 3.2 presents the comparison of key metrics for 15 out of 122 cells that have more

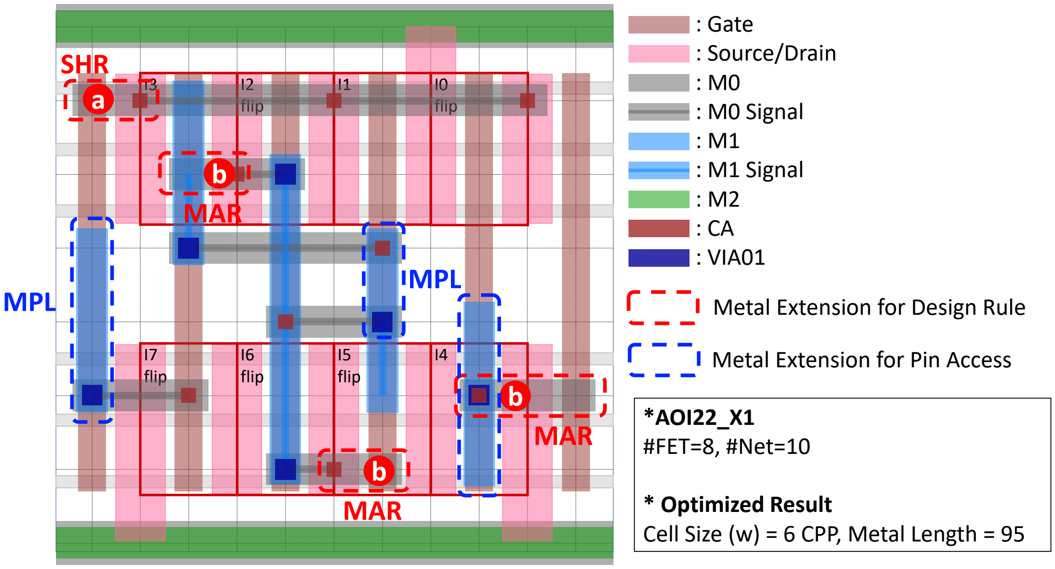


Figure 3.22: An optimized cell layout for DFM and I/O pin accessibility.

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 metal length 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% to 5% increment of resistance, #$M2$ tracks has been reduced 33% under the priority of our key objectives.

Fig. 3.21 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 10 CPP (contacted poly pitch), *SP&R* (Fig. 3.21(b)) respectively reduces the #$M2$ tracks and the total resistance by 50% ($2 \rightarrow 1$) and 5% ($936.8 \rightarrow 892.4$) by virtue of the DPA scheme (Fig. 3.21(a)).

[5] The total parasitic resistance of wires is calculated using sheet resistance information of ASAP7 library published in [83].

3.5.2 Optimization for DFM and I/O pin accessibility

Fig. 3.22 shows an example of a generated AOI22_X1 cell layout satisfying all pre-discussed 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.

3.5.3 Single Diffusion Break in a Crossover

Fig. 3.23(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. 3.23(b)).

3.5.4 Crosstalk Mitigation

Fig. 3.24(a) shows a layout of a DFFHQNx1 from ASAP7 library, displayed by a commercial tool [8]. The clock signals (*CLKN* and *CLKB*) with the opposite directions are routed in the adjacent $M2$ tracks with a parallel run-length over 7 CPPs. This may cause a substantial crosstalk between the clock signals which increases the delay and the power consumption. In constrast, the result of *SP&R* with a crosstalk mitigation parameter $ML = 4$ between the clock signals successfully prevents the crosstalk by restricting the parallel run-length of those signals less than 2 CPPs as shown in Fig. 3.24(b).

3.5.5 Scalability Improvement

Table 3.3 represents the scalability improvement stages of *SP&R* framework. Phase I refers to the base framework of [73]. Phase II includes a simple pre-processing based-on the

[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.

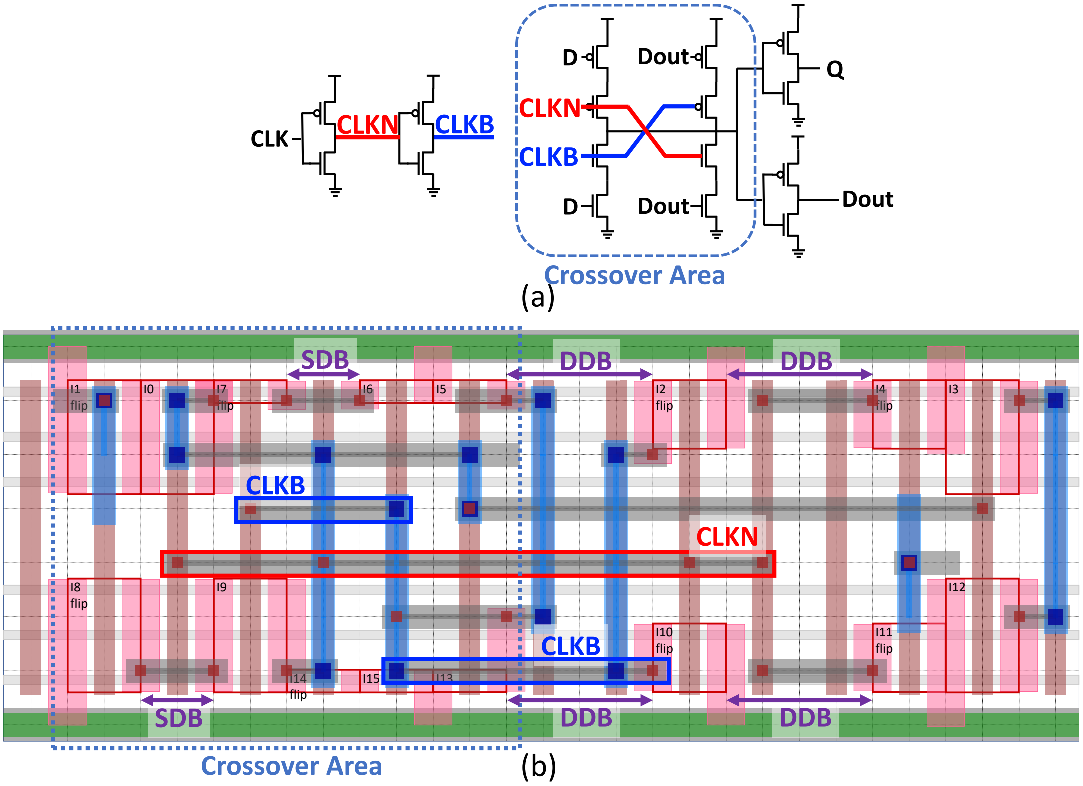


Figure 3.23: Example of SDBs in a crossover area. (a) A schematic of DHLx1, and (b) DHLx1 layout generation.

Table 3.3: Scalability improvement stages.

Stage	**Description**
Phase I	The framework of [73]
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

Boolean constraint propagation (BCP), breaking design symmetry, and conditional assignment based on Phase I. Phase III and Phase IV perform localization of the routing region and objective function partitioning based on Phase II and Phase III, respectively.

Trade-Off between Scalability and Key Metrics

Table 3.4 presents the experimental results showing the trade-off between scalability and key metrics of each improvement stage for 35 NandGate [82] and 30 ASAP7 [7] combinational

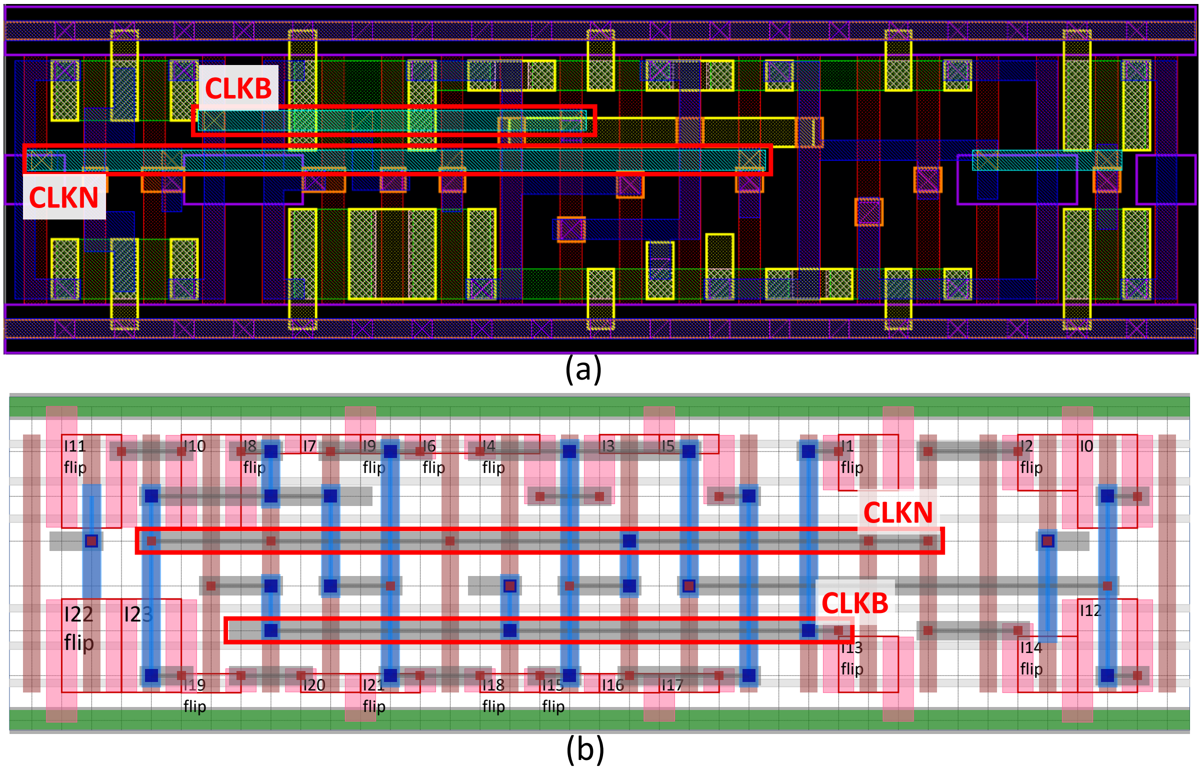
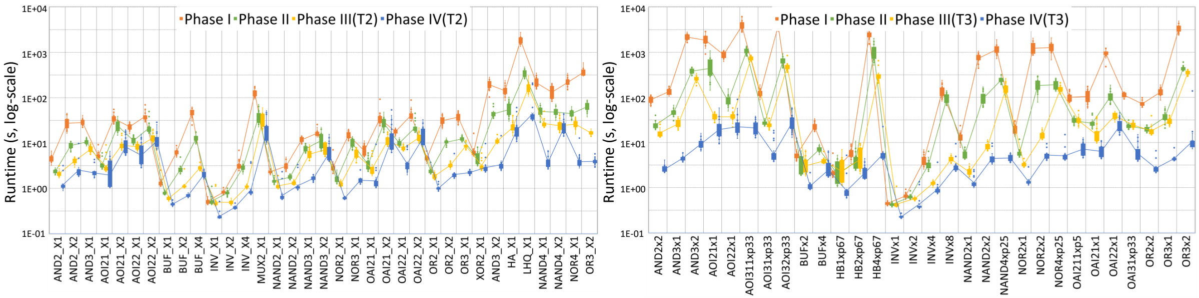


Figure 3.24: Layout of DFFHQNx1. (a) Layout from the standard cell library [7], displayed by a commercial tool [8]. (b) *SP&R*'s layout generation.

Table 3.4: Trade-off 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 ratio (reference = Phase I), 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×**



(a) 35 combinational logic cells of NanGate Library

(b) 30 combinational logic cells of ASAP7 Library

Figure 3.25: Contributions of each scalability improvement phase for runtime reduction with statistical runtime visualization (21 random seeds).

logic cells which can be generated within an hour in Phase I with different design rule sets. Phase 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× (75.14s@Phase I → 3.61s@Phase IV ($T1$)). Compared to Phase I (the simple pre-processing with BCP), the average numbers of variables (#Var)/constraints (#Con) have been reduced by 4.1%/4.8% and 6.7%/7.9% for Phase II and Phase III, respectively. In Phase III, the conditional localization with tolerance $T = 1$ ($T1$) results in 0.17% increment of the total metal length compared to Phase I while all key metrics are the same with tolerance $T = 2$ ($T2$). In Phase IV, the total metal lengths 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× (557.34s@Phase I → 4.23s@Phase IV ($T1$)). In Phase III, the conditional localization with tolerance $T = 1$ ($T1$) results in 0.07% increment of the total metal length 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 metal length 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]

Combinational Logic Cells

Fig. 3.25 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. 3.25(a), 222.3 → 52.0 → 16.5 → 2.6 in Fig. 3.25(b)). This demonstrates that the search space is effectively reduced with the scalability improvement constraints.

Sequential Logic Cells

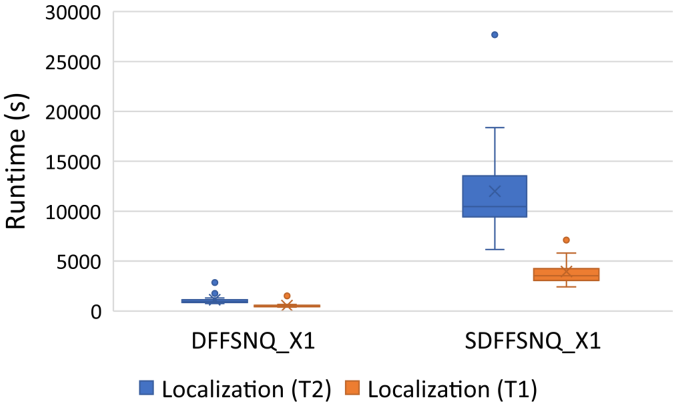
Fig. 3.26(a) shows the key metrics of sequential logic cells in NanGate library [82]. The number of FETs/nets in DFFSNQ_X1 and SDFFSNQ_X1 are 28/19 and 36/27, respectively. Due to the high complexity, the cell-partitioning constraints with six functional modules are applied to the improvement features 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 metal length as well as VIA length is decreased similarly to the Phase IV cases in Table 3.4. The decreased runtime deviation in Fig. 3.26(b) with the decrease of the tolerance shows the reduction

[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 Phase I and II can be achievable with the higher weight (i.e., ¿ 4) of VIA elements.

(T : tolerance of localization)

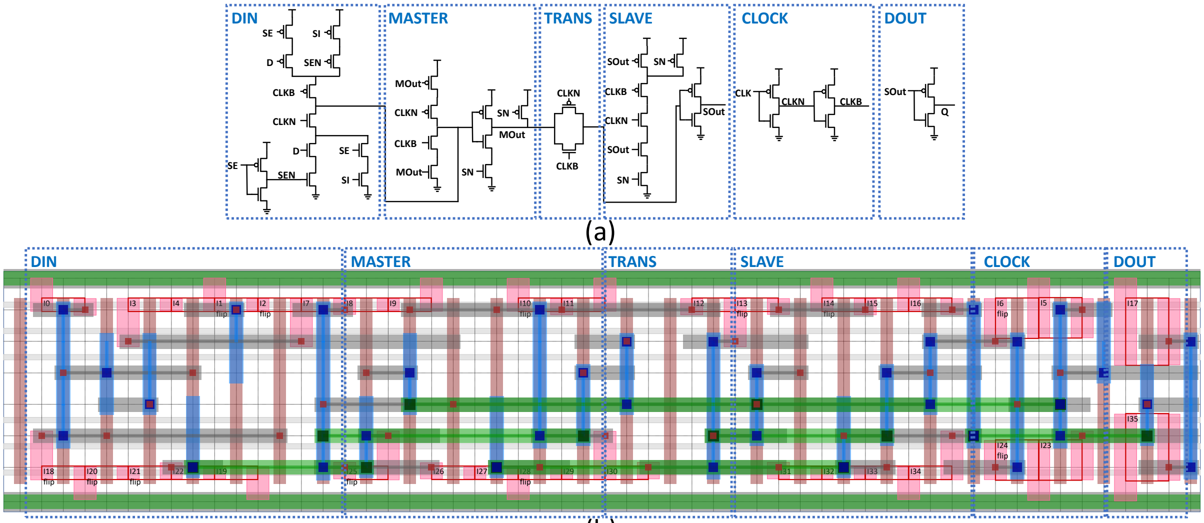
	T	Key Metrics				
		Size	Metal Length			M_2
			Total	VIA	Metal	
DFFSNQ_X1	T1	23	612	340	272	3
	T2		606	336	270	
SDFFSNQ_X1	T1	28	702	388	314	
	T2	28	697	384	313	

(a)



(b)

Figure 3.26: Sequential logic cells. (a) Key metrics statistics. (b) Runtime variation depicted in boxplots (21 random seeds).

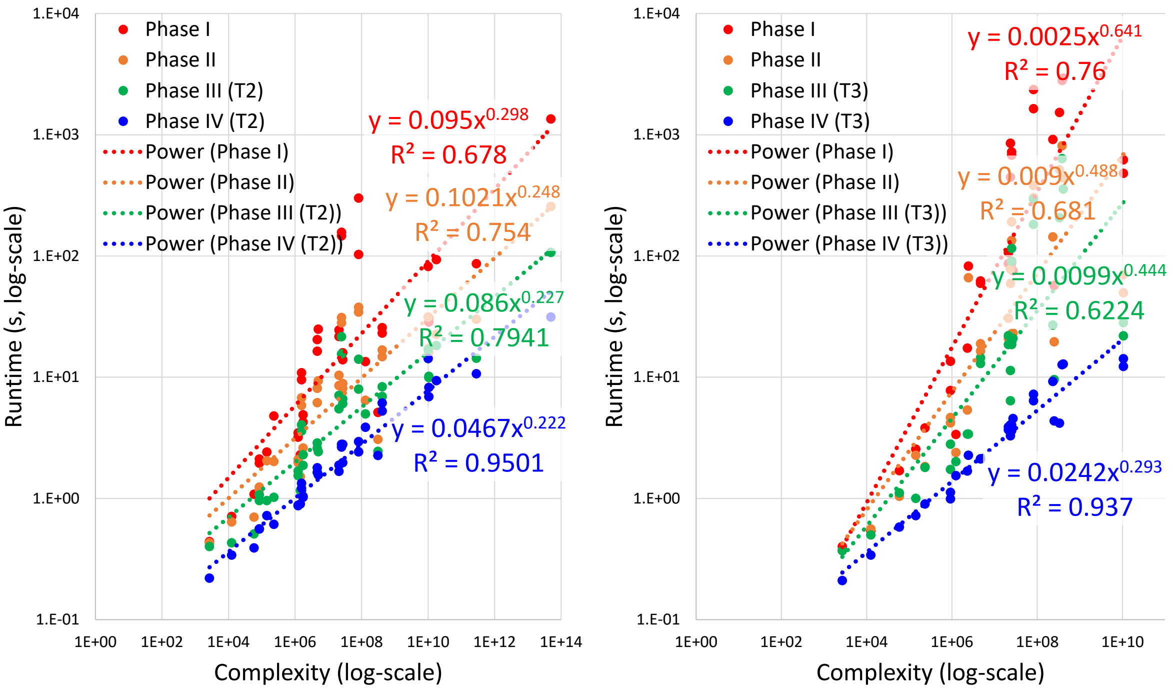


(a)

(b)

Figure 3.27: SDFFSNQ_X1. The largest cell in this work (28 CPPs). (a) Function module partitioning. (b) The generated layout (6168 seconds).

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. 3.27(a) and Fig. 3.27(b), respectively.



(a) NanGate 35 combinational logic cells

(b) ASAP7 30 combinational logic cells

Figure 3.28: Scalability of *SP&R* for combinational logic cells (in log-scale).

Scalability of SP&R Framework

The placement permutation is expressed in Expression (3.13). The number of clauses for routing is derived as Expression (3.14)[27]. In order to predict the runtime of SP&R, we test various structures combining Expressions (3.13) and (3.14) to maximize the correlation R^2. Using Expression (3.15), we achieve $R^2 = 0.9501$ for Phase IV on NanGate 35 combinational logic cells (Fig. 3.28(a)) and $R^2 = 0.937$ for Phase IV on ASAP7 on 30 combinational logic cells (Fig. 3.28(b)).

$$\textbf{Placement permutation : } O\big(\big(\frac{(X/2)!}{(X/2-N/2)!}\big)^2\big) \tag{3.13}$$

$$\textbf{Routing clauses : } O\big(X \cdot P\big) \tag{3.14}$$

$$\textbf{\textit{SP\&R} runtime prediction base : } O\big(\frac{(X/2)!}{(X/2-N/2)!} \cdot X^2 \cdot P^2\big) \tag{3.15}$$

3.5.6 Experimental Statistics of a Practical 7nm Cell Library

We show that the proposed *SP&R* satisfies both practicality and scalability through comparing key metrics and runtime with ASAP7 library [7] and the previous work [70], respectively. Though the apple-to-apple comparison may not possible due to the different netlists and cell architectures, cell size and number of occupied $M2$ tracks can be compared because (i) the important routing resources such as the numbers of horizontal routing tracks and fins are the same and (ii) 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, crosstalk mitigation, and SDB in a crossover constraint are applied.

Combinational Logic Cells

Table 3.5 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, 1 cell has reduced the number of $M2$ tracks by 4, and 2 cells have reduced cell size by 1 to 2 CPPs. Compared to the previous sequential approach [70], 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 minutes.

Table 3.6 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

Table 3.5: *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[7]		SP&R		Runtime(s)	
		Min.	Max.	Min.	Max.	Min.	Max.	Size	M_2	Size	M_2	SP&R	[70]
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 3.6: *SP&R* results of 15 combinational/3 sequential logic cells from ASAP7 library with FET separation.

CellType	#Cell	#FET		#NET		Size		ASAP7[7]		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**

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. 3.29 illustrates an example. By separating FETs, the cell size is reduced by 4 CPPs (Fig. 3.29(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. 3.29(a)). The increment on #$M2$ at the “AND/OR“ cell type is caused by the “AND5x2” cell whose cell size is reduced by 6 CPPs compared to the known layout at the cost of 2 more #$M2$ tracks. The average runtimes of AND/OR and NAND/NOR types of cells are much higher than the other cells in Table 3.5 due to the increased complexity and the larger cell size.

Sequential Logic Cells

Table 3.7 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 *M*2 tracks. Fig. 3.24(b) displays a layout of DFFHQNx1 generated by *SP&R*. With the same cell size, the result of *SP&R* requires less *M*2 routing tracks than the known layout of Fig. 3.24(a). All sequential logic cells are generated within 42 minutes and the average runtime per cell is less

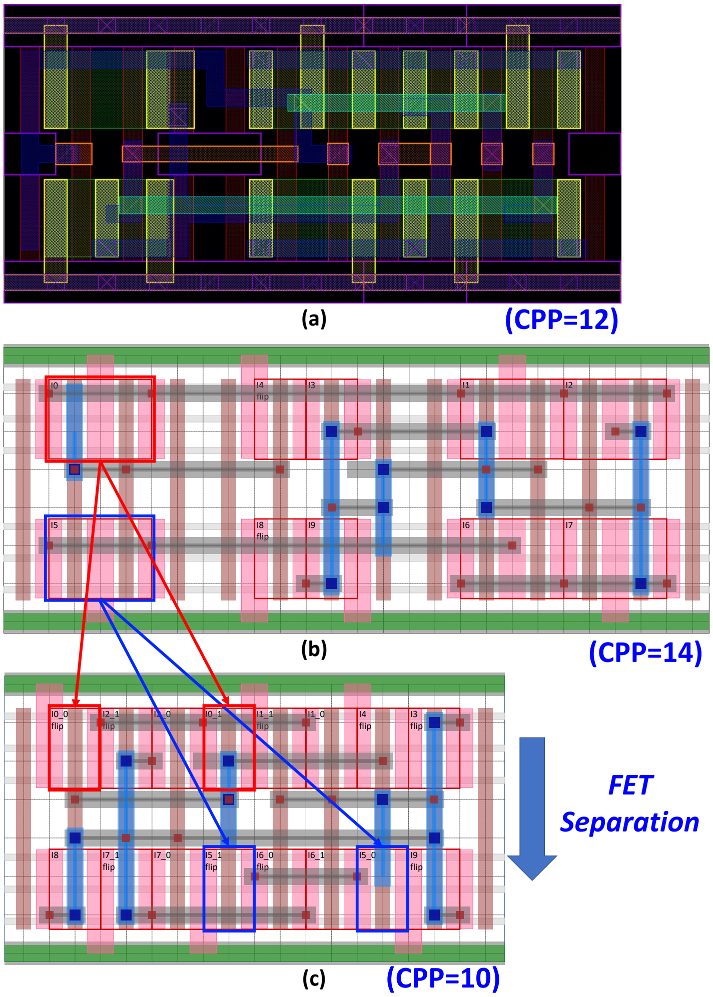


Figure 3.29: Layout of XOR2x1. (a) Layout from ASAP7 standard cell library [7]. *SP&R* layout (b) without and (c) with FET separation.

Table 3.7: *SP&R* results of 23 sequential logic cells from ASAP7 library without FET separation.

CellType	#FET	#NET	ASAP7[7]		SP&R			[70]				
			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**					

than 16 minutes. Compared to the previous work [70], *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 (contacted poly pitch) and 0.3 tracks on average for all 183 ASAP7 cells compared to the known layouts, respectively. Also, *SP&R* generates a whole set of 7nm standard cell library within 19 hours through the scalability improvement (Fig. 3.30). The obvious separation of runtime trends between two types of cells is observed by virtue of the additional scalability improvements such as the cell partitioning.

3.6 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 place-and-route in search space. We have improved the

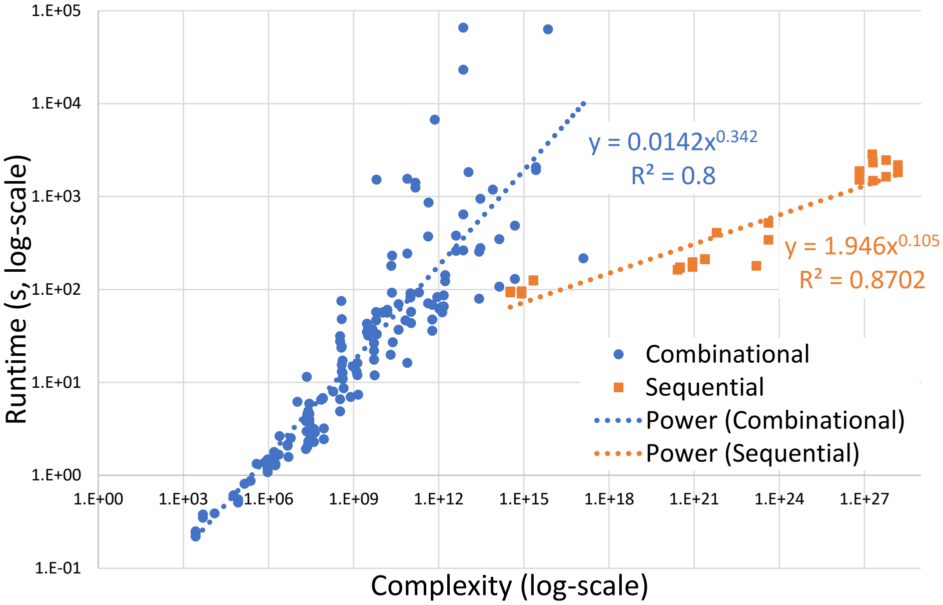


Figure 3.30: Scalability of SP&R framework for 157 combinational and 26 sequential logic cells in ASAP7 Library (in log-scale).

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 [73] by exchanging less than 0.2% of the total metal length. 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 hours for the largest cell (SDFFSNQ_X1, Fig. 3.27).

This chapter contains materials from "SP&R: SMT-based Simultaneous Place-&-Route for Standard Cell Synthesis of Advanced Nodes", by Daeyeal Lee, Dongwon Park, Chia-Tung Ho, Ilgweon Kang, Hayoung Kim, Sicun Gao, Bill Lin, and Chung-Kuan Cheng, which appears

[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 sub-optimal solution (in terms of the total metal length) within an hour.

in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, November 2020. The dissertation author was the primary investigator and author of this paper.

Chapter 4

Standard Cell Synthesis: Monolithic 3D Integration

4.1 Introduction

As device integration-process technologies continue to shrink beyond 5*nm*, scaling of conventional (Conv.) FET device becomes increasingly difficult as facing their limitations due to routing congestions, lateral P-N separations, and performance requirements. Also, design technology co-optimization (DTCO) on pitch scaling and patterning approaches its physical and technical cliff in 2D process technology. To overcome these problems by shifting to a 3-D design paradigm, a new transistor structure called vertical gate-all-around (GAA) nanowire FET (VFET) has emerged as a promising candidate for sub-5*nm* nodes [84, 85, 86, 9]. Also, the authors of [87] and [88] have demonstrated the feasibility of stacked logic transistors along the vertical nanowire.

Fig. 4.1 illustrates schematics of nanowire FETs. Whereas the gate length and spacer thickness of a conventional lateral GAA FET (LFET) are confined by the device's footprint, VFET is less constrained as they are oriented vertically. Furthermore, the freedom of device ordering in VFET layouts leads to better layout optimization in terms of routing resources and area density. 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 from transistor-level placement and in-cell routing. Therefore, the automation of standard cell (SDC)-layout design takes essential roles for achieving seamless technology transition and design-based equivalent scaling through manufacturability-aware standard cell layout design [1, 57].

Conventional SDC Synthesis

For transistor-level placement problem, many approaches have been proposed to reduce the search space by adopting heuristic approaches such as “Eulerian trail” [58], “Branch and Bound” [60], “Transistor connection pruning” [61], etc. For in-cell routing problems, several approaches based on traditional “Maze Routing algorithm” [62] are suggested but inapplicable to modern multiple-patterning technologies because of the complex design rules. An SADP-aware

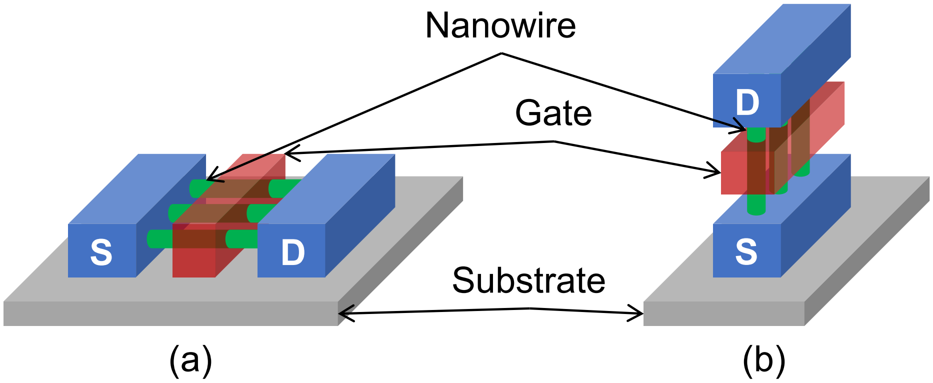


Figure 4.1: Schematics of nanowire FETs. (a) Lateral GAA FET (LFET), and (b) Vertical GAA FET (VFET) [9].

routing solution is presented [64], and several pin-accessibility optimization techniques have attracted considerable attention to improve the pin-accessibility of standard cells in sub-7*nm* technology [64, 65, 66, 67, 70, 71, 72]. However, these approaches which rely on solving sub-problems are hard to reach the optimal cell-layout solution because of the intractable search space partitioning and the sub-optimal heuristic methodology.

Recently, Lee *et al.* [26] have proposed a Satisfiability Modulo Theories (SMT)-based SDC synthesis automation framework that simultaneously solves the place-and-route (P&R) problem without deploying any sequential procedures by using a novel dynamic pin allocation (DPA) approach, resulting in the generation of SDCs with optimal cell areas. However, this work is unsuitable for the VFET SDC generation due to the distinctive spatial cell structure.

VFET SDC Synthesis

To capture the unique VFET cell architecture, [89] and [90] have discussed layout generation algorithms and search space reduction techniques based on the bipartite tree representation of VFET devices for out-bound and in-bound power rail architectures. In [91], the authors have presented area and routing optimization strategies. Furthermore, recent literatures [92, 10] describe a guideline with an interconnect structure to harvest the maximum advantages of 1-tier as well as many-tier VFETs which stack multiple transistors on the same transistor footprint. They also show that 2-tier VFET SDCs provide a 36.5% reduction of the cell area compared to 1-tier VFET SDCs. However, due to the intrinsic limitation of the sequential P&R approach and the freedom during the placement of the devices on top of the other, there are still rooms to further improve the areal benefit of many-tier VFETs.

In this paper, we propose a novel SMT-based many-tier VFET cell synthesis framework which provides minimum-sized cell layouts by performing concurrent place-and-route (P&R) of FETs, inspired by [26].

Our contributions are summarized as follows.

- We propose a novel many-tier VFET standard cell (SDC) synthesis framework which concurrently performs transistor placement and in-cell routing.
- We develop horizontal/vertical relative positioning constraints and dummy gate control scheme to model the unique VFET cell architecture.
- We devise efficient objectives to enhance the pin-accessibility and reduce the number of vertical interconnections.
- We compare the optimized cell area by using our concurrent P&R approach over the previous work [10].
- We explore the substantial gain of many-tier VFET configurations up to 4 tiers on the cell-level metrics and block-level areas, throughout the various experiments.

The remaining sections are organized as follows. Section 4.2 discusses preliminary information. Section 4.3 describes our proposed synthesis framework. Section 4.4 presents experimental results. Section 4.5 concludes the article.

[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.

Table 4.1: Notations for the proposed VFET SDC Synthesis framework.

Term	Description
X	The number of vertical tracks in the given cell bounding box
H	The number of horizontal tracks in the given cell bounding box
T	The number of tiers
F	Set of FETs
F_s	s^{th} FET
x_s	x-axis coordinate of s^{th} FET
y_s	y-axis coordinate of s^{th} FET
t_s	tier location of s^{th} FET
P^s	Set of internal pins of s^{th} FET
p^s_i	i^{th} pin of s^{th} FET
$n(p)$	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, v_{x_v,y_v,z_v}	A vertex with the coordinate (x_v, y_v, z_v)
v_d	A d-directional[1]adjacent vertex 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_{EX}	Set of external-pin nets in the given routing box
n	n^{th} multi-pin net
f^n_m	A two-pin subnet connecting s^n and d^n_m, i.e., a commodity
v^n	0-1 indicator if v is used for n
$e^n_{v,u}$	0-1 indicator if $e_{v,u}$ is used for n
$f^n_m(v,u)$	0-1 indicator if $e_{v,u}$ is used for commodity f^n_m
$m_{v,u}$	0-1 indicator if there is a metal segment on $e_{v,u}$

4.2 Framework Preliminary

4.2.1 Overview of VFET SDC Synthesis Framework

We formulate an SDC layout process as a single constraint satisfaction problem (CSP) with variables and constraints to integrate SDC place-and-route steps into a multi-objective optimization problem. We deploy the state-of-the-art *lazy*-approach SMT solver *Z3* [18] to solve the given optimization problem. Given netlist information and cell architecture, our framework directly obtains an optimal solution that precisely satisfies the constraints of transistor placement, in-cell routing, and conditional design rules. The individual placement and routing problems

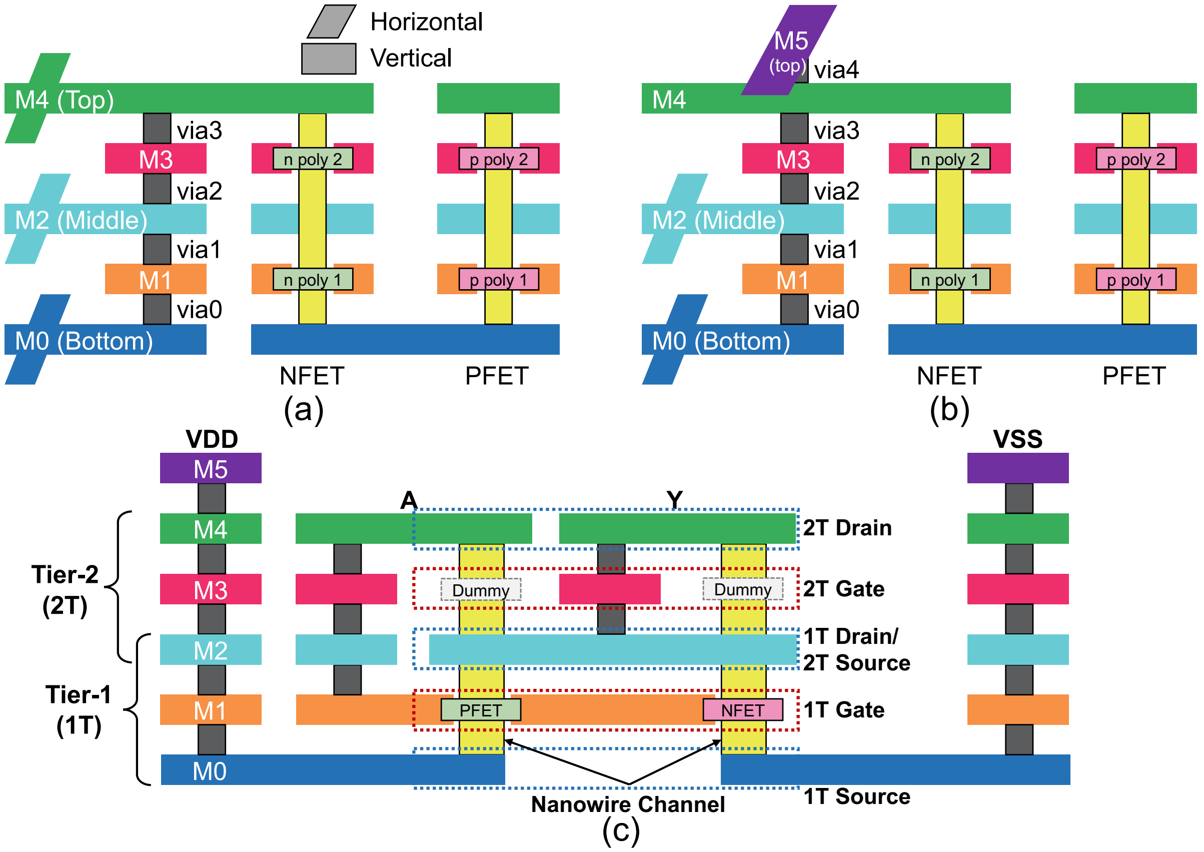


Figure 4.2: Interconnections for VFET. (a) 2-tier interconnection [10], and (b) 2-tier interconnection with top-layer separation. (c) A profile view of 2-tier VFET inverter.

combine into the dynamic formulation targeting conditional pin allocation (i.e., DPA) [26]. The notations are shown in Table 4.1.

4.2.2 Interconnect Structure

Inspired by [10], we adopt the assumptions on the interconnect structure for many-tier VFETs as illustrated in Fig. 4.2(a). First, each VFET device has vertically oriented source, gate, and drain nodes which have separated interconnection layers. For example, a VFET on the bottom tier has M0, M1, and M2 metal layers for source, gate, and drain interconnections. Second, we assume that the gate poly layers are directly connected to the corresponding metal interconnection layers (i.e., M1, M3). Third, we apply bi-directional routing to all source/drain nodes in the interconnection layers (i.e., M0, M2, and M4) and uni-directional vertical routing to the gate terminals (i.e., M1 and M3). Lastly, we allow stacked vias for connecting all of the metal layers between M0 and M4 (i.e., via0, via1, via2, and via3). On top of the assumptions above, we separate the bi-directional top layer into two uni-directional layers (i.e., vertical and horizontal layers) to match them with the preferred directions of BEOLs in the block-level routing as shown in Fig. 4.2(b).

Fig. 4.2(c) illustrates a profile of a 2-tier VFET inverter. VFET transistors in each tier consist of three metal interconnection layers for Gate/Source/Drain nodes. The inverter cell has two active VFET transistors on tier-1 level whereas there are two dummy VFET transistors on tier-2 level that do not have any connections from the Gate poly to the interconnection layer (i.e., M3). M0/M2/M4 layers are used to interconnect Source/Drain nodes of each transistor and power rails. The external I/O pins (i.e., A and Y) are generated on M4/M5 layers.

4.2.3 VFET Cell Architecture

Our framework considers 5*nm* standard cell architecture (e.g., layer/track information) of [10], as depicted in Fig. 4.3.

Layer Configuration

Fig. 4.3 shows an example of our framework's grid-based placement and 3-D routing graph composed of four metal layers (i.e., M0, M1, M2, and M3) for 1-tier cell architecture. In the placement grid, there are three placement layers (i.e., M0, M1, and M2) per tier, aligned with the corresponding routing grids for an allocation of FETs in the P-FET/N-FET region. The placement tracks of P-FET/N-FET are given by the input parameters. In this work, we assign P-FETs and N-FETs on 2^{nd} and 4^{th} placement tracks, respectively[2]. The routing grid has five horizontal tracks, and each layer follows its predefined direction (Section 4.2.2).

[2] The placement track of P-FET/N-FET may affect the final layout solutions in terms of the cell area and metal length. In this work, we refer to the same placement tracks of [10] for generating SDCs with 5 horizontal tracks.

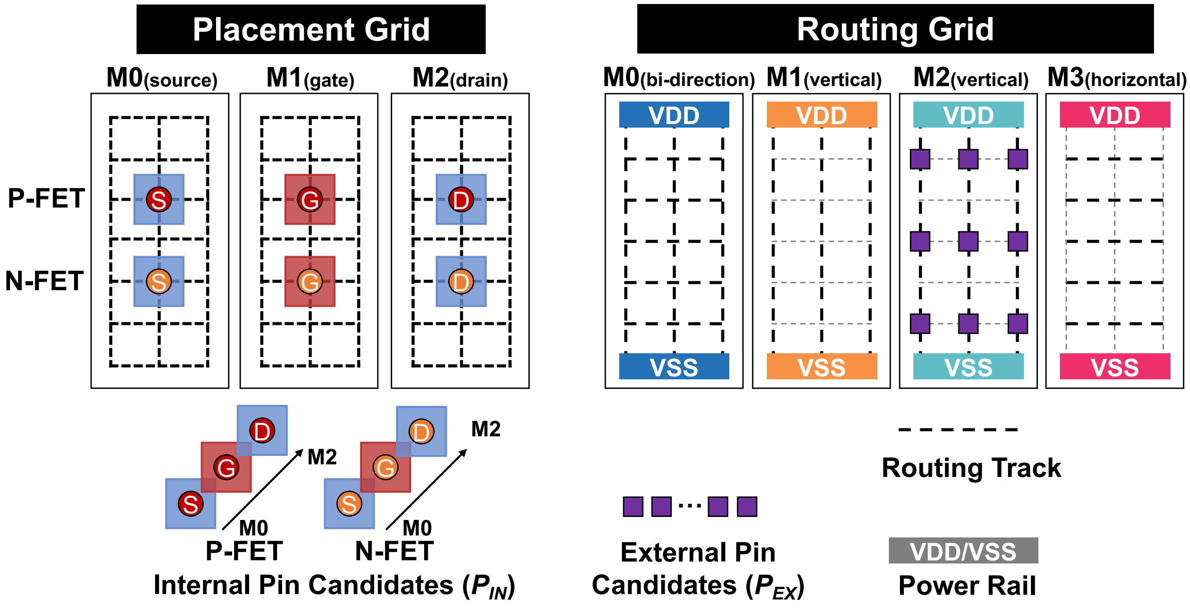


Figure 4.3: Grid-based placement & 3-D routing graph for 1-tier cell architecture.

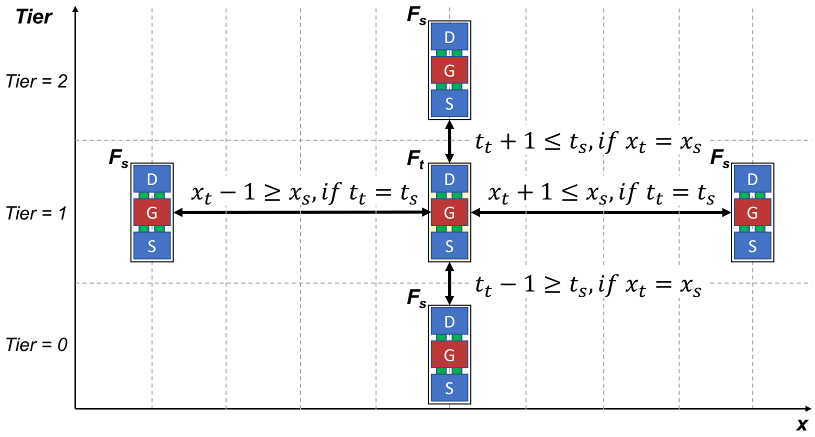

Tier
Tier = 2
Tier = 1
Tier = 0
x
F_s
F_t
D
G
S
$t_t + 1 \leq t_s, if\ x_t = x_s$
$x_t - 1 \geq x_s, if\ t_t = t_s$
$x_t + 1 \leq x_s, if\ t_t = t_s$
$t_t - 1 \geq t_s, if\ x_t = x_s$

Figure 4.4: Relative positions between two FETs.

Internal Pin (P_{IN}) for FET

P_{IN} refers to the source, drain, and gate of each FET. Each pin's location is dynamically determined by the placement formulation (Section 4.3.1) and is associated with the flow formulation to solve routing by applying the dynamic pin allocation (DPA) scheme [26].

External Pin (P_{EX}) for I/O Pin Access

P_{EX} represents standard cell's I/O pins. Vertices (depicted in purple squares in Fig. 4.3) interconnected to P_{EX} on the M2 layer are defined as candidates for each I/O pin-access point. One of these candidates is assigned as the I/O pin-access point by our flow formulation. The routed metal segments on M2 and M3 layers including the assigned vertices represent the I/O pin of a standard cell for the block-level routing phase.

4.3 Proposed VFET SDC Synthesis Framework

We implement flow formulations and dynamic pin allocation (DPA) constraints for the concurrent P&R by utilizing the same methodology of [26]. Based on the framework, we further develop new constraints which characterize the unique placement and routing features of many-tier VFETs such as the absence of diffusion sharing and multiple stacking of transistors. Furthermore, we devise efficient objective functions to improve pin-accessibility and reduce the number of (high-resistance) vertical interconnections.

4.3.1 Transistor Placement

We utilize the conventional floorplanning approach (i.e., *Relative Positioning Constraint (RPC)*) for the transistor placement problem [77]. All transistor positions are represented by four RPCs as shown in Fig. 4.4. Unlike the conventional FET architectures, there are no diffusions sharing between horizontally adjacent FETs on the same tier because of the vertically oriented shape of VFETs. However, vertically stacked FETs on the same *x*-coordinate form a diffusion sharing between adjacent source/drain nodes. Therefore, we set additional constraints to avoid the placement of two FETs that have no identical net information on the adjacent tier when those

FETs' *x*-coordinates are the same as expressed in (4.1).

$$\begin{cases} |t_t - t_s| \geq 1, & \text{if } \bigvee_{p_i^t \in P^t, p_i^s \in P^s} \left(n(p_i^t) = n(p_i^s) \right) = true \\ |t_t - t_s| \geq 2, & \text{otherwise} \end{cases} \tag{4.1}$$

4.3.2 In-cell Routing

We apply grid-based conditional design rule-aware multi-commodity network flow theory to formulate the in-cell routing problem as described in [25, 27].

Dummy FET Gate Control

In the placement of many-tier VFET SDC, the number of transistors in each P/N-region of a cell is not always equal to the number of maximum placement grids that can accommodate transistors (i.e., $|X| \times |T|$). In this case, there exist dummy FETs that have no gate connections and behave like regular conducting channels. Therefore, we set a conditional constraint to prevent from utilizing gates of dummy FETs as a normal routing path, presented in Algorithm 8. If a dummy FET's gate is detected on a certain vertex v (Line 5), all of the vertical flow variables that are routed from/to the vertex v for a net n and commodity m are set to false (Lines 6, 7). Expression (4.2) detects a dummy FET on a certain vertex v_{x_v,y_v,z_v}.

$$\begin{gathered} \neg\Big(\bigvee_{\forall F_s \in F} m_{v_{x_v,y_v,z_v},p_1^s} \Big) \wedge \Big(\bigvee_{\forall F_s \in F, z_{v2} \neq z_v, z_{v2} \in L} m_{v_{x_v,y_v,z_{v2}},p_1^s} \Big), \\ \forall x_v \in X, \forall y_v \in \{2,4\}, \forall z_v \in L, L = \{2 \cdot t - 1 | t = 1,2,...,T\} \end{gathered} \tag{4.2}$$

4.3.3 Multi-Objective Optimization

Our framework has multiple objectives associated with the P&R problems for SDC layout design. The cell size is defined as the maximum occupation of vertical tracks by FETs as shown

Algorithm 8 Dummy FET Gate Control

```
for $t = 0$ to $|T| - 1$ do
    for $x_v = 0$ to $|X| - 1$ do
        for $y_v = 2, 4$ do                                        ▷ 2 for P-FET, 4 for N-FET
            $z_v \leftarrow 2 \cdot t + 1$;                        ▷ $z_v$: gate interconnect layer
            if (Dummy FET is detected on a vertex $v_{x_v,y_v,z_v}$ (4.2)) then
                $f_m^n(v,u) = false, \forall n \in N, u \in \{v_F, v_B\}$;
            end if
        end for
    end for
end for
```

in (4.3). The tier distribution (TD) is defined as the sum of each FET's tier location as shown in (4.4). The horizontal pin-cost (HP) and vertical pin-cost (VP) are defined as the sum of the number of adjacent I/O pins within an interference distance d_{int} in the horizontal and vertical directions as shown in (4.5) and (4.6), relatively. The number of TM (i.e., top metal layer) tracks is defined as the number of occupied top-metal layer routing tracks in a generated cell as shown in (4.7). The total metal length (ML) is the weighted sum of the routed metal segments as shown in (4.8)[3]. We optimize these multiple objectives in the light of "lexicographic" order with an optimization feature of the optimization modulo theories (OMT) [18][74]. The objectives are ranked in the order of emphasized, as described in (4.9).

$$\textbf{Placement (Cell Size) :}\ \max\left\{x_s \,\middle|\, F_s \in F\right\} \tag{4.3}$$

$$\textbf{Placement (TD) :}\ \sum_{Fs \in F} \left(T - t_s\right) \tag{4.4}$$

[3] Substantially, routing layers' multiple interchanges on timing critical paths are undesirable due to the severe performance loss by the high resistance of VIA elements. Thus, we determine the weighted cost $w_{v,u}$ of VIA metal segments by 4× higher than that of horizontal and vertical metal segments.

Pin-accessibility (HP/VP) :

$$\sum_{k=1}^{X} \sum_{n \in N_{EX}} \sum_{l \neq n, l \in N_{EX}} \bigvee_{v_x = k, p \in {P_{EX}}_k^l} e_{v,p}^{l} \tag{4.5}$$

$${P_{EX}}_k^l = \{p | p \in P_{EX}, k - d_{int} \leq x_p \leq k + d_{int}\}$$

$$\sum_{k=1}^{H} \sum_{n \in N_{EX}} \sum_{l \neq n, l \in N_{EX}} \bigvee_{v_y = k, p \in {P_{EX}}_k^l} e_{v,p}^{l} \tag{4.6}$$

$${P_{EX}}_k^l = \{p | p \in P_{EX}, k - d_{int} \leq y_p \leq k + d_{int}\}$$

$$\textbf{Routability (\#}TM\textbf{ Track) :} \sum_{k=1}^{H} \bigvee_{e_{v,u} \in E_k} m_{v,u} \tag{4.7}$$

E_k=Set of Top-metal Layer Edges in k_{th} Track

$$\textbf{Routing (Total ML) :} \sum_{e_{v,u} \in E} \left(w_{v,u} \times m_{v,u}\right) \tag{4.8}$$

Lexicographic Minimization:

(a) CellSize, (b) TD, (c) HP/VP, (d) #TMTrack, (e) TotalML (4.9)

4.4 Experimental Results

We have implemented the proposed framework in *Perl/SMT-LIB* 2.0 standard-based formula and validated it on a Linux desktop with 3.6GHz AMD Ryzen5 3600 CPU and 32GB memory. The SMT Solver *Z3* (ver. 4.8.5) [20] is used to produce the optimized solution.

4.4.1 Experimental Environment

SDC Generation

We first generate 12 1-tier and 2-tier SDCs in Table 4.2 by using cell netlists from Silvaco 45*nm* open cell library [93]. To explore the scaling impact of many-tier VFETs on cell-level and block-level area, we select and generate 30 representative SDCs [94, 95] as specified in Table 4.3 for many-tier 7T VFET architecture with the number of tiers ranging 1-4 tiers by using ASAP7 [83] SDC netlists. We also generate 4.5T GAA Nanosheet FET (GAAFET) SDCs with buried power rails by scaling 4.5T FinFET SDCs [96], based on the effective width ratio [97] between FinFET and Nanosheet structures with the concurrent P&R framework [26] for the comparison with VFETs.

We adopt the same number of horizontal routing tracks (i.e. 5 tracks), location of FETs (i.e., $2^{nd}/4^{th}$ horizontal tracks for P-FET/N-FET), and conditional design rules with [10] to compare our proposed concurrent P&R results to the known results for the conventional sequential P&R approach. The conditional design rule [27, 26] parameters are specified (MAR/EOL/VR/PRL/SHR/MPO=1/0/0/1/1/2) to have the same routing results as [10] for fair comparisons[4].

Block-level P&R

We employ three open-source RTL designs [98], M0 Core, M1 Core, and AES, which respectively have 17K, 20K, and 14K instances. We perform the block-level analysis through the commercial P&R tool [8]. We generate LEF format from our framework's SDC layout solutions by matching the top-layer of each many-tier VFET SDC to the M2 layer and the following layers to VIA12 and the M1 layers, regardless of the number of tiers to obtain the simplification of

[4] We employ the grid-based representative conditional design rules, e.g., Minimum Area Rule (MAR), End-of-Line spacing (EOL), Via Rule (VR), Parallel Run Length (PRL), Step Heights (SHR), and Minimum Pin Opening (MPO) following the same principles of [25, 27, 26].

Table 4.2: 1-tier/2-tier VFETs cell area comparison. Area Impr. = Cell Area Improvement (reference = [10]).

Cell	#FET	Cell Area (#CPP)				Area Impr.(%)	
		Previous Work [10]		Proposed Work			
		1-tier	2-tier	1-tier	2-tier	1-tier	2-tier
AND2	6	3	2	3	2	0.0	0.0
AOI211	8	4	3	4	3	0.0	0.0
BUF	4	2	2	2	2	0.0	0.0
INV	2	1	1	1	1	0.0	0.0
MUX2	12	6	4	6	3	0.0	25.0
NAND4	8	4	2	4	2	0.0	0.0
NOR3	6	3	2	3	2	0.0	0.0
OAI21	6	3	2	3	2	0.0	0.0
XOR2	10	5	3	5	3	0.0	0.0
HA	16	8	5	8	5	0.0	0.0
FA	28	14	10	14	7	0.0	30.0
DFF	28	14	10	14	7	0.0	30.0
Average		**5.58**	**3.83**	**5.58**	**3.25**	**0.0**	**15.2**

the routing configuration. For example, the M3/M5/M7/M9 layers of 1/2/3/4-tier VFETs are converted into an M2 layer in the generated LEF formats for the block-level P&R.

We set the number of masks for each BEOL according to [1]. We use 36*nm* and 24*nm* for the contacted poly pitches (CPPs)/M1 pitch and M0/M2 metal pitches, respectively, by applying the design parameters from [85][5]. The pitches and widths of layers above M2 are set by referring [99]. We use the same power delivery network for both GAAFET and VFET, which consists of the top metal-layer power meshes (M8 and M9), intermediate power stripes (M3), and standard cell power rails (M2). Then, the power is delivered from M3, which is 4× wider than signal wires, to M2 using stacked vias. The M3 power stripes for standard cell power rails are placed per every 64 CPPs [100]. We use the 300 #DRVs threshold to measure the valid block-level area. As a common industrial practice, once the number of DRVs increases beyond 300, the block layout is considered as too expensive to fix with laborious (sometimes, manual) engineering change orders.

[5] For GAAFET, the CPP/M1 pitches are 42*nm* and M0/M2 pitches are 24*nm*.

Table 4.3: Many-tier VFETs cell metric comparisons. #CPP Impr. = Cell Area Improvement Ratio (reference = 1-tier), ML/TT Incr. = Metal Length/#Top-layer Track Increment Ratio (reference = 1-tier).

Cell Specification			Cell Area (#CPP)				Metal Length (ML)				#Top-layer Track (TT)				#CPP Impr. (%)			ML Incr. (%)			TT Incr. (%)			Average
Name	#FET	#Net	1-tier	2-tier	3-tier	4-tier	1-tier	2-tier	3-tier	4-tier	1-tier	2-tier	3-tier	4-tier	2-tier	3-tier	4-tier	2-tier	3-tier	4-tier	2-tier	3-tier	4-tier	Runtime
AND2x2	6	7	4	3	2	2	95	100	119	119	2	3	3	3	25.0	50.0	50.0	5.3	25.3	25.3	50.0	50.0	50.0	(s)2.1
AND3x1	8	9	4	3	2	2	116	95	108	108	4	5	1	1	25.0	50.0	50.0	-18.1	-6.9	-6.9	25.0	-75.0	-75.0	16.0
AND3x2	8	9	5	3	3	2	113	141	141	154	3	5	5	3	40.0	40.0	60.0	24.8	24.8	36.3	66.7	66.7	0.0	14.3
AOI21x1	6	8	6	4	3	3	131	180	155	155	2	3	1	1	33.3	50.0	50.0	37.4	18.3	18.3	50.0	-50.0	-50.0	35.0
AOI22x1	8	10	8	4	4	3	176	197	244	243	2	3	3	3	50.0	50.0	62.5	11.9	38.6	38.1	50.0	50.0	50.0	2584.7
BUFx2	4	5	3	2	2	2	67	67	67	67	2	2	2	2	33.3	33.3	33.3	0.0	0.0	0.0	0.0	0.0	0.0	0.8
BUFx3	4	5	4	3	3	3	115	118	118	118	1	1	1	1	25.0	25.0	25.0	2.6	2.6	2.6	0.0	0.0	0.0	1.4
BUFx4	4	5	5	4	4	4	136	137	137	137	1	1	1	1	20.0	20.0	20.0	0.7	0.7	0.7	0.0	0.0	0.0	2.2
BUFx8	4	5	10	8	8	8	224	232	230	230	1	1	1	1	20.0	20.0	20.0	3.6	2.7	2.7	0.0	0.0	0.0	15.9
DFFHQNx1	24	17	12	6	5	4	277	305	328	265	3	5	4	5	50.0	58.3	66.7	10.1	18.4	-4.3	66.7	33.3	66.7	12547.6
FAx1	24	17	12	6	4	4	254	263	273	273	2	5	5	5	50.0	66.7	66.7	3.5	7.5	7.5	150.0	150.0	150.0	2607.6
INVx1	2	4	2	1	1	1	22	31	31	31	0	0	0	0	50.0	50.0	50.0	40.9	40.9	40.9	0.0	0.0	0.0	0.4
INVx2	2	4	2	2	2	2	42	42	42	42	2	2	2	2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.6
INVx4	2	4	4	4	4	4	107	107	107	107	1	1	1	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.7
INVx8	2	4	8	8	8	8	191	191	191	191	1	1	1	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	20.4
NAND2x1	4	6	4	2	2	2	79	71	71	71	0	3	3	3	50.0	50.0	50.0	-10.1	-10.1	-10.1	300.0	300.0	300.0	1.4
NAND2x2	4	6	8	4	4	4	139	187	198	198	0	2	1	1	50.0	50.0	50.0	34.5	42.4	42.4	200.0	100.0	100.0	5.5
NAND3x1	6	8	9	6	3	3	166	174	181	181	0	0	5	5	33.3	66.7	66.7	4.8	9.0	9.0	0.0	500.0	500.0	23.4
NAND3x2	6	8	18	12	6	6	309	276	307	351	0	0	2	1	33.3	66.7	66.7	-10.7	-0.6	13.6	0.0	200.0	100.0	12552.2
NOR2x1	4	6	4	2	2	2	79	71	71	71	0	3	3	3	50.0	50.0	50.0	-10.1	-10.1	-10.1	300.0	300.0	300.0	1.2
NOR2x2	4	6	8	4	4	4	139	187	250	250	0	2	1	1	50.0	50.0	50.0	34.5	79.9	79.9	200.0	100.0	100.0	6.4
NOR3x1	6	8	9	6	3	3	166	174	168	168	0	0	5	5	33.3	66.7	66.7	4.8	1.2	1.2	0.0	500.0	500.0	30.0
NOR3x2	6	8	18	12	6	6	309	276	319	426	0	0	2	1	33.3	66.7	66.7	-10.7	3.2	37.9	0.0	200.0	100.0	6029.8
OAI21x1	6	8	6	4	3	3	137	194	164	164	2	2	1	1	33.3	50.0	50.0	41.6	19.7	19.7	0.0	-50.0	-50.0	30.9
OAI22x1	8	10	8	4	4	3	176	197	244	243	2	3	3	3	50.0	50.0	62.5	11.9	38.6	38.1	50.0	50.0	50.0	2254.1
OR2x2	6	8	4	3	2	2	95	100	119	119	2	3	3	3	25.0	50.0	50.0	5.3	25.3	25.3	50.0	50.0	50.0	2.1
OR3x1	8	9	4	3	2	2	116	95	108	108	4	5	1	1	25.0	50.0	50.0	-18.1	-6.9	-6.9	25.0	-75.0	-75.0	12.4
OR3x2	8	9	5	3	3	2	113	140	140	154	3	5	5	3	40.0	40.0	60.0	23.9	23.9	36.3	66.7	66.7	0.0	14.1
XNOR2x1	10	9	8	5	3	3	190	165	206	261	1	2	5	3	37.5	62.5	62.5	-13.2	8.4	37.4	100.0	400.0	200.0	73.0
XOR2x1	10	9	8	5	3	3	190	165	206	261	1	2	5	3	37.5	62.5	62.5	-13.2	8.4	37.4	100.0	400.0	200.0	62.9
Average			**7.0**	**4.5**	**3.5**	**3.3**	**149.0**	**155.9**	**168.1**	**175.5**	**1.4**	**2.3**	**2.5**	**2.2**	**35.2**	**50.0**	**52.4**	**4.7**	**12.8**	**17.8**	**66.7**	**81.0**	**59.5**	**1298.3**

4.4.2 Sequential vs. Concurrent P&R

Table 4.2 presents comparisons of cell area across 1-tier/2-tier VFET cells. Compared to the previous sequential P&R approach [10], we observe a 15.2% cell-area reduction (3.83 → 3.25) on average for 2-tier SDCs, whereas both SDCs in 1-tier architecture have the same minimum cell area. In particular, MUX2/FA/DFF cells are generated with the minimum-achievable cell width (i.e., $|F|/(2{\cdot}T)$). These results demonstrate that the proposed concurrent P&R approach achieves a smaller cell area than that of the sequential approach as the number of tiers increases.

4.4.3 Single Cell Comparisons

Cell Area

Table. 4.3 presents cell metric comparisons of many-tier VFET SDCs. The average cell area (i.e., #CPP) respectively reduces by 35.2%, 50.5%, and 52.4% for 2-tier, 3-tier, and 4-tier VFETs over 1-tier VFET. The average reduction rates of 2/3/4-tier VFETs are less than the theoretical maximum rates of 50%/66.7%/75%, respectively. This is due to (i) the structural restriction that only allows vertical stacking of serially connected FETs, (ii) the existence of large FETs whose widths are larger than one, (iii) the netlist configuration which can not be evenly distributed through the vertical tiers, and (iv) the minimum pin-opening constraint (MPO) [26] for improving the pin-accessibility. For example, we observe that INVs have the same area and metal lengths regardless of the number of tiers, because they have only 2 FETs with different widths[6]. NAND3x2 results in the maximum-achievable cell area reduction of 66.7% in the 3-tier configuration. However, it does not show a further reduction in the 4-tier configuration because it has three N-FETs with a width of six. Therefore, we can estimate the trajectory by further optimization efforts of the cell area through the appropriate separation and distribution of large width FETs along the vertical tiers.

Fig. 4.5 shows the comparisons of cell area for 4.5T GAAFET and many-tier VFETs, estimated by applying the design parameters described in Section 4.4.1. Since (i) the unit cell-height of VFET (i.e., 7T=168*nm*) is larger than 4.5T GAAFET (i.e., 4.5T=108*nm*) and (ii) GAAFET cells with higher drive strength (i.e., NAND3x2, NOR3x2) have the smaller number of CPPs because of the different power rail type, more number of horizontal tracks and the higher effective width [97], some VFET SDCs result in the larger cell area. As a result, the average cell area for all 30 SDCs has been increased by 4.9% for 1-tier VFET whereas the average cell area has been reduced by 32.1%, 47.6%, and 50.1% for 2-tier, 3-tier, and 4-tier VFETs, respectively.

[6] 1-tier INVx1 uses two vertical tracks to satisfy MPO constraint that requires at least two pin-openings for each I/O pins.

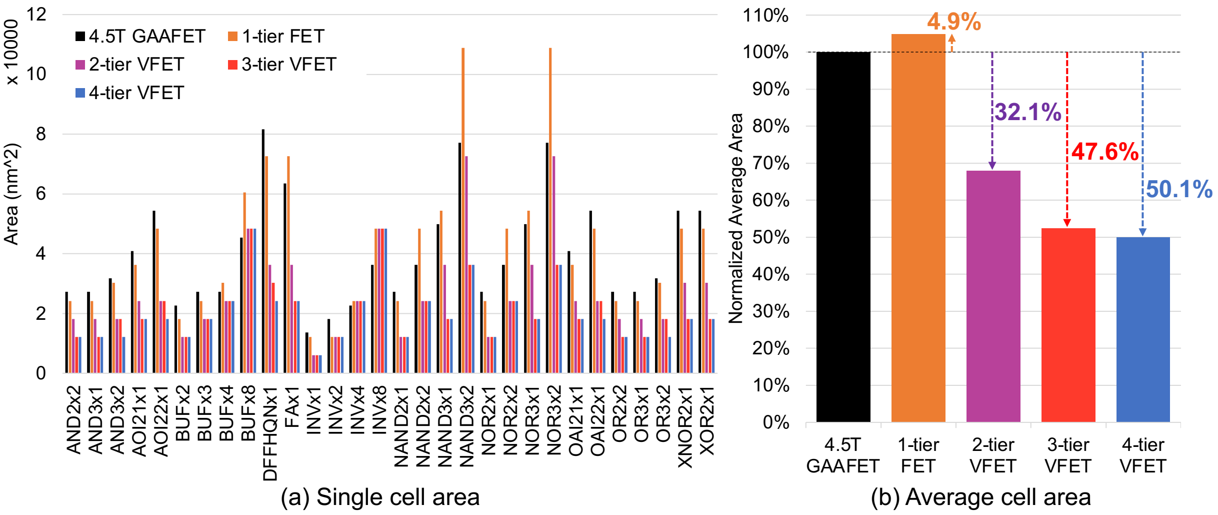


Figure 4.5: Cell area comparison of 30 representative SDCs. (a) Single-cell area, and (b) Average-cell area.

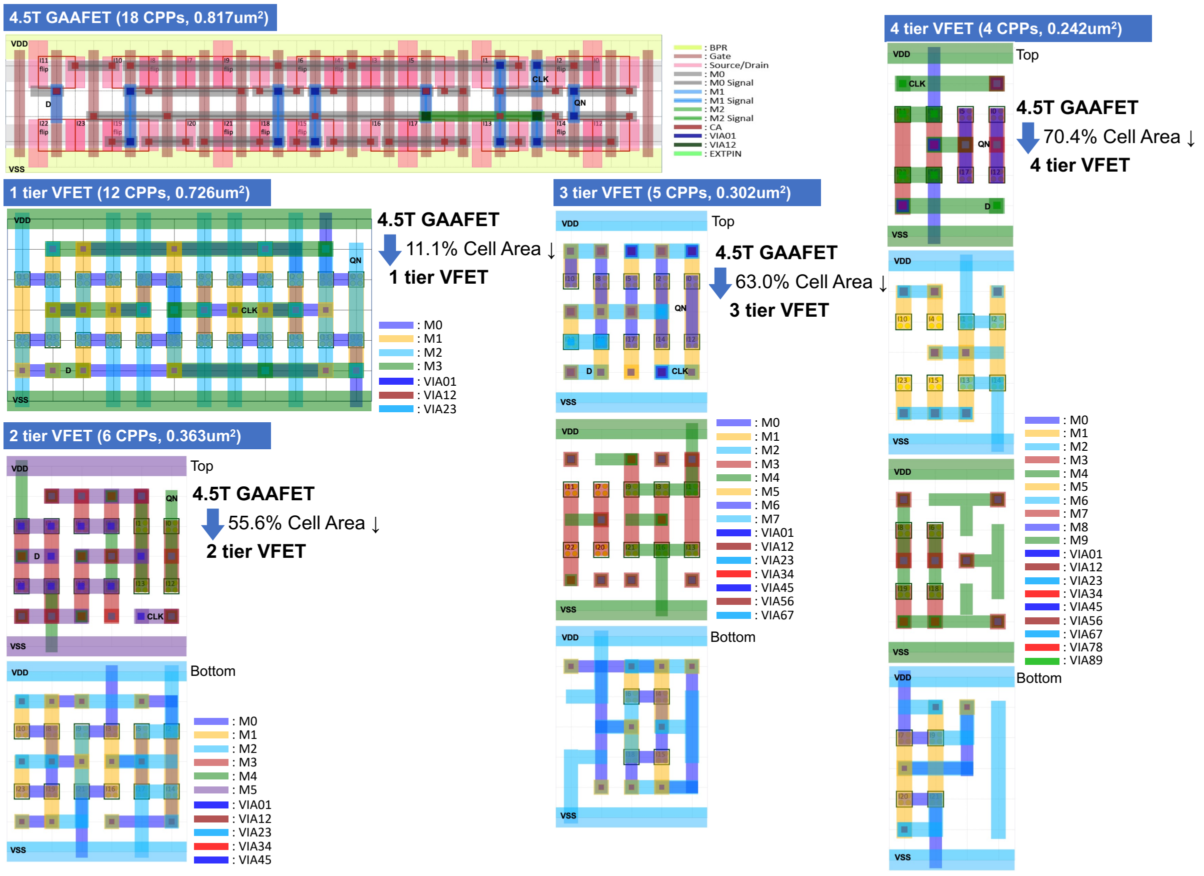


Figure 4.6: Layouts of 4.5T GAAFET and many-tier VFET DFFHQNx1 SDC.

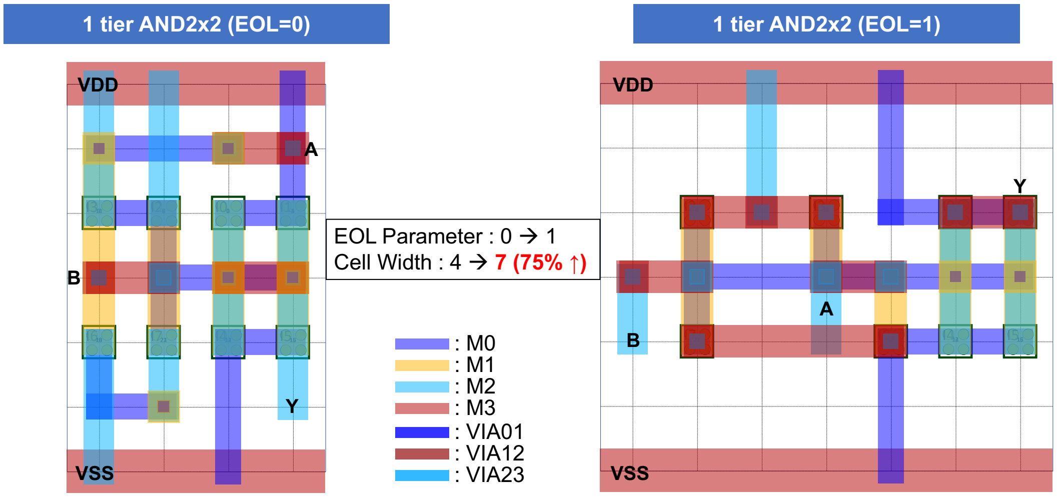
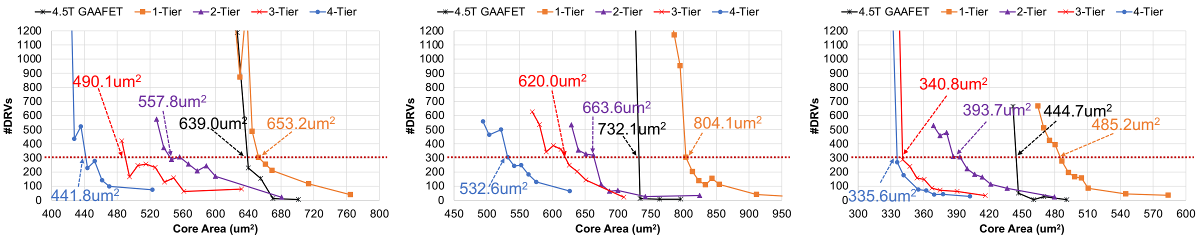


Figure 4.7: Layouts of 1-tier AND2x2 VFET SDC with different EOL parameters.



M0 Core M1 Core AES core

Figure 4.8: Block-level core area comparison of M0 Core, M1 Core, and AES for 4.5T GAAFET and many-tier VFETs.

Fig. 4.6 depicts the layouts of DFFHQNx1 for 4.5T GAAFET and many-tier VFET architectures. Compared to the GAAFET, VFETs respectively achieve the footprint reductions by 11.1%, 55.6%, 63.0%, and 70.4% for each increment of tiers.

The EOL design rule has the most significant impact on the width of VFET SDCs because the NFETs and PFETs reside on the same track, resulting in a reduced number of vertical tracks. Fig. 4.7 shows that the cell width increases by 75% with the EOL=1 for 1-tier AND2x2 VFET cell[7]. Therefore, VFET requires EOL=0 to achieve the maximum area reduction over the conventional structure.

[7] The EOL parameter defines the minimum number of empty grids for EOL spacing. For example, EOL=1 defines that the EOL spacing between two metal segments needs to be at least one grid. Thus, the actual minimum EOL spacing for EOL=1 is 60*nm* (i.e., 2·CPP-M2 width) for M2 layer.

Table 4.4: Metal length decomposition. Metal = lateral routing metal segments, Via = vertical routing metal segments.

Cell	Metal Length										Increment Ratio (Reference = 1-tier)					
	4.5T GAAFET		1-tier		2-tier		3-tier		4-tier		2-tier		3-tier		4-tier	
	Metal	Via	Metal	Via	Metal	Via	Metal	Via	Metal	Via	Metal	Via	Metal	Via	Metal	Via
AND2x2	17	52	39	56	36	64	31	88	31	88	-7.7%	14.3%	-20.5%	57.1%	-20.5%	57.1%
AND3x1	23	60	40	76	31	64	28	80	28	80	-22.5%	-15.8%	-30.0%	5.3%	-30.0%	5.3%
AND3x2	25	64	45	68	41	100	41	100	38	116	-8.9%	47.1%	-8.9%	47.1%	-15.6%	70.6%
AOI21x1	48	88	51	80	52	128	43	112	43	112	2.0%	60.0%	-15.7%	40.0%	-15.7%	40.0%
AOI22x1	78	144	68	108	61	136	64	180	51	192	-10.3%	25.9%	-5.9%	66.7%	-25.0%	77.8%
BUFx2	13	44	27	40	23	44	23	44	23	44	-14.8%	10.0%	-14.8%	10.0%	-14.8%	10.0%
BUFx3	22	56	39	76	38	80	38	80	38	80	-2.6%	5.3%	-2.6%	5.3%	-2.6%	5.3%
BUFx4	22	56	48	88	45	92	45	92	45	92	-6.3%	4.5%	-6.3%	4.5%	-6.3%	4.5%
BUFx8	47	84	92	132	88	144	86	144	86	144	-4.3%	9.1%	-6.5%	9.1%	-6.5%	9.1%
DFFHQNx1	121	212	133	144	93	212	96	232	93	172	-30.1%	47.2%	-27.8%	61.1%	-30.1%	19.4%
FAx1	104	156	126	128	87	176	73	200	73	200	-31.0%	37.5%	-42.1%	56.3%	-42.1%	56.3%
INVx1	5	24	10	12	11	20	11	20	11	20	10.0%	66.7%	10.0%	66.7%	10.0%	66.7%
INVx2	7	28	18	24	18	24	18	24	18	24	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
INVx4	16	40	39	68	39	68	39	68	39	68	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
INVx8	37	64	75	116	75	116	75	116	75	116	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
NAND2x1	19	52	31	48	23	48	23	48	23	48	-25.8%	0.0%	-25.8%	0.0%	-25.8%	0.0%
NAND2x2	34	72	59	80	51	136	50	148	50	148	-13.6%	70.0%	-15.3%	85.0%	-15.3%	85.0%
NAND3x1	46	96	62	104	62	112	45	136	45	136	0.0%	7.7%	-27.4%	30.8%	-27.4%	30.8%
NAND3x2	93	152	121	188	112	164	95	212	95	256	-7.4%	-12.8%	-21.5%	12.8%	-21.5%	36.2%
NOR2x1	19	52	31	48	23	48	23	48	23	48	-25.8%	0.0%	-25.8%	0.0%	-25.8%	0.0%
NOR2x2	46	76	59	80	51	136	62	188	62	188	-13.6%	70.0%	5.1%	135.0%	5.1%	135.0%
NOR3x1	46	96	62	104	62	112	44	124	44	124	0.0%	7.7%	-29.0%	19.2%	-29.0%	19.2%
NOR3x2	93	152	121	188	112	164	95	224	110	316	-7.4%	-12.8%	-21.5%	19.1%	-9.1%	68.1%
OAI21x1	54	120	57	80	54	140	44	120	44	120	-5.3%	75.0%	-22.8%	50.0%	-22.8%	50.0%
OAI22x1	78	144	68	108	61	136	64	180	51	192	-10.3%	25.9%	-5.9%	66.7%	-25.0%	77.8%
OR2x2	17	52	39	56	36	64	31	88	31	88	-7.7%	14.3%	-20.5%	57.1%	-20.5%	57.1%
OR3x1	23	60	40	76	31	64	28	80	28	80	-22.5%	-15.8%	-30.0%	5.3%	-30.0%	5.3%
OR3x2	25	64	45	68	40	100	40	100	38	116	-11.1%	47.1%	-11.1%	47.1%	-15.6%	70.6%
XNOR2x1	79	120	82	108	73	92	54	152	57	204	-11.0%	-14.8%	-34.1%	40.7%	-30.5%	88.9%
XOR2x1	78	128	82	108	73	92	54	152	57	204	-11.0%	-14.8%	-34.1%	40.7%	-30.5%	88.9%
Average	**44.5**	**86.9**	**60.3**	**88.7**	**53.4**	**102.5**	**48.8**	**119.3**	**48.3**	**127.2**	**-11.4%**	**15.6%**	**-19.1%**	**34.6%**	**-19.8%**	**43.5%**

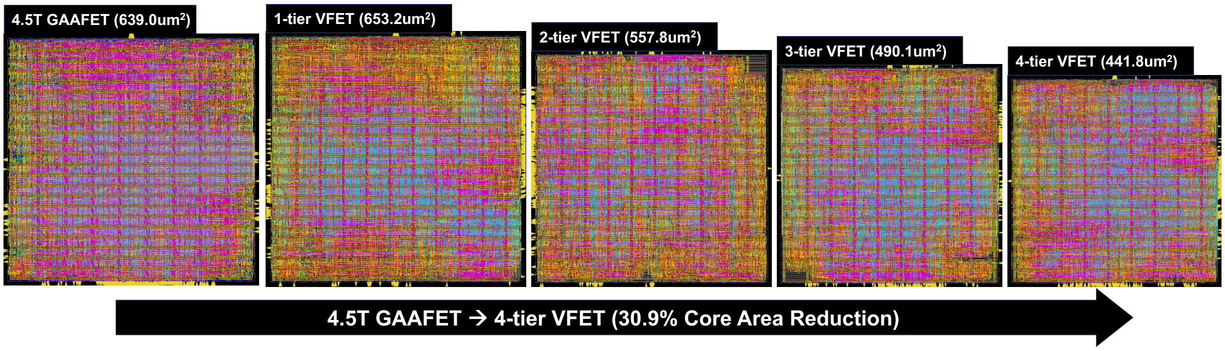
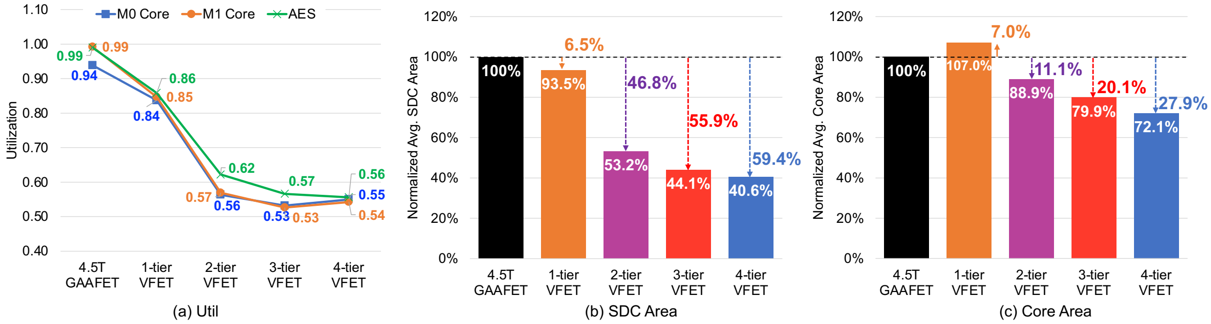

4.5T GAAFET (639.0um^2)
1-tier VFET (653.2um^2)
2-tier VFET (557.8um^2)
3-tier VFET (490.1um^2)
4-tier VFET (441.8um^2)
4.5T GAAFET → 4-tier VFET (30.9% Core Area Reduction)

Figure 4.9: Block-level P&R results of M0 Core.

Metal Length

The average metal length of VFET SDCs (as shown in Table. 4.3) increases by 4.7%, 12.8%, and 17.8% for 2-tier, 3-tier, and 4-tier VFETs over 1-tier VFET, respectively. Table 4.4



(a) Util

(b) SDC Area

(c) Core Area

Figure 4.10: Block-level design utilization, SDC, and core area comparisons for 4.5T GAAFET and Many-tier VFETs. SDC and core area present a normalized average area of M0 Core, M1 Core, and AES.

presents the decomposition of each SDC's total metal length into lateral (i.e., Metal) and vertical (i.e., Via) direction routing segments[8]. The number of vias (Via) respectively increases by 15.6%, 34.6%, 43.5%, whereas the number of metal segments (Metal) respectively decreases by 11.4%, 19.1%, and 19.8% for 2-tier, 3-tier, and 4-tier VFETs over 1-tier VFET. Compared to the 4.5T GAAFET (i.e., conventional structure), VFETs show higher usage of average metal length on both Metal and Via. In particular, 1-tier VFET requires 35.5% more Metal on average (i.e., 44.5 → 60.3) because of the unique cell structure that does not have Source/Drain nodes sharing. These results show a clear trade-off between the area and the performance of SDCs across the cell structures. The increment of VFET tiers provides a smaller cell area at the cost of higher parasitic resistance due to the increasing usage of vertical interconnections.

Top-layer Track Occupation

The average number of occupied tracks (i.e., #TT) on the top-metal layer (in Table. 4.3) increases by 66.7%, 81.0%, and 59.5% for 2-tier, 3-tier, and 4-tier VFETs, respectively. #TT does not have monotonic increment by increasing the number of tiers because we minimize the track occupation of the top-metal layer. The routing congestion is highly correlated with the effective cell area which depends on several factors such as the netlist (i.e., the number and width of

[8] We set the weighted cost of Via by 4× higher than that of Metal.

FETs) and tier configuration (i.e., the number of tiers and minimum vertical tracks) of each SDC. For example, #TT of XOR2x1 increases from 1 to 5 as the #CPP decreases from 8 to 3 by the increment of #tier. However, the #TT reduces again from 5 to 3 because the 4-tier configuration has the same #CPP but demanding more routing layers. These additional routing layers contribute to the #TT reduction at the cost of more metal length (i.e., 206@3-tier→261@4-tier).

4.4.4 Block-level Area Comparisons

Fig. 4.8 illustrates block-level P&R results of 4.5T GAAFET and many-tier VFETs for three open-source designs. The red dotted lines represent the 300 #DRVs, indicating the threshold for valid block-level area. Fig. 4.9 shows that 4-tier VFET achieves 30.9% reduction in core area over 4.5T GAAFET for M0 Core. For all designs, the average core areas are reduced by 11.1%, 20.1%, and 27.9%, whereas the SDC areas respectively reduced by 46.8%, 55.9%, and 59.4% for 2-tier, 3-tier, and 4-tier VFETs, as shown in Fig. 4.10.

Despite the 6.5% of average SDC area reduction, 1-tier VFET shows a 7.0% increment in the average core area. This is because of the larger unit cell height as well as the relative pin-accessibility loss induced by the reduced cell width, resulting in the diminished block-level area benefit. The steep reduction of the design utilization for many-tier VFETs (Fig. 4.10(a), i.e., resulting in the diminished core area improvement over SDC area), indicates that the further scaling of vertical BEOL pitches can pull up the utilization and maximize the core area reduction by efficiently exploiting the SDC-area benefit of many-tier VFETs.

4.5 Conclusion

We propose an SMT-based many-tier VFET standard cell synthesis framework. The proposed framework simultaneously performs FET P&R with the extended constraints for many-tier VFET structures. We demonstrate that the proposed concurrent P&R approach obtains 15.2%

of average cell area reduction for 2-tier VFET compared to the sequential P&R [10]. Throughout the exploration for various many-tier VFET configurations up to four tiers, we show that the 4-tier VFET respectively achieves 50.1% and 27.9% of average area reduction on chip-level and block-level, over 4.5T GAAFET. Lastly, we find that there are still rooms to further improve, e.g., the higher parasitic resistance of many-tier VFETs and the thermal issue in a stacked logic transistors [88] call future research topics to obtain the maximum-achievable PPAC (power, performance, area, and cost) benefits through VFET.

This chapter contains materials from "Many-Tier Vertical Gate-All-Around Nanowire FET Standard Cell Synthesis for Advanced Technology Nodes", by Daeyeal Lee, Chia-Tung Ho, Ilgweon Kang, Sicun Gao, Bill Lin, and Chung-Kuan Cheng, which appears in IEEE Journal on Exploratory Solid-State Computational Devices and Circuits, June 2021. The dissertation author was the primary investigator and author of this paper.

Chapter 5

Engineering Change Order (ECO) Automation

5.1 Introduction

With the relentless scaling toward advanced technology nodes, increasingly sophisticated IC fabrication constraints (e.g., fewer routing tracks, higher pin density, and complicated conditional design rules) bring rapidly increasing design complexity [1]. This leads to non-trivial challenges for physical design, particularly in the routing stage. The number of remaining design rule violations (DRVs) after place-and-route (P&R) has become one of the most crucial metrics for an automatic IC layout solution, since back-end designers must manually stitch/modify all DRVs via implementation of engineering change orders (ECOs) at the post-layout stage. In particular, achieving the pin accessibility needed to resolve DRVs is a critical, time-consuming engineering task just before tapeout, and is therefore a critical bottleneck in the advanced-node IC design process [101, 102].

To mitigate pin accessibility-induced DRVs, several approaches are proposed to improve pin accessibility through detailed placement (DP) optimization [103, 65, 104, 105, 106, 107] and standard cell layout optimization [108, 109]. The authors of [103] perform DP optimization using a global routing solution as guidance, with pin accessibility modeled only in the form of pin density. Dynamic programming and deep learning-based DP optimizations considering each pin's access are developed in [104, 105, 106, 107]. In [65], the authors introduce a measurement of inaccessible pins in a cell to optimize DP. However, these models have limited capability of comprehending design rules in detailed routing (DR). The works of [109] and [108] have proposed pin accessibility-driven cell layout optimization frameworks for improving routability at block-level. Recently, the authors of [110] perform replacement of inaccessible cells with diverse cell layouts in terms of pin locations and number of access points, in the ECO stage. The applicability of these works is intrinsically limited due to the lack of holistic consideration for the block-level design steps (e.g., DP and DR).

In this work, we propose *CoRe-ECO*, a **Co**ncurrent **Re**finement framework which simultaneously performs incremental DP and DR, along with cell replacement, at the **ECO** stage. To our knowledge, this is the first work to present a concurrent co-optimization of DP, DR, and cell replacement within block-level P&R. Our main contributions are summarized as follows.

- We propose a concurrent refinement framework which simultaneously performs DP, DR, and cell replacement for each local window (i.e., switchbox) covering DRV locations, to determine ECOs at post-layout stage. We devise a novel dynamic cell allocation (DCA), merging the refinement steps into a single-step optimization.
- *CoRe-ECO* performs (i) placement adjustment such as horizontal/vertical shifting, horizontal flipping, and cell swapping; (ii) pin-length extension with a recommendation of adopting alternative master cells while maintaining the same functionality; and (iii) routing optimization to seek the best-quality DRV-clean solution.
- *CoRe-ECO* minimizes perturbation of the given layout by utilizing a satisfiability modulo theories (SMT) solver, enabling multi-objective optimization.
- We validate the proposed *CoRe-ECO* framework with various testcases, demonstrating successful fixing of pin accessibility-induced DRVs through the proposed ECO flow along with total wirelength optimization.

The remaining sections are organized as follows. Section 5.2 describes the proposed *CoRe-ECO* framework. Section 5.3 discusses our experimental setup and results. Section 5.4 concludes the paper.

5.2 CoRe-ECO Framework

This section introduces an overview of the proposed framework, grid-based P&R architecture, refinement operations, switchbox generation, perturbation-minimized optimization, and SMT formulation.

Table 5.1: Notations for the proposed *CoRe-ECO* framework.

Term	Description
T	Set of instances in a switchbox
t	t^{th} instance
l_t	0-1 indicator if instance t is flipped
x_t	x-axis coordinate of lower-left corner of t
y_t	y-axis coordinate of placement row of t
y^t_{org}	Initial y-axis coordinate of placement row of t
x^t_{org}	Initial x-axis coordinate of lower-left corner of t
w_t	Width of instance t
P^t	Set of internal pins of instance t
p^t_i	i^{th} pin of instance t
$a_{ext}(p^t_i)$	Set of extended vertices of pin p^t_i
$V(V_i)$	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)
$a(v)$	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
f^n_m	A two-pin subnet connecting a source and sink, i.e., a commodity
$e^n_{v,u}$	0-1 indicator if $e_{v,u}$ is used for n
$f^n_m(v,u)$	0-1 indicator if $e_{v,u}$ is used for commodity f^n_m
$m_{v,u}$	0-1 indicator if there is a metal segment on $e_{v,u}$
$C^n_m(v,u)$	Capacity variable for $e_{v,u}$ of commodity f^n_m

5.2.1 Framework Overview

We formulate a conventional (sequential) layout refinement process as a constraint satisfaction problem (CSP) with variables and constraints to integrate placement adjustment and routing steps into a single multi-objective optimization problem. We adopt the SMT solver *Z3* [20] to solve the given optimization problem. Fig. 5.1 illustrates an overview of our framework. Given standard cell library, switchbox, and instance slack information,[1] our framework simultaneously obtains the optimal solution that strictly satisfies the constraints integrated into our novel DCA scheme. Our notations are described in Table 5.1.

[1] We define *instance slack* as the worst slack among pins of a given instance.

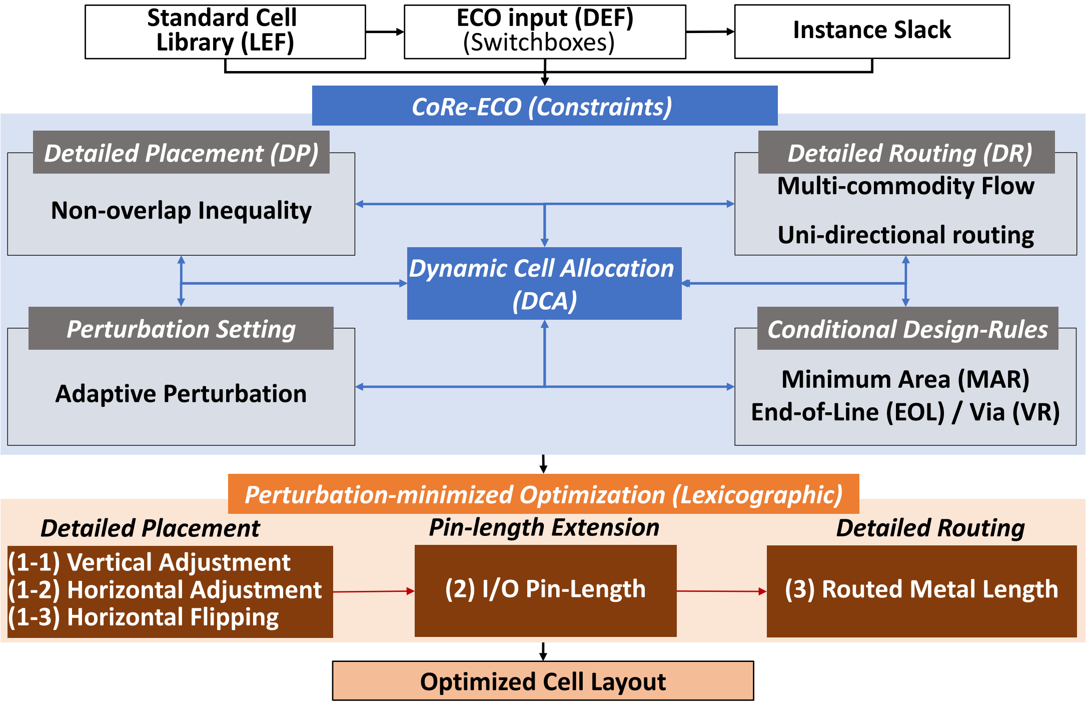


Figure 5.1: Framework Overview.

5.2.2 Grid-based Place-and-Route Architecture

We define the grid-based placement and 3-D routing graph composed of four metal layers (i.e., $M1$-$M4$) as shown in Fig. 5.2. Cell instances and I/O pins are aligned with $M1$ vertical tracks and gate poly of the standard cell. Inspired by [25], we adopt supernodes to cover the multiple candidates for each pin, either the I/O pin of a standard cell (i.e., P_{IN}) or the outer pin of a switchbox (i.e., P_{EX}). The location of P_{IN} is dynamically determined by placement formulation and is associated with the flow formulation for routing through DCA. P_{EX} interconnects the internal pins inside the switchbox to the outer pins and is located along the boundary of the switchbox which corresponds to the pre-routed result.

The horizontal routing grid (i.e., $M2$ and $M4$) consists of eight tracks per placement row and the vertical routing grid (i.e., $M3$) is aligned with the cell placement grid[2]. Note that we focus

[2] We assume an on-grid routing scheme for each routing layer, consistent with sub-7nm multi-patterning technologies and IC practitioners' restriction of preferred routing direction per each layer [24].

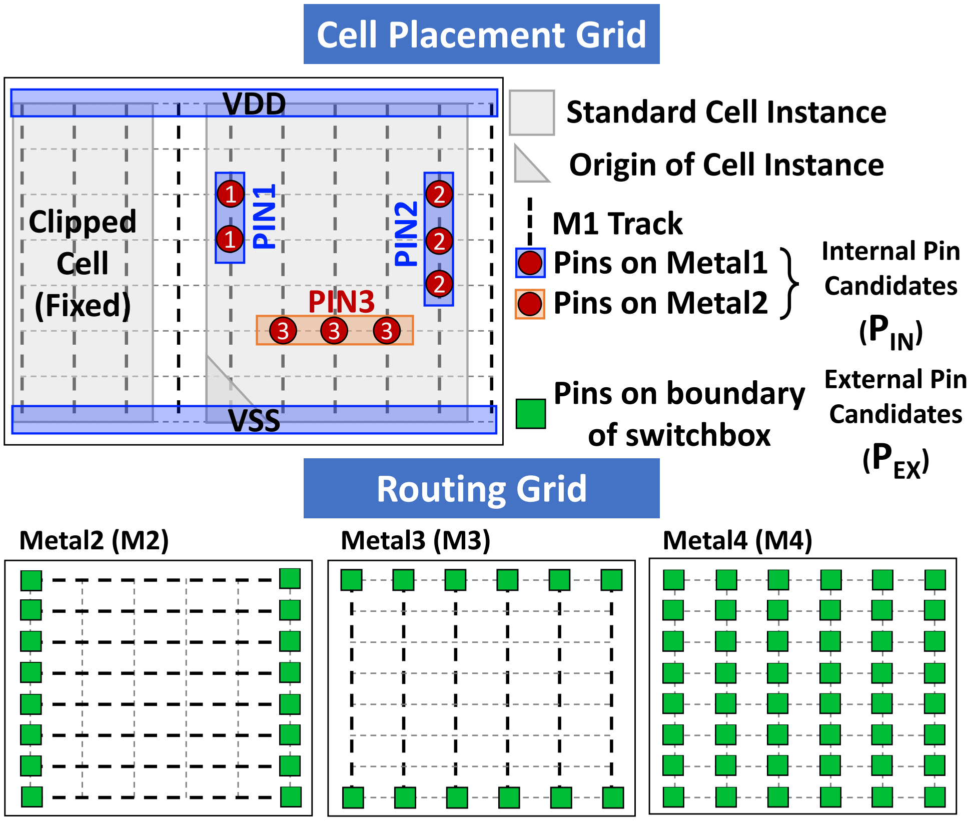


Figure 5.2: Grid-based Place-and-Route Architecture.

on *M*2-*M*3 layers assuming that *M*2 and *M*3 have the same metal pitches, because our proposed framework targets the correction of the pin accessibility-induced DRVs. Thus, *M*4 layer only contains VIA34 elements as the external pins for connections to the upper layer.

5.2.3 Refinement Operations

CoRe-ECO uses placement adjustment and pin-length extension as refinement operations within its adaptive perturbation method.

Placement Adjustment

Fig. 5.3 illustrates possible adjustments during DP to solve the pin accessibility-induced DRV. When the given placement layout in a switchbox (Fig. 5.3(a)) does not have feasible routing

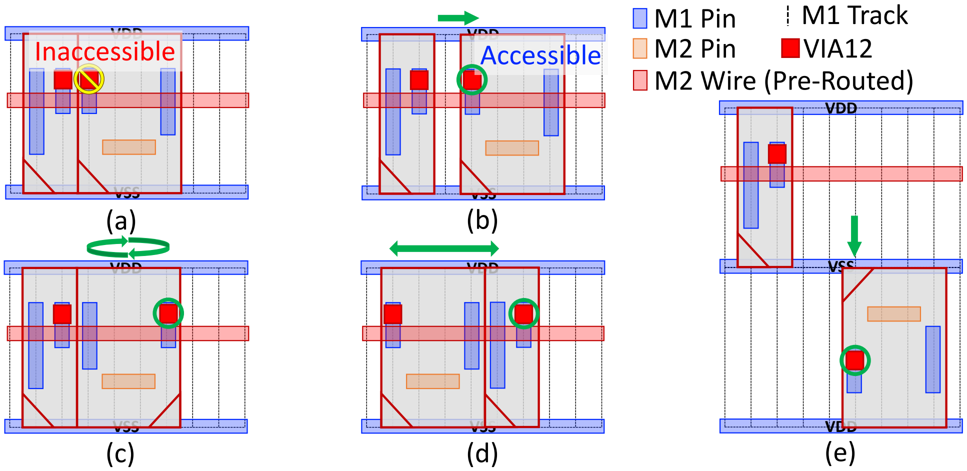


Figure 5.3: Placement Adjustment. (a) Initial Placement. (b) Horizontal Shifting. (c) Horizontal Flipping. (d) Cell Swapping. (e) Vertical Shifting.

solutions due to an inaccessible *M*1 pin, *CoRe-ECO* adjusts the placement of instances in the switchbox by horizontal/vertical shifting, horizontal flipping, and cell swapping as shown in Figs. 5.3(b)-(e). Note that cell swapping is only performed between two adjacent instances.

Pin-length Extension

As a rule, master cells with minimum pin-length are preferred for use during initial P&R, e.g., to achieve better timing optimization. However, in the ECO stage, engineers should consider adopting alternative master cells for specific instances, so as to improve pin accessibility or to achieve the target design specification. While works of 20+ years ago [111, 112, 113] pursued *liquid library* approaches, today it is well-understood that a library cell must be qualified before it is used in a production chip. Thus, our framework suggests minimum-achievable pin-length extensions needed to fix DRVs: by only extending the metal segments of I/O pins, we enable engineers to adopt (i.e., swap in) alternative master cells that maintain the same functionality while minimizing the magnitude of undesired timing impact on each cell. Furthermore, our framework minimizes the effective number of alternative master cells by the perturbation-minimized optimization described in Section 5.2.5. Note that this simple pin-extension satisfies all the

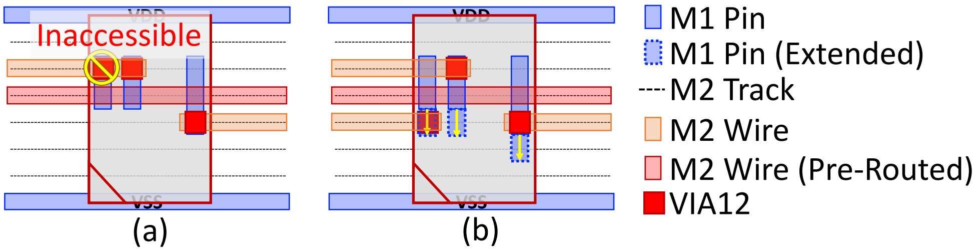


Figure 5.4: Pin-length Extension. (a) Initial Placement. (b) Pin-length Extension.

conditional design rules for generating standard cell libraries that are described below in Section 5.3.1. Fig. 5.4 illustrates pin-length extension that resolves a DRV caused by the inaccessible pin (Fig. 5.4(a)) by extending the instance's I/O pins (Fig. 5.4(b)) without DP adjustments. We only apply the pin-extension for $M1$ pins because (i) we do not allow routing on $M1$ layer, and (ii) $M2$ pins are directly accessible from BEOL layers.

Adaptive Perturbation

We utilize instance timing slacks to set the perturbation range (i.e., the range of the vertical and horizontal adjustments) by considering timing margins of each instance. For the instances with the worst slacks, we fix the placement and the routed wires to prevent the deterioration of timing characteristics. For the rest of the instances, the applicable range of perturbation is set by the input parameter settings. Flipping of instances and extension of pin-lengths are allowed for all instances except for the fixed instances.

5.2.4 Switchbox Generation

The proposed *CoRe-ECO* generates a switchbox by extracting the instance, pin, net, and obstacle information from each local window covering DRVs. Fig. 5.5(b) visualizes the generated switchbox representation from the local window depicted in Fig. 5.5(a).

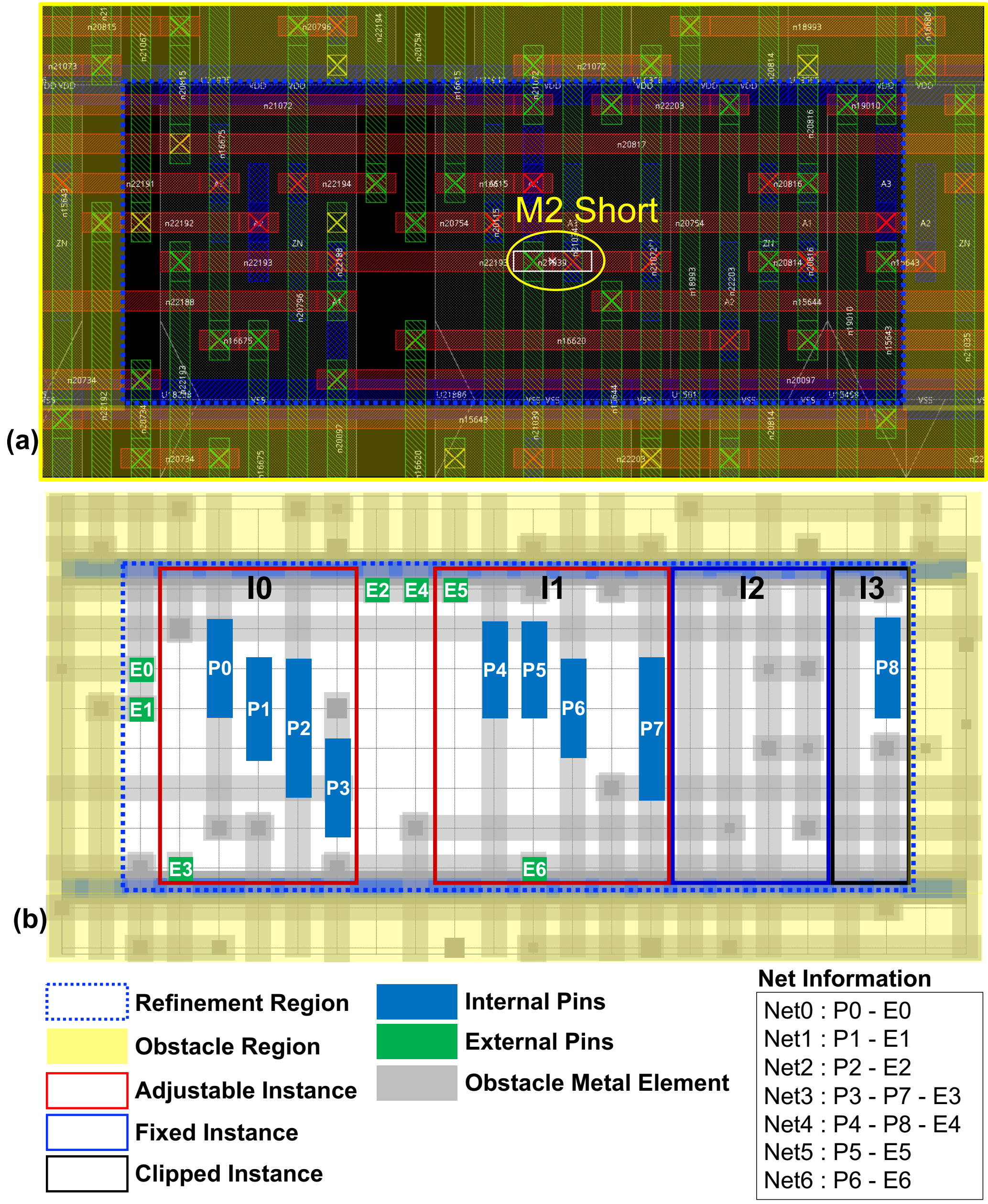


Figure 5.5: Switchbox Generation. (a) Local window with 1 DRV, displayed by the commercial tool [8]. (b) Visualization of the generated switchbox.

Instances

We separate the instances in the refinement region (i.e., blue dotted box) according to the given P&R results. We first fix the placement and the corresponding pre-routed nets of the instances (i.e., I2) whose timing slacks are less than a predefined worst-slack upper bound. Then, the instances inside the region are extracted as adjustable instances (i.e., I0, I1) that are allowed for the refinement operations. The clipped instances (i.e., I3) are partially included in the refinement region. Therefore, those instances are not adjustable, but their partial pins/nets are extracted for the routing optimization.

Pins/Nets

The I/O pins in the adjustable and clipped instances are extracted as internal pin candidates (i.e., P0 - P8). The external pins (i.e., E0 - E6) are extracted along the boundary of the refinement region if there is a connection from the internal pins to the outside of the refinement region. The net information defines new interconnections between internal pins and external pins.

Obstacle Elements

The switchbox has two types of obstacle elements (i.e., gray rectangles). First, we extract the routed metal elements in the obstacle region (i.e., yellow solid box) and set those elements as obstacles to check for DRVs on the boundary of the refinement region. Second, we consider the routed elements inside the refinement region as obstacles if those elements do not have any connections to the internal/external pins or they are connected to the fixed instances.

5.2.5 Perturbation-minimized Optimization

The proposed *CoRe-ECO* has multiple objectives associated with the refinement operations and routing problems. To honor the given (initial) DP and DR solution, we minimize the perturbations made by refinement operations as well as the total metal length. The vertical adjustment (ΔV) and horizontal adjustment (ΔH) are respectively defined as the total amount of vertical and horizontal shifts of the adjustable instances as shown in (5.1) and (5.2). The horizontal flipping (ΔF) is defined as the total number of flipped instances as shown in (5.3). The pin-length extension (ΔP) is defined as the sum of extended pin-lengths as shown in (5.4). The routing (ML) is the sum of routed VIA/Metal elements (i.e., $VIA12$, $M2$, $VIA23$, and $M3$). Each element has the same weight in the calculation of ML because we separate the objective functions for each type of element as shown in (5.5).

CoRe-ECO simultaneously optimizes these multiple objectives in light of the "lexicographic" order described in (5.6). In other words, the objectives are optimized according to the priority order given by **LexMin**; for each given objective, this effectively induces a single-objective optimization problem under the constraining condition that optimizes the higher-priority objectives.

$$\textbf{Vertical Adjustment } (\Delta\textbf{V})\textbf{ : } \sum_{t \in T} (\left| y_t - y^t_{org} \right|) \tag{5.1}$$

$$\textbf{Horizontal Adjustment } (\Delta\textbf{H})\textbf{ : } \sum_{t \in T} (\left| x_t - x^t_{org} \right|) \tag{5.2}$$

$$\textbf{Horizontal Flipping } (\Delta\textbf{F})\textbf{ : } \sum_{t \in T} l_t \tag{5.3}$$

$$\begin{gathered}\textbf{Pin-length Extension } (\Delta\textbf{P})\textbf{ : } \sum_{e_{p,r} \in E} \left(w_{p,r} \times m_{p,r} \right), \\ \forall p \in P^t, \forall t \in T, \forall r \in a_{ext}(p)\end{gathered} \tag{5.4}$$

$$\begin{gathered}\textbf{Routing } (\textbf{ML}\{\#VIA12\textbf{, }\#M2\textbf{, }\#VIA23\textbf{, }\#M3\})\textbf{ : } \sum_{e_{v,u} \in E} m_{v,u} \\ E\text{=Sets of each } VIA12\text{, } M2\text{, } VIA23\text{, and } M3 \text{ Element}\end{gathered} \tag{5.5}$$

$$\begin{gathered}\textbf{LexMin:} \text{ (a) } \Delta\text{V, (b) } \Delta\text{H, (c) } \Delta\text{F, (d) } \Delta\text{P,} \\ \text{(e) ML } \{\text{(1) } \#VIA12\text{, (2) } \#M2\text{, (3) } \#VIA23\text{, (4) } \#M3\}\end{gathered} \tag{5.6}$$

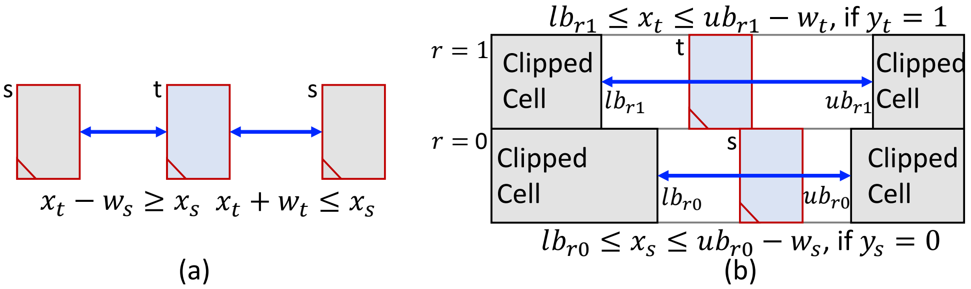


Figure 5.6: Placement Constraints. (a) Relative positions between two instances in the same placement row. (b) Boundary condition of each placement row.

Algorithm 9 Set RPC Constraint (Instances t, s)

```
if $y_t = y_s$ then                              ▷ t and s are on the same placement row
    if $x_t >= x_s + w_s$ then
        $x_t \geq x_s + w_s$;                    ▷ t is on the right side of s
    else if $x_t + w_t < x_s$ then
        $x_t + w_t \leq x_s$;                    ▷ t is on the left side of s
    else
        Unsatisfiable condition;
    end if
end if
```

5.2.6 SMT Formulation

Placement Formulation

We utilize the conventional floorplanning approach (i.e., *Relative Positioning Constraint (RPC)*) for the placement problem [77]. All instance positions in a specific placement row can be represented by two RPCs as shown in Fig. 5.6(a). At least one of the two inequalities holds for each pair $t \neq s$ through the SMT expression described in Algorithm 9. The maximum adjustable column boundary of each instance t is determined by the placement row y_t due to the different composition of clipped instances in each row as shown in Fig. 5.6(b). These geometric conditions determine the position and the flip status of the instance.

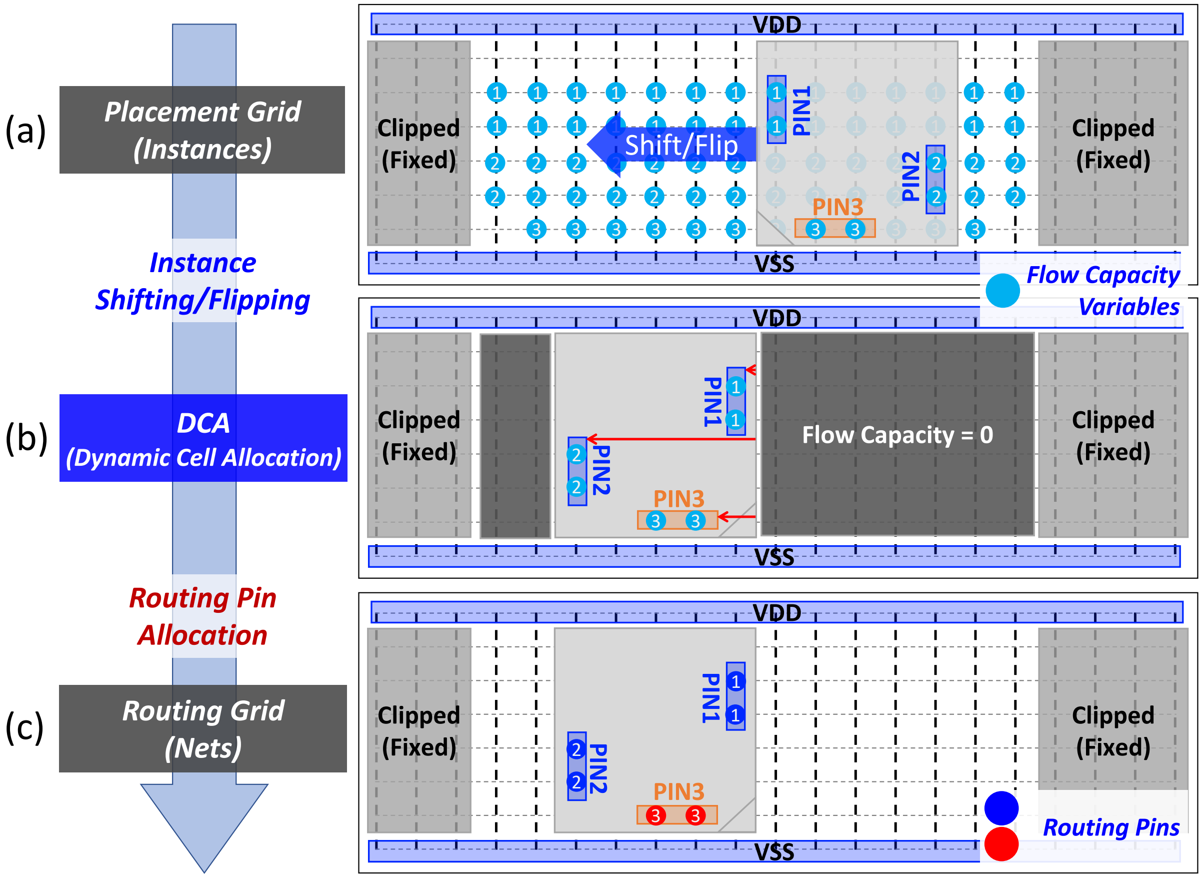


Figure 5.7: Dynamic Cell Allocation (DCA).

Dynamic Cell Allocation (DCA)

Every pin in each instance has its corresponding flow capacity variable $C_m^n(p,r)$ for certain net n and commodity m on the corresponding vertices of the placement grid, according to the shape and relative position of the pin in the instance as well as the possible adjustment range of each instance (see Fig. 5.7(a)). When locations of instances are determined by the placement formulation, the flow capacity variables of each instance's pins are conditionally assigned to the corresponding locations according to the placement status of each instance (i.e., shifted, flipped) as described in Algorithm 10. First, the coordinates of each pin are determined by the location of each instance and flip status (Lines 1–6). Then, all capacities $C_m^n(p,r)$ outside the range of each pin are assigned to zero (Lines 7–11) as depicted in Fig. 5.7(b).

Algorithm 10 Set Flow Capacity Control Constraint $(C^n_m(p,r))$

```
/* x coordinate (resp. placement row) of a routing grid r: $x_r$ (resp. $y_r$) */
/* x coordinate (resp. placement row) of a pin p: $x_p$ (resp. $y_p$) */
/* p is either source or sink of a net n and commodity m */
/* origin's column, origin's row, flipping of a instance i: $i_x$,$i_y$,$i_f$ */
/* x offset from the instance origin of a pin p: $o_{x_p}$ (resp. $o_{y_p}$) */
if $i_f$ = False then
    $x_p = i_x + o_{x_p}$;
else
    $x_p = i_x - o_{x_p}$;
end if
$y_p = i_y + o_{y_p}$;
if $(x_r \neq x_p) \mid (y_r \neq y_p)$ then
    $C^n_m(p,r) = 0$;
else
    $C^n_m(p,r)$ is Determined by Routing Formulation;
end if
```

$$f^n_m(v=p,u=r) \leq C^n_m(p,r), \quad \forall r \in a(p), \forall r \in V_0 \tag{5.7}$$

Equation (5.7) associates the flow variable $f^n_m(v,u)$ with the flow capacity variable $C^n_m(p,r)$. Each $f^n_m(v,u)$ 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 $M1,M2$ (i.e., V_1,V_2). This enables our routing formulation to recognize the feasible sets of r in V_1,V_2 layers as routing pins, as depicted in Fig. 5.7(c).

Pin-length Extension

We generate flow variables for extendable pin candidates to enable pin-length extension when finding a routable solution, as illustrated in Fig. 5.8. The extendable pins are generated in both up/down directions from the uppermost/lowermost vertex of each I/O pin. We set different weights for the extendable pin candidates, proportional to their distance from the nearest I/O pins. These weights are used as the priority in our objective function (Equation (5.4)) for minimizing

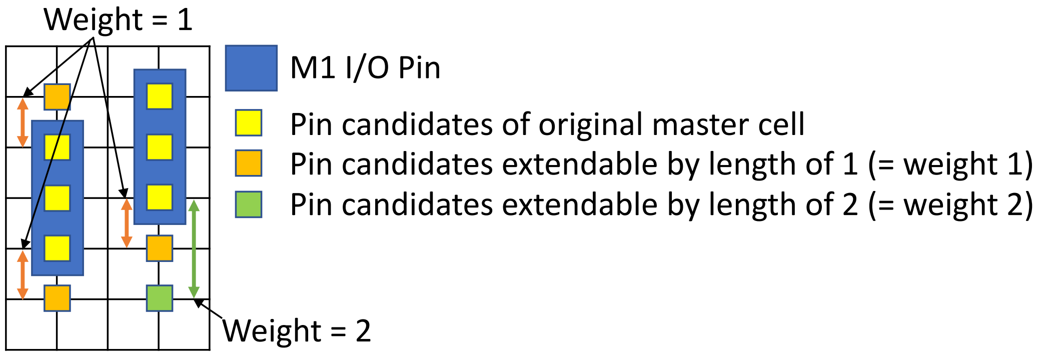

Weight = 1
M1 I/O Pin
Pin candidates of original master cell
Pin candidates extendable by length of 1 (= weight 1)
Pin candidates extendable by length of 2 (= weight 2)
Weight = 2

Figure 5.8: Weighted extendable pin candidates.

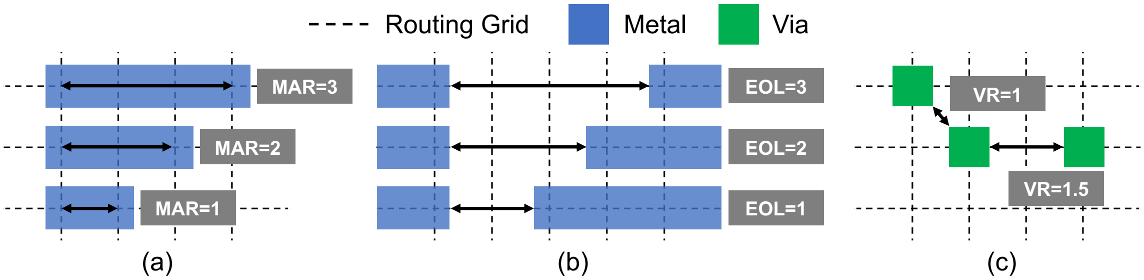


Figure 5.9: Grid-based conditional design rules. (a) MAR, (b) EOL, (c) VR.

the total length of the extended pins.

Routing Formulation

We use multi-commodity network flow and conditional design rules to formulate the DR problem, following the same principles as [114]. The flow formulation secures the routing path between the source and the sink for each commodity. Specifically, the refined constraints for *commodity flow conservation* and *vertex exclusiveness* in uni-directional edges are implemented in our framework to reduce the search space of the routing formulation. The conditional design rules work as constraints to route using design-rule violation-free paths. *CoRe-ECO* implements three fundamental grid-based design rules[3], namely, *Minimum Area (MAR)*, *End-of-Line Spacing (EOL)* and *Via Rule (VR)*, as illustrated in Fig. 5.9. MAR (Fig. 5.9(a)) defines the minimum

[3] In this work, we assume sub-7*nm* technologies that are based on Extreme Ultraviolet (EUV) lithography as in the previous work [115]. However, our framework is applicable to additional multi-pattern-aware design rules, such as Parallel Run Length (PRL) and Step Height Rule (SHR).

Table 5.2: Standard cell architectures.

Cell Architecture			Design Constraint	
#Fin	#Routing Tracks	Cell Height	Design Rule	Pin accessibility
3	6	8T	EUV-Loose (EL)	MPO3
3	6	8T	EUV-Tight (ET)	MPO3
2	4	6T	EUV-Loose (EL)	MPO2
				MPO3
			EUV-Tight (ET)	MPO2
				MPO3

number of grids that should be covered by the metal segments. EOL (Fig. 5.9(b)) defines the minimum number of grids between two metal segments. VR (Fig. 5.9(c)) defines the minimum distance (in L_2 norm) between vias.

5.3 Experiments

We have implemented the proposed *CoRe-ECO* framework in *Perl/SMT-LIB* 2.0 standard-based formula and validated on a Linux workstation with Intel (R) Xeon E5-2560L at 1.8GHz and 128GB memory. The SMT Solver *Z3* (version 4.8.5) [20] is used to produce the optimized solution.

5.3.1 Experimental Environment

Standard Cell Library Preparation

Using an SMT-based cell layout automation [116], we prepare six types of standard cell libraries with various cell architecture and design constraints presented in Table 5.2. We adopt two design rule sets that comprise combinations of specific design rule settings, as follows. EUV-Loose (EL) consists of MAR/EOL/VR = 1/1/1. EUV-Tight (ET) consists of MAR/EOL/VR = 1/2/1, inspired by [115]. We also generate two different types of cell libraries ensuring at least two and three I/O pin access points (i.e., MPO2 and MPO3). Then, for design enablement, we convert the primitive layout solutions of the SMT to LEF format. We assume the contacted poly

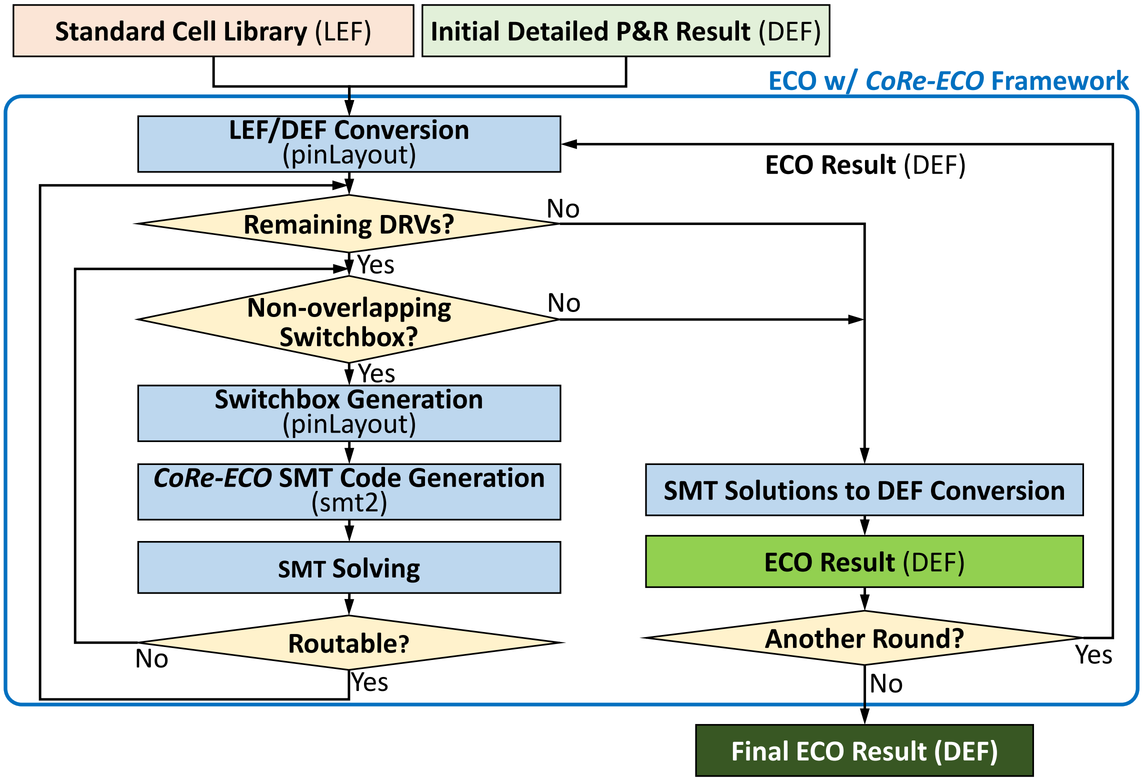


Figure 5.10: Overall ECO flow using *CoRe-ECO* framework.

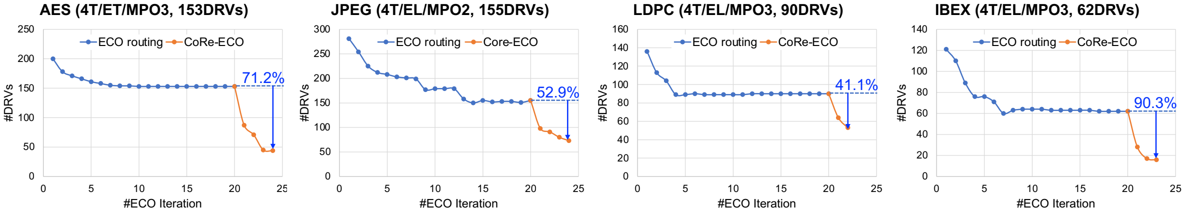


Figure 5.11: Example trends of the number of DRVs by the ECO routing and the proposed *CoRe-ECO* iterations.

pitch (CPP), metal pitch (MP), and cell height of 40, 40, and 280nm, respectively.[4] We also generate three additional LEFs that have cells with pin-lengths extended by 1, 2 and 3 grids, respectively, consistent with the cell height of the corresponding standard cell library.

[4] Since the layouts are fully grid-based, we consider the CPP and MP as pitches of grids.

Table 5.3: Experimental Statistics. RT = #routing tracks, DRSet = Design rule set, MPO = Minimum pin-opening parameter, WL = Total wirelength, Impr./Incr. = Improvement/Increment Ratio over ECO routing, #V/#H/#F/#P = #refined instances by Vertical/Horizontal Shifting/Flipping/Pin-Length Extension.

Design			Cell Library			ECO routing		Proposed *CoRe-ECO* Refinement								#ECO	Runtime
								#Remaining DRVs		WL		#refinements					
Name	#nets	#cells	RT	DRSet	MPO	#DRVs	WL(*um*)	#DRVs	Impr.	WL(*um*)	Incr.	#V	#H	#F	#P	Round	(h)
AES	13,958	13,694	6T	ET	3	40	46,272.16	2	95.0%	46,262.73	-0.020%	2	17	5	36	3	9.2
	13,912	13,648	4T	ET	2	74	45,968.76	13	82.4%	45,958.58	-0.022%	3	32	31	60	5	9.9
	13,938	13,674	4T	EL	3	41	45,797.96	0	100.0%	45,781.98	-0.035%	2	19	12	28	4	4.3
	13,802	13,538	4T	EL	2	146	45,035.54	56	61.6%	45,025.73	-0.022%	8	51	23	86	5	16
	13,867	**13,603**	**4T**	**ET**	**3**	**153**	**44,548.12**	**44**	**71.2%**	**44,528.05**	**-0.045%**	**25**	**104**	**45**	**79**	**4**	**18.2**
JPEG	70,543	70,518	6T	ET	3	79	144,905.98	30	62.0%	144,892.66	-0.009%	0	12	15	51	3	13.2
	71,491	**71,466**	**4T**	**EL**	**2**	**155**	**134,901.91**	**73**	**52.9%**	**134,893.09**	**-0.007%**	**3**	**30**	**14**	**73**	**4**	**25.3**
	71,177	71,152	4T	EL	3	37	133,558.22	18	51.4%	133,555.19	-0.002%	2	14	8	18	3	2.8
	70,932	70,907	4T	ET	2	198	132,739.50	132	33.3%	132,738.24	-0.001%	4	22	12	49	3	41.4
	70,008	69,983	4T	ET	3	68	136,852.58	42	38.2%	136,849.64	-0.002%	1	13	5	12	2	8.7
LDPC	57,133	55,081	6T	EL	3	21	732,917.12	8	61.9%	732,908.83	-0.001%	2	5	3	7	2	1.3
	57,138	55,086	4T	ET	2	12	737,957.04	7	41.7%	737,955.47	0.000%	0	3	1	1	1	0.24
	57,106	**55,054**	**4T**	**EL**	**3**	**90**	**751,812.13**	**53**	**41.1%**	**751,797.69**	**-0.002%**	**7**	**19**	**7**	**9**	**2**	**5.3**
IBEX	15,540	12,225	4T	EL	2	166	48,734.35	56	66.3%	48,730.00	-0.009%	12	40	16	80	5	15.8
	15,432	**12,117**	**4T**	**EL**	**3**	**62**	**48,426.72**	**6**	**90.3%**	**48,425.94**	**-0.002%**	**3**	**11**	**9**	**8**	**3**	**4.8**
	15,502	12,187	4T	ET	2	141	47,042.10	80	43.3%	47,045.31	0.007%	3	18	10	48	4	13.9
	15,179	11,864	4T	ET	3	93	49,346.74	32	65.6%	49,367.76	0.043%	0	9	14	13	3	5.1
Average						92.7	195695.11	38.4	**58.6%**	195689.23	**-0.003%**	**4.5**	**24.6**	**13.5**	**38.7**	**3.3**	**11.5**

Place-and-Route (P&R)

We validate our framework by using four open-source RTL designs AES, JPEG, LDPC [98], and IBEX [117]. We utilize $M2$-$M7$ layers as BEOL (Back End of Line). We assume the power/ground pins on $M1$ layer for the initial detailed routing (DR). However, since our proposed framework targets grid-based architecture and correction of the pin accessibility-induced DRVs, we focus on $M2$-$M3$ layers assuming that $M2$ and $M3$ layers have the same metal pitches. Two commercial tools [8][118] are used to generate the initial P&R layouts and to execute the following ECO routing. In the commercial tool, we perform 20 iterations of ECO routing until the commercial tool is unable to further reduce the number of DRVs (i.e., #DRVs) for most of the benchmark cases. The blue lines in Fig. 5.11 show the example trends of #DRVs through the ECO routing iterations for four representative cases from Table 5.3 (i.e., cases in bold). Note that we compare our work with the results of ECO routing because we are not able to fairly compare our work with the previous works, [103, 65, 104, 105, 106, 107, 110], due to (i) the different target design stage (i.e., DP optimization vs. ECO) and (ii) the lack of exact experimental settings.

Setting up the Perturbation Range

We set the 1% of worst slack cells and the routed nets connected to those cells as fixed instances and obstacle elements, respectively. For the rest of the instances, we set the perturbation range of the vertical and horizontal adjustments to two placement rows and eight poly pitches, respectively.

5.3.2 Design of Experiments

Fig. 5.10 illustrates an overview of the ECO flow utilizing our *CoRe-ECO* framework. Given a cell library and initial detailed P&R result, the new ECO round starts with converting these layout information to a *pinLayout* format for the proposed framework in the **LEF/DEF Conversion** step. Then, if there exist any remaining DRVs in the non-overlapping regions, we rip up the region and generate a switchbox representation in the **Switchbox Generation** step. Note that the switchboxes in the same ECO round cannot overlap because our framework could change the P&R in each switchbox, and it also refers to the horizontal/vertical obstacle regions for checking the design rules on the boundary of each switchbox. Given the switchbox representation, we generate an SMT code and solve the problem through the ***CoRe-ECO*** **SMT Code Generation** and **SMT Solving** steps. We iterate these refinement steps until we find a routable solution or there are no remaining DRVs or feasible switchboxes in non-overlapping regions. After the iterations, in the **SMT Solutions to DEF Conversion** step, we apply the DRV-clean solutions to the original DEF and generate a revised DEF for the next ECO round or publish as the final ECO result.

The ECO flow described above is fully automated, and each sequence (i.e., switchbox generation to SMT solving) can be executed in parallel through the multi-threaded operation. In this work, up to 24 threads are used for all testcases. For each DRV, our framework examines multiple switchboxes of various sizes (i.e., 10 - 25 vertical tracks and 1 - 5 placement rows) for several relative locations to the target DRV. The size of a routable switchbox for each DRV varies according to the existing P&R results, routing congestion, and locations of the adjacent DRVs. To minimize the perturbation of the placement as well as the runtime of **SMT Solving** step, our framework increases the size of the switchbox from the minimum (i.e., 10 vertical tracks × 1 row) to the maximum (i.e., 25 vertical tracks × 5 rows) in **Switchbox Generation** step until it finds a routable solution or fails.

5.3.3 Experimental Results

Statistics on the Proposed ECO Flow

Table 5.3 summarizes the experimental statistics of the proposed ECO flow for benchmark cases which consist of four base design circuits synthesized with various cell libraries described in Table 5.2. Column "ECO routing" represents the total number of DRVs (i.e., #DRVs) in target layers (i.e., $M2$-$M3$) and the total wirelength (i.e., WL) after the 20 iterations of ECO routing. The target DRVs mainly include "Cut Spacing" on $M1$–$M2$ layers and "Metal End-of-Line Spacing", and "Metal Short" on $M2$–$M3$ layers. Our *CoRe-ECO* framework reduces the remaining DRVs after ECO routing by 58.6% on average, with reductions ranging from 33.3% to 100.0%. Fig. 5.11 shows the trend of #DRVs versus iterations of ECO routing (i.e., the blue line), along with the following *CoRe-ECO* flow (i.e., the orange line) for four representative cases of each base benchmark circuit from Table 5.3 (i.e., cases in bold). The figure demonstrates that our framework can further improve the routability with the concurrent cell refinements and the routing optimization. The reduction of the average total wirelength by 0.003% shows that our framework has successfully minimized the wirelength despite the refinement of cell placement and pin-length extension. We observe that the number of ECO rounds and the total runtime depend on #DRVs and the benchmark circuit configurations. For all benchmark cases, *CoRE-ECO* performed 3.3 ECO rounds on average with an average runtime of 11.5 hours.

Table 5.4 presents the detailed refinement results of the AES benchmark circuit with

Table 5.4: Detailed Results (AES, 4T/ET/MPO2, 74 DRVs). #VTrack/#Row = #vertical tracks/placement rows, #Adj.Inst./#FixedInst. = #adjustable/fixed or clipped instances, #Inst. = #perturbed instances, #V/#H/#F/#P = #refined instances by Vertical/Horizontal Shifting/Flipping/Pin-Length Extension.

Index	ECO Round	Switchbox								#Cell Refinement				SMT Runtime(s)
		#Vtrack	#Row	#DRV	#Total Inst.	#Adj. Inst.	#Fixed Inst.	#Net	#Pin	#V	#H	#F	#P	
0	1st	18	2	1	5	3	2	15	33	0	2	0	1	11.0
1		19	1	2	4	3	1	9	22	0	0	0	2	1.3
2		18	2	1	4	3	1	13	29	0	1	1	1	8.5
3		14	1	1	3	2	1	6	17	0	0	0	1	0.6
4		14	1	2	3	2	1	6	15	0	0	0	0	0.6
5		21	2	1	7	3	4	15	43	0	0	0	3	25.4
6		17	1	1	3	3	0	7	17	0	2	0	1	3.4
7		15	2	1	4	2	2	9	21	0	1	0	0	28.8
8		14	1	1	3	2	1	8	17	0	0	0	1	0.7
9		14	1	1	2	0	2	6	14	0	0	0	0	0.4
10		18	2	1	8	5	3	16	41	0	0	3	3	11.8
11		16	1	1	2	1	1	6	13	0	0	0	1	0.6
12		15	1	2	4	2	2	6	16	0	0	1	2	0.7
13		22	2	3	9	6	3	19	42	0	3	2	5	20.2
14		16	1	1	3	3	0	8	18	0	2	1	2	1.4
15		14	1	1	2	2	0	5	11	0	1	0	0	0.4
16		14	1	1	2	1	1	4	12	0	0	0	0	0.5
17		14	1	1	2	2	0	4	10	0	0	1	0	0.4
18		14	1	1	2	1	1	4	9	0	1	0	0	0.4
19		14	1	1	2	2	0	6	14	0	0	0	0	1.0
20		18	2	1	6	3	3	16	38	0	0	1	2	5.7
21		14	1	1	3	2	1	6	14	0	2	1	0	0.5
22		16	1	1	3	2	1	7	16	0	1	2	2	0.8
23	2nd	18	1	1	4	3	1	10	25	0	0	1	1	2.2
24		14	1	1	3	2	1	7	15	0	0	0	1	0.7
25		14	1	1	2	1	1	3	7	0	0	0	0	0.3
26		15	2	1	5	2	3	12	29	0	0	0	0	13.7
27		14	1	1	2	1	1	7	18	0	0	1	1	0.8
28		14	1	1	2	1	1	5	11	0	1	0	0	0.4
29		19	2	1	7	6	1	15	42	0	1	1	3	54.6
30		16	2	1	5	4	1	16	34	0	1	0	0	5.6
31	3rd	29	2	1	10	8	2	27	63	0	1	1	5	588.9
32		18	5	2	13	11	2	34	77	0	0	3	2	859.9
33		17	2	7	7	5	2	14	38	2	3	2	2	6.3
34		19	3	3	10	7	3	24	54	0	1	1	4	109.1
35		20	3	1	9	8	1	25	67	0	1	4	4	1465.8
36		14	3	3	7	5	2	16	41	0	0	0	1	10.8
37		17	2	1	5	2	3	18	39	0	1	0	2	8.3
38		18	5	3	13	7	6	33	71	1	2	2	3	1841.8
39	4th	14	1	2	2	2	0	5	12	0	2	1	1	0.6
40		21	2	1	9	6	3	16	43	0	0	1	2	30.7
41	5th	16	2	1	7	3	4	12	28	0	2	0	1	6.5
Average		**16.6**	**1.7**	**1.5**	**5.0**	**3.3**	**1.6**	**11.9**	**28.5**	**0.1**	**0.8**	**0.7**	**1.4**	**122.2**
Total				61	208	139	69	500	1196	3	32	31	60	5131.84

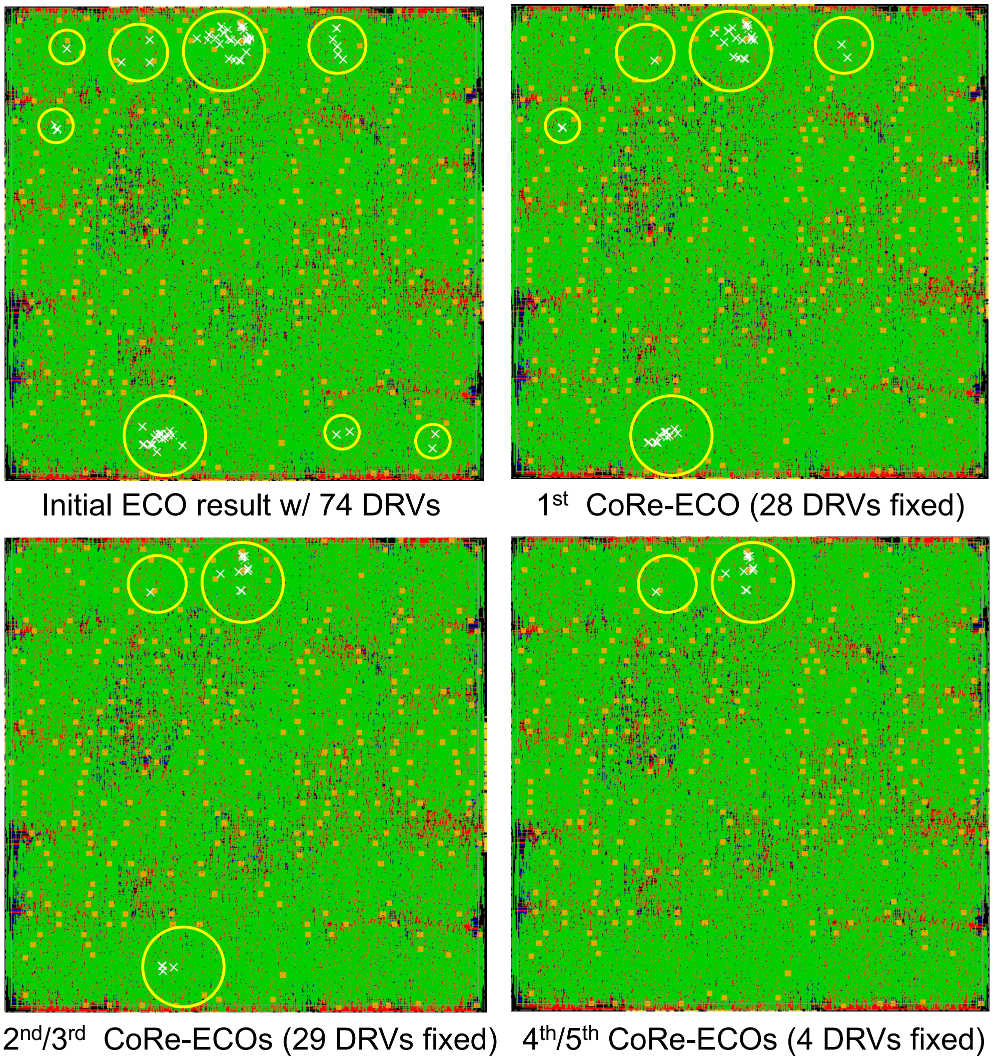
4T/ET/MPO2 cell library and 74 DRVs. The total of 5 ECO rounds with 42 switchboxes have been performed to resolve 61 out of all 74 DRVs. The average number of vertical tracks, placement rows, and DRVs in the switchboxes are 16.6, 1.1, and 1.5, respectively. Each switchbox includes 5.0 total/3.3 adjustable/1.6 fixed or clipped instances and 11.9 nets/28.5 pins on average. Through the 5 rounds of ECO flow, 126 out of 139 adjustable instances have been perturbed in the placement or the length of pins. 6 out of 42 ECO cases have been fixed without any perturbation. And 7 and 7 cases require the extension of pin-lengths or the re-placement of instances, respectively. The remaining 22 cases are routable by only changing both instance placement and pin-lengths. The average runtime per switchbox is less than 3 minutes and the switchboxes up to 18×34 vertical/horizontal tracks, 7 adjustable instances, 33 nets, and 71 pins (i.e., Index 38) have been solved within 31 minutes. Fig. 5.12 shows the reduction of DRVs in full-chip layouts by multiple ECO rounds utilizing *CoRe-ECO* framework, displayed by a commercial tool [8]. The yellow circles indicate regions with DRVs.

Example Refinement Operations

Fig. 5.13 shows an example of refinement operations in our proposed ECO framework. Fig. 5.13(a) depicts a switchbox of index 38 case in Table 5.4. The switchbox consists of 7 adjustable / 6 clipped(i.e. fixed) cell instances with 3 'M3 Short' DRVs in 18×34 vertical/horizontal tracks. Fig. 5.13(b) illustrates the DRV-clean solution with the refinement operations (i.e., placement adjustment and pin-length extension) and the optimized routing in terms of the metal length. Note that the elements in gray color represent the obstacles inside the switchbox and $M1$ I/O pins are not displayed in Fig. 5.13(b). The pre-routed wires, that (i) are connected to the fixed instances or (ii) have no internal connection inside the switchbox or (iii) exist outside the switchbox, are regarded as obstacles.

Placement Adjustment

The placement of the instance I2 in Fig. 5.13(b) has been adjusted from the placement row 0 to 1 and horizontally shifted from the vertical track 12 to 10. The instance I0 has shifted in the same placement row from the vertical track 8 to 10. Instances I0 and I5 have been flipped on


Initial ECO result w/ 74 DRVs

1st CoRe-ECO (28 DRVs fixed)

2nd/3rd CoRe-ECOs (29 DRVs fixed)

4th/5th CoRe-ECOs (4 DRVs fixed)

Figure 5.12: DRV reductions by *CoRe-ECO* rounds for AES (4T/ET/MPO2).

the same placement locations.

Pin-length Extension

The I/O pins of the instance I1, I3, and I5 in Fig. 5.13(b) have been extended by 1–2 to maximize the pin accessibility. In the proposed ECO flow, the respective master cell of each of these instances is replaced with the additional master cell with extended pin-lengths, in the **SMT Solutions to DEF Conversion** stage.

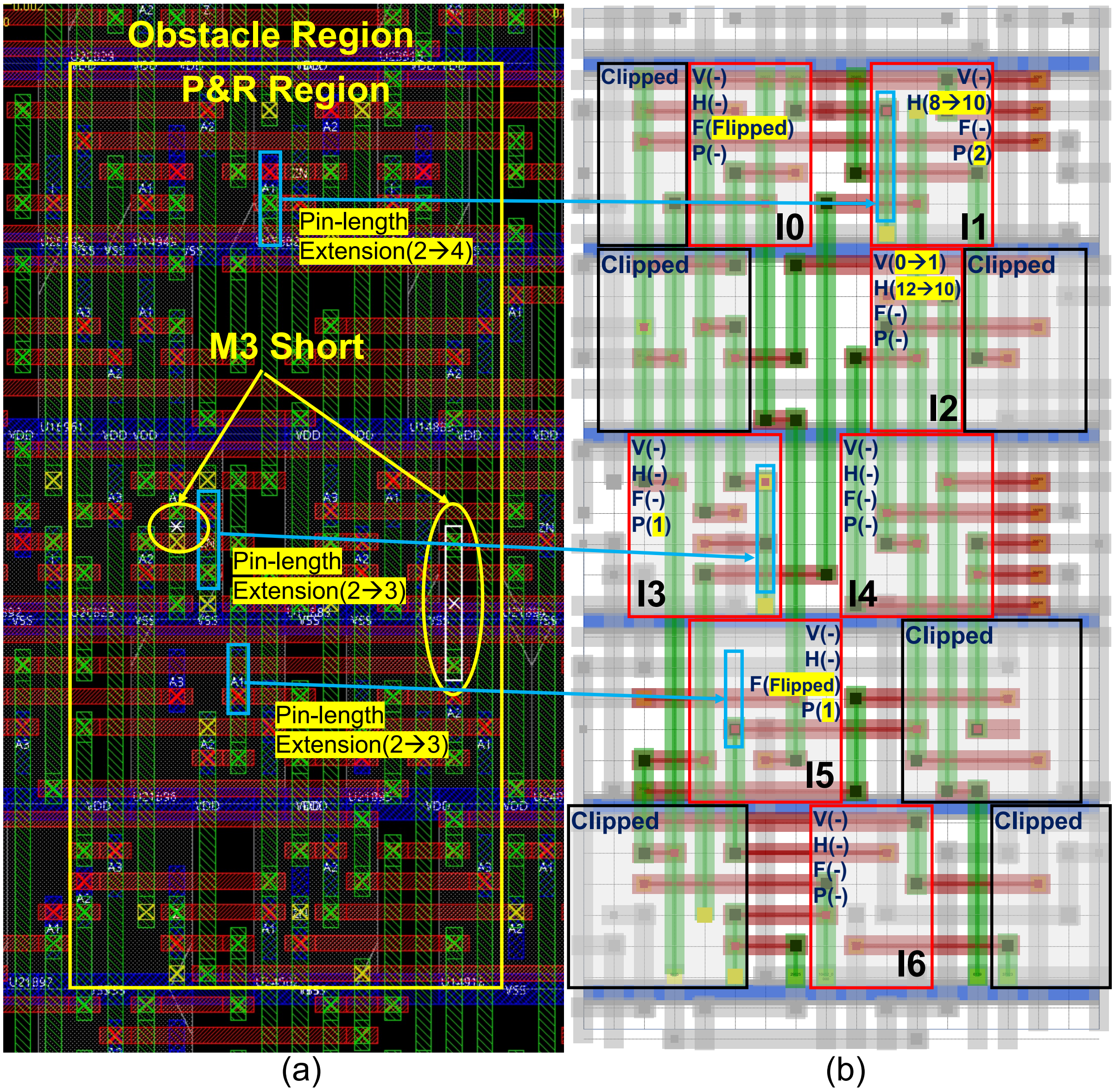


Figure 5.13: Example of refinement operations in the proposed ECO flow (Index 38 case in Table 5.4). (a) Switchbox with 3 DRVs. (b) Routable solution with placement adjustments and pin-length extensions.

5.4 Conclusion

We have described a novel concurrent refinement framework for the automated ECO flow. Our framework provides simultaneous and perturbation-minimized refinements of DP-, DR-, and cell-optimized layout solutions to address the DRVs during the ECO stage. By ripping up and refining a local window of the whole layout design, *CoRe-ECO* is capable of achieving a DRV-clean layout solution. We have demonstrated that our framework successfully resolves an average of 58.6% (range: 33.3% to 100.0%) of remaining post-ECO route DRVs on $M1$-$M3$ layers, across a range of benchmark circuits with various cell architectures, with no adverse effect on total routed wirelength (average of 0.003% reduction).

This chapter contains materials from “CoRe-ECO: Concurrent Refinement of Detailed Place-and-Route for an Efficient ECO Automation”, by Chung-Kuan Cheng, Andrew B. Kahng, Ilgweon Kang, Minsoo Kim, Daeyeal Lee, Bill Lin, Dongwon Park, and Mingyu Woo, which appears in International Conference on Computer Design, December 2021. The dissertation author was the primary investigator and author of this paper.

Chapter 6

Summary

This thesis describes automated frameworks utilizing logical reasoning techniques in three topics of VLSI applications; (i) NoC task mapping and scheduling, (ii) standard cell synthesis, and (iii) engineering change order (ECO).

Chapter 2 presents an SMT-based task mapping and scheduling framework that guarantees contention-free data transmissions to achieve the optimal latency for 2D/3D SMART NoCs. Also, we develop link overlap detection constraints for the mixed dimension-order routing. We have reduced the formulation complexity by utilizing SMT's efficient modeling capability for the conditional constraints and also improved the scalability by introducing efficient search-space reduction techniques. We demonstrated that our SMT framework achieves 10× higher scalability than ILP, solving the problem within 12 hours up to 500 tasks for 2D and the 3D extension. Also, the 2D and 3D extensions of our SMT framework with the mixed dimension-order routing maintain the improved scalability with the diversified routing paths, resulting in reduced latency through various application benchmarks. Lastly, we find that there are still rooms to further improve, e.g., the static task execution and data transmission time calls for future research topics to accommodate the variability of real systems.

Chapter 3 and Chapter 4 present new SMT-based standard cell synthesis frameworks for conventional FinFET and many-tier VFET architectures. Our framework provides fully automated procedures for generating the optimal cell layouts that combine the place-and-route in search space. In Chapter 3, 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 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 [73] by exchanging less than 0.2% of the total metal length. We demonstrate that our framework accomplishes a wide variety of cell-layout designs, up to 28 CPPs, 36 FETs, 27 nets, and 92 commodities, within

1.75 hours for the largest cell (SDFFSNQ_X1, Fig. 3.27). In Chapter 4, we demonstrate that the proposed concurrent P&R approach obtains 15.2% of average cell area reduction for 2-tier VFET compared to the sequential P&R [10]. Throughout the exploration for various many-tier VFET configurations up to four tiers, we show that the 4-tier VFET respectively achieves 50.1% and 27.9% of average area reduction on chip-level and block-level, over 4.5T GAAFET. Lastly, we find that there are still rooms to further improve, e.g., the higher parasitic resistance of many-tier VFETs and the thermal issue in a stacked logic transistors [88] call future research topics to obtain the maximum-achievable PPAC (power, performance, area, and cost) benefits through VFET.

Chapter 5 presents a novel concurrent refinement framework for the automated ECO flow. Our framework provides simultaneous and perturbation-minimized refinements of DP-, DR-, and cell-optimized layout solutions to address the DRVs during the ECO stage. By ripping up and refining a local window of the whole layout design, *CoRe-ECO* is capable of achieving a DRV-clean layout solution. We have demonstrated that our framework successfully resolves an average of 58.6% (range: 33.3% to 100.0%) of remaining post-ECO route DRVs on $M1$-$M3$ layers, across a range of benchmark circuits with various cell architectures, with no adverse effect on total routed wirelength (average of 0.003% reduction).

Bibliography

[1] S. M. Y. Sherazi, M. Cupak, P. Weckx, O. Zografos, D. Jang, P. Debacker, D. Verkest, A. Mocuta, R. H. Kim, A. Spessot, and J. Ryckaert, "Standard-cell design architecture options below 5nm node: The ultimate scaling of finfet and nanosheet," in *Design-Process-Technology Co-optimization for Manufacturability XIII, vol. 10962*. SPIE, 2019.

[2] J. Ryckaert, A. Gupta, A. Jourdain, B. Chava, G. Van der Plas, D. Verkest, and E. Beyne, "Extending the roadmap beyond 3nm through system scaling boosters: A case study on buried power rail and backside power delivery," in *2019 Electron Devices Technology and Manufacturing Conference (EDTM)*. IEEE, 2019, pp. 50–52.

[3] Z. Tokei, https://www.imec-int.com/en/imec-magazine/imec-magazine-november-2018/the-supervia-a-promising-scaling-booster-for-the-sub-3nm-technology-node.

[4] IEEE, "IRDS International Roadmap for Devices and Systems: Executive Summary," 2020.

[5] R. Dick, D. Rhodes, and W. Wolf, "Tgff: task graphs for free," in *Proceedings of International Workshop on Hardware/Software Codesign. (CODES)*, 1998, pp. 1–6. [Online]. Available: https://doi.org/10.1109/HSC.1998.666245

[6] T. Tobita and H. Kasahara, "A standard task graph set for fair evaluation of multiprocessor scheduling algorithms," *Journal of Scheduling*, vol. 5, no. 5, pp. 379–394, 2002. [Online]. Available: http://dx.doi.org/10.1002/jos.116

[7] V. Vashishtha, M. Vangala, and L. T. Clark, "ASAP7 predictive design kit development and cell design technology co-optimization," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2017, pp. 992–998.

[8] Cadence IC Digital Implementation Tool, *Innovus*, http://www.cadence.com.

[9] T. Huynh-Bao, J. Ryckaert, S. Sakhare, A. Mercha, D. Verkest, A. Thean, and P. Wambacq, "Toward the 5nm technology: layout optimization and performance benchmark for logic/srams using lateral and vertical gaa fets," in *Proceedings of international society for optics and photonics (SPIE)*. SPIE, 2016, pp. 1–12.

[10] T. Song, “Many-tier vertical gaafet (v-fet) for ultra-miniaturized standard cell designs beyond 5nm,” *IEEE Access*, pp. 149 984–149 998, 2020.

[11] G. E. Moore, “Cramming more components onto integrated circuits.” McGraw-Hill New York, NY, USA:, 1965.

[12] R. H. Dennard, F. H. Gaensslen, H.-N. Yu, V. L. Rideout, E. Bassous, and A. R. LeBlanc, “Design of ion-implanted mosfet’s with very small physical dimensions,” vol. 9, no. 5. IEEE, 1974, pp. 256–268.

[13] IEEE, “IRDS International Roadmap for Devices and Systems: More-than-Moore White Paper,” 2020.

[14] S. K. Moore, “The node is nonsense,” *IEEE Spectrum*, vol. 57, no. 8, pp. 24–30, 2020.

[15] R.-h. Kim, Y. Sherazi, P. Debacker, P. Raghavan, J. Ryckaert, A. Malik, D. Verkest, J. U. Lee, W. Gillijns, L. E. Tan, V. Blanco, K. Ronse, and G. McIntyre, “Imec n7, n5 and beyond: Dtco, stco and euv insertion strategy to maintain affordable scaling trend,” in *Design-Process-Technology Co-optimization for Manufacturability XII*, vol. 10588. International Society for Optics and Photonics, 2018, p. 105880N.

[16] IEEE, “IRDS International Roadmap for Devices and Systems: Lithography,” 2020.

[17] C. Barrett and C. Tinelli, “Satisfiability modulo theories,” in *Handbook of Model Checking*. Springer, 2018, pp. 305–343.

[18] N. Bjørner, A.-D. Phan, and L. Fleckenstein, “νz-an optimizing smt solver,” in *Proceedings of International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 2015, pp. 194–199.

[19] R. Sebastiani and P. Trentin, “Optimathsat: A tool for optimization modulo theories,” in *Proceedings of International Conference on Computer Aided Verification (CAV)*, 2015, pp. 447–454.

[20] L. De Moura and N. Bjørner, “Z3: An efficient smt solver,” in *Proceedings of International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 2008, pp. 337–340.

[21] Y. A. Badr, K.-w. Ma, and P. Gupta, “Layout pattern-driven design rule evaluation,” *Journal of Micro/Nanolithography, MEMS, and MOEMS*, vol. 13, no. 4, p. 043018, 2014.

[22] K. Lucas, C. Cork, B. Yu, G. Luk-Pat, B. Painter, and D. Z. Pan, “Implications of triple patterning for 14nm node design and patterning,” in *Proceedings of international society for optics and photonics (SPIE)*, vol. 8327, 2012, p. 832703.

[23] Y. Ma, J. Sweis, H. Yoshida, Y. Wang, J. Kye, and H. J. Levinson, “Self-aligned double patterning (SADP) compliant design flow,” in *Proceedings of international society for optics and photonics (SPIE)*, vol. 8327, 2012, p. 832706.

[24] Cadence Design Systems, Inc., Sr Software Engineering Group Director, *personal communication*, 2019.

[25] I. Kang, D. Park, C. Han, and C.-K. Cheng, "Fast and precise routability analysis with conditional design rules," in *Proceedings of System Level Interconnect Prediction (SLIP)*, 2018, p. 4.

[26] D. Lee, D. Park, C. T. Ho, I. Kang, H. Kim, S. Gao, B. Lin, and C. K. Cheng, "SP&R: SMT-based Simultaneous Place-and-Route for Standard Cell Synthesis of Advanced Nodes," *IEEE Transactions on Computer Aided Design of Integrated Circuits Systems (TCAD)*, 2020.

[27] D. Park, D. Lee, I. Kang, C. Holtz, S. Gao, B. Lin, and C. Cheng, "Grid-based framework for routability analysis and diagnosis with conditional design rules," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2020.

[28] W. J. Dally and B. P. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., 2003.

[29] T. Krishna, C.-H. O. Chen, W. C. Kwon, and L.-S. Peh, "Breaking the on-chip latency barrier using smart," in *Proceedings of IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2013, pp. 378–389. [Online]. Available: https://doi.org/10.1109/HPCA.2013.6522334

[30] ——, "Smart: Single-cycle multihop traversals over a shared network on chip," *IEEE Micro*, vol. 34, no. 3, pp. 43–56, 2014. [Online]. Available: https://doi.org/10.1109/MM.2014.48

[31] L. Yang, W. Liu, P. Chen, N. Guan, and M. Li, "Task mapping on smart noc: Contention matters, not the distance," in *Proceedings of ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2017, pp. 1–6. [Online]. Available: https://doi.org/10.1145/3061639.3062323

[32] C.-H. O. Chen, S. Park, T. Krishna, S. Subramanian, A. P. Chandrakasan, and L.-S. Peh, "Smart: A single-cycle reconfigurable noc for soc applications," in *Proceedings of Design, Automation Test in Europe Conference Exhibition (DATE)*, 2013, pp. 1–6. [Online]. Available: https://doi.org/10.7873/DATE.2013.080

[33] B. S. Feero and P. P. Pande, "Networks-on-chip in a three-dimensional environment: A performance evaluation," *Transactions on Computers*, vol. 58, no. 1, pp. 32–45, 2008. [Online]. Available: https://doi.org/10.1109/TC.2008.142

[34] X. Chen and N. K. Jha., "Reducing wire and energy overheads of the smart noc using a setup request network," *IEEE Transactions on Very Large Scale Integration*, vol. 24, no. 10, pp. 3013–3026, 2016. [Online]. Available: https://doi.org/10.1109/TVLSI.2016.2538284

[35] B. K. Joardar, K. Duraisamy, and P. P. Pande, “High performance collective communication-aware 3d network-on-chip architectures,” in *Proceedings of Design, Automation Test in Europe Conference Exhibition (DATE)*, 2018, pp. 1351–1356. [Online]. Available: https://doi.org/10.23919/DATE.2018.8342223

[36] Y. Asgarieh and B. Lin, “Smart-hop arbitration request propagation: Avoiding quadratic arbitration complexity and false negatives in smart nocs,” *ACM Transactions on Design Automation of Electronic Systems*, vol. 24, no. 64, pp. 64:1–64:25, 2019. [Online]. Available: https://doi.org/10.1145/3356235

[37] D. Lee, D. Sourav, J. Rao Doppa, P. Pratim Pande, and K. Chakrabarty, “Performance and thermal tradeoffs for energy-efficient monolithic 3d network-on-chip,” *ACM Transactions on Design Automation of Electronic Systems*, vol. 23, no. 5, pp. 1–25, 2018. [Online]. Available: https://dl.acm.org/doi/10.1145/3223046

[38] G. Van der Plas, P. Limaye, I. Loi, A. Mercha, H. Oprins, C. Torregiani, S. Thijs, D. Linten, M. Stucchi, G. Katti, D. Velenis, V. Cherman, B. Vandevelde, V. Simons, I. De Wolf, R. Labie, D. Perry, S. Bronckers, N. Minas, M. Cupac, W. Ruythooren, J. Van Olmen, A. Phommahaxay, M. de Potter de ten Broeck, A. Opdebeeck, M. Rakowski, B. De Wachter, M. Dehan, M. Nelis, R. Agarwal, A. Pullini, F. Angiolini, L. Benini, W. Dehaene, Y. Travaly, E. Beyne, and P. Marchal, “Design issues and considerations for low-cost 3-d tsv ic technology,” *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 293–307, 2010. [Online]. Available: https://doi.org/10.1109/JSSC.2010.2074070

[39] D. Seo, A. Ali, W.-T. Lim, and N. Rafique, “Near-optimal worst-case throughput routing for two-dimensional mesh networks,” in *Proceedings of International Symposium on Computer Architecture (ISCA)*, 2005, pp. 1–12. [Online]. Available: https://doi.org/10.1109/ISCA.2005.37

[40] Gurobi Optimization, LLC, “Gurobi optimizer reference manual,” 2021. [Online]. Available: http://www.gurobi.com

[41] J. Hu and R. Marculescu, “Energy-aware mapping for tile-based noc architectures under performance constraints,” in *Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2003, pp. 1–6. [Online]. Available: https://dl.acm.org/doi/10.1145/1119772.1119818

[42] W. Liu, J. Xu, X. Wu, Y. Ye, X. Wang, W. Zhang, M. Nikdast, and Z. Wang, “A noc traffic suite based on real applications,” in *Proceedings of IEEE Computer Society Annual Symposium on VLSI*, 2011, pp. 66–71. [Online]. Available: https://doi.org/10.1109/ISVLSI.2011.49

[43] A. Lodi and A. Tramontani, “Performance variability in mixed-integer programming,” in *TutORials in Operations Research*, 2014, pp. 1–12. [Online]. Available: https://doi.org/10.1287/educ.2013.0112

[44] E. A. Lee and D. G. Messerschmitt, "Static scheduling of synchronous data flow programs for digital signal processing," *IEEE Transactions on computers*, vol. 100, no. 1, pp. 24–35, 1987. [Online]. Available: https://doi.org/10.1109/TC.1987.5009446

[45] ——, "Synchronous data flow," *Proceedings of the IEEE*, vol. 75, no. 9, pp. 1235–1245, 1987. [Online]. Available: https://doi.org/10.1109/PROC.1987.13876

[46] S. Murali and G. D. Micheli, "Bandwidth-constrained mapping of cores onto noc architectures," in *Proceedings of Design, Automation Test in Europe Conference Exhibition (DATE)*, 2004, pp. 896–901. [Online]. Available: https://doi.org/10.1109/DATE.2004.1269002

[47] J. Hu and R. Marculescu, "Energy-aware communication and task scheduling for network-on-chip architectures under real-time constraints," in *Proceedings of Design, Automation Test in Europe Conference Exhibition (DATE)*, 2004, pp. 234–239. [Online]. Available: https://doi.org/10.1109/DATE.2004.1268854

[48] G. Chen, F. Li, S. Son, and M. Kandemir, "Application mapping for chip multiprocessors," in *Proceedings of ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2008, pp. 620–625. [Online]. Available: https://doi.org/10.1145/1391469.1391628

[49] H. Yu, Y. Ha, and B. Veeravalli, "Communication-aware application mapping and scheduling for noc-based mpsocs," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010, pp. 3232–3235. [Online]. Available: https://doi.org/10.1109/ISCAS.2010.5537920

[50] L. Yang, W. Liu, N. Guan, and N. Dutt, "Optimal application mapping and scheduling for network-on-chips with computation in stt-ram based router," *IEEE Transactions on Computers*, vol. 68, no. 8, pp. 1174–1189, 2019. [Online]. Available: https://doi.org/10.1109/TC.2018.2864749

[51] R. Yan, A. Cai, H. Gao, F. Ma, and J. Yan, "Smt-based multi-objective optimization for scheduling of mpsoc applications," in *Proceedings of International Symposium on Theoretical Aspects of Software Engineering (TASE)*, 2019, pp. 160–167. [Online]. Available: https://doi.org/10.1109/TASE.2019.000-5

[52] D. Lee, B. Lin, and C.-K. Cheng, "Smt-based contention-free task mapping and scheduling on smart noc," *Embedded Systems Letters*, vol. 1, no. 1, pp. 1–4, 2021. [Online]. Available: https://doi.org/10.1109/LES.2021.3049774

[53] A. P. Jacob, R. Xie, M. G. Sung, L. Liebmann, R. T. Lee, and B. Taylor, "Scaling challenges for advanced cmos devices," *International Journal of High Speed Electronics and Systems*, vol. 26, no. 01n02, p. 1740001, 2017.

[54] L. Liebmann, J. Zeng, X. Zhu, L. Yuan, G. Bouche, and J. Kye, "Overcoming scaling barriers through design technology cooptimization," in *Proceedings of IEEE Symposium on VLSI Technology and Circuits (VLSI)*, 2016, pp. 1–2.

[55] C.-K. Cheng, D. Lee, and D. Park, “Standard-cell scaling framework with guaranteed pin-accessibility,” in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020 (to appear).

[56] A. B. Kahng, H. Lee, and J. Li, “Measuring progress and value of ic implementation technology,” in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2016, pp. 1–8.

[57] 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 Transactions on Very Large Scale Integrated System (TVLSI)*, vol. 27, no. 8, pp. 1933–1946, 2019.

[58] C. Lazzari, C. Santos, and R. Reis, “A new transistor-level layout generation strategy for static cmos circuits,” in *Proceedings of IEEE International Conference on Electronics Circuits and Systems (ICECS)*, 2006, pp. 660–663.

[59] X. Xu, N. Shah, A. Evans, S. Sinha, B. Cline, and G. Yeric, “Standard cell library design and optimization methodology for asap7 pdk,” in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2017, pp. 999–1004.

[60] B. Taylor and L. Pileggi, “Exact combinatorial optimization methods for physical design of regular logic bricks,” in *Proceedings of ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2007, pp. 344–349.

[61] P.-H. Wu, M. P.-H. Lin, T.-C. Chen, T.-Y. Ho, Y.-C. Chen, S.-R. Siao, and S.-H. Lin, “1-d cell generation with printability enhancement,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 3, pp. 419–432, 2013.

[62] N. Ryzhenko and S. Burns, “Standard cell routing via boolean satisfiability,” in *Proceedings of ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2012, pp. 603–612.

[63] C. J. Poirier, “Excellerator: custom cmos leaf cell layout generator,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 8, no. 7, pp. 744–755, 1989.

[64] 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 *Proceedings of Great Lakes Symposium on VLSI (GLSVLSI)*, 2015, pp. 289–294.

[65] J. Seo, J. Jung, S. Kim, and Y. Shin, “Pin accessibility-driven cell layout redesign and placement optimization,” in *Proceedings of ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2017, pp. 1–6.

[66] 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 Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 21, no. 3, p. 42, 2016.

[67] 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 *Proceedings of International Symposium on Physical Design (ISPD)*, 2019, pp. 41–47.

[68] M. Guruswamy, R. L. Maziasz, D. Dulitz, S. Raman, V. Chiluvuri, A. Fernandez, and L. G. Jones, “Cellerity: A fully automatic layout synthesis system for standard cell libraries,” in *Proceedings of ACM/EDAC/IEEE Design Automation Conference (DAC)*, 1997, pp. 327–332.

[69] A. M. Ziesemer and R. A. da Luz Reis, “Simultaneous two-dimensional cell layout compaction using milp with astran,” in *Proceedings of IEEE Computer Society Annual Symposium on VLSI*, 2014, pp. 350–355.

[70] P. Van Cleeff, S. Hougardy, J. Silvanus, and T. Werner, “Bonncell: Automatic cell layout in the 7nm era,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2019.

[71] Y.-L. Li, S.-T. Lin, S. Nishizawa, H.-Y. Su, M.-J. Fong, O. Chen, and H. Onodera, “Nctucell: A dda-aware cell library generator for finfet structure with implicitly adjustable grid map,” in *Proceedings of ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2019, p. 120.

[72] A. Sorokin and N. Ryzhenko, “Sat-based placement adjustment of finfets inside unroutable standard cells targeting feasible drc-clean routing,” in *Proceedings of Great Lakes Symposium on VLSI (GLSVLSI)*, 2019, pp. 159–164.

[73] 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-7nm,” in *Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2020, pp. 345–350.

[74] M. Rentmeesters, W. Tsai, and K.-J. Lin, “A theory of lexicographic multi-criteria optimization,” in *Proceedings of International Conference on Engineering of Complex Computer Systems (ICECCS)*, 1996.

[75] 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 *Proceedings of International Symposium on Physical Design (ISPD)*, 2019, pp. 65–72.

[76] 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 *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2017, pp. 667–674.

[77] S. Sutanthavibul, E. Shragowitz, and J. B. Rosen, “An analytical approach to floorplan design and optimization,” in *Proceedings of ACM/EDAC/IEEE Design Automation Conference (DAC)*, 1991, pp. 187–192.

[78] 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 Transactions Computer-Aided Design Integrated Circuits and Systems*, vol. 34, no. 5, pp. 699–712, 2015.

[79] T. Stöhr, M. Alt, A. Hetzel, and J. Koehl, “Analysis, reduction and avoidance of crosstalk on vlsi chips,” in *Proceedings of International Symposium on Physical Design (ISPD)*, 1998, pp. 211–218.

[80] D. Déharbe, P. Fontaine, S. Merz, and B. W. Paleo, “Exploiting symmetry in smt problems,” in *Proceedings of international conference on Automated deduction (CADE)*, 2011, pp. 222–236.

[81] 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 Transactions on Software Engineering*, vol. 39, no. 9, pp. 1283–1307, 2013.

[82] M. Martins, J. M. Matos, R. P. Ribas, A. Reis, G. Schlinker, L. Rech, and J. Michelsen, “Open cell library in 15nm freepdk technology,” in *Proceedings of International Symposium on Physical Design (ISPD)*, 2015, pp. 171–178.

[83] 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 *Proceedings of International Conference on Microelectronic Systems Education (MSE)*, 2017, pp. 1–4.

[84] S. Maheshwaram, S. K. Manhas, G. Kaushal, B. Anand, and N. Singh, “Vertical silicon nanowire gate-all-around field effect transistor based nanoscale cmos,” *IEEE Electron Device Letters*, pp. 1011–1013, 2011.

[85] T. H. Bao, D. Yakimets, J. Ryckaert, I. Ciofi, R. Baert, A. Veloso, J. Boemmels, N. Collaert, P. Roussel, S. Demuynck, P. Raghavan, A. Mercha, Z. Tokei, D. Verkest, A. V.-Y. Thean, and P. Wambacq, “Circuit and process co-design with vertical gate-all-around nanowire fet technology to extend cmos scaling for 5nm and beyond technologies,” in *Proceedings of European Solid-state Circuits and Devices Conference (ESSDERC)*. IEEE, 2014, pp. 102–105.

[86] D. Yakimets, G. Eneman, P. Schuddinck, T. Huynh Bao, M. Garcia Bardon, P. Raghavan, A. Veloso, N. Collaert, A. Mercha, D. Verkest, A. Voon-Yew Thean, and K. De Meyer, “Vertical gaafets for the ultimate cmos scaling,” *IEEE Transactions on Electron Devices*, pp. 1433–1439, 2015.

[87] H. Na and E. T., “A new compact sram cell by vertial mosfet for low-power and stable operation,” in *Proceedings of IEEE International Memory Workshop (IMW)*. IEEE, 2011, pp. 1–4.

[88] M. Li, J. Shi, M. Rahman, S. Khasanvis, S. Bhat, and C. A. Moritz, “Skybridge-3d-cmos: A fine-grained 3d cmos integrated circuit technology,” *IEEE Transactions on Nanotechnology*, pp. 639–652, 2017.

[89] W.-C. Wang and P. Gupta, "Efficient layout generation and design evaluation of vertical channel devices," *IEEE Transactions on Computer Aided Design of Integrated Circuits Systems (TCAD)*, pp. 1449–1460, 2015.

[90] W.-C. Wang, C. Zhao, and P. Gupta, "Assessing layout density benefits of vertical channel devices," *IEEE Transactions on Computer Aided Design of Integrated Circuits Systems (TCAD)*, pp. 3211–3215, 2018.

[91] H. Liu, C. Hong, T. Han, J. Zhou, and Y. Chen, "Structural design, layout analysis and routing strategy for constructing ic standard cells using emerging 3d vertical mosfets," in *Proceedings of international society for optics and photonics (SPIE)*. SPIE, 2016, pp. 13–23.

[92] T. Song, "Opportunities and challenges in designing and utilizing vertical nanowire fet (v-nwfet) standard cells for beyond 5 nm," *IEEE Transactions on Nanotechnology*, pp. 240–251, 2019.

[93] PDK 45nm Open Cell Library. Accessed: Aug. 2021. [Online], https://www.silvaco.com/.

[94] L. Liebmann, D. Chanemougame, P. Churchill, J. Cobb, C.-T. Ho, V. Moroz, and J. Smith, "Dtco acceleration to fight scaling stagnation," in *Design-Process-Technology Co-optimization for Manufacturability XIV*, vol. 11328. International Society for Optics and Photonics, 2020, p. 113280C.

[95] C.-K. Cheng, C.-T. Ho, D. Lee, B. Lin, and D. Park, "Complementary-fet (cfet) standard cell synthesis framework for design and system technology co-optimization using smt," *to appear in IEEE Transactions on Very Large Scale Integration Systems*, pp. 1–14, 2021.

[96] J. Ryckaert, A. Gupta, A. Jourdain, B. Chava, G. Van der Plas, D. Verkest, and E. Beyne, "Extending the roadmap beyond 3nm through system scaling boosters: A case study on buried power rail and backside power delivery," in *Proceedings of IEEE Electron Devices Technology and Manufacturing (EDTM)*. EDTM, 2019, pp. 50–52.

[97] N. Loubet, T. Hook, P. Montanini, C.-W. Yeung, S. Kanakasabapathy, M. Guillom, T. Yamashita, J. Zhang, X. Miao, J. Wang, A. Young, R. Chao, M. Kang, Z. Liu, S. Fan, B. Hamieh, S. Sieg, Y. Mignot, W. Xu, S.-C. Seo, J. Yoo, S. Mochizuki, M. Sankarapandian, O. Kwon, A. Carr, A. Greene, Y. Park, J. Frougier, R. Galatage, R. Bao, J. Shearer, R. Conti, H. Song, D. Lee, D. Kong, Y. Xu, A. Arceo, Z. Bi, P. Xu, R. Muthinti, J. Li, R. Wong, D. Brown, P. Oldiges, R. Robison, J. Arnold, N. Felix, S. Skordas, J. Gaudiello, T. Standaert, H. Jagannathan, D. Corliss, M.-H. Na, A. Knorr, T. Wu, D. Gupta, S. Lian, R. Divakaruni, T. Gow, C. Labelle, S. Lee, V. Paruchuri, H. Bu, and M. Khare, "Stacked nanosheet gate-all-around transistor to enable scaling beyond finfet," in *Proceedings of Symposium on VLSI Technology (VLSIT)*. IEEE, 2017, pp. 1–2.

[98] OpenCores: Open-Source IP Cores., https://opencores.org/, 2020.

[99] LEF/DEF Language Reference, http://www.InternationalSymposiumonPhysicalDesign(ISPD).cc/contests/18/lefdefref.pdf, 2020.

[100] B. Chava, J. Ryckaert, L. Mattii, S. M. Y. Sherazi, P. Debacker, A. Spessot, and D. Verkest, "Dtco exploration for efficient standard cell power rails," in *Design-Process-Technology Co-optimization for Manufacturability XII*, vol. 10588. International Society for Optics and Photonics, 2018.

[101] X. Qiu and M. Marek-Sadowska, "Can pin access limit the footprint scaling?" in *Proceedings of ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2012, pp. 1100–1106.

[102] M.-K. Hsu, N. Katta, H. Y.-H. Lin, K. T.-H. Lin, K. H. Tam, and K. C.-H. Wang, "Design and manufacturing process co-optimization in nano-technology," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2014.

[103] W.-H. Liu, C.-K. Koh, and Y.-L. Li, "Optimization of placement solutions for routability," in *Proceedings of ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2013.

[104] Y. Ding, C. Chu, and W.-K. Mak, "Pin accessibility-driven detailed placement refinement," in *Proceedings of International Symposium on Physical Design (ISPD)*, 2017, pp. 133–140.

[105] T.-C. Yu, S.-Y. Fang, H.-S. Chiu, K.-S. Hu, P. H.-Y. Tai, C. C.-F. Shen, and H. Sheng, "Pin accessibility prediction and optimization with deep learning-based pin pattern recognition," in *Proceedings of ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE, 2019, pp. 1–6.

[106] A. B. Kahng, L. Wang, and B. Xu, "The tao of pao: anatomy of a pin access oracle for detailed routing," in *Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2020, pp. 1–6.

[107] A. B. Kahng, J. Kuang, W.-H. Liu, and B. Xu, "In-route pin access-driven placement refinement for improved detailed routing convergence," *IEEE Transactions on Computer Aided Design of Integrated Circuits Systems (TCAD)*, pp. 1–5, 2021.

[108] 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 Transactions on Computer Aided Design of Integrated Circuits Systems (TCAD)*, pp. 699–712, 2015.

[109] C.-K. Cheng, C.-T. Ho, D. Lee, and D. Park, "A routability-driven complimentary-FET (CFET) standard cell synthesis framework using smt," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2020.

[110] S. Kim, K. Jo, and T. Kim, "Boosting pin accessibility through cell layout topology diversification," in *Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2021, pp. 1–6.

[111] D. Bhattacharya, V. Boppana, R. Roy, and J. Roy, “Method for automated design of integrated circuits with targeted quality objectives using dynamically generated building blocks,” *US Patent, US7225423B2*, 2007.

[112] L. Burns, Jeffrey, and J. A. Feldman, “C5m-a control-logic layout synthesis system for high-performance microprocessors,” *IEEE Transactions on Computer Aided Design of Integrated Circuits Systems (TCAD)*, pp. 14–23, 1998.

[113] P. Majumder, B. Kumthekar, N. R. Shah, J. Mowchenko, P. A. Chavda, Y. Kojima, H. Yoshida, and V. Boppana, “Method of ic design optimization via creation of design-specific cells from post-layout patterns,” *US Patent, US7941776B2*, 2011.

[114] D. Park, D. Lee, I. Kang, C. Holtz, S. Gao, B. Lin, and C.-K. Cheng, “Grid-based framework for routability analysis and diagnosis with conditional design rules,” *IEEE Transactions on Computer Aided Design of Integrated Circuits Systems (TCAD)*, 2020.

[115] C.-K. Cheng, A. B. Kahng, H. Kim, M. Kim, D. Lee, D. Park, and M. Woo, “Probe2.0: A systematic framework for routability assessment from technology to design in advanced nodes,” *IEEE Transactions on Computer Aided Design of Integrated Circuits Systems (TCAD)*, pp. 1–14, 2021.

[116] C.-K. Cheng, D. Lee and D. Park, SMT-Based Standard-Cell Layout Generator, https://github.com/ckchengucsd/SMT-based-STDCELL-Layout-Generator-for-PROBE2.0.

[117] lowRISC, https://www.lowrisc.org.

[118] Synopsys IC Compiler II User Guide, http://www.synopsys.com.