TeMACLE: A Technology Mapping-Aware Area-Efficient Standard Cell Library Extension Framework

Rongliang Fu, Chao Wang, Bei Yu, and Tsung-Yi Ho *Fellow, IEEE*

***Abstract*—Standard cell libraries play a crucial role in modern VLSI design by providing pre-designed, pre-characterized, and pre-verified building blocks to simplify the design process. However, the increasing complexity of circuits demands more specialized and optimized cells, thereby necessitating the extension of standard cell libraries. This paper proposes TeMACLE, a technology mapping-aware area-efficient framework to extend the standard cell library. Aiming at the area optimization of digital circuits, TeMACLE extends the given original standard cell library through two feasible: (i) the area compaction of standard cells and (ii) the area-efficient facilitation for technology mapping. TeMACLE employs K-feasible cones to extract sub-circuits and designs a sub-circuit encoding method to divide them. Then, an SAT-based sub-circuit matching algorithm is proposed to identify all equivalent sub-circuits further. Finally, new standard cells are determined by a technology mapping-aware area-efficient strategy. The experimental results on the EPFL benchmark using the FreePDK45 process design kit show the effectiveness and efficiency of TeMACLE. Notably, TeMACLE is available at https://github.com/Flians/TeMACLE.**

***Index Terms*—Standard cell library extension, technology mapping, area optimization, circuit encoding, circuit matching**

I. Introduction

THE field of very large-scale integration (VLSI) design has witnessed significant advancements in recent years, leading to the creation of highly complex integrated circuits. Standard cell libraries have emerged as a fundamental component in IC design, providing designers with pre-designed, pre-characterized, and pre-verified cells that facilitate the design process. However, with the increasing complexity of circuit design, there arises a necessity to extend standard cell libraries to include additional cell variants tailored to specific design requirements [1].

The significance of standard cell library extension lies in the enhanced flexibility it offers to designers. By extending the original standard cell library to encompass a broader range of standard cell options, designers gain the ability to select standard cells that better align with their design specifications. This flexibility empowers designers to optimize their circuits for critical metrics such as power, performance, and area (PPA). However, the simple usage of large standard cells may result in a significant area overhead in the final design [2]. To mitigate this issue, the generation of regular complex cells [3], particularly compound gates [4], has become increasingly appealing. Nonetheless, the efficient and scalable synthesis methodologies for these complex cells still require further research to address the emerging challenges.

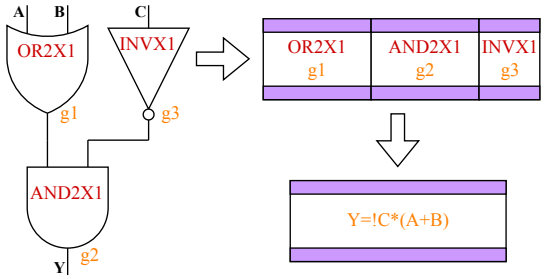


Fig. 1 An example of merging and compacting a sub-circuit comprising three standard cells.

Existing research primarily addresses the automated generation of standard cells, as evidenced by works such as [4]–[10]. These works have achieved and optimized the synthesis, placement, and routing of standard cells. Additionally, several investigations [11]–[13] have explored methods to expand the standard cell library. For instance, Pilato *et al.* [11] focused on the identification of functionalities to extend the given standard-cell library. Kiamehr *et al.* [12] undertook the replication and redesign of library cells with a focus on balancing the rise and fall delays at the expected lifetime to mitigate bias temperature instability, thereby delaying transistor aging. Moreover, AutoCellLibX [13] investigated the impact of the standard cell library extension on the back-end design. Given that merging several standard cells into a new cell can lead to area reduction [14], AutoCellLibX directly selected several mutually non-overlapping sub-circuits as the extension part of the original standard cell library by a frequent subgraph mining algorithm, thereby reducing the circuit area. It is worth noting that when any updates to the gate-level circuit design

The research work described in this paper was conducted in the JC STEM Lab of Intelligent Design Automation funded by The Hong Kong Jockey Club Charities Trust and was supported in part by the Research Grants Council of Hong Kong SAR (Grant No. CUHK14207523); in part by the Research Grants Council of Hong Kong SAR (Grant No. CUHK14208021); in part by the Key Research and Development Program of Jiangsu Province (Grant No. SBE2023020263).

Rongliang Fu, Bei Yu, and Tsung-Yi Ho are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong SAR. E-mail: {rlfu, byu, tyho}@cse.cuhk.edu.hk.

Chao Wang is with the School of Integrated Circuits, Southeast University, China, and the National Center of Technology Innovation for EDA, China. E-mail: wangc@seu.edu.cn.

necessitate a complete repetition of the standard cell library extension process in AutoCellLibX. Furthermore, the standard cells extended by AutoCellLibX exhibit irregular functionalities, which complicate their reuse, and possess complex transistor netlists, which pose challenges for their re-synthesis using existing standard cell synthesis tools.

However, these existing methods lack adequate consideration for the mutual promotion and constraints between circuit design and standard cell library when extending new standard cells. On one hand, an increase in the complexity of functions offered by new standard cells may lead to a more compact gate-level netlist of the circuit. On the other hand, the circuit design can guide the extension of the standard cell library, while its requirement for new standard cells with complex functionalities may pose significant challenges to the process of standard cell layout synthesis. Consequently, the co-optimization of circuit design and standard cell library can align with the principle of Design and Technology Co-Optimization (DTCO) [15], which uses design-level power, performance, and area (PPA) metrics analyses to assess design rules, cell architecture, and design architectures.

Furthermore, as a crucial step in logic synthesis, technology mapping plays a pivotal role in the selection of standard cells during the circuit implementation process with a specific technology. The intricate nature of new standard cells, characterized by a substantial number of inputs and outputs along with complex functionalities, may cause their inefficient or even unusable utilization by existing technology mapping tools. This is where the proposition of a technology mapping-aware standard cell library extension algorithm becomes crucial.

To support the DTCO, this paper proposes TeMACLE, a technology mapping-aware standard cell library extension framework, which aims to reduce the circuit area after technology mapping with the extended standard cell library. To fulfill this aim, TeMACLE considers two potential ways to generate new standard cells, including (i) merging standard cells for the area compaction and (ii) incorporating the feedback from technology mapping. Fig. 1 shows an example of merging three standard cells for the area compaction. The SPICE netlist of this sub-circuit can be extracted and then used to synthesize a new cell with smaller area and equivalent functionality. In summary, this paper makes the following contributions:

- To the best of our knowledge, this paper proposes the first technology mapping-aware framework for standard cell library extension, which is available on GitHub.
- A sub-circuit encoding method is proposed to efficiently group K-feasible cone-based sub-circuits.
- Sub-circuit exact matching is formulated as the Boolean satisfiability (SAT) problem, with an SAT-based sub-circuit matching algorithm proposed to identify all the same sub-circuits.
- TeMACLE enables the generation of the function expression, SPICE netlist, and GDSII layout for each sub-circuit, while also enabling the extraction of the corresponding physical information through standard cell layout synthesis tools.
- Furthermore, a technology mapping-aware strategy is proposed to select a given number of patterns as final extended cells.
- Experimental results on EPFL benchmark [16] using the FreePDK45 [17] process design kit (PDK) show that following optimization by TeMACLE, the circuit area is significantly reduced, with an average reduction of 13.97%. Moreover, TeMACLE achieves an average area reduction of 1.96 times that of the state-of-the-art while only requiring nearly one-third of its runtime on average.

The remainder of this paper is organized as follows. Section II discusses the preliminaries, including the introduction to standard cell layout synthesis and the reason why the feedback of technology mapping should be considered and graph matching techniques are required in the standard cell library extension process. Section III presents the problem formulation. Section IV introduces the proposed standard cell library framework. Section V demonstrates our experiment setting and evaluation of our framework, followed by a conclusion in Section VI.

II. Preliminaries

A. Standard Cell Layout Synthesis

As a critical step in the design and manufacturing of integrated circuits, standard cell layout synthesis entails the automatic generation of the physical layout of the corresponding standard cell in terms of an abstract description, such as a transistor-level netlist or a Boolean function. This process involves three primary phases: (i) transistor folding, (ii) transistor placement, and (iii) in-cell routing. Currently, significant efforts have been devoted to automatic standard cell layout synthesis generation. For instance, CELLERITY [5] possesses compatibility with a wide range of process technologies and provides flexibility in the cell layout template. CDF [4] can synthesize physical layouts of logic cells from truth table descriptions or Boolean equations. ASTRAN [6], an open-source framework for standard cell layout synthesis, adopts mixed integer linear programming (ILP) to efficiently compact cell layouts and is competitive with manually designed cells, thereby selected as the standard cell synthesis tool in this paper. Additionally, in the face of increasingly complex design rules associated with standard cell layout in 7 nm technology and beyond, various studies [7]–[9], [18] have utilized ILP or satisfiability modulo theories (SMT) solvers to optimize the standard cell layout synthesis.

In addition to traditional standard cell layout synthesis methods tailored to specific functions and process technologies, there is also a growing interest in merging existing standard cells to generate new standard cells. Previous studies [4], [13] have demonstrated that this operation can result in area savings in the layout. Consequently, the abstraction of the sub-circuit from the gate-level netlist for re-synthesis as a new standard cell with complex functionality emerges as an attractive approach for the standard cell library extension.

B. Technology Mapping

Technology mapping [19] involves the process of translating a given logic-level representation of a circuit into a specific target technology, such as the field-programmable gate array (FPGA), the application-specific integrated circuit (ASIC), and emerging circuit technologies [20]–[23]. This process typically contains three distinct phases: circuit decomposition, matching, and covering. Circuit decomposition translates the circuit into a technology-independent representation. Then, the matching phase analyzes this representation to determine suitable standard cell implementations for each part. Finally, the covering phase selects a set of these alternatives to construct the final gate-level circuit.

Efficient technology mapping open-source tools [24], [25] have been developed to address the challenges associated with mapping complex logic circuits onto target technologies. These tools take into consideration various design factors including technology-specific standard cell libraries, circuit functionality, and resource availability. Their primary objective is to minimize the key metrics such as area, power, and delay of the resulting circuit implementation while satisfying the design constraints. The standard cell library serves as a crucial input for technology mapping tools and significantly impacts the quality of the generated gate-level circuit. Therefore, the extension of the standard cell library should take into account the feedback obtained from the technology mapping process, with the aim of continuously enhancing the quality of the resulting gate-level circuits.

C. Graph Matching

Graph matching is a fundamental problem that involves the identification of similarities or correspondences between graphs. Its related algorithms find extensive applications across various fields, including computer vision, pattern recognition, bioinformatics, and circuit design. Efficient algorithms [26] have been developed to address the challenges associated with graph matching. These algorithms employ techniques such as graph traversal, pattern matching, and indexing to efficiently search for subgraph instances within large graphs while optimizing the computational complexity and memory requirements and ensuring accuracy and scalability.

In the specific context of circuit matching, graph matching techniques play a crucial role in identifying equivalent or similar sub-circuits within a larger circuit. Since the gate-level circuit consists of numerous standard cells, the occupancy ratio of each standard cell can serve as an indicator of its significance for the circuit implementation, particularly in terms of the circuit area. Consequently, the application of graph matching techniques can facilitate the identification of high-frequency sub-circuits that, following layout compaction, can become potential candidates for the standard cell library extension.

III. Problem Formulation

This section first introduces several related concepts, including the graph, Boolean satisfiability, and circuit matching. Subsequently, our problem formulation is presented.

A. Terminology

Definition 1 (Directed Acyclic Graph)**.** *A directed acyclic graph (DAG) is defined as $G(V, E)$ with a vertex set V, also denoted as V_G, a directed edge set $E \subseteq V \times V$, also denoted as E_G, and no directed cycles. For a vertex $v \in V$, $E_i(v)$ is the set of its input edges, and $E_o(v)$ is the set of its output edges. For an edge $e \in E$, e_s is its source, and e_t is the set of its sinks.*

A gate-level circuit can be represented by a directed acyclic graph $G(V, E)$, where vertex set $V = I \cup O \cup C$ consists of the set I of primary inputs (PIs), the set O of primary outputs (POs), and the set C of logic gates, and edge set E consists of the circuit nets. The graph has vertex labels corresponding to the type of logic gates and has no edge labels. For a vertex $v \in V$, $\text{FI}(v)$ and $\text{FO}(v)$ represent the set of its fan-in vertices and its fan-out vertices, respectively. For any vertex $u \in \text{FI}(v) \cup \text{FO}(v)$, there exists a directed edge between u and v. In addition, for any PI $i \in I$, the set of its fan-in vertices is empty. For any PO $o \in O$, the set of its fan-out vertices is also empty. Moreover, for a vertex $v \in V$, $\text{TFI}(v)$ and $\text{TFO}(v)$ represent the set of its transitive fan-in vertices and its transitive fan-out vertices, respectively. For any vertex $u \in \text{TFI}(v)$ ($u \in \text{TFO}(v)$), there is a path from u to v (from v to u).

Definition 2 (Cut)**.** *A cut C_v of a vertex v is a vertex set such that any path from a PI to v must pass through at least one vertex in C_v. These vertices in C_v are called its leaves.*

The size of a cut C_v is denoted as the set size $|C_v|$. When the size $|C_v|$ of a cut C_v is equal to or less than K, cut C_v is called a K-feasible cut.

Definition 3 (Cone)**.** *A cone N_v of a vertex v is a subgraph surrounded by v and its cut C_v, containing v and some of its transitive fan-in vertices, such that a path from any vertex $u \in N_v$ to v exists in N_v.*

When the cut size of a cone N_v is equal to or less than K, *i.e.*,$|C_v| \leq K$, cone N_v is called a K-feasible cone. We can use a K-feasible cone N_v to construct a sub-circuit, detailed in Section IV-A.

Definition 4 (Boolean Satisfiability)**.** *Given a Boolean formula $f(L)$ of n literals $L = (l_1, l_2, \ldots, l_n)$, where $l_i \in \{\mathbf{True}, \mathbf{False}\}$, the Boolean satisfiability (SAT) problem determines whether there exist the values of literals L that satisfies $f(L)$, i.e., $\exists L \in \{\mathbf{True}, \mathbf{False}\}^n, f(L) = \mathbf{True}$.*

For example, an SAT instance with three literals $L = (l_1, l_2, l_3)$ can be formulated as $f(L) = (l_1 \vee l_2) \wedge (\neg l_2 \vee l_3)$, where $\vee$ is logical "or", $\wedge$ is logical "and", and $\neg$ is logical "not". Each parenthesized part, *e.g.*, $(l_1 \vee l_2)$, defines a clause. This example exists literal assignments to satisfy f, *e.g.*, $l_1 = \mathbf{True}$, $l_2 = \mathbf{True}$, and $l_3 = \mathbf{True}$.

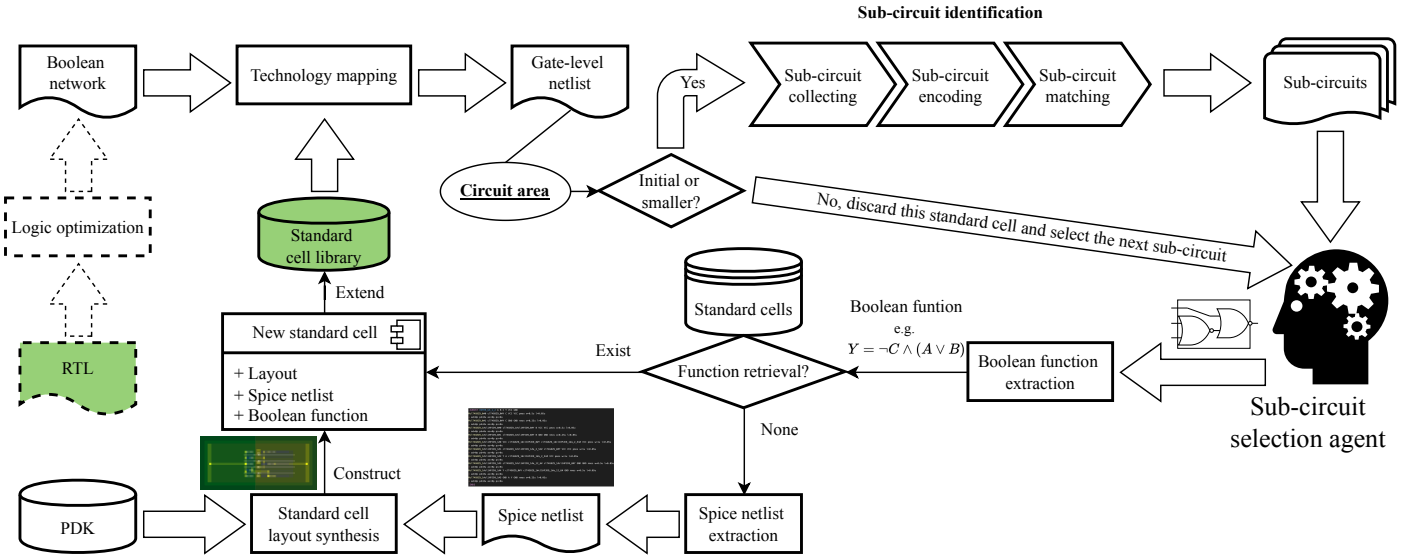


Fig. 2 The workflow of our proposed TeMACLE.

Definition 5 (Boolean Function Matching)**.** *Given a Boolean function $f(X)$ and a target Boolean function $g(X)$, where $|X| = n$, they are matched if there exists a one-to-one permutation map ρ: $\{1, 2, \ldots, n\} \rightarrow \{1, 2, \ldots, n\}$, where $\forall X \in \{\mathbf{True}, \mathbf{False}\}^n, f(X) \equiv g(\rho(X))$.*

For example, there are two Boolean functions $f(X) = (x_1 \wedge x_2) \vee x_3$ and $g(X) = (x_1 \wedge x_3) \vee x_2$ with Boolean variables $X = (x_1, x_2, x_3)$. There exists a one-to-one permutation map $\rho = \{x_1 : x_1, x_2 : x_3, x_3 : x_2\}$ that makes functions $f(X)$ and $g(X)$ matched.

Definition 6 (Circuit Matching)**.** *Given a gate-level circuit S and a target gate-level circuit G, they are matched if there is a function: $f : V_S \rightarrow V_G$, where $\forall (u, v) \in E_S, (f(u), f(v)) \in E_G$ and $\forall (u, v) \notin E_S, (f(u), f(v)) \notin E_G$.*

When two gate-level circuits are matched, they not only have the same topology, characterized by an equivalent number of vertices and edges, but their corresponding Boolean functions are also matched. Circuit matching can benefit to identify the sub-circuits with the same SPICE netlist,

B. Problem Formulation

This paper considers the feedback of technology mapping on standard cell library extension. In this way, even if updates are made to the gate-level circuit design, an effective gate-level circuit can still be generated well using the extended standard cell library. Hence, the problem of **technology mapping-aware standard cell library extension** based on merging existing standard cells can be defined as follows:

Input:

1) A given Boolean network G.
2) An original standard cell library L including l cells $\{c_1, \cdots, c_l\}$.
3) A process technology.

Output: An extended standard cell library L', where the logic function and layout can be generated for each new standard cell.

Constraints:

1) At most T new standard cells $\{nc_1, \cdots, nc_t\}$ can be added into the final standard cell library L', *i.e.*, $t \leq T$.
2) Any new standard cell only contains at most N original standard cells from the original standard cell library L, *i.e.*, $\forall i \in [1, t], |nc_i| \leq N$ and $\forall c_j \in nc_i, c_j \in L$.
3) Any new standard cell has only one functional output to support current technology mapping tools.

Goal: Minimize the circuit area of the gate-level circuit $G'(V', E')$ re-generated with extended standard cell library L', formulated as follows:

$$\min \sum_{v \in V'} area(v), \tag{1}$$

where $area(v)$ denotes the area of gate v in library L'.

This problem presents significant challenges to the standard cell extension. Firstly, the layout synthesis process of standard cells is time-consuming. It is impractical to carry out the layout synthesis for all sub-circuit candidates. Moreover, not all sub-circuits can be synthesizable into new standard cells via the standard cell layout generator due to constraints imposed by the given process technology. Hence, Constraints 1 and 2 are introduced to enable a feasible solution to this problem. Secondly, the characteristics of technology mapping must be taken into account. Notably, current technology mapping tools mainly support standard cells with a single output. This necessitates the introduction of Constraint 3 to ensure that the generated new standard cells are compatible with the existing technology mapping tools. Lastly, the structure of the gate-level circuit may exhibit variability following each technology mapping iteration, rendering previously identified sub-circuits as new standard cells obsolete. Consequently, the generated

new standard cells must maintain compatibility with the gate-level circuit with varying structures.

IV. TeMACLE

To address these challenges comprehensively, this section proposes a technology mapping-aware area-efficient framework, namely TeMACLE, designed for automatic standard cell library extension. Its overall flow is illustrated in Fig. 2. TeMACLE integrates the processes of logic optimization, technology mapping, and standard cell layout synthesis, with a specific concentration on the identification process of the sub-circuits to generate new standard cells. Notably, TeMACLE incorporates the feedback on the circuit area from the technology mapping process. Following the execution of the logic optimization and technology mapping processes for a given Boolean network G using open-source tools and the original standard cell library, we can obtain an initial gate-level netlist. Then, we can identify distinct sub-circuits from the gate-level netlist as the candidates for new standard cells, as detailed in Sections IV-A to IV-C. Subsequently, we select specific sub-circuits by the sub-circuit selection strategy proposed in Section IV-E to attempt the synthesis of their corresponding standard cell, as detailed in Section IV-D. Finally, generated standard cells beneficial to the circuit area reduction are extended into the final standard cell library.

A. Sub-Circuit Collecting

The focus of this paper is on identifying the sub-circuits that can potentially be candidates for new standard cells. Since current technology mapping tools mainly support standard cells with a single functional output, the identification of sub-circuits can be effectively handled through the utilization of K-feasible cones, a technique commonly employed in covering a given Boolean network during technology mapping. As defined in Definition 3, a cone of a vertex v refers to a subgraph of its transitive fan-in network, encompassing all vertices situated along any path from any vertex in the corresponding cut C_v to v. A K-feasible cone is a special cone surrounded by a K-feasible cut C_v, where the cut size $|C_v|$ does not exceed K. In Fig. 3(a), there is a 3-feasible cut of gate g_6, consisting of its three transitive fan-in gates, *i.e.*, $\{g1, g2, g5\}$. Correspondingly, a 3-feasible cone surrounded by this cut consists of five gates, *i.e.*, $\{g_1, g_2, g_5, g_4, g_6\}$.

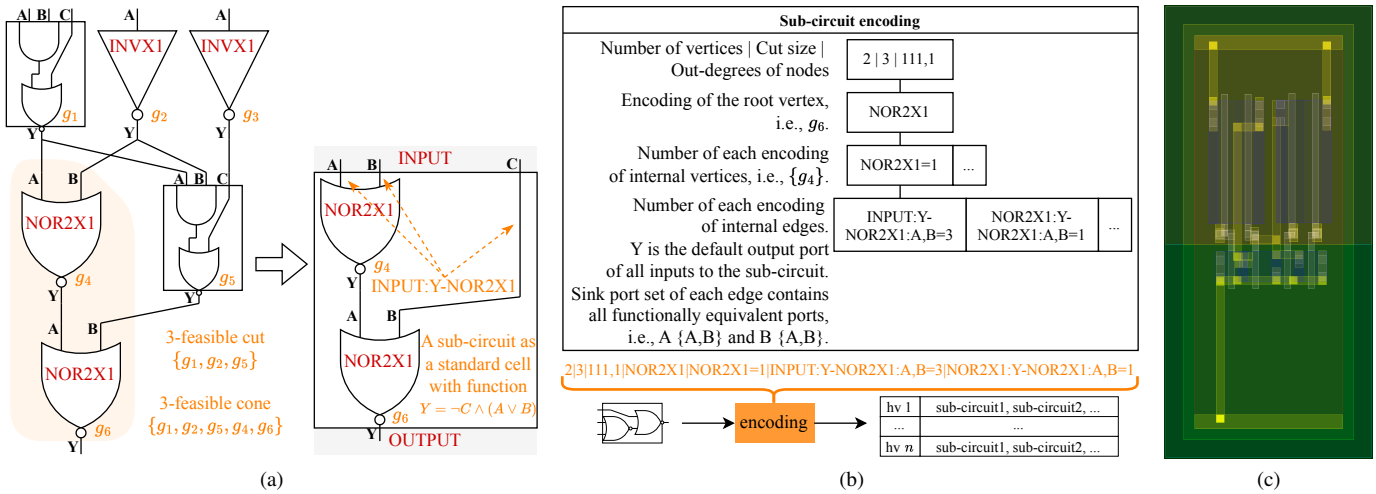
A K-feasible cone N_c associated with a vertex v and its K-feasible cut C_v can be utilized to construct a sub-circuit. This sub-circuit has one output port derived from the output of the root vertex v and $|C_v|$ input ports derived from the K-feasible cut C_v. The internal vertices within the sub-circuit maintain the same connection relationships and labels as those in the K-feasible cone N_c. Notably, it has been experimentally observed that too many input ports of a standard cell may result in its low utilization by technology mapping tools. Therefore, the vertices within the K-feasible cut are regarded as the primary inputs of the sub-circuit, with their functions and inputs ignored. This approach helps avoid introducing an excessive number of input ports for the sub-circuit. Taking Fig. 3(a) as an example, the vertices in the 3-feasible cut $\{g1, g2, g5\}$ are regarded as the primary inputs of the sub-circuit, which only introduces three input ports for this sub-circuit. In contrast, directly using the input ports of these vertices as primary inputs would introduce five input ports. Furthermore, for the subsequent cell layout synthesis, it is necessary to collect the gate information, connection relationship, and circuit function of the sub-circuit to generate its corresponding SPICE netlist. Hence, how to quickly collect complete information of all sub-circuits becomes a challenge.

Algorithm 1: The generation of K-feasible cuts and K-feasible cones.

Input: A gate-level circuit $G(V, E)$ and cut size K.
Output: K-feasible cuts Cs and K-feasible cones Ns.

$Cs \leftarrow \{\}$, $Ns \leftarrow \{\}$;
for $v \leftarrow$ *topological sort*(G) **do**
  $Cs[v] \leftarrow \{\{v\}\}$;
  $Ns[\{v\}] \leftarrow \{v\}$;
  if $FI(v) = \emptyset$ **then**
    continue;
  $P \leftarrow \text{FI}(v) \setminus \{p_1\}$;
  $Cs_v \leftarrow Cs[p_1]$;
  $Ns_v \leftarrow \{\}$;
  for $C_{p_1} \in Cs_v$ **do**
    $Ns_v[C_{p_1} \cup \{v\}] \leftarrow Ns[C_{p_1} \cup \{p_1\}] \cup \{v\}$;
  for $p_2 \leftarrow P$ **do**
    $Cs_t \leftarrow \emptyset$;
    $Cs_{p_2} \leftarrow Cs[p_2]$;
    for $C_1, C_2 \leftarrow Cs_v \times Cs_{p_2}$ **do**
      $C_t \leftarrow C_1 \cup C_2$;
      if $|C_t| \leq K$ **then**
        $Cs_t \leftarrow Cs_t \cup \{C_t\}$;
        $Ns_v[C_t \cup \{v\}] \leftarrow$
          $Ns_v[C_1 \cup \{v\}] \cup Ns[C_2 \cup \{p_2\}]$;
    $Cs_v \leftarrow Cs_t$;
  for $C_v \leftarrow Cs_v$ **do**
    $Ns[C_v \cup \{v\}] \leftarrow Ns_v[C_v \cup \{v\}]$;
  $Cs[v] \leftarrow Cs[v] \cup Cs_v$;
return Cs, Ns;

To collect all sub-circuits, how to enumerate all K-feasible cones becomes critical. Algorithm 1 presents our method of generating K-feasible cuts and K-feasible cones for a given gate-level circuit $G(V, E)$ and cut size K. The data structure Cs functions as a mapping, where each vertex serves as the key, while the corresponding set of its K-feasible cuts constitutes the value. This structure is employed to systematically record all K-feasible cuts within the circuit. Similarly, the data structure Ns operates as a mapping, where the key is a set that includes the root vertex along with its K-feasible cut, while the value corresponds to the respective K-feasible cut. This structure is utilized to meticulously record all K-feasible cones within the circuit. We first traverse all vertices within the circuit in order of the topological sort from primary



(a) (b) (c)

Fig. 3 (a) initially acquires a 3-feasible cone, and then transforms it into a sub-circuit. (b) shows how to encode the corresponding sub-circuit. (c) shows the cell layout corresponding to this sub-circuit after automatic layout synthesis by ASTRAN [6] with the FreePDK45 library [17].

inputs to primary outputs. Since each vertex can construct a K-feasible cut and a K-feasible cone, we can first record them (lines 3-4). In cases where vertex v has not fan-in, we simply move on to the subsequent vertex (lines 5-6). Conversely, if vertex v possesses a fan-in, we can get its one fan-in vertex p_1 (line 7) and retrieve the corresponding K-feasible cuts of vertex p_1 to partially initialize the cut set Cs_v for vertex v (line 8). Then, we can partially initialize the cone set Ns_v for vertex v using the K-feasible cuts of vertex p_1 (lines 9-11). Subsequently, we proceed to traverse the remaining fan-in vertices P of vertex v (lines 12-20). For each vertex p_2, we use all its K-feasible cuts to complement current partial K-feasible cuts of vertex v (lines 15-18,20), while generating the corresponding K-feasible cones (line 19). Finally, after completing the traversal of all fan-in vertices, we achieve a comprehensive collection of all K-feasible cuts and all K-feasible cones associated with vertex v (lines 21-23). The time complexity of Algorithm 1 is $O(|V| \cdot F \cdot C^2 \cdot K \log K)$, where F is the maximum fan-in of a node and C is the maximum number of cuts of a node.

B. Sub-Circuit Encoding

Following the completion of the sub-circuit collecting, a lot of sub-circuits are generated based on K-feasible cones. To find high-frequency items within these sub-circuits, how to distinguish and match these sub-circuits becomes critical. Notably, in the context of the gate-level circuit, since vertices typically represent standard cells and thereby have various types, the directed graph constructed by the gate-level circuit is heterogeneous. Moreover, each vertex possesses different input and output ports, which are usually non-equivalent. Consequently, these characteristics pose huge challenges in the process of sub-circuit matching.

To address this problem, this section proposes a novel encoding method to generate a hash value (*hv*) for each sub-circuit. In a sub-circuit structure, it consists of vertices and directed edges. So, the feature description of vertices and edges is beneficial for distinguishing different sub-circuits. For consistency, the primary inputs of the sub-circuit are also viewed as a kind of vertex with type "INPUT", while the output port of the sub-circuit is labeled as 'Y'. First, vertices and edges need to be encoded as follows.

- Vertex encoding: Since a vertex is determined by its corresponding gate type, it is encoded as the gate type. For instance, the vertex encoding of the vertex g_4 in Fig. 3(a) is "NOR2X1".
- Edge encoding: Since each gate has multiple ports and not all its ports are functionally equivalent, the identification of each edge within a circuit depends not only on its source and sink ports, but also on its source and sink vertices. Notably, the edge in the circuit can have multiple sinks, thereby divided into multiple connections, each of which features one source and one sink. Besides, there exists functional equivalence between the input ports of the gate. For example, the inputs of the "NOR2X1" gate with the NOR function can be interchanged arbitrarily without affecting functionality. Consequently, each connection can be encoded as the formulation: "*the type of source vertex*:*the name of source port*–*the type of sink vertex*:*the set of equivalent ports with the sink port*". For instance, the edge between the vertex g_4 and the vertex g_6 in Fig. 3(a) can be encoded as "NOR2X1:Y–NOR2X1:A,B", where two inputs 'A' and 'B' to vertex g_6 with type "NOR2X1" have functional equivalence.

The sub-circuit can be encoded after determining the encoding of vertices and edges. As shown in Fig. 3(b), the encoding of the sub-circuit consists of four parts. The first part reflects the scale of the sub-circuit, composed of the number of non-leaf vertices, the cut size, and the out-degrees of non-root vertices sorted by vertex encoding. The second

part records the encoding of the root vertex. The third part records the number and encoding of internal vertices and sorts them by vertex encoding. These two parts reflect the vertex distribution in the sub-circuit. The last part records the number and encoding of edges and sorts them by edge encoding. In this way, a sub-circuit can be represented by a string. For instance, the sub-circuit in Fig. 3(a) can be encoded as "2—3—111,1—NOR2X1—NOR2X1=1—INPUT:Y-NOR2X1:A,B=3—NOR2X1:Y-NOR2X1:A,B=1", where "111,1" represents that all three primary inputs have an out-degree of 1, and vertex g_4 has an out-degree of 1.

After encoding all sub-circuits, TeMACLE categorizes them into distinct groups based on their generated hash values, as shown in the bottom of Fig. 3(b). Each group type can become a candidate for generating a new standard cell. To exactly count the number of the same sub-circuits within each group, an exact matching process among all sub-circuits in each group is essential. Given that the utilization of sub-circuit encodings has effectively distinguished most of the distinct sub-circuits, the scale of each group is acceptable. Therefore, this facilitates reducing the workload associated with the subsequent sub-circuit exact matching process, thereby speeding up the matching process.

C. SAT-based Sub-Circuit Matching

Although TeMACLE groups all sub-circuits in terms of the sub-circuit encoding, it is not guaranteed that all sub-circuits within the same group are identical. Therefore, a fast sub-circuit function-and-structure-matching approach becomes imperative. As described in Section II-C, despite the availability of numerous graph matching methods for directed graphs, they may not be suitable for sub-circuit matching when considering the unique characteristics of circuits. With the advent of the Davis-Putnam-Logemann-Loveland algorithm [27] and conflict-driven clause learning [28], modern SAT algorithms have significantly improved and are widely used in the electronic design automation field, such as equivalence checking and automatic test pattern generation. Furthermore, SATMargin [29] has also demonstrated that the SAT-based method exhibits superior efficiency compared to depth-first search (DFS)-based methods in frequent subgraph testing. Consequently, referring to SATMargin, this section proposes an SAT-based sub-circuit matching approach to exactly match sub-circuits in terms of function and structure.

First, the construction of the SAT model is required to determine whether the two sub-circuits G_1 and G_2 are identical. Given that each sub-circuit can be viewed as a heterogeneous graph comprising vertices, edges, and their associated attributes, which determine the function and structure of a gate-level circuit, a meticulous matching process for these elements is essential. Consequently, according to Definition 6, this SAT model contains four parts: (i) vertex matching, (ii) vertex constraints, (iii) edge matching, and (iv) edge constraints. The specific definition of these four parts is as follows.

- Vertex matching: For vertices $v \in G_1$ with type $t(v)$ and $u \in G_2$ with type $t(u)$, they may be matched only if they have the same vertex type, *i.e.*, $t(v) = t(u)$. Otherwise, they cannot be matched. The corresponding clause C_1 can be formulated as:

$$C_1 = \bigwedge_{v \in G_1} \left(\left(\bigvee_{\substack{u \in G_2 \\ t(v)=t(u)}} l_{v,u} \right) \wedge \left(\bigwedge_{\substack{u \in G_2 \\ t(v)\neq t(u)}} \neg l_{v,u} \right) \right), \tag{2}$$

  where $l_{v,u}$ is a Boolean literal that indicates whether vertices $v \in G_1$ and $u \in G_2$ are matched. This clause ensures that each vertex in sub-circuit G_1 can find a matched vertex with the same type in sub-circuit G_2.
- Vertex constraints: Since there exists a situation that multiple vertices in sub-circuit G_1 match the same vertex in sub-circuit G_2 after introducing the vertex matching clause, the vertex constraints are introduced to ensure that any two vertices v_1 and v_2 in one sub-circuit cannot simultaneously match the same vertex u in the other sub-circuit, *i.e.*, $(\neg l_{v_1,u} \vee \neg l_{v_2,u})$ must be true. The corresponding clause C_2 can be formulated as:

$$C_2 = \bigwedge_{\substack{v_1,v_2 \in G_1, u \in G_2 \\ \text{or} \\ v_1,v_2 \in G_2, u \in G_1}} (\neg l_{v_1,u} \vee \neg l_{v_2,u}). \tag{3}$$

- Edge matching: Any two connected vertices v_1 and v_2 in one sub-circuit cannot simultaneously match any two unconnected u_1 and u_2 in the other sub-circuit, *i.e.*, $(\neg l_{v_1,u_1} \vee \neg l_{v_2,u_2})$ must be true when v_1 and v_2 are connected and u_1 and u_2 are not connected. The corresponding clause C_3 can be formulated as:

$$C_3 = \bigwedge_{\substack{v_1,v_2 \in G_1, (v_1,v_2) \in G_1, u_1,u_2 \in G_2, (u_1,u_2) \notin G_2 \\ \text{or} \\ v_1,v_2 \in G_1, (v_1,v_2) \notin G_1, u_1,u_2 \in G_2, (u_1,u_2) \in G_2}} (\neg l_{v_1,u_1} \vee \neg l_{v_2,u_2}), \tag{4}$$

  where $(v_1, v_2) \in G_1$ represents an edge in the sub-circuit G_1 from vertex v_1 to vertex v_2. On the contrary, $(v_1, v_2) \notin G_1$ represents no edge between vertices v_1 and v_2 in the sub-circuit G_1. This clause ensures that the matched vertices have the same connection structure.
- Edge constraints: Given that the edge matching clause only considers the connection relationship, the attributes of edges also need to be further matched, including source and sink ports and the type of source and sink vertices, *i.e.*, the edge encoding introduced in Section IV-B. Therefore, only if one edge in sub-circuit G_1 has the same edge encoding as the other edge in sub-circuit G_2 may they be matched. The corresponding clause C_4 is formulated as:

$$C_4 = \bigwedge_{\substack{v_1,v_2 \in G_1, (v_1,v_2) \in G_1 \\ u_1,u_2 \in G_2, (u_1,u_2) \in G_2 \\ t(v_1,v_2) \neq t(u_1,u_2)}} (\neg l_{v_1,u_1} \vee \neg l_{v_2,u_2}), \tag{5}$$

  where $t(v_1, v_2)$ represents the edge encoding of the edge from vertex v_1 to vertex v_2. This clause ensures that the characteristics of the two sub-circuits match.

Now all clauses are defined. Consequently, according to Definition 4, the SAT model of sub-circuit matching can be

defined as:

$$\exists\{l_{v,u} \mid v \in G_1, u \in G_2\}, C_1 \wedge C_2 \wedge C_3 \wedge C_4 \equiv \mathbf{True}, \quad (6)$$

where there are a total of $|G_1|\cdot|G_2|$ Boolean literals. C_1 and C_2 ensure a one-to-one mapping relationship between the vertices in the two sub-circuits. C_3 and C_4 guarantee the edges within two sub-circuits can be matched under the consideration of circuit features. Therefore, the determination of the existence of a solution for this model using the SAT solver can make sure whether the two sub-circuits G_1 and G_2 are identical.

Consequently, the proposed SAT-based sub-circuit matching method can further subdivide each group clustered by the sub-circuit encoding. Finally, each group represents a specific type of sub-circuit characterized by identical topology and Boolean function, with the corresponding number of occurrences of each sub-circuit being counted. In this way, all distinct sub-circuits and their respective quantities can be identified.

D. Standard Cell Synthesis

As illustrated in Section II-A, merging existing standard cells can produce a new standard cell to save the circuit area. TeMACLE also applies this strategy to generate new standard cells. After identifying the sub-circuit, TeMACLE first automatically extracts the Boolean function expression between the inputs and the output of the sub-circuit according to the topology of the sub-circuit and the Boolean functions of vertices within the sub-circuit. However, the resulting Boolean function expression is usually complex and incomprehensible. To address this problem, TeMACLE simplifies the extracted Boolean function expression into the sum-of-products (SOP) form by the Quine-McCluskey algorithm [30], [31]. The Quine-McCluskey algorithm is a method used for the minimization of Boolean functions, which facilitates the simplification of Boolean expressions into a reduced form through the utilization of prime implicants. Furthermore, it provides a deterministic methodology for verifying the attainment of the minimal form of a Boolean function. Hence, the final Boolean function expression of the sub-circuit in Fig. 3(a) can be simplified as $Y = \neg C \wedge (A \vee B)$.

Furthermore, in order to acquire the physical information of the newly generated standard cell, TeMACLE has the capability to integrate the automatic standard cell layout synthesis tool. Specifically, when dealing with a specific sub-circuit, TeMACLE first uses the SPICE netlists of its internal vertices to automatically construct its SPICE netlist according to the interconnections among these internal vertices. Subsequently, the layout synthesis tool is applied to generate its cell layout with the given process technology. Fig. 3(c) shows the layout schematic of the sub-circuit derived from Fig. 3(a), which is generated by ASTRAN with the FreePDK45 library. In this way, the area of this new standard cell can be obtained.

It is worth noting that layout synthesis is a time-consuming process. To minimize redundancy during layout synthesis, the layout synthesis results of sub-circuits are stored in an auxiliary standard cell library with their respective Boolean function expressions as the identifying key. To improve the efficiency of sub-circuit retrieval, we utilize the Quine-McCluskey algorithm to simplify the Boolean function expression for each sub-circuit, thereby significantly improving the success rate of retrieval. Before performing the layout synthesis for a given sub-circuit, a function retrieval is conducted within this library to determine the presence of the sub-circuit to be synthesized. Only in cases where the Boolean function corresponding to this sub-circuit is not found in the library will the layout synthesis be performed. Although the introduction of a predefined library may incur extra memory costs, it can significantly speed up the standard cell library extension process. Hence, it is essential to achieve the Boolean function matching. According to Definition 5, we can verify whether a one-to-one permutation map ρ can make two Boolean functions f

Algorithm 2: The final flow of TeMACLE.

Input: A given Boolean network G, an original standard cell library L, a process technology, K, N, and T.
Output: An extended standard cell library.

```
Map G into gate-level circuit G' using library L;
Calculate the circuit area s of gate-level circuit G';
for i ← 0 to T do
    Collect all sub-circuits with at most N original vertices from G' by Section IV-A;
    Encode and group these sub-circuits by Section IV-B;
    Refine the grouping of all sub-circuits within each group by Section IV-C to obtain all distinct sub-circuits Q;
    Sort Q in descending order based on their quantity;
    flag ← 0;
    while Q ≠ ∅ do
        Q ← Q \ {g};
        if the number of g < 2 then
            flag ← 1;
            break;
        Generate g's function expression, SPICE netlist, and cell layout using the given process technology, detailed in Section IV-D;
        New a standard cell c using g's function and layout;
        L ← L ∪ {c};
        Remap G into G' using new library L;
        Calculate circuit area s' of gate-level circuit G';
        if s > s' then
            s ← s';
            break;
        else
            L ← L \ {c};
    if Q ≠ ∅ ∨ flag = 1 then
        break;
return extended standard cell library L;
```

TABLE I Experimental results on the EPFL benchmark [16] and FreePDK45 library [17].

Circuit	Original			AutoCellLibX [13]				TeMACLE				labellist
	gates	area (μm^2)	depth	area (μm^2)	area (%)	time (s)	ECSize	area (μm^2)	area (%)	depth	time (s)	ECSize
adder128	768	2373.25	193	1900.56	19.92	1381.01	2/4/4/3/3	1884.15	20.61	129	5.18	3/5
bar	2283	5426.52	10	4975.61	8.31	754.39	3/3/3/3/2	4607.51	15.09	9	22.22	3/2/2/4
cavlc	532	1232.38	11	1194.37	3.08	71.97	2/2	1187.73	3.62	10	7.16	2/2/2/2/2
div	42220	109675.88	2213	97629.82	10.98	410.18	2/2/3/3/2	86576.50	21.06	2200	164.19	2/2/2/2/3
hyp	187027	505059.72	10749	468522.11	7.23	367.52	2/2/3/2	415107.84	17.81	10733	818.84	2/2/2/2/2
i2c	1094	2461.95	11	2400.11	2.51	88.63	2/3	2368.51	3.80	11	11.16	2/2/3/2/3
int2float	182	420.49	9	412.43	1.92	439.60	5	407.74	3.03	9	4.08	2/2/3
log2	21827	58706.61	232	55187.01	6.00	604.45	5/2/2/2/3	52795.23	10.07	221	187.50	2/2/2/2/2
max	2694	6006.57	178	5475.46	8.84	311.32	3/2/2/2/3	5002.03	16.72	176	14.91	2/2/3/2/2
mem_ctrl	36517	81936.96	74	78879.38	3.73	85.21	2/2/2/2/2	73855.93	9.86	74	233.73	2/2/2/3/2
multiplier	16684	48909.51	179	44385.50	9.25	2353.00	5/4/2/3/2	43687.63	10.68	178	82.97	2/2/2/3/3
priority	1072	2344.15	64	2026.28	13.56	702.76	2/4/3/4/2	1691.72	27.83	64	8.09	2/2/3/2
router	240	569.73	25	551.39	3.22	1121.60	3/2	542.09	4.85	25	2.70	2/2/3/2/2
sin	4145	10882.60	116	10386.81	4.56	243.91	2/2/2/2/3	9609.23	11.70	113	31.75	2/2/2/2/3
sqrt	18140	47752.68	3039	43455.75	9.00	2428.33	3/2/4/4/3	32940.58	31.02	3038	95.61	2/2/2/5/4
square	13895	37336.10	167	35389.53	5.21	363.72	2/3/3/2/3	33215.22	11.04	167	81.64	2/2/2/4/2
voter	9973	28922.49	38	26996.74	6.66	1698.92	2/2/2/2/2	23512.72	18.70	33	51.55	2/2/2/3/2
Ave. ratio		1	1	0.93				0.86		0.96		

and g matched by the following SAT model:

$$\exists X \in \{\mathbf{True}, \mathbf{False}\}^n, f(X) \oplus g(\rho(X)) = \mathbf{True}. \quad (7)$$

When this model yields a solution, it indicates that the permutation map ρ cannot make f and g matched; otherwise, it can. Consequently, we can traverse all one-to-one permutation maps (totally $n!$) to determine whether two Boolean functions f and g can be matched.

E. Final Flow

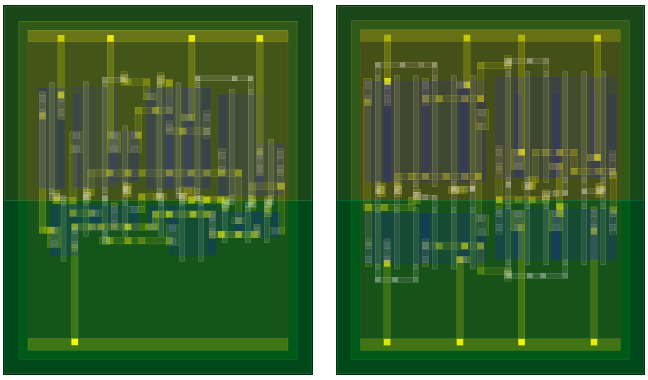
Up to this point, this paper has already introduced the method through which TeMACLE acquires all the different sub-circuits along with their corresponding quantities. Nonetheless, the determination of the final extended standard cells poses a significant challenge. As discussed at the beginning of Section IV, it is impractical to perform a comprehensive exploration of the entire solution space, involving the synthesis of all sub-circuits into new standard cells, the selection of any T among them for technology mapping iteration, and the subsequent determination of the optimal combination as the final extend standard cells. Consequently, this section proposes a technology mapping-aware strategy to choose a maximum of T different kinds of sub-circuits as the final standard cells, as outlined in Algorithm 2.

Initially, a given Boolean network G is mapped into a gate-level circuit G' using the original standard cell library L, leading to the determination of the original circuit area s (lines 1-2). Following collecting and grouping all the sub-circuits within circuit G' (lines 4-6), $|Q|$ distinct sub-circuits can be derived, thereby forming the set Q. The set Q is then arranged in descending order according to the quantity of each sub-circuit and the number of original standard cells it contains (line 7). Given the objective of minimizing circuit area and more occupation of circuit area by sub-circuits with higher quantities, the area compaction on these sub-circuits may yield greater area savings. Consequently, each iteration focuses on selecting the sub-circuit with the highest quantity as an extended standard cell (lines 9–25). If one sub-circuit occurs only once, the process of selecting extended standard cells is prematurely terminated (lines 11-13, 24-25). Once a sub-circuit is chosen, TeMACLE automatically generates its function expression using the Quine-McCluskey algorithm (line 14). Moreover, TeMACLE automatically generates its SPICE netlist and then synthesizes its cell layout using the given process technology (line 14). Subsequently, TeMACLE creates a new standard cell corresponding to the function and layout of this sub-circuit to extend the standard cell library and then remaps the original Boolean network with this extended standard cell library (lines 15–18). If the resulting gate-level circuit has a smaller area, TeMACLE proceeds to the next iteration (lines 19–21). Otherwise, TeMACLE discards this standard cell and continues to select the next sub-circuit (line 23). Finally, the final extended standard cell library containing a maximum of T new standard cells can be obtained (line 26).

V. Experimental Results

The proposed standard cell library extension framework TeMACLE[1] was implemented using Python language, and CryptoMiniSat [32], an advanced incremental SAT solver, was selected to solve our proposed SAT-based sub-circuit matching model. Furthermore, the experiments chose ASTRAN as the standard cell layout tool. It can support FreePDK45 [17], an open-source generic process design kit with a predictive 45 nm CMOS technology process. Meanwhile, this paper selected several original standard cells from the FreePDK45 standard cell library to construct the original standard cell library. These cells included "AND2X2", "AOI21X1", "BUFX2", "INVX1", "NAND2X1", "NAND3X1", "NOR2X1", "NOR3X1", "OAI21X1", "OR2X2", "XNOR2X1", and "XOR2X1". Specifically, the SPICE netlists of new standard cells are

[1] https://github.com/Flians/TeMACLE



(a) $Y = (B \wedge C) \vee (B \wedge \neg A) \vee (C \wedge \neg A)$

(b) $Y = (A \wedge B \wedge \neg C) \vee (A \wedge C \wedge \neg B) \vee (B \wedge C \wedge \neg A) \vee (\neg A \wedge \neg B \wedge \neg C)$

Fig. 4 New standard cells generated for "adder128".

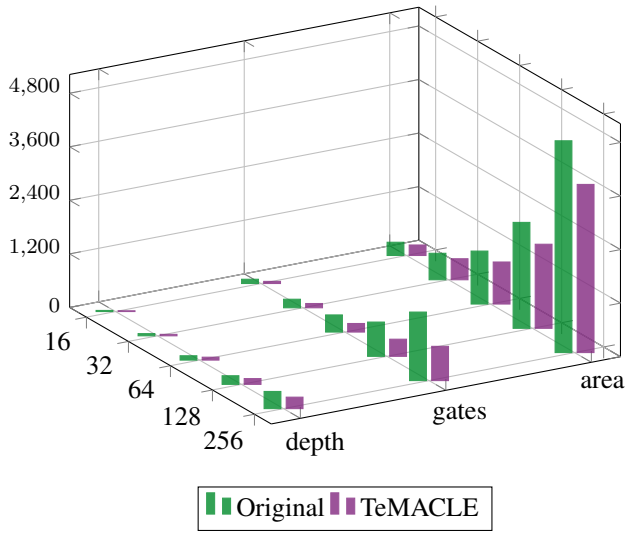


Fig. 5 The generation of full adders with varying bit numbers using the original and extended standard cell libraries, respectively.

automatically constructed based on the SPICE netlists of FreePDK45 standard cells and then are synthesized by ASTRAN [6] to generate their corresponding layouts that adhere to the design rules of FreePDK45 [17]. The experiments used the EPFL benchmark [16] to evaluate TeMACLE and used mockturtle [25], an advanced open-source logic network library, to carry out logic synthesis and technology mapping for all benchmark circuits. First, the node resynthesis algorithm `node_resynthesis` with the resynthesis function based on DSD decomposition `dsd_resynthesis` was employed to optimize all benchmark circuits. Then, these optimized circuits were mapped using the standard cell library through the cut-based technology mapping algorithm `map`, where the maximum number of cuts per node is 24.

In terms of the experimental parameter configuration, the parameter T serves to establish the upper limit on the number of new standard cells to be generated. Given the limitation of practical standard cell library generation cost, this paper adopts a value of 5 for T. On the other hand, the parameter N determines the maximum number of original standard cells contained within the sub-circuit corresponding to each newly created standard cell. To mitigate the intricacies inherent in the standard cell layout synthesis process stemming from the SPICE netlist of new standard cells, N was set to 5. Lastly, the parameter K indicates the maximum number of inputs to each new standard cell. Given that an excessive number of inputs can pose challenges for the existing technology mapping tools, K was set to 3.

The experiments were executed on the machine with Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz and 256.0 GB memory running Ubuntu 22.04. AutoCellLibX [13] was selected as the baseline, which directly extracts non-overlapping sub-circuits to generate new standard cells. AutoCellLibX used the same configuration of the parameters T and N. Table I shows the experimental results. The "Original" part shows the attributes of benchmark circuits, including the number of used standard cells ("gates"), the circuit area ("area (μm^2)"), and the circuit depth ("depth") after technology mapping using the original standard cell library. The "AutoCellLibX" and "TeMACLE" parts show the results of the baseline and our proposed TeMACLE, respectively. Since AutoCellLibX clusters standard cells to identify sub-circuits for compaction, the resulting extended standard cells are not regular and unsuitable for technology mapping. Meanwhile, due to the irregular circuit structure after cell compaction, the circuit depth is not statistical for AutoCellLibX. Therefore, the "area (μm^2)" in "AutoCellLibX" denotes the circuit area after compacting the area of sub-circuits corresponding to the extended standard cells, while the "area (μm^2)" and "depth" in "TeMACLE" denote the circuit area and circuit depth after technology mapping using the extended standard cell library, respectively. "area (%)" denotes the average reduction in circuit area compared to one obtained using the original standard cell library. "time (s)" denotes the runtime in seconds for performing the standard cell library extension. Lastly, "ECSize" denotes the number of original standard cells within the sub-circuit corresponding to each new cell. For instance, in the case of circuit "adder128" in TeMACLE, "3/5" denotes that TeMACLE has generated two new standard cells for circuit "adder128", and their corresponding sub-circuit contains three and five original standard cells from the original standard cell library, respectively.

Overall, compared to the circuits based on the original standard cell library, TeMACLE achieved a significant reduction in the circuit area, specifically by 13.97% on average. Meanwhile, the circuits generated using the standard cell library extended by TeMACLE exhibited an average reduction of 1.96 times in the circuit area compared to those generated using the standard cell library extended by AutoCellLibX. Specifically, TeMACLE demonstrated a significant advantage over AutoCellLibX across all testcases. In particular, TeMACLE demonstrated superior performance on large circuits, such as "div", "hyp", "mem_ctrl", and "sqrt". TeMACLE showcased

notable improvements on these four circuits, with circuit area reductions of 1.92, 2.46, 2.64, and 3.45 times those achieved by AutoCellLibX, respectively.

As for efficiency, the runtime on each testcase was acceptable, and all tasks were completed by TeMACLE within less than 900 s. In general, the runtime required by TeMACLE is nearly one-third that of AutoCellLibX on average. The cell layout synthesis process in AutoCellLibX accounts for a significant portion of the runtime, primarily due to the extraction of irregular and large sub-circuits. In contrast, TeMACLE mitigates this issue by employing a predefined library of pre-synthesized sub-circuits, resulting in a reduced runtime contribution from this process. Instead, the runtime occupation of the collection and encoding of sub-circuits becomes apparent in TeMACLE. Although AutoCellLibX exhibits a shorter runtime on the "hyp" and "mem_ctrl" circuits, it primarily focuses on extracting non-overlapping sub-circuits without accounting for the compatibility of new standard cells across diverse circuit structures and the complexity involved in standard cell layout synthesis. Consequently, AutoCellLibX tends to require more time to achieve less area reduction in most circuits. In contrast, TeMACLE considers these factors and thus needs basic time to produce and handle sub-circuits. Nevertheless, TeMACLE still demonstrates an acceptable runtime on these two circuits, particularly given its significant advantages in terms of circuit area savings.

Moreover, the compatibility and effectiveness of the standard cell library extended by TeMACLE were substantiated. TeMACLE introduced two new cells for the "adder128" circuit, whose respective functions and layouts are illustrated in Fig. 4. Notably, the cell in Fig. 4(a) implements a 3-input majority function with an inverter in front of the input port A, while the other in Fig. 4(b) implements a 3-input exclusive-OR (XOR) function with an inverter in front of input port C. This just aligns with the multi-digit binary coded decimal (BCD) adder design [34], where the summary bit and the carry-out bit can be derived by a 3-input majority function and a 3-input exclusive-OR function, respectively. Subsequently, this extended standard cell library was used for technology mapping of 16, 32, 64, and 256-bit full adders, respectively. Fig. 5 shows the circuit depth, gate count, and circuit area of generated full adders with varying bit numbers using the original and extended standard cell library, respectively. Noteworthy, the generated circuits still had an average area reduction of 20.64% compared to those generated using the original standard cell library. This substantial decrease in the circuit area demonstrated the compatibility and effectiveness of TeMACLE's results.

Finally, To demonstrate the improvement of our framework on the high-density standard cell library, this paper also selected ASAP7 [33], an open-source generic process design kit with a predictive 7 nm CMOS technology process, to synthesize the design rule check (DRC)-clean standard cell layouts by an industrial standard cell layout generation tool. This paper initially selected the above standard cells from the ASAP7 7.5-track standard cell library [33] to construct the original standard cell library, and then used AutoCellLibX and TeMACLE to extend the original library. Table II shows the experimental results. Overall, TeMACLE demonstrated a significant advantage over AutoCellLibX in all testcases. Specifically, compared to the circuits based on the initial standard cell library, TeMACLE had an average circuit area reduction of 18.30%. Furthermore, the circuits generated using the standard cell library extended by TeMACLE achieved an average reduction in the circuit area of 6.12 times compared to those generated using the standard cell library extended by AutoCellLibX, while maintaining an average runtime of only one-seventh.

VI. Conclusion

This paper highlighted the importance and necessity of standard cell library extension in VLSI design and proposed TeMACLE, a technology mapping-aware area-efficient framework for standard cell library extension. TeMACLE uses a novel sub-circuit encoding to group all K-feasible cuts and then uses an SAT-based sub-circuit matching method to calculate the frequency of each sub-circuit. Subsequently, TeMACLE iteratively selects sub-circuits in descending order of their frequency, where the layout of the new cell derived from each sub-circuit is synthesized by ASTRAN with FreePDK45 technology. The final standard cells beneficial to area-efficient technology mapping are selected to extend the standard cell library. The experimental results on the EPFL benchmark demonstrated the effectiveness and efficiency of TeMACLE.

This paper focuses on circuit area optimization through the extension of the standard cell library. While the circuit area is a primary concern, other metrics, including the timing and power consumption, also play significant roles in circuit design. In future work, we will explore timing and power consumption optimization by the extension of the standard cell library. This necessitates that standard cell layout tools provide more comprehensive data to construct the liberty file containing cell timing and power information. Additionally, the selection of sub-circuits also needs to consider the metrics of standard cell layout synthesis for better layouts, such as design rule compliance, pin accessibility, area, and timing.

References

[1] R. Reis, "Design automation of transistor networks, a new challenge," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2011, pp. 2485–2488.

[2] J.-s. Seo, I. L. Markov, D. Sylvester, and D. Blaauw, "On the decreasing significance of large standard cells in technology mapping," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2008, pp. 116–121.

[3] M. Lefebvre and D. Marple, "The future of custom cell generation in physical synthesis," in *ACM/IEEE Design Automation Conference (DAC)*, 1997, pp. 446–451.

[4] J. Togni, F. Schneider, V. Correia, R. Ribas, and A. Reis, "Automatic generation of digital cell libraries," in *Symposium on Integrated Circuits and Systems Design (SBCCI)*, 2002, pp. 265–270.

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

TABLE II Experimental results on the EPFL benchmark [16] and ASAP7 process design kit [33].

Circuit	Original			AutoCellLibX [13]				TeMACLE				
	gates	area (μm^2)	depth	area (μm^2)	area (%)	time (s)	ECSize	area (μm^2)	area (%)	depth	time (s)	ECSize
adder128	769	94.17	193	85.07	9.66	5467.74	2/4/4/4	65.32	30.64	129	6.54	2/5/3
bar	2456	265.02	10	253.71	4.27	2935.29	3/3	204.02	23.02	9	13.24	2/2/2/3
dec	316	27.88	4	27.12	2.72	843.84	4/3	26.30	5.65	3	3.66	2/2
div	43755	4480.59	2213	4241.57	5.33	6729.66	3/3/3/2/2	3551.27	20.74	2200	212.67	2/2/3/2/2
hyp	185740	21314.22	10749	20696.91	2.90	2022.29	2/2/2/2	17881.03	16.11	10733	853.90	2/2/2/2/3
i2c	1081	112.81	11	112.81	0.00	14.51	3/3/3/2	94.65	16.09	11	7.14	2/3/4/2/2
int2float	182	19.68	9	19.68	0.00	11.05	3/2/3/3/2	16.42	16.59	9	4.70	2/3/2/4
log2	22168	2588.82	232	2524.79	2.47	1292.03	2/2/2/2/2	2123.45	17.98	221	134.78	2/3/4/2/2
max	2744	258.49	178	245.10	5.18	1578.50	3/2/2/2/3	215.14	16.77	177	38.03	2/2/2/3/3
mem_ctrl	35661	3607.95	74	3521.38	2.40	337.78	2/3/3/2/2	3380.17	6.31	74	169.82	2/2/2/3/3
multiplier	17852	2069.38	179	2000.04	3.35	3303.80	2/2/2/3/2	1866.87	9.79	178	76.69	2/2/2/3/3
priority	1195	110.41	64	103.24	6.50	3034.58	3/2/2002	73.53	33.41	64	7.46	2/2/3/4
router	235	27.10	25	26.81	1.08	4448.85	3	22.60	16.62	25	4.10	2/3/4/3/2
sin	4134	473.38	116	460.82	2.65	1344.33	2/2/2/2/2	381.44	19.42	113	28.34	2/3/4/2/2
sqrt	18103	2140.17	3039	2082.99	2.67	2745.66	2/3/2/2	1375.56	35.73	3038	135.95	2/2/2/3/4
square	14290	1595.18	167	1567.19	1.75	512.27	2/3/2	1472.46	7.69	167	106.26	2/2/2/2/2
voter	9780	1150.04	38	1124.09	2.26	7338.22	3/2/2/2	915.58	20.39	32	64.63	3/2/2/4/5
Ave. ratio		1	1	0.97				0.82		0.95		

[6] A. M. Ziesemer and R. A. d. L. Reis, "Simultaneous two-dimensional cell layout compaction using MILP with ASTRAN," in *IEEE Annual Symposium on VLSI (ISVLSI)*, 2014, pp. 350–355.

[7] P. Van Cleeff, S. Hougardy, J. Silvanus, and T. Werner, "BonnCell: Automatic cell layout in the 7-nm Era," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 39, no. 10, pp. 2872–2885, 2020.

[8] 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," *IEEE Transactions on Very Large Scale Integration Systems (TVLSI)*, vol. 29, no. 6, pp. 1178–1191, 2021.

[9] Y.-L. Li, S.-T. Lin, S. Nishizawa, H.-Y. Su, M.-J. Fong, O. Chen, and H. Onodera, "NCTUcell: A DDA- and delay-aware cell library generator for FinFET structure with implicitly adjustable grid map," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 41, pp. 5568–5581, 2022.

[10] 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 and Systems (TCAD)*, vol. 41, no. 5, pp. 1495–1508, 2022.

[11] C. Pilato, F. Ferrandi, and D. Pandini, "A fast heuristic for extending standard cell libraries with regular macro cells," in *IEEE Annual Symposium on VLSI (ISVLSI)*, 2010, pp. 23–28.

[12] S. Kiamehr, M. Ebrahimi, F. Firouzi, and M. B. Tahoori, "Extending standard cell library for aging mitigation," *IET Computers & Digital Techniques*, vol. 9, no. 4, pp. 206–212, 2015.

[13] T. Liang, J. Chen, L. Li, and W. Zhang, "AutoCellLibX: Automated standard cell library extension based on pattern mining," 2022, arXiv:2207.12314.

[14] K. A. C. Albinagorta, C. Conceição, C. S. Cardenas, and R. Reis, "Exploring area and total wirelength using a cell merging technique," in *IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, 2019, pp. 329–334.

[15] G. Northrop, "Design technology co-optimization in technology definition for 22nm and beyond," in *Symposium on VLSI Technology (VLSIT)*, 2011, pp. 112–113.

[16] L. Amarù, P.-E. Gaillardon, and G. De Micheli, "The EPFL combinational benchmark suite," in *IEEE/ACM International Workshop on Logic Synthesis*, 2015.

[17] J. E. Stine, I. Castellanos, M. Wood, J. Henson, F. Love, W. R. Davis, P. D. Franzon, M. Bucher, S. Basavarajaiah, J. Oh, and R. Jenkal, "FreePDK: An open-source variation-aware design kit," in *IEEE International Conference on Microelectronic Systems Education (MSE)*, 2007, pp. 173–174.

[18] K. Baek and T. Kim, "CSyn-fp: Standard cell synthesis of advanced nodes with simultaneous transistor folding and placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 43, no. 2, pp. 627–640, 2024.

[19] E. Testa, M. Soeken, L. G. Amar, and G. De Micheli, "Logic synthesis for established and emerging computing," *Proceedings of the IEEE*, vol. 107, no. 1, pp. 165–184, 2019.

[20] G. De Micheli, "Logic synthesis for emerging technologies," in *IEEE International Conference on ASIC (ASICON)*, 2023, pp. 1–4.

[21] R. Fu, M. Wang, Y. Kan, O. Chen, N. Yoshikawa, B. Yu, and T.-Y. Ho, "Buffer and splitter insertion for adiabatic quantum-flux-parametron circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, pp. 1–14, 2024.

[22] R. Fu, R. Wille, and T.-Y. Ho, "RCGP: An automatic synthesis framework for reversible quantum-flux-parametron logic circuits based on efficient cartesian genetic programming," in *ACM/IEEE Design Automation Conference (DAC)*, 2024.

[23] R. Fu, O. Chen, N. Yoshikawa, and T.-Y. Ho, "Exact logic synthesis for reversible quantum-flux-parametron logic," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2023, pp. 1–9.

[24] R. Brayton and A. Mishchenko, "ABC: An academic industrial-strength verification tool," in *International Conference on Computer-Aided Verification (CAV)*. Springer Berlin Heidelberg, 2010, pp. 24–40.

[25] M. Soeken, H. Riener, W. Haaswijk, E. Testa, B. Schmitt, G. Meuli, F. Mozafari, S.-Y. Lee, A. Tempia Calvino, and G. Marakkalage, Dewmini Sudara De Micheli, "The EPFL logic synthesis libraries," 2022, arXiv:1805.05121v3.

[26] S. Bouhenni, S. Yahiaoui, N. Nouali-Taboudjemat, and H. Kheddouci, "A survey on distributed graph pattern matching in massive graphs," *ACM Computing Surveys*, vol. 54, no. 2, 2021.

[27] M. Davis, G. Logemann, and D. Loveland, "A machine program for theorem-proving," *Communications of the ACM*, vol. 5, no. 7, pp. 394–397, 1962.

[28] J. Marques Silva and K. Sakallah, "GRASP-A new search algorithm for satisfiability," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 1996, pp. 220–227.

[29] M. Liu and P. Li, "SATMargin: Practical maximal frequent subgraph mining via margin space sampling," in *The Web Conference (WWW)*, 2022, pp. 1495–1505.

[30] W. V. Quine, "A way to simplify truth functions," *The American Mathematical Monthly*, vol. 62, no. 9, pp. 627–631, 1955.

[31] E. J. McCluskey Jr., "Minimization of boolean functions*," *The Bell System Technical Journal*, vol. 35, no. 6, pp. 1417–1444, 1956.

[32] M. Soos, K. Nohl, and C. Castelluccia, "Extending SAT solvers to cryptographic problems," in *The International Conferences on Theory and Applications of Satisfiability Testing (SAT)*, O. Kullmann, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 244–257.

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

[34] Z. Chu, Z. Li, Y. Xia, L. Wang, and W. Liu, "BCD adder designs based on three-input XOR and majority gates," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 6, pp. 1942–1946, 2021.

Rongliang Fu received his BS degree in software engineering from the Northwestern Polytechnical University, Xi'an, China, in 2018 and his MS degree in computer science and technology from the University of Chinese Academy of Sciences, Beijing, China, in 2021. He is currently studying for his Ph.D degree in the Department of Computer Science and Engineering, The Chinese University of Hong Kong. His research interests include electronic design automation and computer architecture.

Zhao Wang (M'24) received the M.S. and Ph.D. degrees from Southeast University in 2002 and 2010, respectively. He is currently an Associate Professor at Southeast University. His primary research interests include EDA, VLSI DSP, and embedded systems.

Bei Yu (M'15-SM'22) received the Ph.D. degree from The University of Texas at Austin in 2014. He is currently an Associate Professor in the Department of Computer Science and Engineering, The Chinese University of Hong Kong. He has served as TPC Chair of ACM/IEEE Workshop on Machine Learning for CAD, and in many journal editorial boards and conference committees. He received eleven Best Paper Awards from ICCAD 2024 & 2021 & 2013, IEEE TSM 2022, DATE 2022, ASPDAC 2021 & 2012, ICTAI 2019, Integration, the VLSI Journal in 2018, ISPD 2017, SPIE Advanced Lithography Conference 2016, six ICCAD/ISPD contest awards, and many other awards, including DAC Under-40 Innovator Award (2024), IEEE CEDA Ernest S. Kuh Early Career Award (2022), and Hong Kong RGC Research Fellowship Scheme (RFS) Award (2024).

Tsung-Yi Ho (F'24) is a Professor in the Department of Computer Science and Engineering, The Chinese University of Hong Kong (CUHK). He received his Ph.D. in Electrical Engineering from National Taiwan University in 2005. His research interests include several areas of computing and emerging technologies, especially in the design automation of microfluidic biochips. He was a recipient of the Best Paper Award at the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems in 2015. Currently, he serves as the VP Conferences of IEEE CEDA, and the Executive Committee of ASP-DAC and ICCAD. He is a Distinguished Member of ACM and a Fellow of IEEE.