SUMMARY This study introduces a pattern-matching method to enhance the efficiency and accuracy of physical verification of cell libraries. The pattern-matching method swiftly compares layouts of all I/O units within a specific area, identifying significantly different I/O units. Utilizing random sampling or full permutation can improve the efficiency of verification of I/O cell libraries. All permutations within an 11-unit I/O unit library can produce 39,916,800 I/O units (11!), far exceeding the capacity of current IC layout software. However, the proposed algorithm generates the layout file within 1 second and significantly reduces the DRC verification time from infinite duration to 63 seconds executing 415 DRC rules. This approach effectively improves the potential to detect layer density errors in I/O libraries. While conventional processes detect layer density and DRC issues only when adjacent I/O cells are placed due to layout size and machine constraints, in this work, the proposed algorithm selectively generates multiple distinct combinations of I/O cells for verification, crucial for improving the accuracy of physical design.

key words: pattern matching; physical verification; cell libraries; layer density; physical design.

1. Introduction

System-on-Chip (SoC) integrates diverse electronic system IPs onto a single chip, necessitating the concurrent utilization of I/O Library, standard cell, and memory IP. In the SoC chip design process, discovering design rule check (DRC) errors only during the final IP integration stage poses a significant and intricate engineering problem, especially concerning I/O cells. Unlike standard cell libraries or memory IPs, I/O cells in SoC design cannot easily adjust their positions to resolve DRC errors or layer density rule issues. Density DRC errors often exhibit regional and local characteristics rather than arising entirely due to the specific contiguity of I/O cells. Therefore, incorporating potentially diverse permutations into the physical verification process during I/O cell library design helps reduce the likelihood of layer density errors during SoC design, especially at the I/O PAD Ring.

Advancements in semiconductor manufacturing have enabled system-on-a-chip (SoC) design. To meet short time-to-market requirements, fast and accurate SoC design and verification is essential. A reusable intellectual property (IP) platform was developed that uses a framework and reusable design procedure to efficiently combine IPs from various sources into an SoC [1]-[2]. A common coding style for soft IPs was designed that provides design guidelines for integrating and accurately verifying IPs from various sources [3]. In the case of mixed-signal IPs, a pipelined analog-to-digital converter was demonstrated [4]. This highlights that designing mixed-signal IPs involves both a top-down approach for high-level parametric behavioral models involving system-level integration and a bottom-up approach for verifying and extracting features from cells and back-annotating them into the high-level model [5]-[6]. Silicon IP libraries, such as standard cell libraries and input/output (I/O) libraries, are essential components in technology [7]-[10]. Designing a fully verified and reusable library is challenging. In 2003, Zhang et al. used NEC Corporation’s advanced 0.25-μm complementary metal-oxide semiconductor (CMOS) process to design a cell library and I/O library [4]. The cell library contained more than 190 cells, each including simulation, symbol, and layout libraries necessary for SoC integration. As part of their study, they used a simple adder designed with the Verilog hardware description language to demonstrate the feasibility of the cell library. Although they made proposals for physical layout, reusable design methods, and IP design solutions, they did not address the importance of verification of the physical designs for reusable IPs. Moreover, they failed to address the issue with I/O libraries where once arranged on an SoC, I/O cells cannot be rearranged without violating design rule checks (DRCs). Flaws in physical design can lead to delays in the back-end flow of SoC development.

To achieve reusability, universal standard cell and I/O libraries have been designed with consideration of standard physical layout design rules, such as layer spacing, width, and enclosure, and potential DRC violations when cells are arranged in parallel (known as butting rules). Despite the fact that butting rules are well established and have been effective in reducing DRC violations in cells repeated randomly in parallel, they have become inadequate for addressing the increasing design considerations for physical layout as manufacturing processes evolve. For example, layer density plays a crucial role in etching and chemical mechanical polishing quality, which directly affects yield[11]-[12]. Checking layer density involves both global density (amount of wafer area to be etched) and local density. An excessively high or low global density can result in substandard etching outcomes, with die-to-die variations on...
the wafer (global variations) and varying line widths across different parts of the wafer. Inconsistent etching during an aluminum back-end-of-line process can lead to resistance variations across different parts of a wafer [13]–[15].

Local density has become a significant focus in advanced manufacturing processes [15]–[16]. An area with high local density (meaning it is heavily loaded) contains fewer etch reactants, resulting in a slower etching rate and variations in the lines [13]. These variations can lead to differences in line resistance and short circuits, which reduce the yield; this phenomenon is known as the micro loading effect. Chemical mechanical polishing involves removing deposited films through friction and chemical reactions. The removal rate may vary depending on the density of different layers. For example, high-density areas may be polished away faster than low-density areas, creating local dents in high-density regions, as shown in the “High M1 Coverage” area indicated in Fig. 1(a). This poses challenges when polishing the second metal layer (M2; Fig. 1). Reducing the polish level may cause a short circuit between the first and second metal layers at locations where dents exist (Fig. 1(d)). Conversely, increasing polish level to prevent short circuits may increase resistance within the second layer in areas with a low-density first layer, as shown in the “Low M1 Coverage” area in Fig. 1(e) [14]. These examples highlight how violating layer density rules can affect product yield. Generally, local density checks are conducted using overlapping windows that repeatedly examine whether layer densities within a specified area adhere to design guidelines (Fig. 2). By ensuring comprehensive density checks, chip yield can be improved. Due to the above phenomenon, this type of layer density DRC is prevalent in contemporary manufacturing processes in order to improve yield. By ensuring comprehensive density checks, chip yield can be improved.

Global density rule checks are typically conducted on a completed physical chip design, and a dummy layer can be used to equalize the global density [18]. Chip manufacturing practices have been moving from mature processes (e.g., 0.25 μm) to more leading-edge processes (22 nm) [19]–[20]. Verification of local layer density can be time-consuming. Adding a dummy layer in a later stage of design to address inadequate density in local areas can create parasitic capacitance in key signal paths, which may compromise circuit performance and necessitate redesign if the circuit specifications do not satisfy requirements [21].

These trade-offs make the verification of layer density in physical design the most challenging task in DRCs [22]–[25]. These design risks and costs can be mitigated with rigorous physical design verification of the reusable standard cell and I/O libraries. For example, in standard cell libraries, it is common to find that a cell typically consists of two metal layers. Layer density violations can be prevented if the density is designed in the range of 10%–80% according to the design rules. Such violations can also be addressed by adjusting the location of standard cells in the SoC design based on design requirements or needs related to solving this DRC issue. Verification of I/O libraries is challenging because the arrangement of I/O cells is often restricted by (a) the pin definition, which is determined by the client’s specification requirements and system board design guidelines, (b) the power plan, and (c) the electrostatic discharge protection. Unlike standard cells where design rule violations in layer density can be resolved simply by changing the positions of cells, I/O cells may experience electrostatic discharge or arcing due to small metal spacing when a dummy layer is added. Furthermore, window-based local density checks (Fig. 2) often reveal density violations at the interface of neighboring I/O cells.

The following discusses global and local density check results on I/O cells in various arrangements. Fig. 3 presents three cases of density checks on 11 differently arranged I/O cells. When the 11 I/O cells are arranged sequentially on the basis of their numbers (Case A), the density check reveals a density violation only on the rightmost part. When I/O cells 5, 6, and 7 are reordered (Case B), the check also reveals a density violation only on the rightmost part. When I/O cells 5, 6, 7, and 8 are reordered (Case C), a larger area of density violation is observed, spanning I/O cells 3, 4, 7, and 8. According to the results in Fig. 3, the arrangement and combination of I/O cells can affect the density check result in advanced manufacturing processes, and the density violations may not be fully visible when observing two random cells placed together.

A physical verification on different arrangements and
combinations of I/O cells at the time of developing an I/O library could facilitate early modifications to the locations of potential local density violations, thereby reducing the time required in back-end physical verification and the cost arising from design modifications during SoC development. However, studies have predominantly focused on the physical verification processes in traditional wafer foundries, with few discussions on the physical verification of I/O pads. For example, Infineon proposed an automated verification process for I/O libraries [26], which involves verifying various arrangements and combinations of different I/O pads (i.e., digital, analog, functional, and power supply pads). However, the study [26] discussions were limited to mature verification processes, such as DRCs, antenna DRCs, and electrical rule checks; it did not cover verification considerations regarding density or propose possible solutions for maximizing verification efficiency or the sample size of I/O pad arrangements and combinations. Conducting comprehensive verifications of all possible arrangements and combinations of cells within a large library is impossible. This study proposed a method capable of quickly comparing I/O cells across a library that exhibited considerably variable integrated circuit (IC) layouts and automatically generating various cell arrangements and combinations for verification. The proposed method could maximize the number of relevant cell arrangements verified while decreasing the time, resources, and costs required for the verification and minimizing errors observed in the physical verification conducted at the end of a chip development process.

2. Current challenges with physical verification of I/O libraries

Fig. 4 presents an example of common design rule violations observed during physical verification on input and output pads arranged together; the metal and open layers are indicated in the figure, and the corresponding errors are illustrated in Table 1. Type 1 errors are common DRC errors that are only seen when I/O cells are integrated with other cells. Type 2 errors occur as a result of unaligned horizontal metal lines at the edges of butted I/O cells; these errors are often associated with DRC problems concerning inadequate spacing or sharp edges. Type 3 errors arise due to an excessively high local layer density in cell arrangements. These types of errors may not be visible when verifying small I/O cells individually because the problem is only seen when an excessive portion of a local area is covered by the metal content of I/O cells placed together. Type 4 errors involve open circuits or short circuits in the horizontal metal lines spanning across various cells arranged together.

Fig. 3. Density checks on three arrangements of various I/O cell types. (Exclusive Layout of Metal 2)

Fig. 4. Common DRC violations in physical I/O pad verifications.

Table 1. I/O cell general error types during physical verification.

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Description of error content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type1</td>
<td>Inadequate spacing at the IC's opening in the top metal layer</td>
</tr>
<tr>
<td>Type2</td>
<td>Unaligned horizontal metal lines causing DRC errors in the spacing or sharp edges</td>
</tr>
<tr>
<td>Type3</td>
<td>Layer density is too high or too low after integration</td>
</tr>
<tr>
<td>Type4</td>
<td>Open circuit or short circuit in horizontal metal lines</td>
</tr>
</tbody>
</table>

Various types of I/O cells exist [27], [28]. The choice of I/O cell type depends on the signal attributes in question. A common chip configuration involves one I/O cell butted on each side of the core circuit (Fig. 5(a)). With this configuration, a layer density violation would be particularly difficult to resolve. This is because I/O cells cannot be reordered, and in this configuration, a dummy metal layer cannot be added, and removing part of the metal layer is not feasible. Current physical verification methods for cell libraries (including standard cell and I/O libraries) predominantly adopt the approach of verifying cells in pairs or individually on the basis of design rules. However, layer density problems cannot be seen with only two cells placed together, and sometimes density DRC problems may not even be detectable by more than one.
Developers of I/O libraries have mostly conducted DRC verifications by arranging I/O cells together (Fig. 5(b)). However, the arranging methods and coverage of I/O cell combinations can affect the accuracy of layer density rules when a library is used in chip design. A simple solution to this issue is to adopt the exhaustive method to verify all possible cell arrangements in pairs (Fig. 6). Nevertheless, the design rules for wafer foundries typically include the dimensions of photomask patterns and also the verification of density uniformity [29]. The uniformity of this density cannot be observed through paired checks of I/O cells because density-related rules typically require region-based checks, as illustrated in Fig. 3. In the experiments shown in Fig. 3, we have verified that the probability of layer density errors is not only due to the proximity of two I/O cells. Sometimes it is affected by the arrangement of the I/O cells in a specific area, and it may require a specific arrangement of three or four cells in some configurations for the error to occur.

Fig. 6. Exhaustive verification on cell arrangements in pairs.

Region-based (Local density) verification requires checks on arrangements of more than two cells. However, a combination comprising an excessive number of cells may be difficult to verify physically and means only some of the possible arrangements can be verified given the restrictions imposed by software and hardware (e.g., computing power of the equipment, software memory, and number of licenses) and development schedules. Take a 10–I/O cell combination for example; an exhaustive approach will involve verification of more than three million possible arrangements. As shown in Fig. 3, the density DRC problems are revealed only in certain cell arrangements, meaning that the presence of such problems cannot be determined through a random arrangement of the 10 I/O cells. Rather, it requires verification of all possible arrangements of the 10 I/O cells, which is a daunting task if conducted manually. When performed by a computer program with the assumption that each I/O cell is 50 μm wide, the total length of the IC layout for the I/O library with 10 I/O cells will be 3,628,800 × 10 × 50 μm = 1,814,400,000 μm (Fig. 7), which is currently unattainable by any chip development hardware or software and is simply not economically viable.

Fig. 7. All possible arrangements in exhaustive verification.

Technological advancements have enabled deep learning applications in chip manufacturing. Supervised reinforcement learning was used to train a classifier [30]. To facilitate the use of the classifier, the effects of relevant factors, including cell area, process window overlaps, and block-level usage on layer density, were evaluated. A process density compliance analysis was conducted on the standard cell library used in Intel’s 10-nm process [30]. In total, 85% of the failed cells were successfully predicted by the proposed classifier. However, the verification in that study was conducted in auto-place-and-route software at the chip integration stage rather than in a physical design environment at the cell development stage, meaning that the verified IC density would still require a physical design verification for density layer DRC checks at tapeout signoff. A novel power-efficient design process that involves a 16-nm fin field-effect transistor technology was developed [31]. The process used the Calibre verification tool to complete the standard layout verification. Standard physical verification software provided the most accurate verification results on layer density checks. In Intel’s 10-nm process, cells were arranged using auto-place-and-route software, whose signal dependency limitation precludes the generation of all possible arrangements and combinations for each cell [30]. Therefore, density violations can still occur when the developed chip is reused because density verification is closely related to the arrangements and combinations of cells, and verifying only some of the possible arrangements or combinations provides only partial information on density violation. In addition, the unique qualities of I/O libraries were not considered in the verification process.

Because exhaustive verification of a multicell configuration results in the layout having an excessively high order of magnitude and an excessively long length (Fig. 7), the present study proposed a pattern matching–based (PMB) method that involves the following: cells with similar layouts within an I/O library are clustered, one cell from each cluster is randomly arranged, the process is then repeated for the remaining cells from each cluster. This method could facilitate efficient and high-coverage verification of cells across all structural types in a library without consuming excessive resources.

3. Proposed pattern matching–based (PMB) physical verification

To minimize the incidence of layer density violations in the reuse of cells, as many cell arrangements as possible must be verified when developing an I/O cell library. In addition, the need to conduct layer density checks in regions requires verification of cells arranged together. For such verification, the exhaustive method will result in a total of more than three million possible arrangements with just 10 cells (10! = 3,628,800 arrangements).
3,628,800), which is computationally unfeasible, particularly for modern I/O libraries, which may comprise dozens of I/O cells (Fig. 8(a)). Accordingly, the PMB method first classifies cells before selecting cells from each class on the basis of various settings (i.e., random selection for exhaustive verification) and arranges them, revealing potential arrangements with errors (Fig. 8(b)). This method could considerably reduce the required number of arrangements for verification, prevent repeated verification on the same arrangement, and maximize the identification of problems at the library development stage.

In Step 1, as depicted in Fig. 8(b), the entire I/O cell library is loaded, and in Step 2, the layer information of each I/O cell is extracted. Subsequently, in Step 3, the Pattern Matching-Based (PMB) method is initiated to compare which I/O cell shares similar or identical structural content. Moving to Step 4, after categorizing these cells, one representative cell is selected from each category. These representative I/O cells are then subjected to either the Exhaustive Method or Random Sampling Method. After arranging these representative cells, the remaining I/O cells not selected as representatives are included in the groups for verification, completing the preparation of the entire verification dataset.

Given the large number of cells in an I/O library, the PMB method starts by classifying cells before sampling cells from each class for verification. At the chip design stage, I/O cells are typically arranged side-by-side. For these cells to be arranged as such, the basic principle is to place cells with similar structures and dimensions together or to place filler cells between neighboring cells. Because each I/O cell will have passed DRC verification before being included in a library, the verification at the stage of integrating the included cells into a library focuses on arrangements of cells rather than individual cells, the arrangement similar to Fig. 5(b) will suffice. Accordingly, the PMB method classifies I/O cells on the basis of their left and right edges (Fig. 9(a)(b)), with the sampling range of edge and pattern type adjustable to the user’s needs. The I/O cells are then classified into a limited number of categories, from which cells are selected and put into arrangements and combinations for efficient verification (Fig. 9c).

In IC design, the layout versus layout (LVL) technique is a common approach for comparing two GDS files. The LVL technique typically applies DRC syntax for layer-to-layer XOR comparison and can identify all differences, but it requires substantial preparation—the IC layout must be exported into separate GDS files, and DRC command files must be prepared. For comparing large numbers of GDS files, this process can be time-consuming. Thus, we adopted the Partial Cut Fuzzy Matching Technique, which does not involve comparisons of the contents of GDS files within each cell library and focuses solely on specific ranges of interest. By accessing information directly from the library within IC layout tools, we can more rapidly conduct comparisons.

In our program, after extracting all layer and object-related information from the IC Layout (Fig. 10(a)), we designate the key factors for comparison as hashes. The region of the cells to be compared is temporarily stored, as shown in the rightmost image in Fig. 10(a). Once we obtain all the hash data for the cells to be compared, as illustrated in Fig. 10(b), we can initiate the comparison process. The hash settings include Hash 1, Layout Purpose (LPP); Hash 2, Shape Type; Hash 3, Area; Hash 4, Center Coordinate; and so on. Converting the layout content of each unit into hash values facilitates rapid comparisons (Fig. 10(c)). Upon encountering nonmatching hash values, differences in the declared area content of the two units can be immediately determined. Obtaining immediate results in the traditional LVL technique is not possible.

Furthermore, we can classify units by condition or sampling criteria in accordance with the needs of designers (Fig. 10(d)). For example, classifying units on the basis of hash values further refines the comparison scope. We considered the design process of practical I/O cell libraries, which not only takes into account DRC rules related to process yield but also considers DRC rules related to reliability design. These rules may necessitate specific choices to achieve large-scale rapid calculations. In IC/IP design, considerations extend beyond PPA (power, performance, area) and DRC accuracy. Considerations include timeliness and cost factors and involve balancing IC design progress with machine resource availability.
Fig. 10. Instructions for LVL of Partial Cut Fuzzy Matching Technique

Fig. 11. I/O cell library verification. (a) Traditional method and (b) proposed PMB method.

Fig. 11(a) depicts a flowchart of the traditional exhaustive verification approach, which is limited by the memory and computational capacity of hardware and software. Such an approach allows for only verification of cells arranged in pairs or random I/O rings or rows. However, with a large library, even pair-based exhaustive verification can become impractical. Fig. 11(b) presents the flowchart of the proposed PMB method, in which cells are classified into a limited number of classes on the basis of their physical design similarity with other cells. Exhaustive selection or random sampling can then be performed on these classes to arrange the selected cells in I/O rings or rows for physical design DRC verification.

Fig. 12 presents the interface of a program we developed for performing PMB verification. This program offers five cell-arranging methods to generate Graphic Data Stream (GDS) files for subsequent physical verification. The arranging methods and their order of magnitude can both be adjusted using configuration files. The five arranging methods were named COST 1–5 because the implemented level of verification rigorosity affects the generated database and thereby the cost associated with the environment requirement (software/hardware). The arranging methods are detailed as follows:

(1) COST 1
Sequentially arrange the first half of the library from left to right, on the basis of the value of COST1_MIRROR (0 for left, 1 for right). E.g., Fig. 13(a).

(2) COST 2
Begin by left–right mirroring the odd-numbered pins (1, 3, 5…). Skip even-numbered pins and then mirror the even-numbered pins in reverse order. E.g., Fig. 13(b).

(3) COST 3
Randomly arrange the I/O cells in the library at irregular intervals of the input. E.g., Fig. 13(c).

(4) COST 4
Arrange the I/O cells in the library with irregular random numbers in multiples of the input. Mirror the odd-numbered pins first and then mirror the even-numbered pins in reverse order. E.g., Fig. 13(d).

(5) COST 5
Utilize the exhaustive method for library arrangement.

Import the configuration file into Block 2 on the program interface (Fig. 12). The configuration file provides condition settings for the I/O cells and pattern matching. The file in Fig. 14 defines the layer extraction information used in pattern matching: for example, it may declare the sampling area (distance in μm from the two sides of each I/O cell inward) and layer for matching. The condition settings for matching are the coordinates, area, and layer of the objects compared. Fig. 14 presents an IC layout before and after removing the unwanted part for matching.

As illustrated in Fig. 14, customizable parameters are available for users. For example, the parameter
cell_shrink_inward, which is set to 5,5, indicates that objects are retained within a range of 5 µm inward from both sides of the entire cell for comparison, excluding other objects. The Layerlist declaration specifies the critical layers that users wish to compare, and the Difference_degree parameter indicates the allowable margin of error. Users can determine the settings of these parameters on the basis of different requirements or in accordance with their practical product-specific experience, particularly concerning design rules in the manufacturing process. For example, if latch-up rules are of greater concern than layer density in the casting process, the conditions for pattern matching and classification can specify maintaining an 8-µm inward distance (assuming a latch-up rule space of 16 µm), with emphasis placed on N-Well, Oxide Diffusion, N-type Ion Implant, and P-type Ion Implant layers rather than metal layers. Nevertheless, if layer density rules are of primary concern after the placement of I/O units, one may need to designate metal layers as prerequisites for classification. If multiple design rules need to be taken into account, they can be determined accordingly. Therefore, we provide users with multiple choices, allowing the customization of verification methods depending on machine resources, scheduling constraints, process design rules, and product experiences.

Cells on each side of the ring are arranged through pattern matching and classification. Arranging cells from an I/O library with five I/O cells (Fig. 15; Classes A–E) in pairs in parallel to form a chip-level I/O pad ring (Fig. 5(a)) results in 10 (5 × 4 ÷ 2) combinations amounting to 20 cells. If the number of cells to be repeated (i.e., the number entered in Block 5; M) is set to 5, the total number of cells generated will be N × M, where N refers to the number of classes in a library. Arrangements are then produced by repeating M I/O cells, whose purpose is to simulate the repeated-I/O-cell arrangement in some SoC designs. In this example, initially, a ring consisting of 45 (20 + 5 × 5) I/O cells is produced. To ensure the four sides of the ring have equal length, the total number of cells must be a multiple of four; thus, three more filler cells are added to the ring. To this end, the program randomly selects three cells from the I/O library. An I/O cell array is also produced (For example, Inputting the I/O cell from Fig. 16 generates the array depicted on the left side of Fig. 17.) to maximize the number of I/O cell arrangements within a reasonable IC size. By contrast, the exhaustive method in Fig. 7 yields 36,288,000 (3,628,800 × 10) I/O cells, which, when arranged into a chip-level I/O pad ring, possibly exceeds the allowable size limit in IC design software. Therefore, an array is used to maximize the number of arrangements included.

As soon as the I/O pad ring’s physical layout has been
generated, the program automatically proceeds to the DRC verification of the physical design.

![Diagram](image1)

Fig. 15. Placement of I/O cells in chip level

4. Experimental results and performance evaluation

A. Design Example

The experimental I/O library consists of 11 cells (Fig. 16) classified into five classes on the basis of pattern matching (as shown on the summary report, Fig. 18). COST 4 was selected with an order of magnitude of 500 for sampling. The cells were arranged in an array (as shown on the left in Fig. 17) and an I/O pad ring (as shown on the right in Fig. 17) for physical DRC verification. Fig. 18 presents a summary of the experiment, which generated 2,500 randomly arranged cells in 4 seconds.

Fig. 15 is a schematic of how I/O cells were arranged to form the chip-level I/O pad ring in Fig. 17. Take 11 I/O cells for example:

1. All I/O cells were arranged in pairs; \( C(11,2) = 55 \), and with two cells in each pair, the total number of I/O cells was 110.
2. All I/O cells were repeated on the basis of the number entered in Block 5 of the program interface in Fig. 12. The number of cells repeated was set to five; therefore, the total number of I/O cells was \( 11 \times 5 = 55 \).

The chip-level I/O pad ring on the right of Fig. 17 consists of 165 I/O cells. To ensure equal length on the four sides of the whole chip, filler I/O cells were added to make the total number of cells a multiple of four.

The following equation was used to calculate the total number of I/O cells: \( \text{Total}_{\text{I/O Cell}} = 4 \times \text{cell (I/O Cell)} / 4 \), where the cell function rounds any decimals up to the nearest integer. In the given example, the \( \text{Total}_{\text{I/O Cell}} = 168 \), with 42 I/O cells on each side.

Fig. 16. Case studies used in experiments.

Arranging cells from the 11-cell I/O cell library exhaustively would yield \( 39,916,800 \) (11!) arrangements, and the corresponding layout would be too large to be verified on current software or hardware. Therefore, to showcase the difference between exhaustive permutation and the PMB approach, we used a 9-cell I/O cell library for comparison purposes. An exhaustive arranging approach was first performed without pattern matching, which yielded 9! arrangements, and the corresponding layout for physical verification contained 3,265,920 (9! × 9) cells with dimensions of 424,680 \( \mu m \times 150,020 \mu m \) (as shown on the right in Fig. 19). By contrast, when pattern matching was performed to categorize cells into classes before selecting cells from each class for arrangement, the resulting layout for physical verification was small, with dimensions of only 3,055 \( \mu m \times 3,380 \mu m \). This reduction of layout size saves time in physical design verification.

Fig. 17. IC layout generated by proposed program for physical design verification.

Fig. 18. Summary report generated after program execution.

Verifying this type of cell library involves considering DRC issues between cells, making it impossible to verify each cell individually. Running physical verification on the layouts individually requires the program to repeatedly read and input a GDS file for each layout, thus slowing down the verification process. This method also cannot detect DRC issues between cells or when multiple cells are placed together. Therefore, we believe that consolidating all layouts into a single GDS file can expedite the verification process. Additionally, the arrangement isn't limited to just chip-level
configurations; it can also be structured in an array format (as shown on the left in Fig. 17).

B. Performance Evaluations for Execution Efficiency
The 11-cell I/O cell library (each cell has dimensions of 62.6 \( \mu m \times 179.6 \mu m \)) was employed to compare the conventional method with the proposed PMB method in terms of resulting cell arrangements and layouts. Table 2 illustrates the program settings for the two methods performed using COST 4 (generating random arrangements in multiples of the total number of cells in the given library) and COST 5 (the exhaustive approach). For COST 4, the multiplication factor was set to 500.

Table 2. Performance evaluations.

<table>
<thead>
<tr>
<th>Options</th>
<th>COST 4 (Random)</th>
<th>COST 5 (Exhaustive method)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Cells</td>
<td>Layout size (um(^2))</td>
</tr>
<tr>
<td>Conventional</td>
<td>5500</td>
<td>10,000 × 10,680</td>
</tr>
<tr>
<td>Proposed PMB</td>
<td>2500</td>
<td>10,000 × 4,979</td>
</tr>
</tbody>
</table>

When COST 4 was implemented with a multiplication factor of 500 (i.e., randomly sampling and arranging the cells), the conventional method yielded a 10,000 × 10,680 \( \mu m^2 \) layout comprising 5,500 (11 × 500) cells in 4 seconds (Table 2). The PMB method, which classified the 11 cells into five classes, yielded a 10,000 × 4,979 \( \mu m^2 \) layout consisting of 2,500 (5 × 500) cells in 1 second.

When COST 5 was implemented (i.e., exhaustively arranging the cells), the conventional method yielded 39,916,800 arrangements, whose corresponding layout would not be able to be generated using current software and hardware, and the resulting array took up an unreasonably large area (So Table 2 shows it as N.A.). The proposed PMB method generated an array of 600 cells and a 10,000 × 1,100 \( \mu m^2 \) layout within 1 second. These comparisons reveal the impracticality of the conventional method in layout generation and validate the proposed PMB method’s feasibility and efficiency given its ability to accurately and rapidly produce relevant arrangements for verification.

In addition to cell arrays, the program generates the corresponding I/O pad ring around a chip. In the given example, the pad ring had 42 cells on each side and 172 cells in total, including the corner cells.

Table 3 presents the time the programs took to complete DRCs on the basis of the conditions in Table 2; 415 design rules were applied. When the conventional method was used with COST 4, the program took 4 seconds to generate the layout for verification and 886 seconds to complete the DRCs. When the proposed PMB method was implemented with COST 4, the program took only 1 second to generate the layout and 442 seconds to complete the DRCs, which was half the time required by the conventional method.

When the conventional method was implemented with COST 5, the resulting arrangements were too numerous for the program to generate a layout or proceed to DRCs. When the proposed PMB method was used with COST 5, the program took less than 1 second to complete the arrangements and also generate a layout and took only 63 seconds to finish the DRCs. Accordingly, the proposed method can improve the accuracy of physical design verification when developing a cell library and enable users to efficiently identify potential DRC violations. The proposed method overcomes the lack of coverage of cell arrangements in manual and random arranging methods and avoids generating a computationally infeasible number of arrangements. The overall time difference in using our proposed PMB method might be more significant when dealing with a larger number of I/O cells in a real SoC design.

Table 3. Run time for DRC (415 rules).

<table>
<thead>
<tr>
<th>Algorithm (Option)</th>
<th>Conventional (COST 4)</th>
<th>Proposed PMB (COST 4)</th>
<th>Conventional (COST 5)</th>
<th>Proposed PMB (COST 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td># Cells (Array)</td>
<td>5,500</td>
<td>2,500</td>
<td>39,916,800 × 11</td>
<td>600</td>
</tr>
<tr>
<td># Cells (Ring)</td>
<td>172</td>
<td>172</td>
<td>172</td>
<td>172</td>
</tr>
<tr>
<td>Layout Generation time (Sec.)</td>
<td>4</td>
<td>1</td>
<td>∞</td>
<td>&lt; 1</td>
</tr>
<tr>
<td>DRC Run time (Sec.)(Reduction %)</td>
<td>886</td>
<td>442</td>
<td>∞</td>
<td>63</td>
</tr>
</tbody>
</table>

We validated three combinations from Table 3 that can be implemented together, as displayed in Fig. 20. Conventional (COST 5) employs an exhaustive method without any reduction, resulting in outcomes that surpass the limits of EDA Tools. Therefore, these results are not included in the analyzed data. At Position B (Fig. 20), we set the same cell to repeat five times in chip mode, resulting in the detection of density errors for this cell in all three chip modes. Concurrently, at Position A, we observed the possibility of duplicate placement under random arrangements with COST 4. After analyzing these regions, we removed duplicate cells from Position A and compared the results of the three combinations (Table 4).

We used a program to convert each cell covered by DRC error regions into text, then compared and recorded the remaining combinations after removing duplicates in Table 4. For example, the data 2 for Case 4 VS. Case 2 and the data 27 for Case 2 VS. Case 1 represent the additional combinations obtained after removing duplicates. If there are more additional combinations after removal, it suggests a higher validation accuracy for the method. According to the data in the table, “Case 2 versus Case 1” indicates that in Case 2, using PMB for classification and arranging cells with significant differences could facilitate the identification of issues compared with random sampling without PMB. Additionally, data from “Case 4 versus Case 2” suggest that arranging all cells classified after PMB by using COST 5 could provide a higher chance of identifying problematic points compared with a random sampling arrangement with COST 4. The accuracy of DRC verification is closely related to the complexity of the process and the command files.
provided by the foundry. However, on the basis of our experimental results, we expect that accuracy rates would be higher when one selects I/O units with significant differences for PMB classification and employs exhaustive rather than random arrangements. Our method is superior to traditional manual arrangement or sampling methods, whose accuracy is limited by EDA size, and can enhance error detection coverage.

![Fig. 20. An experiment comparing the correctness of three methods](image)

**Table 4. Comparison of cell arrangement combinations generated by conventional and PMB methods.**

<table>
<thead>
<tr>
<th>Accuracy comparison</th>
<th>Case1</th>
<th>Case2</th>
<th>Case4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm (Option)</td>
<td>Conventional</td>
<td>Proposed PMB</td>
<td>Proposed PMB</td>
</tr>
<tr>
<td>(Cost)</td>
<td>(Cost 4)</td>
<td>(Cost 4)</td>
<td>(Cost 5)</td>
</tr>
<tr>
<td>The remaining number of combinations after deduplication</td>
<td>Case4 VS. Case2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Case2 VS. Case1</td>
<td>27</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**C. Verification Results of Design Rule Check**

Once a layout is generated, the program automatically proceeds to DRC verification. Fig. 21 presents an example of the program’s verification result. In Position A, a layer density rule violation is observed when Cell DIN_18 is arranged with two Cell VSS_PAD2A. Fig. 22 shows the actual IC layout and the DRC violation at Position A in Fig. 21 (an error related to low density). In Position B, no density violation is noted when Cells VSS_PAD2A and VSS_PAD1 are placed together alone; however, a density violation is noted when Cell VSS_PAD2A is placed between Cells AIN_18 and VDD_PAD1. These results demonstrate the proposed method’s ability to reduce potential DRC errors in libraries and accelerate the process of reusing libraries for SoC development.

![Fig. 21. DRC verification results.](image)

Conventional approaches could not have revealed the two aforementioned density errors at the library development stage, and these errors would not be able to be resolved by simply rearranging the cells or adding a filler layer. Therefore, the value of the proposed method lies in its ability to maximize the identification of such errors and modify designs at the library development stage, thereby reducing delays in the subsequent use of the library due to density-related DRC errors that occur in certain combinations of I/O cells. These errors are extremely difficult to rectify and can easily lead to development schedule delays.

In our approach, we assume the internal structure of each I/O cell is correct, focusing instead on potential issues arising from the boundaries between I/O cells or the density resulting from their arrangement. Therefore, our goal is to swiftly generate diverse arrangements and combinations. When exhaustive permutations are not feasible, we prioritize arranging significantly different I/O cells after classification using Pattern Matching (PMB), thus enhancing accuracy. The method we propose effectively alleviates DRC issues at the boundaries between units, enabling designers to concentrate on more complex challenges.

In our method, we assume that the internal structure of each I/O cell is correct, and we focus on potential issues arising from the boundaries of I/O cells or the density resulting from their arrangement. Our goal is to rapidly generate diverse arrangements and combinations. When exhaustive permutations are not feasible, we prioritize arranging significantly different I/O cells after classification using pattern matching (the PMB method), thus enhancing accuracy. The proposed method effectively alleviates DRC issues at the boundaries between units, thus enabling designers to concentrate on more complex challenges.
various library types, including standard cell libraries. Of these libraries, and is applicable to the development of I/O cell libraries, improves the accuracy method can be applied to any IC physical verification area, and reducing layout generation time by four times, achieved with only 2,500 I/O units, requiring half the layout experiments, a combination of 5,500 I/O cells initially seconds, executing 415 DRC rules. In another set of DRC verification time is shortened from indefinite to 63 μlevel verification layouts with dimensions exceeding 10,000 through the proposed PMB method, wafer-level and array-layouts beyond the dimensions of traditional IC designs. impossible for integrated circuit layout tools to produce experiments, the initial exhaustive arrangement made it permutations required for verification. This method can This study proposes a new method of classifying cells based on their structure, significantly reducing the number of permutations required for verification. This method can quickly produce the layout required for verification. In the experiments, the initial exhaustive arrangement made it impossible for integrated circuit layout tools to produce layouts beyond the dimensions of traditional IC designs. Through the proposed PMB method, wafer-level and array-level verification layouts with dimensions exceeding 10,000 μm can be automatically generated within 1 second, and the DRC verification time is shortened from indefinite to 63 seconds, executing 415 DRC rules. In another set of experiments, a combination of 5,500 I/O cells initially required a layout area of 10,000 x 10,680 μm². Using the proposed algorithm, the same verification effect can be achieved with only 2,500 I/O units, requiring half the layout area, and reducing layout generation time by four times, while also halving the execution time of DRC. The proposed method can be applied to any IC physical verification software available on the market. This approach accelerates the development of I/O cell libraries, improves the accuracy of these libraries, and is applicable to the development of various library types, including standard cell libraries.

References


Chen-Liang Wu received his bachelor's degree from Southern Taiwan University of Technology in 2001 and his master's degree from Southern Taiwan University of Technology in 2010. He currently serves as the Director of Design Engineering at M31 Technology Corporation, overseeing CAD/EDA automation, ESD, and device reliability flow. At the same time, he is pursuing a Ph.D. in the department of Engineering and Systems Science at National Tsing Hua University.

Chih-Wen Lu received the B.S. degree in electronic engineering from National Taiwan Institute of Technology, Taipei, Taiwan, in 1991, and the M.S. degree in electro-optics and the Ph.D. degree in electronic engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1994 and 1999, respectively. From 1999 to 2001, he was an Assistant Professor with the Department of Electrical Engineering, Dayeh University, Dacun, Taiwan. He joined National Chi Nan University (NCNU), Puli, Taiwan, in 2001, and was a Professor with the Department of Electrical Engineering, in 2010. He joined National Tsing Hua University, Hsinchu, in 2010, and was a Professor with the Department of Engineering and System Science. He joined National Yang Ming Chiao Tung University, Tainan, Taiwan, in 2020, and is currently a Professor in the Institute of Photonic System. His research interests include analog/mixed-mode IC design. He served as the technical program chair for the 5th International Symposium on Next-Generation Electronics (ISNE 2016) in 2016 and general chair for the 13th IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC 2017) in 2017. He is currently the Dean of the College of Photonics, National Yang Ming Chiao Tung University.