

# PICELF: An Automatic Electronic Layer Layout Generation Framework for Photonic Integrated Circuits

Xiaohan Jiang<sup>1</sup>, Yinyi Liu<sup>1</sup>, Peiyu Chen<sup>2</sup>, Wei Zhang<sup>1</sup>, Jiang Xu<sup>2,\*</sup>

<sup>1</sup>*Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology*

<sup>2</sup>*Microelectronics Thrust, The Hong Kong University of Science and Technology (Guangzhou)*

**Abstract**—In recent years, the advent of photonic integrated circuits (PICs) has demonstrated great prospects and applications to address critical issues such as limited bandwidth, high latency, and high power consumption in data-intensive systems. However, the field of physical design automation for PICs remains in its infancy, with a notable gap in electronic layer layout design tools. Current research on PIC physical design automation primarily focuses on optical layer layouts, often overlooking the equally crucial electronic layer layouts. Although well-established for conventional integrated circuits (ICs), existing EDA tools are inadequately adapted for PICs due to their unique characteristics and constraints. As PICs grow in integration density and size, traditional manual-based design methods become increasingly inefficient and sub-optimal, potentially compromising overall PIC performance.

To address this challenge, we propose PICELF, the first framework in the literature for automatic PIC electronic layer layout generation. Our framework comprises a nonlinear binary programming (NBP)-based netlist generator with scalability optimization and a two-stage router featuring initial parallel routing followed by post-routing optimization. We validate our framework's effectiveness and efficiency using a real PIC chip benchmark established by us. Experimental results demonstrate that our method can efficiently generate high-quality PIC electronic layer layouts and satisfy all design rules, within reasonable CPU times, while related existing methods are not applicable.

## I. INTRODUCTION

As Moore's Law slows down, conventional integrated circuits (ICs) face mounting challenges in meeting the escalating demands of data-intensive applications. This technological plateau necessitates a paradigm shift towards novel solutions to sustain performance growth. Photonic integrated circuits (PICs) have emerged as a promising candidate for next-generation IC technology, offering significant advantages in latency reduction, bandwidth expansion, and energy efficiency [1]. Advancements in manufacturing processes have catalyzed a substantial increase in both the integration density and physical scale of PICs. Some large-scale PIC chips have already been fabricated successfully [2]–[4].

PIC physical design is crucial to the entire design flow and significantly impacts overall performance. As shown in Fig. 1, unlike ICs, PIC layouts primarily consist of optical layers and electronic layers. The optical layer, typically designed first due to its higher priority and complexity, incorporates optical devices such as microring resonators (MRRs). These

components are strategically placed and interconnected via optical waveguides to form functional circuits. The electronic layer, designed after the completion of the optical layer layout, contains ports of phase shifters (PSs) that control active optical devices, pads at the chip's periphery, and metal wires connecting them. This layer's physical design requires meticulous attention to prevent circuit control distortion and performance loss. It presents unique challenges, notably the absence of a predefined netlist and the constraint of a single routing layer due to process and cost constraints.

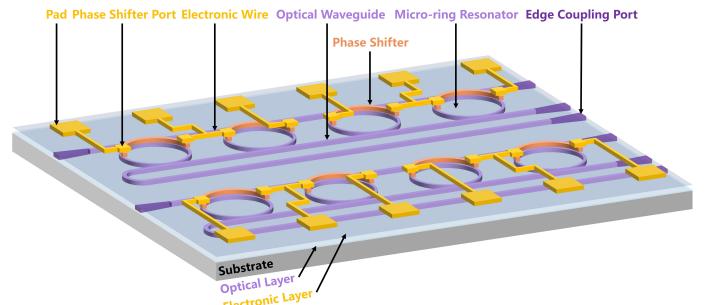


Fig. 1: An illustrative example of a PIC layout, incorporating both the optical and electronic layer designs.

Currently, physical design automation research for PICs predominantly focuses on the optical layer, neglecting the equally important electronic layer. [5], [6] tackles the floorplan problem of optical network-on-chips (ONoCs) based on mixed-integer linear programming (MILP). [7]–[10] have developed automated placement and routing tools for ONoCs but their scope remains confined to optical layer layouts. Similarly, [11]–[17] propose routing algorithms specifically tailored for PIC optical layer layouts. While [18] incorporates electronic layer considerations in the optical routing algorithms, it focuses on constructing optical interconnect networks for photonic-electronic integrated circuits (PEICs) and does not address the automated physical design of PIC electronic layer layouts.

Although the PIC electronic layer physical design problem shares similarities with existing EDA problems: packaging routing [19]–[21] and PCB routing [22]–[24], it presents unique challenges that preclude direct application of current solutions. Packaging routing, while also lacking a pre-defined netlist and constrained to single-layer routing, deals with uniformly arranged ports and pads with strictly one-to-one connections. This uniform arrangement and simpler connection scheme

\*Corresponding author: jiang.xu@hkust-gz.edu.cn

make packaging routing less complex than the PIC electronic layer layout design problem. The heightened complexity of our problem stems from two key factors: multi-connection relationships between PS ports and pads, and non-uniform spatial distribution of PS ports due to diverse optical device placement. Furthermore, PCB routing, despite sharing similar routing scales and scenarios with PIC physical layer design, benefits from a pre-defined netlist and multiple routing layers which significantly simplify the routing task. Therefore, existing EDA tools are inadequate for direct application to automatically generate PIC electronic layer layouts.

As PICs grow in size and integration density, the current manual design method becomes inefficient and often yields sub-optimal layouts, potentially leading to performance degradation. Consequently, there is a critical need for the development of an automated tool capable of efficiently and effectively generating high-quality electronic layer layouts for PICs.

Our major contributions can be summarized as follows:

- We present **PICELF**, an automatic electronic layer layout generation framework for PICs. To the best of our knowledge, this is the first automatic design tool that targets PIC electronic layer layouts.
- We introduce a nonlinear binary programming (NBP)-based netlist generator to generate electronic layer netlists in PICs. Due to the poor scalability of NBP itself, we propose two optimization strategies: region partition and parallelism, as well as locality-based solving space reduction, which facilitate efficient netlist generation for PICs across various scales.
- We propose a two-stage router that utilizes the generated netlist for routing. The first stage employs A\*-based parallel routing to rapidly obtain an initial routing result, ensuring fast convergence and high routability. Then post-routing optimization is performed to eliminate design rule violations and optimize quality metrics like wirelength and bend count. It outperforms conventional single-stage routing in quality and speed for PIC physical layout design while ensuring routability and legality.
- We establish a PIC benchmark derived from real academic and industry PICs and perform experiments. Experimental results demonstrate that our framework can generate legal and high-quality electronic layer layouts for all test cases within reasonable CPU times, outperforming existing related methods.

The rest of this paper is organized as follows. Section 2 introduces the background, design rules, and formulation of the PIC electronic layer layout generation problem. Section 3 details our proposed framework. Section 4 reports the experimental results. Section 5 gives our conclusion.

## II. PRELIMINARIES

### A. Background

We first introduce distinctive features of the PIC electronic layer layout design. As illustrated in Fig. 1, the electronic layer layout comprises pads and PS ports. Pads can be designated as either signal pads for control signals or ground pads. This

functional assignment becomes fixed once the layout design is finalized. Pads are evenly spaced on the top and bottom sides of PIC chips, leaving the left and right sides for optical waveguide ports, whose positions are pre-determined. PSs are two-port devices requiring connections to both signal and ground pads for proper optical device control. They can be categorized as resistive (bidirectional) and PIN (with designated signal and ground ports) types based on control methods. PS port positions are dictated by the pre-designed optical layer layout, with the number of PSs matching the number of active optical devices. Their spatial distribution is non-uniform and irregular due to the diversity of optical layer layout designs.

PIC electronic layer layout generation aims to produce a single-layer, design rule-compliant layout that connects each PS's ports to appropriate pads via metal wires. The layout quality, assessed by metrics like total wirelength and bend count, must also be considered to prevent negative impacts on PIC performance. The generation process is constrained by two key factors: the absence of predefined netlists and the limitation to single-layer routing due to manufacturing and cost constraints.

### B. Design Rules

We clarify three major design rules for the PIC electronic layer layout as follows:

- *Port-Pad Connection Constraint*: Each PS needs to have one of its ports connected to a signal pad and another to a ground pad. Each signal pad must connect exclusively to one PS's signal port as each active optical device needs to be controlled individually. Each ground pad is allowed to connect to multiple PS's ground ports due to the limited overall pad count imposed by PIC chip size.
- *No-Crossing Constraint*: Wire crossing is not allowed due to the single-layer constraint in PIC electronic layer layouts.
- *Spacing Constraint*: Routed metal wires in the layout must adhere to specified minimum spacing requirements from PS ports, pads, and other wires to mitigate interference and meet fabrication requirements.

### C. Problem Formulation

We formally define the electronic layer layout generation problem for PICs as follows:

- Given a PIC layout (with completed optical layer layout design and pad locations) and design rules, efficiently generate the corresponding high-quality electronic layer layout that connects each PS port to an appropriate pad via metal wires while adhering to all design rules.

## III. METHODS

### A. Framework Overview

Fig. 2 presents an overview of our proposed PICELF. The framework takes as input the design rules and a PIC layout in Graphic Design System II (GDSII) format that includes a completed optical layer layout design and specified pad locations. PICELF first utilizes an NBP-based netlist generator

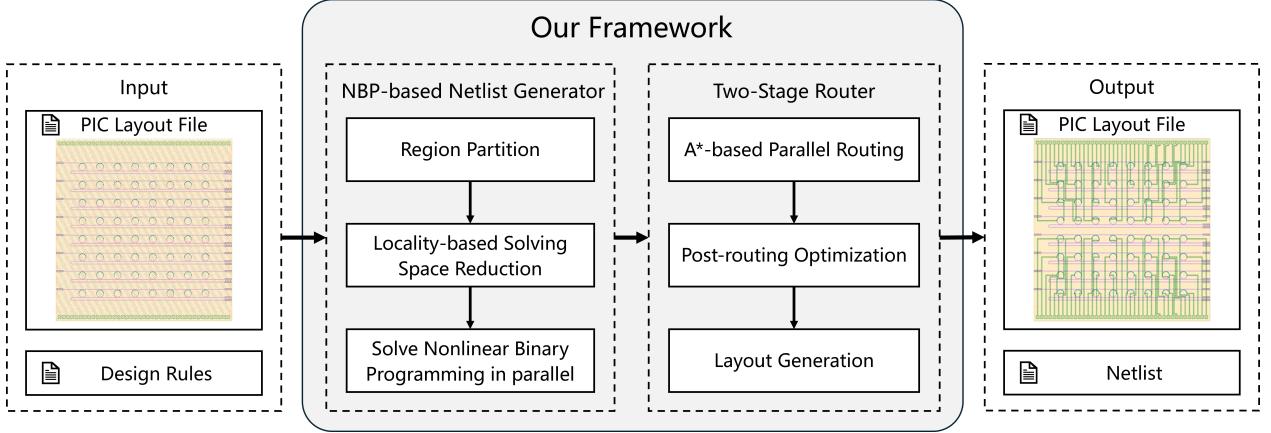


Fig. 2: Overview of PICELF

to create the electronic layer netlist. It then employs the two-stage router to route based on this netlist and generate the layout. The final output is the netlist and input PIC layout with the newly generated electronic layer layout in GDSII format.

### B. NBP-based Netlist Generator

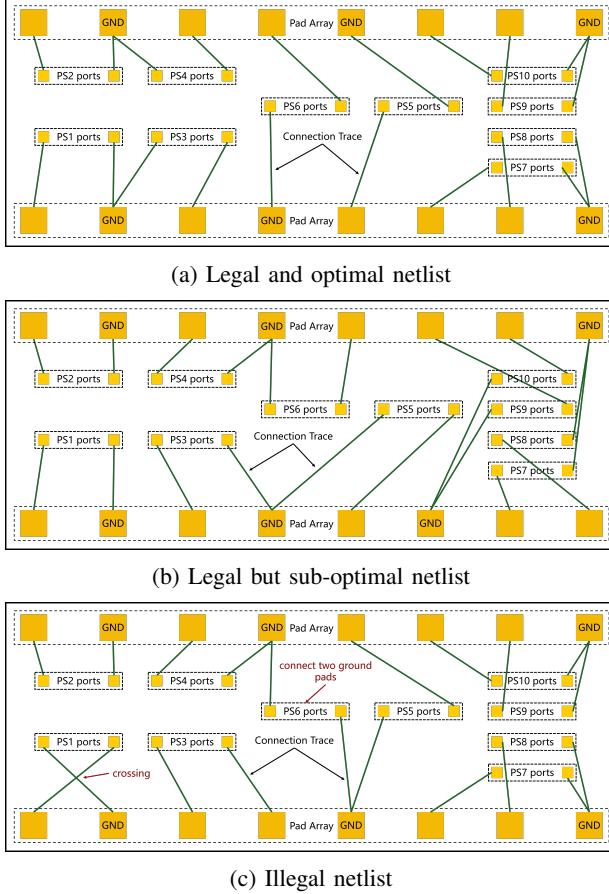


Fig. 3: An illustration of the legality and quality of generated netlist.

Netlist generation is crucial in PIC electronic layer layout design. As illustrated in Fig. 3, the generated netlist must first be legal to prevent design rule violations and ensure the accuracy of the subsequent layout generation. Moreover, the netlist generation process must consider subsequent single-layer

routing to guarantee routability and routing quality. To achieve these objectives, we use an NBP-based method to ensure that legal and high-quality netlists are generated.

TABLE I: Notations of NBP Formulation

$N_p$	The pad number.
$N_s$	The PS number.
$S_p$	A set of pads.
$S_s$	A set of PS ports.
$s_i$	A PS port, $i \in \{1, 2, 3, \dots, 2N_s\}$ ; $s_{2k-1}$ and $s_{2k}$ belong to the same PS, $k \in \{1, 2, 3, \dots, N_s\}$ .
$p_j$	A pad, $j \in \{1, 2, 3, \dots, N_p\}$ .
$\alpha_{i,j}$	0-1 integer variable that denotes whether $s_i$ connects to $p_j$ . $\alpha_{i,j} = 1$ if connected; otherwise, $\alpha_{i,j} = 0$ .
$\beta_j$	0-1 integer variable that denotes whether $p_j$ is a signal pad. $\beta_j = 1$ if it is signal pad; otherwise, $\beta_j = 0$ .
$d_{i,j}$	The Manhattan distance between $s_i$ and $p_j$ .
$C(\alpha_{i,j}, \alpha_{m,n})$	The function that denotes whether the connection between $s_i$ and $p_j$ crosses the connection between $s_m$ and $p_n$ . $C(\alpha_{i,j}, \alpha_{m,n}) = 1$ if crosses and $i \neq m, j \neq n$ ; otherwise, $C(\alpha_{i,j}, \alpha_{m,n}) = 0$ .
$\gamma_i$	User input variable that denotes whether $s_i$ is a signal port. $\gamma_i = 1$ if it is signal pad; otherwise, $\gamma_i = 0$ .

1) *NBP Formulation*: We formulate the netlist generation as a nonlinear binary programming (NBP) problem due to the unique characteristics of PIC electronic layer layout design and the notations used are detailed in TABLE I. We use two binary decision variables:  $\alpha_{i,j}$  for port-pad connections and  $\beta_j$  for pad types. The objective is to minimize total wirelength, as it is the most critical quality metric. The formulation is as follows:

$$\begin{aligned} \min & \sum_{s_i \in S_s} \sum_{p_j \in S_p} \alpha_{i,j} d_{i,j} \\ \text{s.t.} & \sum_{p_j \in S_p} \alpha_{i,j} = 1, \forall s_i \in S_s \end{aligned} \quad (1)$$

$$\sum_{p_j \in S_p} \beta_j = N_s \quad (2)$$

$$\sum_{s_i \in S_s} \alpha_{i,j} \beta_j \leq 1, \forall p_j \in S_p \quad (3)$$

$$\sum_{s_i \in S_s} \sum_{p_j \in S_p} \alpha_{i,j} \beta_j = N_s \quad (4)$$

$$\alpha_{2k-1,j} + \alpha_{2k,j} \leq 1, \forall s_{2k-1}, s_{2k} \in S_s, \forall p_j \in S_p \quad (5)$$

$$\sum_{s_i \in S_s} \alpha_{i,j} \geq 1, \forall p_j \in S_p \quad (6)$$

$$\sum_{p_j \in S_p} (\alpha_{2k-1,j} + \alpha_{2k,j}) \beta_j = 1, \forall s_{2k-1}, s_{2k} \in S_s \quad (7)$$

$$C(\alpha_{i,j}, \alpha_{m,n})(\alpha_{i,j} + \alpha_{m,n}) \leq 1, \forall s_i, s_m \in S_s, \forall p_j, p_n \in S_p \quad (8)$$

$$\alpha_{i,j}(\gamma_i - \beta_j) = 0, \forall s_i \in S_s, \forall p_j \in S_p, \exists \gamma_i \quad (9)$$

The objective function is to minimize the total wirelength while ensuring netlist legality and routability. Equation (1) ensures each PS port must connect to one pad. Equation (2) fixes the total number of signal pads. Equation (3) and (4) jointly ensure that each signal pad connects exactly one PS port. Equation (5) ensures each PS's two ports must connect to two different pads. Equation (6) and (7) collectively allow ground pads to connect to multiple PS ports while ensuring each PS connects to both a signal pad and a ground pad. Equation (8) prevents crossing connections. Equation (9) is specifically constrained for PIN-type PSs due to their pre-designated signal and ground ports. It ensures netlist legality when such PSs are present in the PIC.

Legal and high-quality netlist of the PIC electronic layer can be generated by solving the above NBP problem. However, the inherent poor scalability of NBP poses a significant challenge, as its solving time increases exponentially with problem size. This scalability issue is particularly problematic given the rapid increase in PIC sizes and integration densities. Consequently, it is necessary to optimize its scalability to ensure high efficiency as PIC technology advances.

*2) Scalability Optimization:* Two strategies are used to optimize the scalability of our proposed NBP-based netlist generation method.

*Region Partition and Parallelism:* The unique distribution and connectivity of pads and PS ports in the PIC electronic layer layout allows the NBP problem to be decomposed into multiple sub-problems through region partition. These sub-problems can be solved in parallel, enhancing scalability and efficiency. Therefore, we propose a greedy-based partition algorithm for the PIC physical layer layout that partitions the whole layout into several sub-layouts.

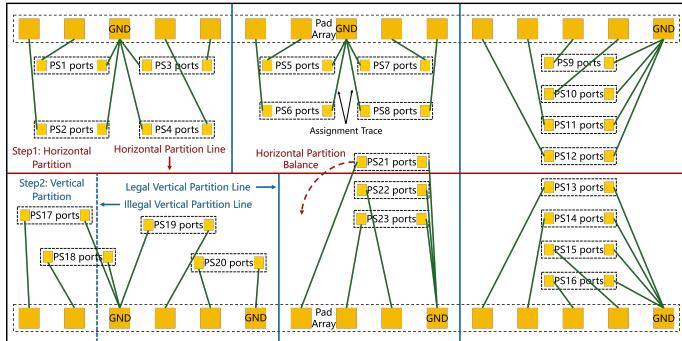


Fig. 4: An illustration of region partition process in the netlist generator.

As shown in Fig. 4, the region partition encompasses horizontal and vertical divisions. The horizontal partition is performed first, leveraging the pad distribution's symmetry by selecting the region's horizontal center line as the partition line. To maintain balance, if the PS count difference between upper and lower sub-regions exceeds a threshold, PSs from the sub-region with a higher PS number near the partition line are grouped into another one. Then, each sub-region is partitioned vertically, employing a greedy algorithm. The algorithm greedily selects vertical partition lines from medians between neighboring pads. The chosen line must not cross any PSs and the ratio of PS number to pad number on its both sides must be within a specified range. Additionally, users can provide custom partition lines, leveraging their design data of the PIC's optical device distribution to easily achieve reasonable region partitions.

After region partition, the NBP problem is decomposed into several sub-problems solvable in parallel, enhancing scalability and efficiency. If there are sub-regions without feasible solutions or with conflicts, they are merged with their neighboring sub-regions and NBP is re-executed on the merged regions. The overall netlist is obtained by integrating the netlist of each sub-region.

*Locality-based Solving Space Reduction:* Sub-regions after region partition may remain large when dealing with PICs with a high number of densely packed PSs, potentially leading to time-consuming NBP solving processes. Therefore, we exploit space locality to constrain the NBP solving space within each sub-region, further enhancing scalability.

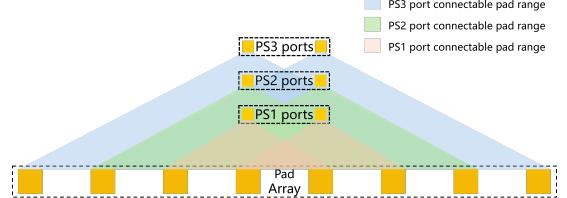


Fig. 5: An illustration of space locality.

Fig. 5 illustrates space locality: PS ports closer to the pad array have a narrower selection range of connectable pads to reduce potential crossings and wirelength. We utilize it to reduce the NBP solving space of each sub-region. Given the number of connectable pads for the farthest and nearest ports, we can limit the number and range of connectable pads for each PS port according to the linear relationship. This approach reduces the number of variables and constraints in the NBP problem and further improves the scalability and efficiency of our netlist generator.

Region partition and parallelism as well as locality-based solving space reduction are applied to the initially formulated NBP to form our netlist generator which is shown in Fig. 2.

### C. Two-stage Router

Routing is executed after PIC electronic layer netlist generation. To ensure the correct and high-quality PIC electronic layer layout, we need the routing result with excellent quality metrics while ensuring routability and legality. Achieving all

these requirements in one routing stage is challenging due to the task's complexity. This approach may fail to satisfy all routing requirements and is time-consuming because of numerous iterations. Therefore, we propose a two-stage router to achieve high-quality routing results efficiently, while ensuring routability and no design rule violations.

1) *Initial Parallel Routing*: We first perform the routing on each sub-region obtained from the netlist generator in parallel. Within each sub-region, multi-terminal nets are decomposed into two-terminal nets by generating the rectilinear minimum spanning tree (RMST) which is faster and more flexible than rectilinear Steiner minimum tree (RSMT) generation. We employ a grid-based A\* routing algorithm [25] to sequentially route each two-terminal net with rip-up and reroute. Since routability is factored into the netlist generation, few iterations are typically needed for successful routing. The cost function  $f(x)$  of the A\* algorithm is defined as:

$$f(x) = h(x) + g(x) \quad (10)$$

where  $h(x)$  is the Manhattan distance from the current location  $x$  to the target, and  $g(x)$  is the actual distance from the source to  $x$ . We use this original format to ensure high pathfinding speed. Routed wires are set as obstacles according to their wirewidth and spacing constraints, ensuring no spacing errors. By merging the routing results of each sub-region, we can rapidly obtain the overall routing result with high routability and few design rule violations mainly due to parallelism and the high-quality generated netlist.

2) *Post-routing Optimization*: In this stage, we conduct a design rule check (DRC) and eliminate all violations through rip-up and reroute, ensuring overall legality. As DRC errors were largely avoided in previous steps, only a few iterations are typically needed to achieve DRC clean. After routability and legality are secured, we optimize the routing result's quality metrics. In this paper, we focus on wirelength and bend count, but our method can support the user to set custom metrics. We evaluate each routed wire using the following cost function:

$$c(x) = a \times C_l + b \times C_b \quad (11)$$

where  $a$  and  $b$  are user-defined parameters,  $C_l$  represents wirelength cost and  $C_b$  denotes bend count. Routed wires are sorted by cost in descending order to form a queue and take out one in turn for optimization. For the wire to be optimized, rip it up and reroute it with the improved A\* algorithm where  $g(x)$  in (10) is replaced by  $c(x)$ . If the new route yields a lower cost, we update the wire's cost and the routing result. When the queue is empty, repeat the above steps until no wire's cost decreases or the iteration limit is reached. With this incremental approach, we can achieve high-quality routing results while ensuring routability and legality.

After obtaining the routing result, we can generate the PIC electronic layer layout, and the overall two-stage router workflow is shown in Fig. 2.

## IV. EXPERIMENTAL RESULTS

### A. Experiment Setup

We implement the overall framework in Python and use the Gurobi optimizer [26] to solve the NBP. We leverage KLayout

[27] to conduct DRC and generate layouts. All the experiments are conducted on a Linux machine with 2.00GHz Intel(R) Xeon(R) Gold 6338 CPUs and 256GB memory.

We established a PIC benchmark to evaluate our framework due to the lack of public ones. Whole PIC chips in the benchmark are composed of real academic and industry PIC modules like optical neural network (ONN) [2]. The detailed information of the PIC benchmark is listed in TABLE II. "L×W" denotes the length and width of each PIC chip. "#PSs" and "#Pads" denote the numbers of PSs and pads respectively.

TABLE II: Benchmark PIC Chips

PIC Chips	L×W (mm)	#PSs	#Pads
Chip01	5×5	16	20
Chip02	5×5	16	26
Chip03	5×5	64	82
Chip04	5×5	72	100
Chip05	5×5	64	82
Chip06	5×5	72	100
Chip07	5×5	71	100
Chip08	10×5	108	152
Chip09	10×5	96	132
Chip10	10×5	100	128
Chip11	10×5	128	164
Chip12	10×5	132	164

### B. Netlist Generator Results

We first perform experiments to validate our NBP-based netlist generator. We compare our approach with the classical ILP-based netlist generate method [19] in packaging routing, focusing on four key metrics: success, legality, half-perimeter wirelength (HPWL), and time. TABLE III presents the experimental results. Compared to [19], our netlist generator guarantees the successful generation of legal PIC electronic layer netlists within reasonable CPU times. [19] fails to generate legal netlists for our scenario, which is more complex than package routing. As the PIC chip size grows, this method also frequently fails due to unacceptable computation times. These limitations preclude the direct application of the netlist generation method in package routing to PIC electronic layer netlist generation.

TABLE III: Comparison of the netlist generation result between [19] and Ours

PIC Chips	Success		Legality		HPWL (μm)		Time (s)	
	[19]	Ours	[19]	Ours	[19]	Ours	[19]	Ours
Chip01	T	T	F	T	-	29300	10	1
Chip02	T	T	F	T	-	42850	17	1
Chip03	T	T	F	T	-	125940	3147	2
Chip04	T	T	F	T	-	122400	7151	2
Chip05	T	T	F	T	-	119805	3108	2
Chip06	T	T	F	T	-	115700	7063	3
Chip07	T	T	F	T	-	136600	7127	2
Chip08	T	T	F	T	-	192247	35067	3
Chip09	T	T	F	T	-	188378	20395	3
Chip10	T	T	F	T	-	161040	22596	12
Chip11	F	T	-	T	-	247630	>60000	3
Chip12	F	T	-	T	-	188985	>60000	4

TABLE IV: Effects of the scalability optimization

PIC Chips	#Variables				#Constraints				HPWL ( $\mu\text{m}$ )			Time (s)		
	P	P+Pa	P+Pa+Lo	RR (%)	P	P+Pa	P+Pa+Lo	RR (%)	P	P+Pa	P+Pa+Lo	P	P+Pa	P+Pa+Lo
Chip01	660	180	114	82.73	49418	1104	224	99.55	29300	29300	29300	14	1	1
Chip02	858	226	140	83.68	83046	2930	386	99.54	43200	42850	42850	21	1	1
Chip03	10578	2034	1052	90.05	14366740	182200	13878	99.90	126265	125895	125940	48813	19	2
Chip04	14500	3700	722	95.02	27971826	648058	2038	99.99	122400	123350	122400	7323	31	2
Chip05	10578	2438	956	90.96	14496885	383674	9550	99.93	118740	119805	119805	12744	38	2
Chip06	14500	2272	1212	91.64	27341486	215314	18464	99.93	115780	116230	115700	15003	20	3
Chip07	14300	1538	960	93.29	27049013	37091	7240	99.97	133800	134350	136600	7663	3	2
Chip08	32984	4352	1210	96.33	146496980	484266	4462	99.99	192993	189415	192247	59586	26	3
Chip09	25476	3082	1151	95.48	85787380	705224	7080	99.99	182393	188617	188378	46052	665	3
Chip10	25728	3934	2086	91.89	87719455	659322	58350	99.93	-	160675	161040	>60000	352	12
Chip11	42148	3456	1858	95.59	237919950	310653	19897	99.99	-	247475	247630	>60000	26	3
Chip12	43460	3074	2376	94.53	254750289	128929	41776	99.98	-	189635	188985	>60000	10	4

\* P: original NBP problem without scalability optimization, Pa: region partition and parallelism, Lo: locality-based solving space reduction, RR: reduction rate before and after scalability optimization. '-' fail due to timeout.

TABLE IV illustrates the effects of our scalability optimization. Since all generated netlists are legal, we do not list success and legality in the table. Instead, we focus on the NBP variable number, the NBP constraint number, HPWL, and time under different optimization settings. As shown in the table, our strategies can significantly reduce the NBP problem size and accelerate netlist generation while ensuring success, legality, and quality. Without scalability optimization, generation time escalates exponentially with increasing PIC chip size, rendering the process infeasible for large designs. Consequently, our NBP-based netlist generator exhibits robust scalability and enables legal electronic layer netlist generation for various PIC chip sizes within reasonable CPU times.

### C. Router Results

We evaluate our proposed two-stage router performance. Given the absence of existing PIC electronic layer routing algorithms, we implemented the related work presented in [24], a one-stage A\*-based PCB router, for the comparative study, because the PCB routing problem is very similar to our routing problem in terms of problem sizes and requirements. We compare them using netlists generated by our proposed netlist generator.

TABLE V shows the routing experiment results. Our proposed router can yield routing results with 100% routability and legality much faster than [24] and achieve comparable quality metrics in wirelength and bend count.

### V. CONCLUSION

In this paper, we introduce PICELF, the first framework in the literature to automatically generate PIC electronic layer layouts. PICELF primarily consists of two key components: an NBP-based netlist generator and a two-stage router. Our NBP-based netlist generator, enhanced with scalability optimization, guarantees to generate legal netlists for target PICs within reasonable times. Leveraging the generated netlist, our two-stage router combines initial parallel routing with post-routing optimization to achieve high-quality results rapidly while ensuring routability and legality. Experimental results validate the effectiveness and efficiency of our framework. PICELF

demonstrably outperforms existing EDA methods for PIC electronic layer physical design, contributing to the advancement of photonic-electronic design automation (PEDA).

TABLE V: Comparison of the routing result between [24] and Ours

PIC Chips	Ro (%)		#Err		WL ( $\mu\text{m}$ )		#Bends		Time (s)	
	[24]	Ours	[24]	Ours	[24]	Ours	[24]	Ours	[24]	Ours
Chip01	100.0	100.0	0	0	36600	36600	34	34	15	18
Chip02	100.0	100.0	0	0	49800	49800	37	39	76	38
Chip03	96.3	100.0	0	0	142142	144900	185	200	1316	67
Chip04	100.0	100.0	0	0	143400	143400	78	78	41	18
Chip05	100.0	100.0	0	0	145236	140785	187	198	520	90
Chip06	92.0	100.0	0	0	138336	136380	468	326	1694	111
Chip07	99.0	100.0	0	0	157684	157400	168	146	431	61
Chip08	100.0	100.0	0	0	220900	221100	255	260	84	62
Chip09	97.0	100.0	0	0	227317	223920	339	289	2483	293
Chip10	92.9	100.0	0	0	184520	182992	506	325	2848	165
Chip11	94.5	100.0	0	0	284621	287055	604	400	4490	298
Chip12	93.9	100.0	0	0	216149	224575	393	329	4642	159

\* Ro: routability, #Err: number of design rule errors, WL: wirelength, #Bends: bend count.

### ACKNOWLEDGMENT

This work is partially supported by the National Natural Science Foundation of China (No.62474152), Guangzhou-HKUST(GZ) Joint Funding Program (No. 2023A03J0013), Nansha District Key Area S&T Scheme (No. 2024ZD007), and InnoHK ACCESS.

### REFERENCES

- [1] Z. Xiao, W. Liu, S. Xu, J. Zhou, Z. Ren, and C. Lee, “Recent progress in silicon-based photonic integrated circuits and emerging applications,” *Advanced Optical Materials*, vol. 11, no. 20, p. 2301028, 2023.
- [2] Y. Shen, N. C. Harris, S. Skirla, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Laroche, D. Englund, *et al.*, “Deep learning with coherent nanophotonic circuits,” *Nature photonics*, vol. 11, no. 7, pp. 441–446, 2017.
- [3] F. Ashtiani, A. J. Geers, and F. Aflatouni, “An on-chip photonic deep neural network for image classification,” *Nature*, vol. 606, no. 7914, pp. 501–506, 2022.
- [4] Z. Xu, T. Zhou, M. Ma, C. Deng, Q. Dai, and L. Fang, “Large-scale photonic chiplet taichi empowers 160-tops/w artificial general intelligence,” *Science*, vol. 384, no. 6692, pp. 202–209, 2024.

- [5] A. K. Coskun, A. Gu, W. Jin, A. Joshi, A. B. Kahng, J. Klamkin, Y. Ma, J. Recchio, V. Srinivas, and T. Zhang, “Cross-layer floorplan optimization for silicon photonic noc in many-core systems,” in *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1309–1314, IEEE, 2016.
- [6] M. Li, T.-M. Tseng, D. Bertozzi, M. Tala, and U. Schlichtmann, “Custom-topo: A topology generation method for application-specific wavelength-routed optical noc,” in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–8, IEEE, 2018.
- [7] A. Boos, L. Ramini, U. Schlichtmann, and D. Bertozzi, “Proton: An automatic place-and-route tool for optical networks-on-chip,” in *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 138–145, IEEE, 2013.
- [8] A. V. Beuningen, L. Ramini, D. Bertozzi, and U. Schlichtmann, “Proton—a placement and routing tool for 3d optical networks-on-chip with a single optical layer,” *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 12, no. 4, pp. 1–28, 2015.
- [9] Y.-K. Chuang, K.-J. Chen, K.-L. Lin, S.-Y. Fang, B. Li, and U. Schlichtmann, “Planaronoc: concurrent placement and routing considering crossing minimization for optical networks-on-chip,” in *Proceedings of the 55th Annual Design Automation Conference*, pp. 1–6, 2018.
- [10] A. Truppel, T.-M. Tseng, and U. Schlichtmann, “Psion 2: Optimizing physical layout of wavelength-routed onocs for laser power reduction,” in *Proceedings of the 39th International Conference on Computer-Aided Design*, pp. 1–9, 2020.
- [11] D. Ding, Y. Zhang, H. Huang, R. T. Chen, and D. Z. Pan, “O-router: an optical routing framework for low power on-chip silicon nano-photonic integration,” in *Proceedings of the 46th annual design automation conference*, pp. 264–269, 2009.
- [12] D. Ding, B. Yu, and D. Z. Pan, “Glow: A global router for low-power thermal-reliable interconnect synthesis using photonic wavelength multiplexing,” in *17th Asia and South Pacific Design Automation Conference*, pp. 621–626, IEEE, 2012.
- [13] C. Condrat, P. Kalla, and S. Blair, “A methodology for physical design automation for integrated optics,” in *2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 598–601, IEEE, 2012.
- [14] C. Condrat, P. Kalla, and S. Blair, “Crossing-aware channel routing for integrated optics,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 6, pp. 814–825, 2014.
- [15] J.-T. Yan, “On-chip optical channel routing for signal loss minimization,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 8, pp. 1654–1666, 2017.
- [16] Y.-S. Lu, S.-J. Yu, and Y.-W. Chang, “A provably good wavelength-division-multiplexing-aware clustering algorithm for on-chip optical routing,” in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, IEEE, 2020.
- [17] Y.-S. Lu, S.-J. Yu, and Y.-W. Chang, “On-chip optical routing with provably good algorithms for path clustering and assignment,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 4653–4666, 2021.
- [18] D. Liu, Z. Zhao, Z. Wang, Z. Ying, R. T. Chen, and D. Z. Pan, “Operon: optical-electrical power-efficient route synthesis for on-chip signals,” in *Proceedings of the 55th Annual Design Automation Conference*, pp. 1–6, 2018.
- [19] J.-W. Fang, C.-H. Hsu, and Y.-W. Chang, “An integer linear programming based routing algorithm for flip-chip design,” in *Proceedings of the 44th annual Design Automation Conference*, pp. 606–611, 2007.
- [20] P.-W. Lee, H.-C. Lee, Y.-K. Ho, Y.-W. Chang, C.-F. Chang, I.-J. Lin, and C.-F. Shen, “Obstacle-avoiding free-assignment routing for flip-chip designs,” in *Proceedings of the 49th Annual Design Automation Conference*, pp. 1088–1093, 2012.
- [21] T.-C. Yu, A.-J. Shih, and S.-Y. Fang, “Flip-chip routing with i/o planning considering practical pad assignment constraints,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 8, pp. 1921–1932, 2019.
- [22] H. Kong, T. Yan, and M. D. Wong, “Optimal simultaneous pin assignment and escape routing for dense pcbs,” in *2010 15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 275–280, IEEE, 2010.
- [23] S.-T. Lin, H.-H. Wang, C.-Y. Kuo, Y. Chen, and Y.-L. Li, “A complete pcb routing methodology with concurrent hierarchical routing,” in *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pp. 1141–1146, IEEE, 2021.
- [24] T.-C. Lin, D. Merrill, Y.-Y. Wu, C. Holtz, and C.-K. Cheng, “A unified printed circuit board routing algorithm with complicated constraints and differential pairs,” in *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, pp. 170–175, 2021.
- [25] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [26] <https://www.gurobi.com/solutions/gurobi-optimizer/>.
- [27] M. Köfferlein, “Klayout,” 2020.