A zero-crossing optical waveguide routing method

Ziyue Zhang[1,*], Didier Colle[2,*], Wouter Tavernier[2,*], and Mario Pickavet[1,*]

[1]Department of Information Technology, Gent University - IMEC, Belgium
[2] IMEC - Department of Information Technology, Gent University, Belgium
[*]{First name}.{Family name}@UGent.be

***Abstract*—Photonic Network on Wafer (PNoW) for multi-chip computing systems is a promising solution for providing high inter-chip bandwidth while maintaining low energy consumption. This paper introduces a novel approach for routing optical waveguides in PNoWs on a 2-dimensional wafer plane, eliminating the need for any waveguide crossings. Our proposed method guarantees zero optical waveguide crossings, therefore high signal-to-noise ratio (SNR) and low signal loss in the photonics circuits. Notably, our proposed routing method is applicable to PNoWs with significantly higher inter-chip bandwidth, surpassing current commercial products by one magnitude. In contrast, we demonstrate that classical routing heuristics that allow waveguide crossings might result in a substantial degradation in terms of signal qualities, which is evaluated with certain approximations in this work. Furthermore, our grid-less routing method imposes minimal computational requirements, offering an efficient and practical solution for the implementation of high-performance PNoWs.**

***Index Terms*—Photonic network, optical waveguide routing, network on wafer, network on chip, Optical network management, Optical network automation**

I. Introduction

A multi-chip computing system serves as an imperative solution for augmenting computing power by interconnecting numerous computing or storage modules. Such an architecture stands as a powerful computer, particularly adept at handling large-scale parallel computing tasks such as AI training, physics simulations, and geometric analysis, etc. At the heart of this system lies a network that facilitates inter-chip communication, necessitating high-data-rate, power-efficient and lossless packet delivery. Notable instances of multi-chip computing systems that operate at the wafer-scale include Intel's multi-core CPUs, NVIDIA's DGX-series multi-GPU [1] and AMD's 'crossfire MGPU' technology [2], alongside High Performance Computing (HPC) systems functioning at a significantly larger scale.

In this work we focus on wafer-scale multi-chip computing systems. Throughout this paper, the term '*bidirectional bandwidth per chip*' of $\Theta[GBps]$ denotes that each computing chip possesses $\Theta[GBps]$ ingress bandwidth and $\Theta[GBps]$ egress bandwidth, $2 * \Theta[GBps]$ bandwidth overall. For example, NIVIDA's DGX-2 [1] multi-GPU interconnects up to 16 GPUs using advanced wafer-scale interconnects like NVLink and NVswitch, delivering remarkable $500GBps$ of Θ for every GPU.

The bandwidth demand of the inter-chip communication scales rapidly when scaling computing tasks in multi-chip computing systems. For instance, simulation results highlighted in [3] demonstrated that the bidirectional inter-chip bandwidth demand can surge from $\Theta = 512GBps$ to $\Theta = 2048GBps$ when transitioning from a 4-GPU to a 16-GPU system. This substantial increase of inter-chip bandwidth poses a significant challenge for electronic on-wafer interconnects – it is difficult to achieve *TBps-scale* bidirectional inter-chip bandwidth in a power efficient way using electronic networks on a wafer.

Addressing this challenge, [3] advocates for the adoption of optical waveguides and possible optical switches to connect computing chips on one wafer, introducing the concept of a Photonic-Network-on-Wafer (PNoW). Such a PNoW solution potentially provides over *TBps-scale* inter-chip bandwidth while maintaining about 5 pJ/bit power consumption [3], [4]. Advancements in the industrial frontiers such as 'TeraPHY' [5] optical I/O from *Ayar Labs* and 'PASSAGE' from *Lightmatter* [6] are also actively exploring avenues to upscale inter-chip bandwidth with state-of-the-art silicon photonics technology.

The problem we address in this paper continues the discussion in [3] where a PNoW is deployed for the connection of multiple computing chips on a wafer. The state-of-art CMOS technologies are able to support the fabrication of such PNoWs. For example, the micro transfer printing technology [7] [8] enables massively parallel (co-)integration of a wide range of materials/devices on the wafer scale. Such a PNoW is fabricated in a 2.5-D fashion: Passive optical devices (waveguides, couplers, etc.) are integrated on the bottom layer of the wafer, providing connectivity among the optical transmitters and receivers. Active optical devices and their pairing CMOS chips are interposed on the wafer, where CMOS chips include electronic driver chips for the optical transmitters; or electronic low-noise transimpedance amplifier (TIA) and clock-and-data recovery (CDR) chips for the optical receivers. The computing chips are then interposed on the top of the wafer, their data ports are coupled with the CMOS chips. Figure 1 illustrates an example where four computing chips are interposed on a wafer and each of them owns 9 CMOS chips.

In such an architecture, data originating from a computing chip is encoded into optical signal via an optical transmitter (e.g., optical modulators with external laser source) that is driven by an CMOS electronic driver chip. The optical signal eventually arrives at an optical receiver (e.g., photo diodes) via a certain form of optical connection (discussed later). The output from the optical receiver is further processed by an TIA-CDR CMOS chip module. Finally the data is delivered to the

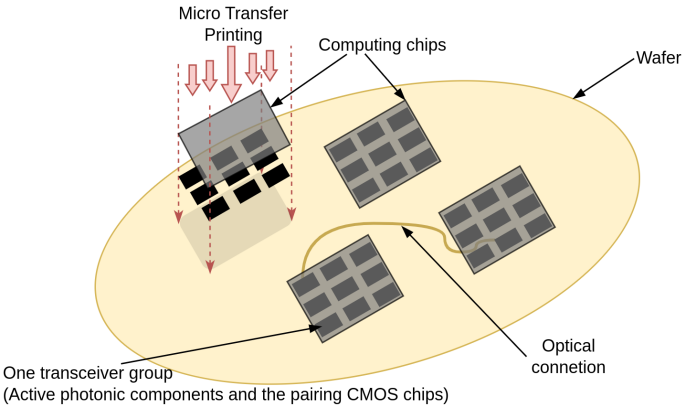

Fig. 1. Micro Transfer Printing (MTP) enables the fabrication of such a PNoW that interposes the processing chips on the top, CMOS chips and active photonic devices in the middle, and passive optical connections underneath.

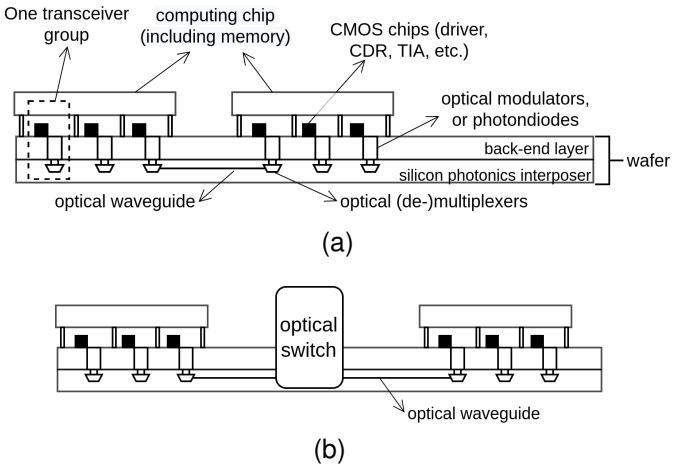

Fig. 2. Sub-figure (a) illustrates the cross-section of PNOW-DW. Sub-figure (b) illustrates the cross-section of PNOW-SW.

destination computing chip for continuing the computation.

Moreover, one optical connection can accommodate optical signals in multiple wavelengths with the help of Wavelength Division Multiplexing (WDM) modules. For simplicity of future discussions, we define a "transmitter group" as the combination of optical transmitters in multiple wavelengths, the WDM multiplexing module, and their electronic driver modules. Similarly, define a "receiver group" as the combination of optical receivers responding in multiple wavelengths, the WDM de-multiplexing module, and their electronic TIA-CDR modules. Therefore, an optical connection is established from a transmitter group to a receiver group. Without loss of generality, further denote a "transceiver group" as the combination of a "transmitter group" and a "receiver group", since it is a common practice to provide bi-directional data ports in a communication network.

We further ~~Define~~ define homogeneous and heterogeneous PNoWs. A homogeneous PNoW is composed by a number of identical computing chips, such that every computing chip owns the same number of transceiver groups. On the other hand, the computing chips in a heterogeneous PNoW could be of different kinds or models, and they could own different numbers of transceiver groups due to their possibly different bandwidth demands or footprint constraints. We will first propose a zero-crossing optical waveguide routing method for homogeneous PNoW in section III, then discuss the generalization of this method for heterogeneous PNoW in section IV.

Furthermore, an optical connection can have different forms of realization. In the scope of this paper, we consider two forms of optical connections which lead to the following two types of PNoW:

- **PNoW with direct waveguides (PNoW-DW)**

As shown in figure 2a, an optical connection between two transceiver groups is realized by direct optical waveguides in PNoW-DW. Such a network has a static bandwidth distribution which cannot be reconfigured after the fabrication of the PNoW, therefore it is difficult for such a network to utilize idle bandwidth capacities. The NVIDIA DGX-1 multi-GPU have adopted such static network topologies [1]. Efforts have also been made in cache management and memory page allocation policies for improving benchmark performances in such static networks [9]–[11].

- **PNoW with optical switches (PNoW-SW)**

As proposed in [3], an optical connection can also be established via optical switches and optical waveguides. As shown in figure 2b, the light wave travels through an optical switch which determines the path it takes. Such a PNoW configuration works in a circuit-switching style as discussed in [12]. Compared to the direct waveguide connections in PNoW-DW, PNoW-SW provides more flexibility that can adapt bandwidth to the network traffic demand by reconfiguring the optical switches, therefore it is likely to improve the overall computational performance compared to a static bandwidth distribution [5], [13]. A synchronous network control scheme and the corresponding algorithms have been proposed in our previous work [12] which can dynamically calculate and reconfigure the chip-to-chip bandwidth distribution in order to match with the estimated network traffic demand. On the down side, the optical switches add more complexity on the PNoW architecture, and more fabrication and operation costs might be induced. The NVIDIA DGX-2 multi-GPU design has adopted such a flexible network topology as well [1], except that the NVSwitches are electronic and work in a packet-routing style. The SiP-ML [5] multi-GPU optical network adopts very similar circuit-switching network architecture as PNoW-SW, but expands such a computation system to the HPC-scale instead of wafer-scale.

In this work, we propose an optical waveguide routing method that leads to strictly zero optical waveguide crossings for the above considered PNoWs. The physical placement of computing chips on the wafer is related to the footprint and cooling aspect of the wafer design, which goes out of the scope of this paper. However, as will be shown later, the physical placement of computing chips is irrelevant to the proposed method, therefore is assumed to be given in the considered problem.

For PNoW-DW, the target PNoW architecture is fully char-

acterized by its bandwidth distribution ~~among~~ (i.e., how many waveguide connections are required by) among each pair of the computing chips. For PNoW-SW, the target PNoW architecture is fully characterized by the bandwidth distribution among computing chips and the optical switches. Therefore, the target PNoW architecture, characterized by the bandwidth distribution, is the only input to the proposed method, and the output is ~~the~~ a zero-crossing waveguide routing result.

The rest of this paper will be organized as the following: Section II briefly discusses how does this problem differ from the electronic wire routing problem, and therefore motivates our proposed zero-crossing method. Section III delves extensively into our proposed method. Section IV discusses the generalization of the proposed method for heterogeneous PNoWs. Section V evaluates our proposed method. Finally section VI concludes this work.

II. Related works and motivations

In this section, we provide a brief overview of the electronic wire routing problem, a topic extensively studied over the past few decades. We then compare it with the optical waveguide routing problem in PNoW, highlighting the key differences between the two. This comparison underscores the need for new methodologies and algorithms specifically for routing optical waveguides on wafers.

• **Electronic wire routing**[1]

The electronic wire routing problem in Integrated Circuits (IC), Printed Circuit Boards (PCB) and more recently Very-large-scale integration (VLSI) designs has been well investigated in the literature. The electronic wires in these systems need to be routed on semiconductor boards, considering the underlying technology and the circuitry's functionality:

Two electronic wires cannot cross on the same layer of the board. Modern electronic boards usually contain multiple layers because often a planar routing is not feasible, or only feasible with long wire detours which conflicts other aspects of design rules, as will be explained later. Therefore, vias are necessary in modern electronic circuits to avoid wire crossing by connecting multiple layers of boards, despite they introduce extra signal loss, power consumption and fabrication costs. However, the number of layers used by the circuit becomes another factor in the fabrication cost. The number of layers is therefore often minimized, while satisfying other design rules. Modern PCBs can contain up to 20 layers [14], [15].

Wire length also plays a crucial role in PCB wire routing due to its direct influence on the electrical performance of the circuitry. Longer wires contribute to increased resistance, and it is a common practice to minimize electronic wire lengths on a chip for achieving low energy consumption and less heat dissipation. Moreover, for high-frequency circuits, longer wires exhibit higher capacitance and inductance, impacting signal propagation and behavior. Therefore, designers often adhere to stringent length constraints for wires in critical paths for high-frequency circuits [15], [16], ensuring optimal signal timing and reduced signal degradation in electronic circuits. Another key aspect of electronic wire routing is managing electromagnetic compatibility. Effective routing, shielding, grounding, and signal separation are vital for ensuring reliable circuit operation.

A heuristic for the electronic wire routing problem is commonly called a 'router'. 'Placement and routing' is a common technique [17] in such routers, it first estimates the best placements on the board for the electronic modules, and then all electronic wires that connect component pins, as per the logical design, must be routed on possible layers. 'Escape routing' is also popular [15], [18], where the wires first escape to the boundaries of the electronic modules, followed by layer assignment and area routing processes. Some state-of-the-art routers also use techniques such as Monte-Carlo Tree Search [19] and 3-D Astar algorithms [20].

• **Optical waveguide routing**

In contrast, the problem of routing optical waveguides on a wafer has only recently drawn attention. In fact, the optical waveguide routing problem has different design rules and trade-offs compared to the electronic wire routing problem. Research on the optical waveguide routing problem is urgently needed due to similarities of silicon photonics technology and electronics technology – increasing complexity and dimension of circuitry. Most photonic circuitry are relatively simple, e.g., those in feed-forward laser modules and basic interferometers. The optical waveguides in these circuitry are usually routed manually with limited human efforts. However, as the field of silicon photonics grew rapidly in the recent decades, photonic circuitry with higher complexity and dimension started to occur, such as photonic neural network circuits [21], programmable photonic circuits [22], and Photonic Network-on-wafer (PNoW), as discussed here.

The divergence between design rules for photonic circuits and for electronic circuits stems from their distinct technological frameworks.

First of all, signals are carried by light (photons) in optical waveguides, which is generally immune to electromagnetic interference since light does not interact with electric and magnetic fields in the same way as electrical signals. Another defining feature of photonic circuits is the inherent potential for waveguide crossings. The insertion loss and crosstalk per crossing can be low ($-20dB$ loss and $37dB$ crosstalk per crossing was shown in [23]). However, the overhead of insertion loss and crosstalk from the waveguide crossings can become intolerable, especially for achieving ultra low Bit-Error-Ratio while the number of waveguide crossings is high.

Furthermore, advancements in optical waveguide technology have led to lower signal attenuation. Materials such as Silicon Nitride (SiN) demonstrate exceptional performance with propagation losses as low as $3\ to\ 10dB/m$ [24]. This not only addresses concerns regarding signal quality in PNoWs but also offers the flexibility to implement extended waveguide paths or detours to circumvent the need for waveguide crossings.

Photonic circuits typically rely on single-layer wafers. Multi-layer photonic wafers are beginning to be discussed in the literature [25], but still remain experimental. Therefore, it

[1]or simply 'wire routing' problem as referred in most literature. Here we use the term 'electronic wire routing' in order to differentiate from 'optical waveguide routing' problem.

is very important to be able to route optical waveguides on this single layer without any crossings.

In the end, the focal points in the optical waveguide routing problem revolve around mitigating optical loss and managing signal crosstalk, which we also used for evaluating our method in section V-B.

Simply applying electronic wire routing heuristics for optical waveguide routing in photonic circuits is not ideal because they have very different design rules and trade-offs. Although escape routing heuristics could be used for optical waveguide, we will show in section V-B that escape routers lead to a considerable number of crossings and therefore lead to intolerable signal performances for the photonics circuits. Grid-based path-finding heuristics might be another candidate solution, learning from the literature of electronic wire routing. However, Grid-based path-finding algorithms, such as Astar algorithm, requires extensive computing power given the problem size: the minimal spacing between two optical waveguides is typically $10\mu m\text{-}scale$, while the wafer is typically $100\,mm\text{-}scale$ in diameter. This leads to 2-D grids to the scale of 1E4-by-1E4, finding routes for a large number of optical waveguides in such a large grid requires heavy computing power, and most importantly, such a method does not guarantee the discovery of zero-crossing solutions. Even if they do, achieving accurate results require extensive fine-tuning of path-finding heuristics, which involves numerous iterations. The proposed method in this paper is grid-less, grounded in observations and heuristics, thereby avoiding such brute-force approaches that require extensive computational resources.

The proposed method functions as an algorithm that takes the target PNoW architecture as the input and generates the corresponding zero-crossing optical waveguide routing as output, promising $5\,TBps\text{-}scale$ bidirectional inter-chip bandwidth in the worst-case scenario for interconnecting 16 identical computing chips while maintaining low optical loss and crosstalk in the network.

Now we delve into the proposed method.

III. Zero-crossing optical waveguide routing for homogeneous PNoW

We first mathematically prove that a zero-crossing solution exists for the optical waveguide routing problem in homogeneous PNoW-DW under certain assumptions and then describe our proposed method with a small example in subsection III-A. We next show that a simple modification leads to our zero-crossing solution for PNoW-SW in subsection III-B. We then provide a generally defined method of zero-crossing optical waveguide routing for homogeneous PNoW-DW and PNOW-SW in subsection III-C, with important technical details in addition to the two examples in the previous two subsections. In the last subsection III-D we present a worst-case analysis on the proposed method.

A. Homogeneous PNoW-DW

Let G be a simple directed graph[2] that represents the optical waveguide connectivity in PNoW-DW, such that each vertex in G represents one transceiver group and each arc in G represents one directional optical waveguide connection from one transceiver group's output port to the other's input port. An example of graph G is shown in figure 3. Because each transceiver group has exactly one input port and one output port, G is a regular graph in which vertices' ingress and egress degrees are exactly 1. Therefore, it is apparent that every connected component in G forms a cycle and thus every connected component in G is planar. Because G is a graph in which every connected component is planar, G is planar.

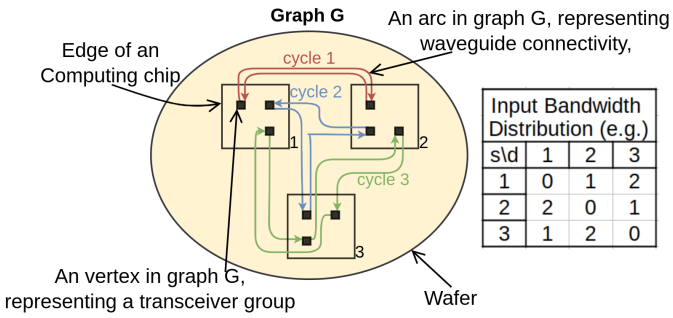

Fig. 3. The directed graph G represents the connectivity among transceiver groups, corresponding to the input bandwidht distribution matrix. G is always planar, since it must be composed by one multiple cycles.

By definition, a planar graph can be drawn without edge intersection on a plane. A planar graph can still be drawn on a plane without edge intersection even if all vertices in the graph are fixed in specific positions on the plane and the edges can be drawn in arbitrary shapes [26]. This however assumes that all edges are infinitely thin and any two edges can be as close to each other as possible.

Similarly, the optical waveguide connectivity among transceiver groups in a PNoW-DW can be seen as a free-drawing of waveguides on the wafer plane in the photonics layer, while keeping transceiver groups at fixed positions on the wafer. However, real optical waveguides have finite thickness and they have to keep a certain distance from each other in order to avoid undesired optical interference. Therefore, assuming that the spacing between adjacent transceiver groups is large enough for accommodating the bypassing waveguides, there exists a zero-crossing routing solution. We will show in section V-A that the footprint of transceiver groups and computing chips, and the minimal spacing between optical waveguides pose limitations on the number of optical waveguides that are able to pass through two adjacent transceiver groups, and therefore pose limitations on the bidirectional bandwidth per computing chip that can be effectively solved by our proposed method. The proposed method is able to achieve one magnitude higher inter-chip bandwidth in the worst-case scenarios compared to the state-of-art products (section V-A).

[2]Simple directed graphs are directed graphs that have no loops

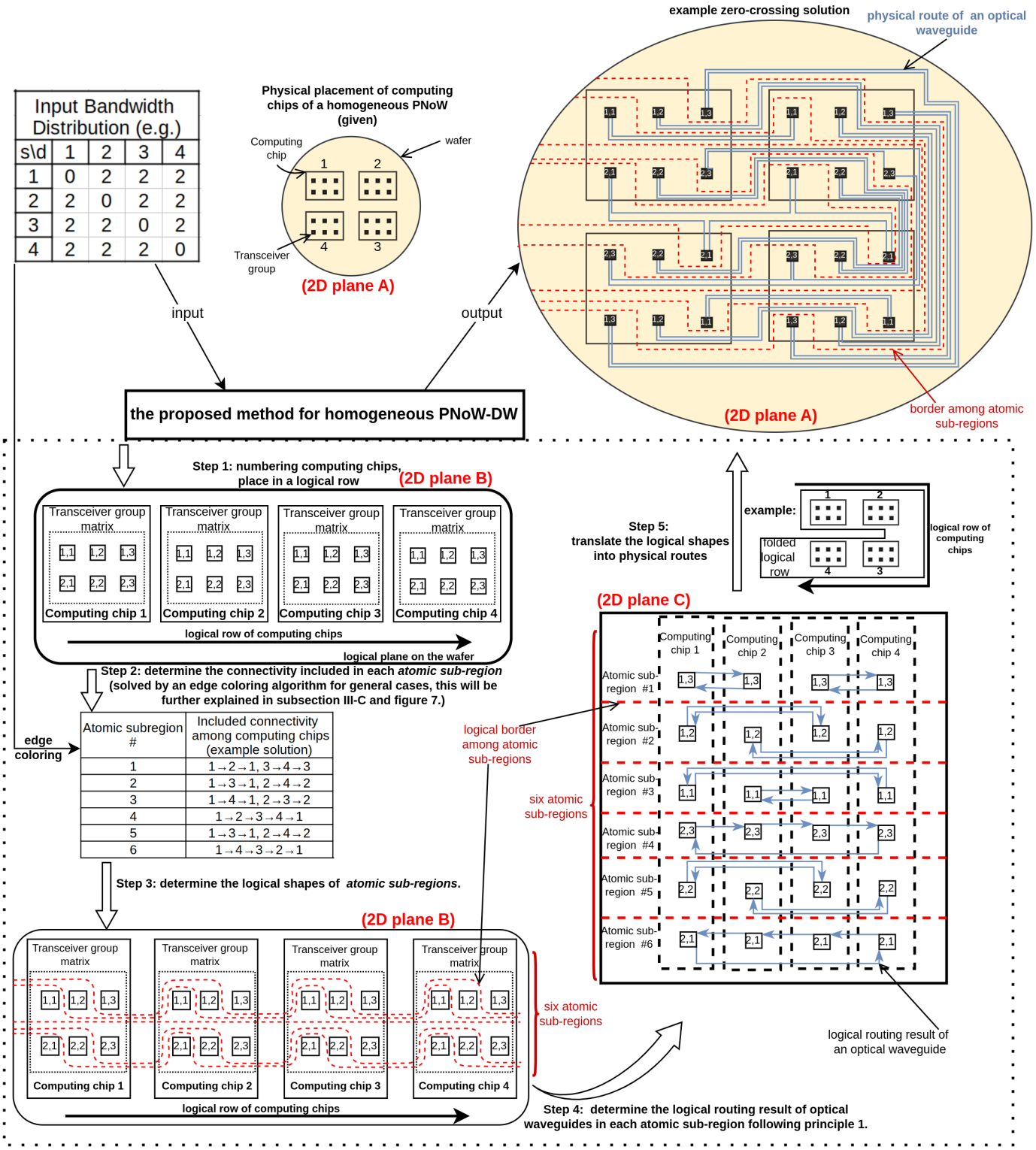

Fig. 4. A small example of the step-by-step zero-crossing method for PNoW-DW. (This figure has been adjusted in the texts of 'step 2', such that there is a reference to subsection III-C and figure 7, when mentioning the edge coloring algorithm.)

For achieving zero-crossing routing in an efficient way, we introduce the concept of *atomic sub-regions* in our method. An *atomic sub-region* is a connected space[3] being a subset of the 2-D plane on the wafer . Any transceiver group is contained in exactly one *atomic sub-region*. Any two atomic sub-regions do not overlap with each other.

Principle 1: An optical waveguide can only be routed within one *atomic sub-region* on the 2-D plane, thus never crosses the

[3]A connected space is a topological space that cannot be represented as the union of two or more disjoint non-empty open subsets.

boundaries of *atomic sub-regions*.

Therefore, each *atomic sub-region* is defined to contain a number of transceiver groups and the optical waveguides connecting them. As will become clear later, the implementation of *atomic sub-regions* and **Principle 1** not only contribute to finding an elegant zero-crossing solution, but also help to limit the number of waveguides passing by the most crowded area of waveguides and therefore determines the inter-chip bandwidth upper-bound in the worst cases.

Without loss of generality, we now use a small example to illustrate the proposed zero-crossing solution for the optical waveguide routing problem in PNoW-DW. A general description of our method will be given in subsection III-C. In this small example, assume that there are four computing chips on the wafer and each of them possesses six transceiver groups. Assume the desired bandwidth distribution among the computing chips is uniform, meaning that there need to be exactly two unidirectional optical waveguides connecting from transceiver groups in one computing chip to that in every other computing chip. Assume that the placement of computing chips is already given on the wafer, a proposed zero-crossing optical waveguide routing solution is obtained through the following five steps. Figure 4 illustrates our proposed method step-by-step for this small example, which will be explained thoroughly in the following steps. The input of the proposed method, the bandwidth distribution, is on the left-top corner of figure 4, and the output (a zero-crossing waveguide routing result) of the proposed method is shown on the right-top corner. The five steps of the proposed methods is illustrated with examples in the rest of figure 4.

Let plane **A** be a finite 2-D plane that represents the plane for routing optical waveguides on the wafer.

As a result of the following steps, several homeomorphic planes of plane **A** will be introduced. A homeomorphism is a continuous and bijective mapping between two topological spaces, with a continuous inverse. Two spaces are said to be homeomorphic if there exists a homeomorphism between them, meaning they are topologically equivalent—they can be deformed into one another without tearing or gluing. In the context of this work, homeomorphic planes are 2-D surfaces that share the same topological properties, even though their specific layouts or arrangements might differ. And a zero-crossing waveguide routing on two homeomorphic planes can be converted into each other with mappings.

step 1: Choose an order for the computing chips and assign them with IDs. Then logically place the computing chips in a row (i.e., an 1-D array), such that the logical row passes by the computing chips following the incremental order of their IDs. In the small example illustrated in figure 4, the logical row passes by the top left, top right, bottom right, and bottom left computing chips consecutively. As a result, we obtain another 2-D plane **B** which is homeomorphic to plane **A**, denote this homeomorphism as $f : \mathbf{A} \rightarrow \mathbf{B}$.

step 2: Define six *atomic sub-regions*, each containing exactly one transceiver group from every computing chip. Determine the connectivity in each *atomic sub-region*, such that the overall desired bandwidth distribution is achieved. It is trivial to achieve the uniform bandwidth distribution for the four computing chips in this small example, as shown in figure 4. ~~In more general terms, the connectivity in *atomic sub-regions* can be solved by an edge coloring algorithm as will be discussed in III-C.~~ In more general terms, the connectivity in *atomic sub-regions* can be solved by an edge coloring algorithm, this will be further explained in subsection III-C.

step 3: Determine the shapes of the *atomic sub-regions*, which is trivial on plane **B**. Figure 4 (left-bottom corner) illustrates an example of such a result, where the borders of two adjacent *atomic sub-regions* merge so that there are no gaps in between. The borders among *atomic sub-regions* are represented as red dashed ~~line~~ lines in the figure.

step 4: Define another homeomorphism $g : \mathbf{B} \rightarrow \mathbf{C}$, mapping from plane **B** to plane **C**. The boundaries of *atomic sub-regions* are straight in plane **C**, which makes routing waveguides in each *atomic sub-region* easy. Within each *atomic sub-region*, determine the logical shapes of optical waveguides such that the connectivity in each *atomic sub-region* (result of **step 2**) is fulfilled while respecting **principle 1**. In this small example, it is trivial to determine the logical shapes of optical waveguides without crossing. Figure 4 illustrates an example solution, where we use blue solid lines to represent the logical routing of a waveguide, and the arrow indicates the direction of signal transmission. An additional technique will be deployed for dealing with general cases, as will be discussed in subsection III-C.

Until now, all optical waveguides have been routed on plane **C**, without any crossings.

step 5: Map the logical waveguide routing obtained from the previous steps onto physical routes of optical waveguides on the wafer plane:

Because plane **A** is homeomorphic to plane **B**, and plane **B** is homeomorphic to plane **C**, plane **A** is homeomorphic to plane **C** with the homeomorphism being the composition of f and g, $f \circ g : \mathbf{A} \rightarrow \mathbf{C}$. Therefore, every logical waveguide routing, which is a curve on plane **C**, can be mapped onto a curve on plane **A** with mapping $(f \circ g)^{-1}$.

Because homeomorphisms preserve topological properties, they also preserve the relative positioning of curves. As a result, any two non-crossing curves on plane **C** will also not cross if they are mapped onto plane **A**. Therefore, step 5 should not induce any waveguide crossings.

The final zero-crossing optical waveguide routing result is shown in the top-right corner of figure 4. The relationship of the three homeomorphic planes **A**, **B** and **C** are illustrated in figure 5.

B. Homogeneous PNoW-SW

Let G be a simple directed graph that represents the optical waveguide connectivity in a PNoW-SW. An optical waveguide cannot be routed through any transceiver group or any optical switch in PNoW-SW, therefore, let a vertex in G represent either one transceiver group or one optical switch. Each edge in G represents one directional optical waveguide connection between one transceiver group and one port of an optical

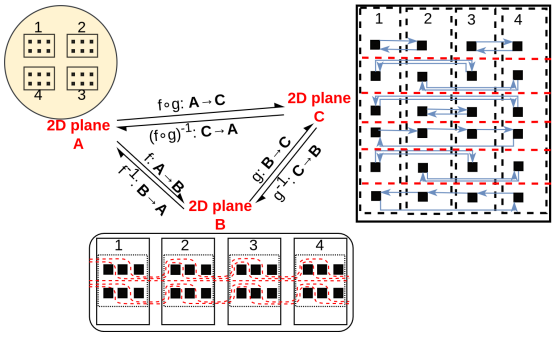

Fig. 5. The homeomorphism among three 2-D planes. (This figure has been adjusted for better readability.)

switch. The resulting graph G has two types of vertices, those who represent transceiver groups have vertex ingress and egress degrees all equal to 1; another type of vertices that represent optical switches have vertex ingress and egress degrees equal to the number of input and output ports per optical switch. Moreover, when drawing such a graph G, the fan-out of the edges from any vertex that represents an optical switch needs to follow a certain clock-wise order, because normally an optical switch has input ports on one side and output ports on another side of the packaging.

Unlike in PNoW-DW, it is nontrivial to determine the planarity of such an embedded connectivity graph G in PNoW-SW. However, we prove G to be planar by demonstrating a zero-crossing drawing of G in this subsection. In the following description, we show that the zero-crossing optical waveguide routing method described for PNoW-DW in subsection III-A can be easily adapted for PNoW-SW to also achieve zero waveguide crossings.

Define another small example for PNoW-SW: there are again four computing chips on the wafer and each of them possesses six transceiver groups. There are six optical switches, each possessing four input ports and four output ports that need to be connected to four transceiver groups in different computing chips. The proposed zero-crossing optical waveguide routing solution is shown in figure 6, obtained by the following five steps:

Let plane **A** be a finite 2-D plane that represents the plane for routing optical waveguides on the wafer.

step 1: Same as **step 1** in subsection III-A. Obtain the homeomorphic plane **B**, $f : \mathbf{A} \rightarrow \mathbf{B}$.

step 2: Define six *atomic sub-regions*, each containing exactly one optical switch and exactly one transceiver group from each computing chip. The connectivity in each *atomic sub-region* is straight-forward: every transceiver groups connects to one input port and one output port of the optical switch within the same *atomic sub-region*.

step 3: Determine the logical shapes of the *atomic sub-regions* on plane **B**. This is the same as **step 3** in subsection III-A.

step 4: Define another homeomorphism $g : \mathbf{B} \rightarrow \mathbf{C}$, mapping from plane **B** to plane **C**. The boundaries of *atomic sub-regions* are straight in plane **C**, which makes routing waveguides in each *atomic sub-region* easy. Within each *atomic sub-region*, determine the logical shapes of optical waveguides such that the connectivity in each *atomic sub-region* (result of **step 2**) is fulfilled while respecting **principle 1**. In this small example, it is trivial to determine the logical shapes of optical waveguides without crossing. Figure 6 (top) illustrates an example solution.

step 5: Same as **step 5** in subsection III-A, map the logical waveguide routing onto physical routes of optical waveguides on the plane **A** with $(f \circ g)^{-1}$.

The final zero-crossing optical waveguide routing result is shown in figure 6 (bottom).

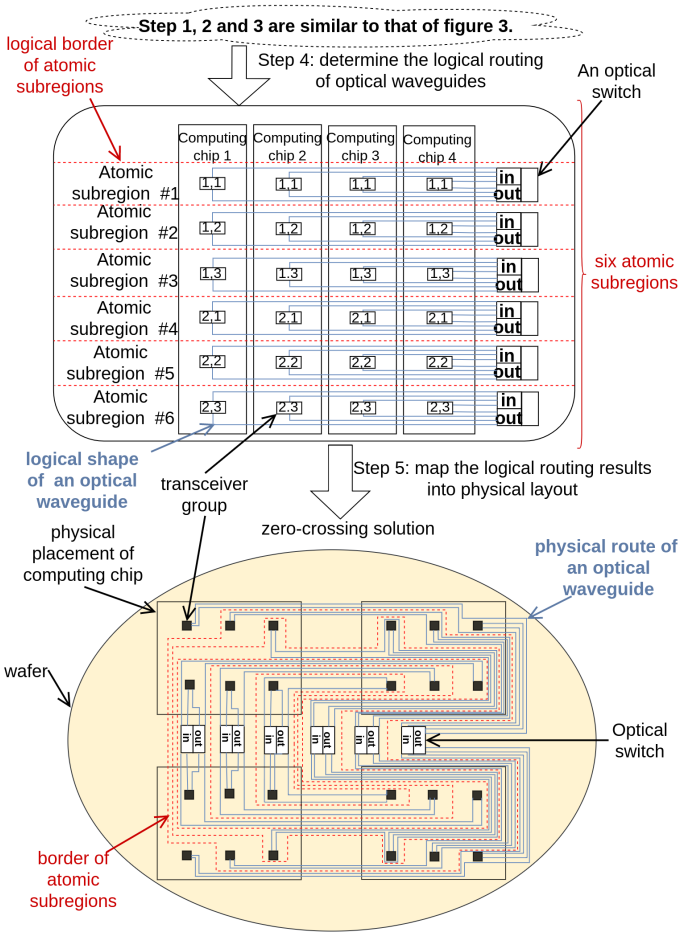

Fig. 6. A small example of the zero-crossing method for PNoW-SW.

C. Generalized method for homogeneous PNoW

This subsections aims at providing a generalized method for the examples described in the previous two subsections. A few more technical details will be elaborated.

Assume we have N computing chips in a PNoW, and each computing chip possesses exactly T transceiver groups due to the homogeneity. For PNoW-DW, assume that there is a desired inter-chip bandwidth distribution that can be represented by an N-by-N matrix M. The value at entry (i, j) in matrix M represents the number of unidirectional optical waveguide connections needed from computing chip i to chip

j. Diagonal values in M are zero because there is no self-loop optical connections for computing chips. PNoW-SW is designed to connect every optical switch to all computing chips evenly, it is therefore not necessary to have such a bandwidth distribution matrix. Our proposed method for achieving a zero-crossing optical waveguide routing solution is described as the following five steps:

Let plane **A** be a finite 2-D plane that represents the plane for routing optical waveguides on the wafer.

step 1: Choose an order for the computing chips and assign them with IDs. Then logically place the computing chips in a row (i.e., an 1-D array), such that the logical row passes by the computing chips following the incremental order of their IDs. As a result, we obtain another 2-D plane **B** which is homeomorphic to plane **A**, denote this homeomorphism as $f : \mathbf{A} \rightarrow \mathbf{B}$.

step 2: This step defines exactly T *atomic sub-regions*, determine the connectivity included in each *atomic sub-region* so that the overall bandwidth distribution is fulfilled.

For PNoW-SW, consider that T optical switches (each with N input ports and N output ports) are used to connect N computing chips. Each *atomic sub-region* contains exactly one optical switch and also one transceiver group from every computing chip. Every transceiver group connects to the optical switch within the same *atomic sub-region*.

For PNoW-DW, this step is able to be solved by an edge coloring algorithm[4]. An edge coloring of a graph is an assignment of "colors" to the edges of the graph so that no two incident edges have the same color; a minimum edge coloring is an edge coloring using the least possible colors. In the proposed method, each "color" corresponds to an *atomic subregion* that implements waveguide connectivity which are represented by edges. Let H be a bipartite multi-graph with 2 times N vertices. Each computing chip is represented by two vertices in H, one in each bipartite vertex set. One bipartite set of vertices represents clusters of transmitter groups in every computing chip (each vertex represents the cluster of all transmitter groups in a computing chip), another bipartite set of vertices represents clusters of receiver groups. Every edge in H represents a unidirectional waveguide connection between two computing chips via their transceiver groups. Graph H is a regular bipartite multi-graph, meaning that all vertices' degree equals to T. It has been proven that graph H has an edge chromatic number equals to T, meaning that it can be edge-colored with T colors [12] in a minimum edge coloring. Due to the homogeneity, each colored edge set in such a minimum edge coloring forms a perfect matching[5] of H. Therefore, each colored edge set can be mapped into the connectivity within one *atomic sub-region* that contains exactly one transceiver group from each computing chip. T colors of edge sets are therefore mapped into exactly T *atomic sub-regions*. This deployment of the edge coloring algorithm is further illustrated in figure 7 with an example where N=3 and T=3.

[4]An edge coloring of a graph is an assignment of "colors" to the edges of the graph so that no two incident edges have the same color; a minimum edge coloring is an edge coloring using the least possible colors.

[5]A matching in a undirected graph is a set of edges without common vertices; a perfect matching is a matching that covers every vertex of the graph.

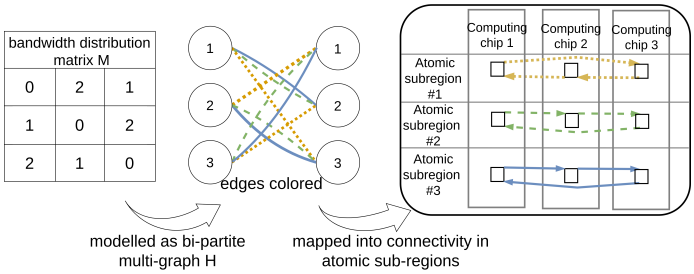

Fig. 7. A small example for the deployment of a minimum edge coloring algorithm to determine the connectivity in each atomic sub-region.

step 3: Determine the logical shapes of the *atomic sub-regions* on plane **B**.

Consider that every T transceiver groups under a computing chip is placed in a $\sqrt{T} - by - \sqrt{T}$ grid and are labeled by their (row, column)-ids in the corresponding grids. Define the logical shape of an *atomic sub-region* such that it contains and only contains transceiver groups with the same label from each computing chip. The shapes of T *atomic sub-regions* are defined one after another, leaving no gaps in between, until all T *atomic sub-regions* are defined. This can be easily generalized from the result of **step 3** in figure 4 as an example.

step 4: Define another homeomorphism $g : \mathbf{B} \rightarrow \mathbf{C}$, mapping from plane **B** to plane **C**. The boundaries of *atomic sub-regions* are straight in plane **C**, which makes routing waveguides in each *atomic sub-region* easy. Within each *atomic sub-region*, determine the logical shapes of optical waveguides such that the connectivity in each *atomic sub-region* (result of **step 2**) is fulfilled while respecting **principle 1**.

For PNoW-SW, the zero-crossing waveguide routing in every *atomic sub-region* can easily be generalized from the result of **step 4** in figure 6 as an example. Therefore, this step is trivial for PNoW-SW.

For PNoW-DW, the connectivity in one *atomic sub-region* must be one or multiple cycles. This can be easily proven by letting graph g be a simple directed graph, each vertex in g represents one transceiver group in the atomic sub-region, each arc represents a directional optical waveguide connection in the atomic sub-region. Graph g must be composed by one or multiple cycles because its vertices' ingress and egress degrees equal to 1, therefore, the connectivity in one *atomic sub-region* must be one or multiple cycles. We deploy the following technique for routing waveguides in one *atomic sub-region* without any crossings:

Let I be the number of cycles of connectivity in an *atomic sub-region*. The transceiver groups in a cycle $i \in \{1, 2, ..., I\}$ is placed at $y = i$, given a fictive y-axis. The two boundaries of the *atomic sub-region* are at $y = 0$ and $y = I + 1$, respectively. The idea of the fictive y-axis is shown in figure 8 with an example. Route waveguides in each connectivity cycle following two rules:

Rule 1: Waveguides in cycle i are routed within the area

where $i-1<y<i+1$.

Rule 2: When routing waveguides in a connectivity cycle i, start from the transceiver group in the computing chip that has the smallest ID (the left-most computing chip in figure 8). Route waveguides one after another following the sequence of the connectivity cycle, avoiding any intersection with already-routed waveguides. When routing a waveguide, route above any transceiver group that has not been connected by any waveguide, while route below any transceiver group that has already been connected.

An example routing that follows these two rules is shown in figure 8(a). The waveguides in connectivity cycle i=2 has been labeled from no.1 to no.5 (labels near the arrows) which is the sequence that they have been routed. For example, when routing waveguide no.2 in cycle i=2, it need to be routed above the transceiver group in computing chip 6 because it has not yet been connected. Rule 1 and 2 ensures two properties of the resulting waveguide routing: First of all, there is no waveguide crossings. Secondly, each waveguide passes by every computing chip at most once. The second property further contributes to the upper bound of the worst-case inter-chip bandwidth as will be discussed in section III-D.

This routing result is then translated into the waveguide routing results on plane **C**, without the fictive y-axis.

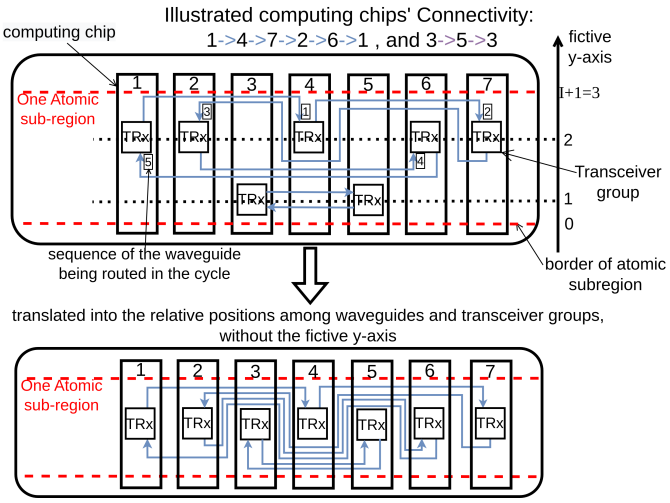

Fig. 8. An example routing that follows the two rules, which leads to the desired properties.

step 5: Map the logical waveguide routing obtained from the previous steps onto physical routes of optical waveguides on the wafer plane. Because plane **A** is homeomorphic to plane **B**, and plane **B** is homeomorphic to plane **C**, plane **A** is homeomorphic to plane **C** with the homeomorphism being the composition of f and g, $f \circ g : \mathbf{A} \rightarrow \mathbf{C}$. Therefore, every logical waveguide routing, which is a curve on plane **C**, can be mapped onto a curve on plane **A** with mapping $(f \circ g)^{-1}$.

In this method, the first homeomorphic mapping $f : \mathbf{A} \rightarrow \mathbf{B}$ simplifies the determination of the shapes of *atomic sub-regions* on plane **B**. The second homeomorphic mapping $f : \mathbf{B} \rightarrow \mathbf{C}$ makes it straight-forward to determine the shapes of waveguides on plane **C** with the proposed two routing rules in **step 4**. This approach leverages the fact that homeomorphic planes preserve the relative positions of curves and points, thereby achieving an elegant zero-crossing routing solution.

D. Worst-case scenario and further optimization

The proposed method induces detours in the routes of optical waveguides for avoiding waveguide crossings. The solution logically determines the relative positioning among the transceiver groups, the optical waveguides and possible optical switches. As a result, there might be a bundle of optical waveguides passing through any two adjacent transceiver groups in the same computing chip. However, it becomes infeasible if the minimal cross-section width of this bundle of optical waveguides is larger than the spacing between the two adjacent transceiver groups, which poses a limitation on the maximum inter-chip bandwidth of computing chips.

For PNoW-DW, the proposed method ensures that every waveguide passes by each transceiver group in the *atomic sub-region* at most once, therefore the worst case in one *atomic sub-region* happens when all optical waveguides pass through a transceiver group exactly once. Because there are at most N optical waveguides in one *atomic sub-region*, this worst-case scenario leads to N number of waveguides passing through the *atomic sub-region* in parallel.

For PNoW-SW, the proposed method also ensures that every waveguide passes by each transceiver group in the *atomic sub-region* at most once. The most crowded area in an *atomic sub-region* is near the optical switches. By logically placing the optical switches in between computing chip N/2 and N/2+1 in the logical row, it leads to $2*N/2 = N$ number of waveguides passing through the *atomic sub-region* near the optical switch.

Consider the neat-floor arrangement in figure 9, where every T transceiver groups under a computing chip are placed in a $\lceil\sqrt{T}\rceil - by - \lceil\sqrt{T}\rceil$ grid. Each horizontal array of $\sqrt{T}$ transceiver groups in one computing chip is therefore included in $\lceil\sqrt{T}\rceil$ *atomic sub-regions*. There are at most $(1+\lceil\sqrt{T}\rceil)$ *atomic sub-regions* passing by two adjacent transceiver groups. As a consequence, there are at most $N(1+\lceil\sqrt{T}\rceil)$ optical waveguides passing between two adjacent transceiver groups, for both PNoW-DW and PNoW-SW.

Assume transceiver groups are interposed strictly underneath their associated computing chips, and the distance between any two adjacent transceiver groups is equal, the distance between two adjacent transceiver groups is approximately $\frac{L_{chip}}{\lceil\sqrt{T}\rceil} - l$. L_{chip} is the length along one side of a rectangular computing chip, l is the the length along one side of a rectangular transceiver group package. l is subtracted in the formula because we assume that an optical waveguide cannot be routed through the transceiver group since there are also passive optical modules (e.g., couplers and WDM (de-)multiplexers) in transceiver groups.

We obtain the following inequality by letting the worst-case waveguide bundle cross-section width smaller or equal to the distance between two adjacent transceiver groups:

$$\frac{L_{chip}}{\lceil\sqrt{T}\rceil} - l \geq (1+\lceil\sqrt{T}\rceil)N(\tau+w) \tag{1}$$

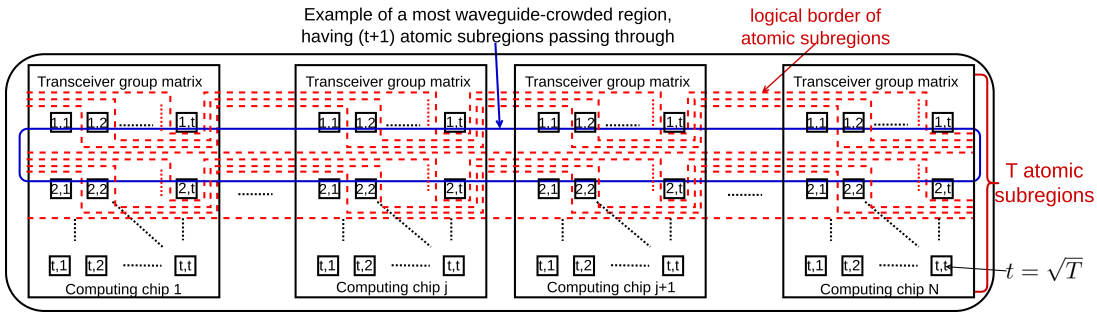

Fig. 9. Illustrating the most crowded area which may lead to the inter-chip bandwidth upper-bound.

In formula 1, τ is the minimum distance between two adjacent waveguides. w is the width of an optical waveguide. Therefore, given L_{chip}, l, N, τ and w, there is an upper bound on T.

Alternatively for PNoW-DW, the *atomic sub-region* may be bounded in ring-shapes, meaning that the *atomic sub-regions* correspond to bounded regions that are placed along a closed-loop structure. Similar to the unbounded structures of *atomic sub-regions* in figure 4 and figure 6, the bounded *atomic sub-regions* are also placed next to each other, however, the bounded regions are not only bounded by neighboring regions, but they are entirely bounded in all directions by definition. As such, an *atomic sub-region* may translate to a row of transceiver groups wherein a first and a last transceiver group of the row are connected via the bounded region, thereby closing the *atomic sub-region* and creating a ring-like shape. For PNoW-DW, adopting a bounded *atomic sub-region* is possible to reduce the number of waveguides passing between two adjacent transceiver groups by half in many cases. A comparison between such bounded *atomic sub-region* and unbounded *atomic sub-region* is illustrated in figure 10.

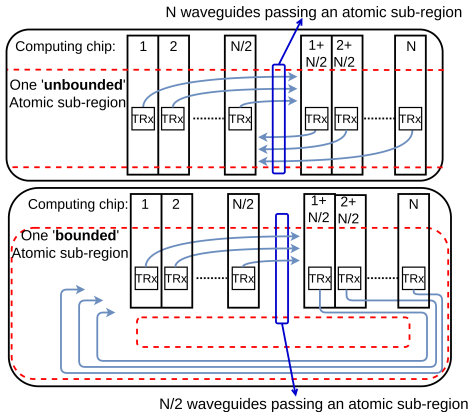

Fig. 10. Illustrating the difference between bounded and unbounded *atomic sub-regions*, in which the bounded *atomic sub-region* could reduce the crowded waveguide bundle.

IV. Zero-crossing optical waveguide routing for heterogeneous PNoW

This section considers heterogeneous PNoW and the generalization of our proposed zero-crossing optical waveguide routing method. Heterogeneity means that the computing chips in a PNoW own different numbers of transceiver groups. Denote T as the maximum number of transceiver groups among all computing chips in a PNoW; thus, there is at least one computing chip owning T transceiver groups. In other words, for heterogeneous PNoW-DW, the summation of a row or a column in the bandwidth distribution matrix M is at most T; and for heterogeneous PNoW-SW, not every optical switch connects to all computing chips, but one optical switch connects to each computing chip with at most one pair of input and output ports.

A. Pitfall of the edge coloring embeddings

Considering heterogeneous PNoW-DW, Directly applying the proposed method means that the vertices in the bipartite graph H in **step 2** might have different degrees. Recall that the two bipartite sets of vertices in H represent computing chips as clusters of transmitter groups and clusters of receiver groups, respectively. Because every transceiver group contains exactly one transmitter group and one receiver group, vertices in H that corresponds to the same computing chip has the same degree.

In mathematical terms, let the two bipartite vertex sets of the bipartite graph H be $X = \{X_1, x_2, ..., x_N\}$ and $Y = \{y_1, y_2, ..., y_N\}$, each vertex x_i in X is bijectively mapped to vertex y_i in Y, and the degree of vertex x_i equals to the degree of the corresponding vertex y_i. The maximum degree of vertices equals to T.

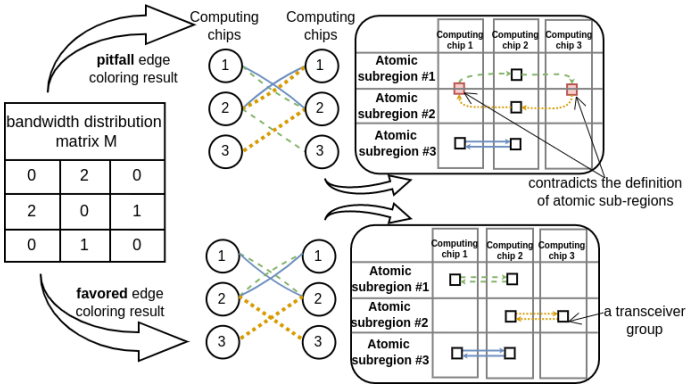

Fig. 11. An example of a pitfall edge coloring and a favored edge coloring for heterogeneous PNoW-DW. As a result of the pitfall edge coloring, there are transceiver groups that are contained in two *atomic sub-regions*.

Let C be a minimum edge coloring of graph H. It is possible that a colored edge set in C is incident on a vertex x_i but not incident on the corresponding vertex y_i, given H is not a regular graph. Therefore, such edge coloring C requires some transceiver groups to be contained in two *atomic sub-region*, which contradicts with the definition of an *atomic sub-region*: one transceiver group is contained in exactly one *atomic sub-region*. An example of such a pitfall is illustrated in figure 11 (top). In contrast, a favored minimum edge coloring is also shown in figure 11 (bottom), where each colored edge set is always incident on two bijectively mapped vertices at the same time.

We now prove that such a favored edge coloring of bipartite graph H can always be found, the corresponding method is in the proof itself.

Theorem 1: There exists at least one minimum edge coloring of H, such that if a colored edge set is incident upon vertex x_i, it is also incident upon the corresponding vertex y_i.

Proof:

Consider the following procedure to construct a T-regular bipartite graph H^* from H:

Procedure 1:
Input: H
pseudo-code:
$H^* \leftarrow H$
For each vertex x_i in set X:
if $degree(x_i) < T$:
Add $(T - degree(x_i))$ parallel edges (x_i, y_i) to H^*
Output: Bipartite multi-graph H^*

Procedure 1 ensures that H^* is a T-regular bipartite graph sharing the same vertex set of H. Denote C* as an arbitrary minimum edge coloring of Graph H^*, C* comprises exactly T colored edge sets. Each colored edge set in C* forms a perfect matching, meaning that it is incident upon all vertices in H^*. Therefore, C* fulfills the property that, if a colored edge set is incident on vertex x_i, it is also incident on the corresponding vertex y_i. Delete all edges that were added during the **procedure 1** from C*, resulting in another collection of colored edge sets C that also contains exactly T colored edge sets. Because the union of all colored edge sets in C forms exactly the edge set of H, C is a minimum edge coloring of Graph H. Because every edge in set $E(H^*) - E(H)$ is incident on two bijectively mapped vertices x_i and y_i, the edge coloring C obtained by removing these edges from C* keeps the property: if a colored edge set in C is incident on vertex x_i, it is also incident upon vertex y_i. Q.E.D.

The proof for Theorem 1 also indicates an algorithm for finding such edge coloring for any heterogeneous PNoW: Add necessary dummy edges between bijectively mapped vertex pairs to the bipartite graph H until it forms a T-regular graph, then perform an arbitrary exact edge coloring algorithm on the T-regular graph to obtain a minimum edge coloring. Removing the added dummy edges from the minimum edge coloring of the T-regular graph leads to an edge coloring of the original graph, such that if a colored edge set is incident on vertex x_i, it is also incident upon vertex y_i. Such an edge coloring algorithm is illustrated in figure 12 with a small example.

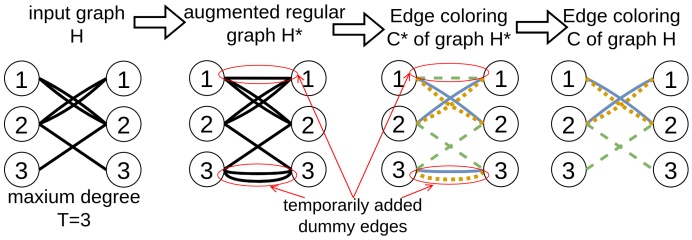

Fig. 12. An example of the edge coloring algorithm, in which every colored edge set is always incident on two bijectively mapped vertices at the same time.

B. Correctly generalized methods for heterogeneous PNoW

Now we describe the complete generalized zero-crossing optical waveguide routing method for heterogeneous PNoW:

For heterogeneous PNoW-DW, increment the diagonal values in the input bandwidth distribution matrix M such that the summation of every row or column equals to T. When modeled by the bipartite multi-graph H, these augmented diagonal values in M is represented by dummy edges that incidents on pairs of bijectively mapped vertices. This augmentation can also be interpreted as dummy transceiver groups with self-loop connectivity, such that it can be seen as a homogeneous PNoW waveguide routing problem. After applying the proposed homogeneous PNoW zero-crossing optical waveguide routing in section III, delete all dummy transceiver groups and their self-loop connectivity from the solution. We have now obtained a solution for the original heterogeneous PNoW zero-crossing optical waveguide routing problem. Furthermore, such heterogeneous solution obviously complies with inequality 1, since the most crowded area in this method can not be worse than the worst-case analysis for the homogeneous PNoW. Figure 13 illustrates the zero-crossing optical waveguide routing method for heterogeneous PNoW with a small example.

For heterogeneous PNoW-SW, the proposed zero-crossing optical waveguide routing method for homogeneous PNoW can be generalized to heterogeneous PNoW similarly. Moreover, the pitfall (discussed in subsection IV-A) in heterogeneous PNoW-SW also lies in the optical switch reconfiguration problem, although we have only discussed on homogeneous PNoW in our previous work [12].

In the optical switch reconfiguration problem, the desired bandwidth distribution matrix M is also represented by the bipartite multi-graph H. A minimum edge coloring of H is then translated into the configuration of T optical switches. The pitfall might occur when an edge coloring contains a colored edge set that is incident only on one of the bijectively mapped vertices in H. Such pitfall edge coloring also means that it cannot be mapped exactly to the configurations of the switches provided in the PNoW. The edge coloring algorithm described in IV-A can also be applied for the optical switch reconfiguration problem to avoid such pitfalls. In additional to any exact edge coloring algorithm for bipartite multi-graph, the edge coloring algorithm for heterogeneous PNoW

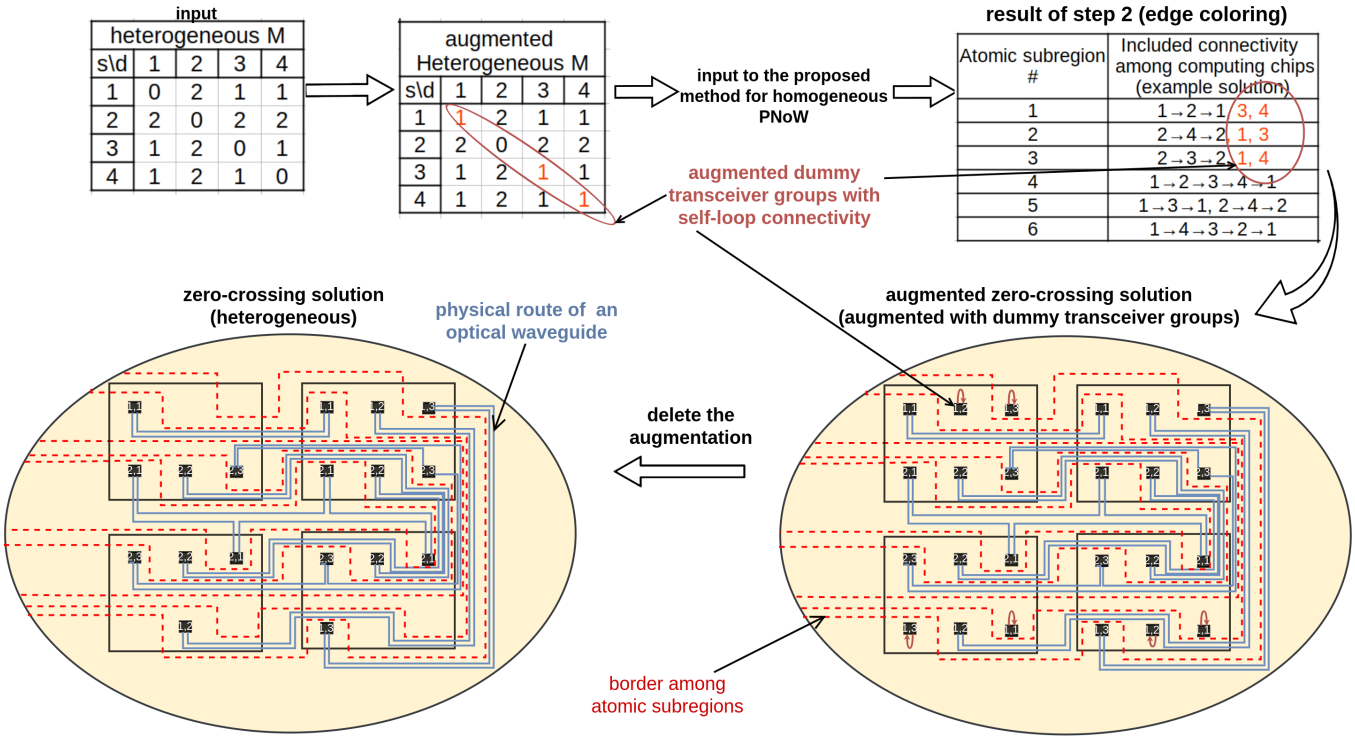

Fig. 13. An example of solving a heterogeneous PNoW zero-crossing optical waveguide routing problem.

only involves adding and removing $O(N*T)$ dummy edges, which is unlikely to become significant overhead for the edge coloring algorithm.

V. Method Evaluation

Here we evaluate the proposed zero-crossing optical waveguide routing method. We consider realistic parameters from the current technology, as summarized in table I. Programs for calculating and plotting the data in this subsection can be found in the git-hub repository [27].

TABLE I
PARAMETERS FROM AVAILABLE SOURCE

Parameter (notation)	Value [unit]	Source of data
wafer diameter (D)	0.3 [m]	[28]
Optical waveguide propagation loss (α_p)	-5 [dB/m]	[24]
Computing chip size ($L_{chip} * L_{chip}$)	$0.03[m] * 0.03[m]$	[29]
Optical waveguide minimum spacing (τ)	$10[\mu m]$	[24]*
Optical waveguide width (w)	$1[\mu m]$	[24]
Number of wavelengths (W) per transceiver group	16	[30], [31]
Data rate (d) per wavelength	$32[Gbps] = 4[GBps]$	[31]
Size of each transceiver group ($l * l$)	$1.6[mm] * 1.6[mm]$**	[32], [33],

*This is a reasonable estimation due to the high confinement factor and thin waveguide width demonstrated in this literature.
**This a reasonable estimation due to the CMOS chip footprints reported in these literature

A. Upper-bound on inter-chip bandwidth

In this sub-section we calculate the upper-bound on inter-chip bandwidth for homogeneous PNoW. The calculated upper-bound will also be admitted by heterogeneous PNoW as discussed in subsection IV-B. The bidirectional inter-chip bandwidth Θ owned by each computing chip can be calculated by the following formula 2:

$$\Theta = T * W * d \tag{2}$$

T is the number of transceiver groups owned by each computing chip, W is the number of wavelength channels in each transceiver group, d is the data rate that each wavelength operates at. Given W and d, the upper bound on Θ comes from the upper bound on T which is due to the finite waveguide thickness and spacings discussed in section III-D. By combining formula 1 and 2, we obtain the following upper bound on Θ which is approximately inversely proportional to the number of computing chips N:

$$\Theta \leq \left(\frac{-b + \sqrt{b^2 + 4L_{chip}N(tau + w)}}{2N(\tau + w)}\right)^2 * Wd, \qquad b = l + N(\tau + w) \tag{3}$$

Figure 14 shows the calculation results of formula 3 by substituting the realistic parameters from table I. We consider the maximum number of computing chips in a PNoW to be 64 due to the realistic geometric parameters considered in table I. Our proposed method is able to achieve $\Theta = 9TBps$ when $N = 16$ and $\Theta = 2.5TBps$ when $N = 64$ for both PNoW-DW and PNoW-SW. This means that, even considering the worst cases, the proposed method could achieve one

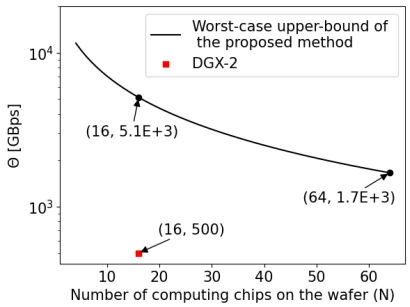

Fig. 14. Upper-bound on Inter-chip bidirectional bandwidth imposed by the proposed method varying with the number of computing chip N.

magnitude higher inter-chip bandwidth compared to the state-of-art wafer-scale multi-chip computing systems such as DGX-2. Because we only calculated the upper-bound worst-case inter-chip bandwidth for general PNoW-DW and PNoW-SW, it is possible to achieve even higher inter-chip bandwidth for specific network designs with or without certain adaptations.

B. Comparative analysis

In this subsection we compare our proposed zero-crossing optical waveguide routing solution with other solutions that may contain optical waveguide crossings.

We focus this comparative analysis on $N = 16$ computing chips with inter-chip bandwidth $\Theta \leq 4TBps$, since this is the bandwidth-sensitive region observed in [3] for several multi-GPU applications. This region is feasible for our proposed method because it is under the upper-bound calculated in sub-section V-A. For comparing with the proposed zero-crossing waveguide routing method, we adopt the 'escape routing' methodology [18] originated for the electronic wire routing heuristics, as introduced in section II.

In the context of the optical waveguide routing problem, an 'escape router' takes the following two steps to route the waveguides: The first step comprises routing waveguides from transceiver groups to the boundary of the corresponding computing chips. Taking the realistic parameters in table I, this first step is trivial due to the relatively small feature size of optical waveguides ($\approx 10\mu m$) compared to the spacing between adjacent transceiver groups under a computing chip ($\geq 3.7mm$ [6]). The second step of the escape router comprises path-finding heuristics to complete the routing of waveguides for achieving the desired connectivity among the computing chips and possible optical switches. The waveguides can no longer be routed underneath any computing chip in the second step. Examples of escape routing for PNoW-DW are illustrated in figure 15, we now further define the two escape routers:

The "escape-straight" router naively routes waveguides as straight lines on the wafer plane after escaping them to the boundary of the computing chips. An example result is shown in figure 15 (a).

The "escape-planarization" router routes waveguides by using a graph-planarization heuristic after escaping them to the boundary of the computing chips. Such a graph-planarization heuristic models every computing chip as a vertex, and every waveguide as an edge in a graph. It then attempts to find a graph layout that minimizes the total number of edge intersections which represents waveguide crossings. An example result is shown in figure 15 (b). In this work, the graph-planarization heuristic has been implemented as a 'Subgraph Planarizer' algorithm from the OGDF c++ library [34].

Figure 16(a) compares the average number of crossings per waveguide achieved by the two escape routers and our proposed zero-crossing method, applied on PNoW-DW. It can be observed that the escape-planarization router achieves about 20 crossings per waveguide for $\Theta \approx 1TBps$ and about 75 crossings per waveguide for $\Theta \approx 4TBps$, which is 50% less crossings per waveguide compared to the escape-straight router. Figure 16(c) compares the average number of crossings per waveguide achieved by the two escape routers and our proposed zero-crossing method, applied on PNoW-SW. The escape-planarization router achieves about 50 crossings per waveguide for $\Theta \approx 1TBps$ and about 200 crossings per waveguide for $\Theta \approx 4TBps$, which is 50% less crossings per waveguide compared to the escape-straight router.

When the escape routers apply on PNoW-DW, the signal crosstalk is dominated by that from the waveguide crossings, while the signal loss mainly comprises the propagation loss along the waveguide and the insertion loss at the waveguide crossings. Denote N_c as the number of crossings on a waveguide, and denote L_{wg} as the waveguide length. With a certain approximations, formula 4 summarizes the calculations for the total optical signal attenuation $\alpha[dB]$ from the transmitter to the receiver; and the output optical noise power N_o.

$$\begin{aligned} \alpha &= N_c\alpha_x + L_{wg}\alpha_p \\ N_o &\approx (P_i * 10^{-SNR_i/10} + N_c P_i * 10^{X_x/10}) * 10^{\alpha/10} \end{aligned} \tag{4}$$

We further estimate the resulting Signal to Noise Ratio (SNR) of the escape routers for PNoW-DW. Assume that the SNR from the transmitter is $SNR_i = 30dB$. Every waveguide crossing could achieve $X_x = -37dB$ crosstalk and $\alpha_x = -0.02dB$ loss [23]. Denote the input optical signal power from the transmitter and the output optical signal power at the receiver as P_i and P_o, respectively. The output SNR_o at the receiver is calculated as in formula 5.

$$SNR_o = 10 * \log_{10}\frac{P_o}{N_o} = 10 * \log_{10}\frac{10^{\alpha/10} * P_i}{N_o}\ [dB] \tag{5}$$

Figure 16 (b) illustrates the estimated SNR_o at the receiver obtained from formulas 5 and 4 for PNoW-DW. It can be observed that the planarization technique is able to improve roughly 3dB SNR at the receivers compared to the naive straight-waveguide solution, by trying to minimize the total number of crossings. However, the estimated output SNR resulting form the escape-planarization router still drops to 23dB for $\Theta \approx 1TBps$, and 18dB for $\Theta \approx 4TBps$ at the receiver.

For PNoW-SW, it is more complicated to estimate SNR because the optical switches also induce significant amount of signal crosstalk and insertion losses. We leave that out in this paper because it need more detailed modeling in the physical layer, which falls out of the scope of this paper.

[6]Deducted from formula 1 and 2, given $\Theta \leq 4TBps$.

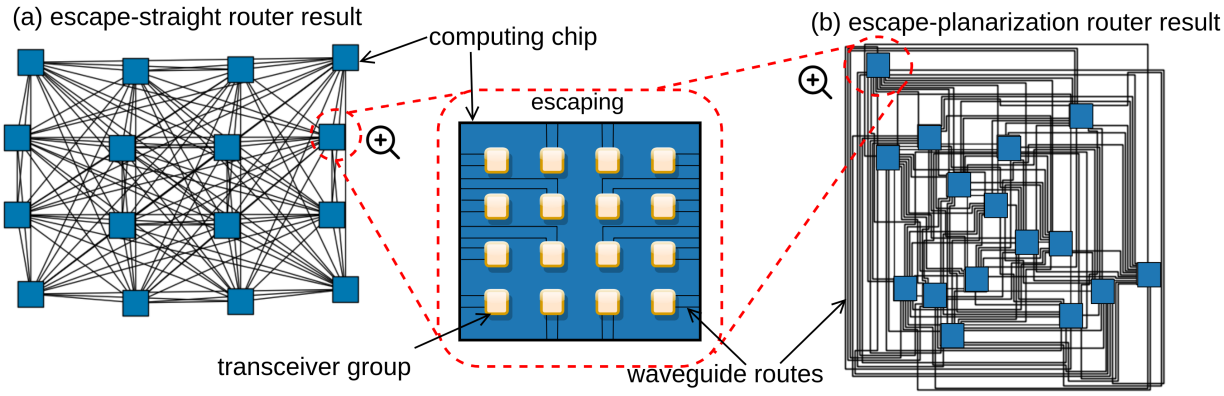

Fig. 15. Examples of routing results of escape routers for PNoW-DW: (a) escape-straight router, and (b) escape-planarization router

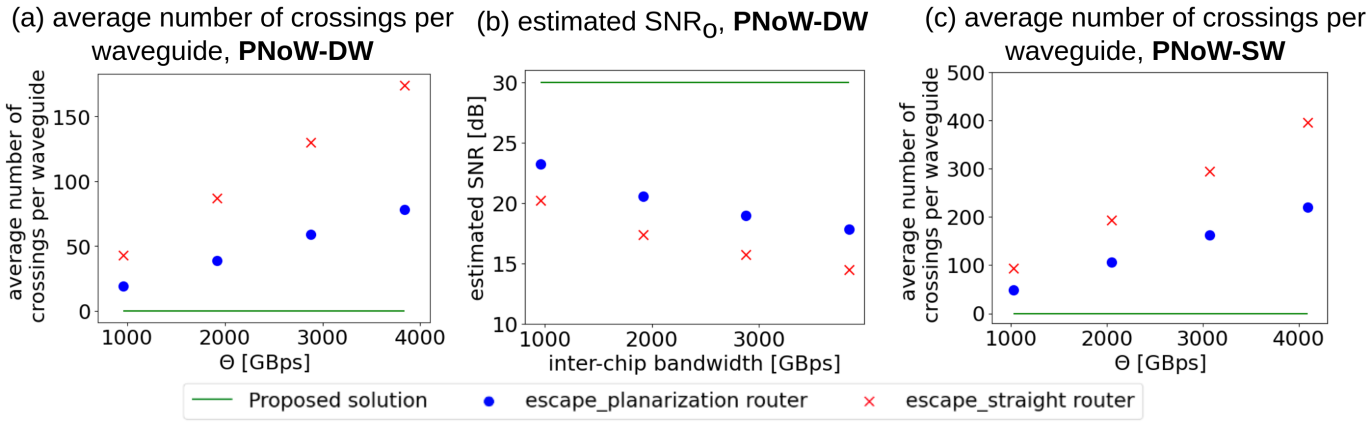

Fig. 16. Comparative analysis between the proposed zero-crossing solution and two escape routers for PNoW-DW

Nonetheless, this comparative analysis clearly shows that the escape routers are not good enough for solving the optical waveguide routing problem in such PNoW, as the low SNR value makes it difficult to support high data-rate loss-less inter-chip communication. This again shows the advantage of our proposed zero-crossing routing method.

In comparison to the escape routers, our proposed method avoids waveguide crossings by routing optical waveguides in detours underneath the computing chips. The propagation loss due to these detours is limited. For instance, in PNoW-DW with 16 computing chips, the average waveguide length is $0.3m$ in our proposed method, leading to only $1dB$ additional propagation loss compared to the escape-straight router, while the insertion loss induced by waveguide crossings from the escape routers are much more dominating when the Θ is high.

VI. Conclusion

In this paper we describe a generic optical waveguide routing method for multi-chip PNoW, that is able to provide an optical waveguide routing solution on a 2D wafer plane without any waveguide crossings.

Heuristic methods from the electronic wire routing literature, such as escape routers, might lead to very high number of waveguide crossings and therefore low SNR at the receiver for our considered PNoWs. Our proposed method, in contrast, leads to very high SNR at the receivers because it provides zero waveguide crossings. Moreover, the inter-chip bandwidth upper-bound is one magnitude higher than the available multi-chip products.

Acknowledgments

The first author is supported by a PhD grant for strategic basic research of the Research Foundation – Flanders (FWO-V, 1S11323N). Part of this research is funded by Ghent University (01G01421).

References

[1] David Schor. A look at nvidia's nvlink interconnect and the nvswitch, 2018. Accessed on: 12-2023.
[2] Akshat Verma. Top graphics card manufacturers and brands for nvidia and amd gpus, 2024. Accessed on: 12-2024.
[3] Shiqing Zhang et al. Photonic network-on-wafer for multichiplet gpus. *IEEE Micro*, 43(2):86–95, 2023.
[4] Yoojin Ban et al. Highly optimized o-band si ring modulators for low-power hybrid cmos-sipho transceivers. In *2023 Optical Fiber Communications Conference and Exhibition (OFC)*, pages 1–3. IEEE, 2023.
[5] Mehrdad Khani et al. Sip-ml: high-bandwidth optical network interconnects for machine learning training. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, pages 657–675, 2021.
[6] Light Matter. https://lightmatter.co/products/passage/. Accessed: February 13, 2024.

[7] Gunther Roelkens et al. Micro-transfer printing for heterogeneous si photonic integrated circuits. *IEEE Journal of Selected Topics in Quantum Electronics*, 29(3: Photon. Elec. Co-Inte. and Adv. Trans. Print.):1–14, 2022.
[8] Gunther Roelkens et al. Present and future of micro-transfer printing for heterogeneous photonic integrated circuits. *APL Photonics*, 9(1), 2024.
[9] Shiqing Zhang et al. Sac: Sharing-aware caching in multi-chip gpus. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, pages 1–13, 2023.
[10] Xia Zhao et al. Nuba: Non-uniform bandwidth gpus. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pages 544–559, 2023.
[11] Jeroen Mampaey. Memory page allocation in multi-chip-module gpus.
[12] Ziyue Zhang et al. On the network design and control of an optical network: interconnecting multiple chips on a wafer. *Journal of Optical Communications and Networking*, 15(2):119–132, 2023.
[13] Shiqing Zhang et al. Balancing performance against cost and sustainability in multi-chip-module gpus. *IEEE Computer Architecture Letters*, 2023.
[14] Jerry C Whitaker et al. *The electronics handbook*. Crc Press, 2018.
[15] Tan Yan. *Algorithmic studies on PCB routing*. University of Illinois at Urbana-Champaign, 2010.
[16] L. W. Ritchey. Busses: What are they and how do they work? *Printed Circuit Design Magazine*, December 2000.
[17] Nadine Abboud et al. Mathematical methods for physical layout of printed circuit boards: an overview. *OR Spectrum*, 30(3):453–468, 2008.
[18] Shih-Ting Lin et al. A complete pcb routing methodology with concurrent hierarchical routing. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 1141–1146. IEEE, 2021.
[19] Youbiao He and Forrest Sheng Bao. Circuit routing using monte carlo tree search and deep neural networks. *arXiv preprint arXiv:2006.13607*, 2020.
[20] Haiyun Li et al. An optimized 3d astar algorithm for multi-layer pcb automatic routing. In *2021 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pages 1–2. IEEE, 2021.
[21] Stijn Sackesyn et al. Experimental realization of integrated photonic reservoir computing for nonlinear fiber distortion compensation. *Optics Express*, 29(20):30991–30997, 2021.
[22] Wim Bogaerts et al. Programmable photonic circuits. *Nature*, 586(7828):207–216, 2020.
[23] Yangjin Ma et al. Ultralow loss single layer submicron silicon waveguide crossing for soi optical interconnect. *Optics express*, 21(24):29374–29382, 2013.
[24] Minh A Tran et al. Ultra-low-loss silicon waveguides for heterogeneously integrated silicon/iii-v photonics. *Applied Sciences*, 8(7):1139, 2018.
[25] Wesley D Sacher et al. Monolithically integrated multilayer silicon nitride-on-silicon waveguide platforms for 3-d photonic circuits and devices. *Proceedings of the IEEE*, 106(12):2232–2245, 2018.
[26] János Pach and Rephael Wenger. Embedding planar graphs at fixed vertex locations. *Graphs and Combinatorics*, 17:717–728, 2001.
[27] Ziyue Zhang. https://github.com/ziyuezzy/zerocrossingpaperexperiments. Accessed: February 13, 2024.
[28] Frédéric Boeuf et al. Silicon photonics r&d and manufacturing on 300-mm wafer platform. *Journal of lightwave technology*, 34(2):286–295, 2015.
[29] NVIDIA Corporation. Nvidia ampere architecture whitepaper, 2020. Accessed on: 12-2024.
[30] Jordan A Davis et al. Silicon photonic chip for 16-channel wavelength division (de-) multiplexing in the o-band. *Optics Express*, 28(16):23620–23627, 2020.
[31] Davide Guermandi et al. Tsv-assisted hybrid finfet cmos—silicon photonics technology for high density optical i/o. In *45th European Conference on Optical Communication (ECOC 2019)*, pages 1–4. IET, 2019.
[32] Marijn Verbeke, Pieter Rombouts, Hannes Ramon, Bart Moeneclaey, Xin Yin, Johan Bauwelinck, and Guy Torfs. A 1.8-pj/b, 12.5–25-gb/s wide range all-digital clock and data recovery circuit. *IEEE Journal of Solid-State Circuits*, 53(2):470–483, 2017.
[33] Michal Rakowski, Marianna Pantouvaki, Peter De Heyn, Peter Verheyen, Mark Ingels, Hongtao Chen, Jeroen De Coster, Guy Lepage, Brad Snyder, Kristin De Meyer, et al. 22.5 a 4× 20gb/s wdm ring-based hybrid cmos silicon photonics transceiver. In *2015 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers*, pages 1–3. IEEE, 2015.
[34] Markus Chimani et al. The open graph drawing framework (ogdf). *Handbook of graph drawing and visualization*, 2011:543–569, 2013.