XRing: A Crosstalk-Aware Synthesis Method for Wavelength-Routed Optical Ring Routers

Zhidan Zheng
Chair of Electronic Design Automation
Technical University of Munich
Munich, Germany
zhidan.zheng@tum.de

Mengchu Li
Chair of Electronic Design Automation
Technical University of Munich
Munich, Germany
mengchu.li@tum.de

Tsun-Ming Tseng
Chair of Electronic Design Automation
Technical University of Munich
Munich, Germany
tsun-ming.tseng@tum.de

Ulf Schlichtmann
Chair of Electronic Design Automation
Technical University of Munich
Munich, Germany
ulf.schlichtmann@tum.de

***Abstract*—Wavelength-routed optical networks-on-chip (WRONoCs) are well-known for supporting high-bandwidth communications with low power and latency. Among all WRONoC routers, optical ring routers have attracted great research interest thanks to their simple structure, which looks like concentric cycles formed by waveguides. Current ring routers are designed manually. When the number of network nodes increases or the position of network nodes changes, it can be difficult to manually determine the optimal design options. Besides, current ring routers face two problems. First, some signal paths in the routers can be very long and suffer high insertion loss; second, to connect the network nodes to off-chip lasers, waveguides in the power distribution network (PDN) have to intersect with the ring waveguides, which causes additional insertion loss and crosstalk noise. In this work, we propose XRing, which is the first design automation method to automatically synthesize optical ring routers based on the number and position of network nodes. In particular, XRing optimizes the waveguide connections between the network nodes with a mathematical modelling method. To reduce insertion loss and crosstalk noise, XRing constructs efficient shortcuts between the network nodes that suffer long signal paths and creates openings on ring waveguides so that the PDN can easily access the network nodes without causing waveguide crossings. The experimental results show that XRing outperforms other WRONoC routers in reducing insertion loss and crosstalk noise. In particular, more than 98% of signals in XRing do not suffer first-order crosstalk noise, which significantly enhances the signal quality.**

***Index Terms*—Wavelength-routed optical networks-on-chip, optical ring router, design automation, crosstalk-aware**

I. Introduction

Enabled by breakthroughs in silicon photonics, optical networks-on-chip (ONoCs) emerge as a promising solution for high-performance multi-core integration [1]. Thanks to the wavelength-division multiplexing technology, multiple signals on different wavelengths can simultaneously travel along a single waveguide in ONoCs [2], [3]. Compared to conventional electrical interconnections, ONoCs provide higher bandwidth with lower latency and power [4], [5]. Among all ONoCs, wavelength-routed ONoCs (WRONoCs) are well-known for reserving collision-free signal paths at design time so that communications between different network nodes can happen simultaneously without wasting energy and time on arbitration.

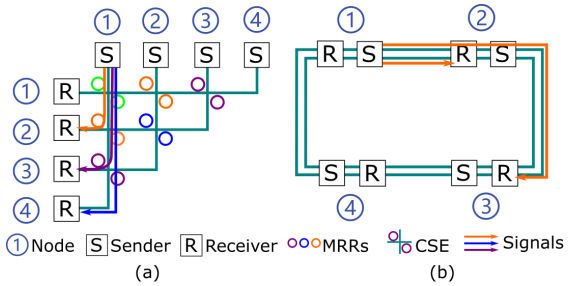

Fig. 1. (a) A 4 × 4 crossbar router (Snake [8]). (b) A 4 × 4 ring router.

Current WRONoC routers can be classified into two types based on their waveguide architecture: crossbar [6]–[9] and ring [10]–[12]. Fig. 1 shows an example of each router type. Specifically, crossbar routers arrange waveguides in a matrix-like structure to interconnect all network nodes, and use microring resonators (MRRs) to switch optical signals between different waveguides, as shown in Fig. 1(a); while ring routers use circular waveguides to sequentially connect all network nodes, and send signals on the same wavelengths along different waveguides, as shown in Fig. 1(b). Compared to crossbar routers, ring routers save MRR-tuning power and avoid the insertion loss and crosstalk noise generated by the MRRs and waveguide crossings [12]. Besides, crossbar routers usually require much placement & routing effort to map their logic topologies onto the physical plane, and the resulting physical designs likely contain many additional waveguide crossings, which further degrade the network performance [13]; on the other hand, ring routers have a simple structure that can be easily projected onto the physical plane without significant changes. Considering these advantages, ring routers are considered an appealing option for WRONoCs.

Currently, waveguide connections between network nodes in ring routers are designed manually. As a result, the network performance depends on the experience of the designers. However, as the number of network nodes increases or the position of network nodes changes, it can be difficult to manually find out the optimal connections. For example, Fig. 2 shows three different options to connect 16 regularly aligned network nodes.

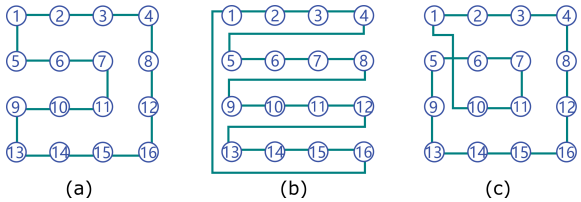

Fig. 2. A ring waveguide with (a) the minimum length and no crossings. (b) a long detour. (c) a waveguide crossing.

Compared to an optimal connection, as shown in Fig. 2(a), a sub-optimal connection, as shown in Fig. 2(b), may contain a long waveguide detour, which results in much propagation loss, or an additional waveguide crossing, as shown in Fig. 2(c), which results in extra crossing loss and crosstalk noise. The connection problem may become more complex when the network nodes are not regularly aligned on the chip. Thus, a design automation method to systematically optimize the waveguide connections in ring routers is necessary.

Besides, current ring routers suffer two problems.

- First, the data transmission paths between certain signals can be quite long, which leads to much propagation loss. For example, as shown in Fig. 2(a), although the nodes 11 and 12 are close to each other on the physical plane, signals sent from 11 need to travel a long distance, via either node 7 or 10 in the clockwise or counter-clockwise order, respectively, to reach node 12. Thus, the worst-case insertion loss of a signal path in ring routers can be quite high.
- Second, to supply the network nodes with laser power from off-chip sources, waveguides in the power distribution network (PDN) have to cross the waveguides in the ring router, which inevitably leads to additional crossing loss and crosstalk noise. For example, as shown in Fig. 3, when the PDN waveguide, coloured in red, wants to reach the sender of a network node at the inner ring waveguide r_1, it has to intersect with the outer ring waveguide r_2. Besides the crossing loss, the intersection will also generate noise signals, which will travel along the ring waveguide and decrease the signal-to-noise ratio (SNR) of other desired signals at the receivers [14].

In this paper, we propose **XRing**, which is the first design automation method to automatically synthesize a ring-based WRONoC router and a crossing-free PDN. XRing is a threefold method: first, we optimize the waveguide connections between the network nodes with a mathematical model; second, we construct shortcuts for signals that suffer long detours; third, we propose a tree-based PDN design and break ring waveguides at selected points to create openings for PDN waveguides to access the senders. We compare XRing to three design tools of crossbar routers: Proton+ [15], PlanarONoC [16], and ToPro [3], and two well acknowledged designs of ring routers:

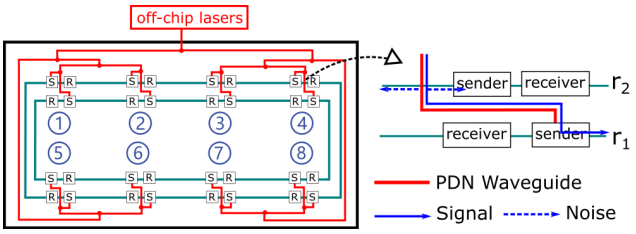

Fig. 3. An 8-node ring router with a PDN.

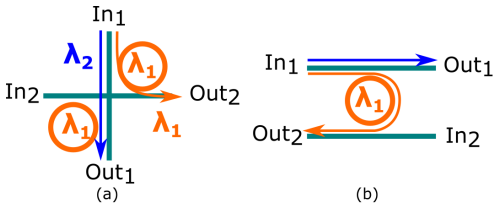

Fig. 4. (a) Crossing switching element. (b) Parallel switching element.

ORNoC [10] and ORing [17]. The experimental results demonstrate the superiority of XRing in reducing power overhead and enhancing signal quality. Compared to the design tools for crossbar routers, XRing decreases the worst-case insertion loss by more than 40%. Compared to ORNoC and ORing, for a 16-node network, XRing reduces the total laser power by more than 10% and increases the worst-case SNR by about 28%. In particular, more than 98% of signals in XRing do not suffer any first-order crosstalk noise.

II. Background

A. Microring Resonators

On WRONoCs, signals are routed based on their wavelengths by microring resonators (MRRs). An MRR consists of a looped waveguide and a coupling mechanism. It can be configured to resonate with certain wavelengths. Fig. 4 shows two optical switching elements (OSEs) constructed with MRRs, which are configured to resonate with wavelength λ_1: Fig. 4(a) shows a crossing switching element (CSE) consisting of MRRs at the intersection of two waveguides, and Fig. 4(b) shows a parallel switching element (PSE) consisting of an MRR between two parallel waveguides. When a signal on λ_1 approaches an OSE, it will be coupled to the MRR and then switched to another waveguide, experiencing a 90° or 180° direction change. In contrast, if a signal on a wavelength other than λ_1 approaches an OSE, it will ignore the MRR and keep its original direction.

B. Performance Factors

Insertion loss and crosstalk noise are two important performance factors for WRONoC routers.

Insertion loss is the loss of signal power during data transmission. In ONoCs, a signal mainly suffer four types of insertion loss: *propagation loss*, which positively relates to the length of waveguides that the signal travels along; *drop loss*, which happens when the signal is coupled to an on-resonance MRR; *through loss*, which happens when the signal passes through an off-resonance MRR; and *crossing loss*, which happens when the signal passes through a waveguide crossing. Besides, photodetectors at the receiver will also cause insertion loss [18]. The total insertion loss of a signal can be calculated as the summation of all these losses. Furthermore, the laser power P of a wavelength λ_x can be calculated as $P^{\lambda_x} = 10^{(il_w^{\lambda_x}+S)/10}$, where $il_w^{\lambda_x}$ and S denote the worst-case insertion loss of signals on λ_x and the receiver sensitivity, respectively [15].

Crosstalk noise refers to noise signals generated at MRRs and waveguide crossings [14]. Specifically, when a signal is coupled to an on-resonance MRR, or when a signal passes through an off-resonance MRR or a waveguide crossing, a portion of the signal power will deviate from the designated transmission direction and become noise, such as the noise shown in Fig. 5(a). Noise signals have the same wavelengths as their original signals and will also travel along waveguides.

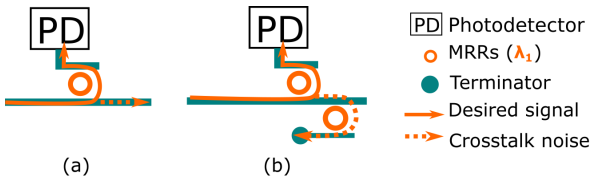

Fig. 5. (a) When the desired signal gets coupled to the MRR at the photodetector, it generates a noise signal. (b) By adding an MRR and a terminator, the noise can be removed.

When being received by photodetectors at receivers, the noise signals will decrease the SNR of the desired signals on the same wavelengths. In practice, we mostly care about noise generated by the original signals, i.e. the first-order noise, but not the noise generated by other noise signals, since their power is relatively small [14]. The SNR of a signal on wavelength λ_x can be calculated as $10\log(\frac{P^{\lambda_x}_{signal}}{P^{\lambda_x}_{noise}})$, where $P^{\lambda_x}_{signal}$ is the power of the desired signal and $P^{\lambda_x}_{noise}$ is the power of noise signals that reach the same photodetector on the same wavelength as the desired signal [9]. To note is that the first-order noise generated by the MRRs at the photodetectors can be removed by adding an additional MRR and a terminator [14], as shown in Fig. 5(b), and will thus not affect the SNR.

III. Methodology

A. Step 1: Ring Waveguide Construction

In a ring router, all nodes are supposed to be sequentially connected by a circular waveguide with the minimum length, and the waveguide segments between different pairs of nodes should not intersect with one another. This can be modelled as a *modified travelling salesman* problem.

Specifically, we model a network with N nodes as a directed graph $G = (V, E)$, where $V = \{v_1, \cdots, v_N\}$ is the set of vertices representing the network nodes, and $E = \{(v_i, v_j), (v_j, v_i) \mid v_i, v_j \in V, v_i \neq v_j\}$ is set of directed edges representing the network connections. Thus, the design requirements are equivalent to the following statement:

Find the shortest path that starts and ends at the same vertex, and visits each vertex exactly once; besides, for any two edges in the path, there is a way to implement the edges as waveguides without crossings.

The former part of this statement can be considered as a travelling salesman problem, and the latter part of this statement can be considered as an additional constraint on this problem. To solve the problem, we construct a mixed integer linear programming (MILP) model: For each vertex $v_i \in V$, we denote the set of its incoming edges as $\mathbb{E}^{v_i}_{\text{in}}$, and the set of its outgoing edges as $\mathbb{E}^{v_i}_{\text{out}}$. For each directed edge $e \in E$, we introduce a binary variable b_e to indicate whether e is selected to construct the ring router, i.e. $b_e = 1$ indicates that e is selected and $b_e = 0$ indicates that e is not selected.

To ensure that each vertex is visited exactly once, i.e. each vertex has exactly one incoming edge and one outgoing edge, we introduce the following constraints:

$$\sum_{e \in \mathbb{E}^{v_i}_{in}} b_e = 1, \quad \sum_{e \in \mathbb{E}^{v_i}_{out}} b_e = 1. \qquad \forall v_i \in V \tag{1}$$

For a network with more than 2 nodes, we want to avoid forming a cycle between two nodes, as shown in Fig. 6(a). Thus, we introduce the following constraints to ensure that

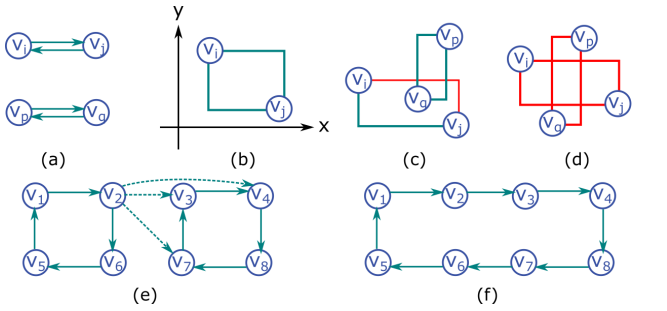

Fig. 6. (a) A cycle between two vertices. (b) Two routing options with Manhattan Distance of the edge (v_i, v_j). (c) One path of the edge (v_i, v_j) does not cross the paths of the edge (v_p, v_q). (d) All paths of both edges are crossed. (e) Non-selected edges from v_2 to the vertices of another cycle. (f) Combining two cycles.

at most one out of the two directed edges between any two vertices can be selected:

$$b_{(v_i,v_j)} + b_{(v_j,v_i)} \leq 1. \qquad \forall v_i, v_j \in V, v_i \neq v_j \tag{2}$$

To ensure that no waveguide crossing will be formed when implementing selected edges, for every two edges, we check their routing options. Specifically, suppose that waveguides are routed either horizontally or vertically, we consider two routing options to implement an edge between two arbitrary vertices v_i and v_j: either we first route in the vertical direction, and then in the horizontal direction, or the other way around, as shown in Fig. 6(b). Thus, there are four options to implement two edges. If there is at least one option to implement the edges without waveguide crossing, as shown in Fig. 6(c), we refer to the edges as *conflict-free*; otherwise, if none of the options can avoid forming a waveguide crossing, as shown in Fig. 6(d), we mark the edges as *conflicting*. For every edge $e \in E$, we construct a set $\mathbb{E}^e_c$ of all its conflicting edges and introduce the following constraints to prevent conflicting edges from being selected together:

$$b_e + b_{e'} \leq 1. \qquad \forall e \in E, e' \in \mathbb{E}^e_c \tag{3}$$

To minimize the total length of the selected edges, we set the optimization objective of our model as follows:

$$Minimize : b_{(v_i,v_j)} * \delta_{(v_i,v_j)} \qquad \forall (v_i, v_j) \in E \tag{4}$$

where $\delta_{(v_i,v_j)}$ is the Manhattan Distance between the network nodes nodes corresponding to v_i and v_j.

To note is that the above model is not a complete formulation of our modified travelling salesman problem, since we have not explicitly modelled the connectivity requirement, i.e. each node should be reachable from every other node. Thus, at the end of the optimization, we may obtain multiple independent sub-cycles, as shown in Fig. 6(e), instead of a path connecting all vertices. However, if we want to model the connectivity requirement with MILP, we will need to introduce an additional constraint to exclude the formation of every possible sub-cycle, which, in the worst case, will result in $\mathcal{O}(2^N)$ additional constraints for a network with N nodes. To improve the efficiency of our method, we propose a heuristic approach to combine the potential sub-cycles to obtain the desired path. Specifically, for every two sub-cycles S_1 and S_2, we find all unordered pairs of conflict-free edges e_1, e_2 with $e_1 \in S_1, e_2 \in S_2$, and select the pairs of edges with minimum lengths to combine the cycles, as shown in Fig. 6(f).

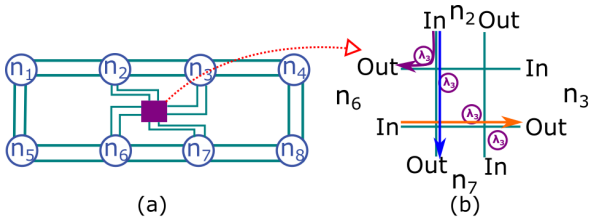

Fig. 7. (a) An 8-node ring router with two feasible shortcuts $p^s_{n_2,n_7}$ and $p^s_{n_3,n_6}$. (b) Two CSEs are formed by two shortcuts.

At last, we route waveguides based on the optimization results so that the length of each waveguide segment between a pair of nodes is equal to the Manhattan distance between the two nodes. By default, we construct two ring waveguides: one for clockwise data transmission denoted as r_1; and the other for counter-clockwise data transmission denoted as r_2. The spacing distance between the two ring waveguides is defined as $A_1 + \lceil \log_2(N) \rceil \times A_2$, where A_1 and A_2 denote the size of a modulator and a splitter, respectively [17]. The way to determine this spacing distance is introduced in Sec. III-D.

B. Step 2: Shortcut Construction

In a classic ring router, a node can transmit signals to another node in either the clockwise or the counter-clockwise direction, depending on the lengths of the signal paths. However, due to the circular waveguide structure, some nodes may suffer long signal paths in both directions, although they may be physically close to each other. To solve this problem, we create shortcuts between certain nodes with waveguides and optionally MRRs. In particular, since senders dedicated to the shortcuts also require power supplied by the PDN, we will only introduce shortcuts when they benefit the network performance and we force that a network node can only have at most one shortcut.

First, we collect potential shortcut options. For every two nodes n_i and n_j, we check whether we can connect their senders and receivers with two additional waveguides without crossing any existing ring waveguides. If so, we consider the shortcut denoted as $p^s_{n_i,n_j}$ to be feasible and calculate its length as the Manhattan distance between the two nodes.

Next, for every two nodes n_i and n_j that have a feasible shortcut, we calculate the lengths of the signal paths on ring waveguides and compare them to the length of the shortcut. For a signal from n_i to n_j, we denote the *clockwise path* and the *counter-clockwise path* on ring waveguides as $p^c_{n_i,n_j}$ and $p^{cc}_{n_i,n_j}$, respectively. We use a gain function $g(n_i, n_j)$ to indicate the gain to map this signal onto its shortcut. This function is formulated as $min(len(p^c_{n_i,n_j}), len(p^{cc}_{n_i,n_j})) - len(p^s_{n_i,n_j})$, where $len(p^c_{n_i,n_j})$, $len(p^{cc}_{n_i,n_j})$, and $len(p^s_{n_i,n_j})$ denote the length of $p^c_{n_i,n_j}$, $p^{cc}_{n_i,n_j}$, and $p^s_{n_i,n_j}$. If the gain value of a signal is negative, we will mark the shortcut as invalid.

According to the gain values of the signals, we sort all shortcuts with positive gains and select the shortcuts with maximum gain values. Besides, we can make use of CSEs to merge shortcuts if two shortcuts form crossings. For example, as shown in Fig. 7(a), $p^s_{n_2,n_7}$ and $p^s_{n_3,n_6}$ form crossings. By replacing the crossings with CSEs, the signals on shortcuts $p^s_{n_2,n_6}$ and $p^s_{n_3,n_7}$ can be transmitted along the shortcuts $p^s_{n_2,n_7}$ and $p^s_{n_3,n_6}$ and directed to their destinations by the MRRs in CSEs. As shown in Fig. 7(b), the signal from n_2 modulated on wavelength λ_3 is coupled to the MRR, which is configured to resonate with λ_3, and reaches n_6. Therefore, the shortcuts $p^s_{n_2,n_6}$ and $p^s_{n_3,n_7}$ are merged with $p^s_{n_2,n_7}$ and $p^s_{n_3,n_6}$. To reduce crosstalk noise, we force that a shortcut can form crossings with at most one other shortcut. For all selected shortcuts, we implement them as waveguides with Manhattan distance.

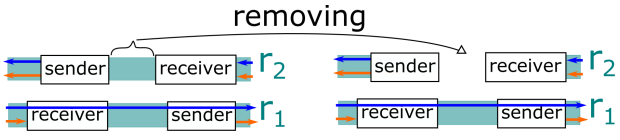

Fig. 8. An opening on ring waveguide r_2.

C. Step 3: Signal Mapping and Ring Waveguide Opening

First, we map the signals that are not supported by shortcuts onto ring waveguides and decide their wavelengths. We apply the methods from [17] to set the maximum wavelength usage ($\#wl$) for each ring waveguide and map signals onto ring waveguides with the shortest length. For example, the signal from n_1 to n_2 shown in Fig. 7(a) has the shortest length of the path when it is clockwise transmitted. To map the signal from n_1 to n_2, we find out all existing clockwise ring waveguides and check their wavelength usages. If we can find an available wavelength for this signal without exceeding the $\#wl$ of a clockwise ring waveguide, we will map this signal onto the ring waveguide and assign the wavelength to the signal. Otherwise, we create a new clockwise ring waveguide and map the signal onto the ring waveguide.

Second, we decide the wavelengths of the signals that are supported by shortcuts. Since waveguides forming the shortcuts will not overlap with the ring waveguides, we can safely use the same set of wavelengths that are assigned to the signals along ring waveguides, and do not need to introduce new wavelengths. For the shortcuts that do not cross other shortcuts, we assign the same wavelengths to the signals, such as λ_1. On the contrary, we assign the signal transmitted along two crossed shortcuts with different wavelengths, such as λ_1 and λ_2, to prevent the noise on the same wavelengths from reaching the receivers to degrade signal quality. For the signals routed by the CSEs formed by two shortcuts, we assign them with wavelengths other than λ_1 and λ_2.

After we determine all signal paths, we count the number of passing signals at each node and find the nodes passed by the least number of signals along each ring waveguide. Those nodes are the candidates when we create an opening on the ring waveguide. If there are multiple candidates, we can pick one of them. For example, as shown in Fig. 8, the node candidate for the ring waveguide r_2 is not passed by any signal, and we open r_2 by removing the waveguide segment between the sender and receiver. Through that opening on r_2, we can route PDN waveguides to reach the senders on r_1 without crossings. If a node candidate is passed by some signals, we will move those signals to other ring waveguides to create an opening. Moving a signal to another ring waveguide should not exceed the $\#wl$ or pass the opening node of the ring waveguide.

D. Step 4: PDN Design

Through the openings on ring waveguides, we route the PDN waveguides to access senders without crossing any ring waveguide. We propose a new PDN design, where the PDN waveguides are routed between a pair of parallel ring waveguides. For example, Fig. 9(a) shows that a pair of ring

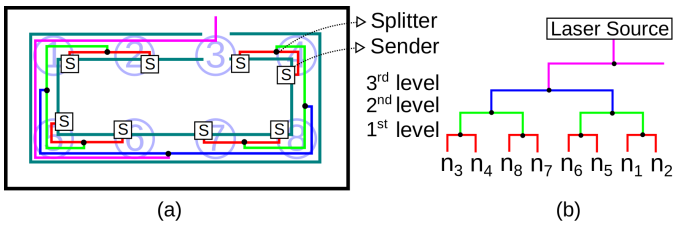

Fig. 9. (a) An example of routing the waveguide of a PDN for the inner ring waveguide. (b) The PDN design for the inner ring waveguide.

waveguides, represented by the turquoise lines, has openings on n_3. We denote two ring waveguides as an *inner* and an *outer* ring waveguide, respectively. Along the inner ring waveguide, each pair of senders is connected to a splitter, represented by the black spot. In order to reserve enough space for routing the PDN waveguides, we set the spacing distance between two ring waveguides as $A_1 + \lceil \log_2(N) \rceil \times A_2$, where N, A_1, and A_2 denote the number of network nodes, the width of a modulator and a splitter.[1]

For each ring waveguide, we model its PDN as a complete binary tree. As shown in Fig. 9(b), all senders along the inner ring waveguide are considered as leaves and the laser source is considered as the root. To prevent the PDN waveguides from blocking the opening of the inner ring waveguide, such as a waveguide connecting n_3's and n_2's senders, we first connect the sender of the opening node on the inner ring waveguide, i.e. n_3's sender, with another sender. By following the signal transmission direction of the inner ring waveguide, we find the closest neighbouring sender of n_3's sender, i.e. n_4's sender, and connect them using a waveguide. We place a splitter at the central point of the waveguide and denote it as the first-level splitter. After connecting all senders along the inner ring waveguide, we use the same method to find the closest neighbouring splitter of the first-level splitter and connect them using a waveguide, such as the green lines shown in Fig. 9(a). We repeat this process until we have only one top splitter for the inner ring waveguide. When we finish routing the PDN waveguides for every ring waveguide, we connect the top splitters of all ring waveguides through their opening nodes without crossing ring waveguides.

IV. Experimental Results

We implemented XRing in C++ and solved our mathematical model using Gurobi [19] as the MILP solver. All experiments are implemented in C++ and carried out on a 2.6 GHz CPU.

A. Comparison to WRONoC Routers Without PDNs

To evaluate the performance of XRing, we compare it to three design tools: Proton+ [15], PlanarONoC [16], and ToPro [3], for the crossbar routers: λ-router [6], GWOR [7], and Light [9]. Besides, we implement the ring routers, ORNoC [10] and ORing [17], for comparison. For ring routers, we try different settings of #wl and pick the one with the minimized the worst-case insertion loss. Since the design tools for crossbar routers have not synthesized the PDNs for their routers, for a fair comparison, we do not perform PDN design for XRing and other ring routers. We compare all routers for two cases: an 8-

[1]Modulators and splitters are optical components for sending optical signals. A modulator converts electric signals into optical signals, and a splitter delivers the power of all wavelengths from laser sources to senders. Typically, a splitter splits an input power into two equal portions, i.e. 50% splitting ratio.

TABLE I
Results for 8-, 16-node WRONoC routers without PDNs

8-node network						
Tool/Method	Router	#wl	il_w	L	C	T
Proton+	λ-router	8	6.6	12.7	27	134
PlanarONoC	λ-router	8	5.2	24.1	7	0.3
ToPro	GWOR	7	4.0	13.5	10	0.17
ORNoC		8	2.6	10.6	0	0.03
ORing		7	2.5	10.0	0	0.03
XRing		7	2.2	7.8	0	0.03

16-node network						
Tool/Method	Router	#wl	il_w	L	C	T
Proton+	λ-router	16	44.0	28.6	255	24425
PlanarONoC	λ-router	16	16.2	89.5	14	0.5
ToPro	Light	15	9.6	44.9	16	0.96
ORNoC		16	9.5	56.3	0	0.23
ORing		16	8.0	46.9	0	0.09
XRing		16	5.6	31.0	0	0.12

#wl: the number of wavelengths. il_w: the worst-case insertion loss value denoted in dB. L: the path length of the signal with maximum insertion loss denoted in mm. C: the number of crossings passed by the signal with maximum insertion loss. T: optimization time denoted in s.

and 16-node network. In each network, a node sends signals to all other nodes except for itself. For all tests, we apply the same node locations and loss parameters as applied in [15].

Table I shows the results of the comparison. In general, XRing outperforms other WRONoC routers in reducing the worst-case insertion loss. For example, for a 16-node network, XRing decreases the worst-case insertion loss by 41% and 30% than Light synthesized by ToPro and the ring router designed by ORing. The reduction in worst-case insertion loss is driven by preventing ring waveguides from crossings and avoiding long detours with shortcuts. XRing decreases the path length of the signal with the worst-case insertion loss by more than 12% compared to other ring routers in a 16-node network. Therefore, signals along ring waveguides in XRing do not suffer crossing loss and much propagation loss.

B. Comparison to ORNoC

In this section, we include the PDNs for ring routers and compare XRing to ORNoC [10]. Since ORNoC has not proposed the method to construct ring waveguides and design PDNs, we implement its algorithm for wavelength assignment, synthesize the ring routers based on our ring waveguide connection results, and perform the same PDN design as applied in [17]. Then, we compare XRing with ORNoC for 8-, 16-, and 32-node networks. For 8-node and 16-node networks, we used the same node locations and die dimension as applied in [20]. For 32-node networks, we extended the node locations and die dimension of the 16-node networks. For both ring routers, we vary the settings of #wl and pick the one with the minimum power and maximum SNR. For all tests, we use the loss and the crosstalk parameters proposed in [17] and [14], respectively.

Table II shows the results in total laser power and SNR for ORNoC and XRing. Generally, XRing requires less laser power than ORNoC, which is achieved by decreasing the worst-case insertion loss. For example, for a 32-node network, XRing decreases the laser power by 64% and the worst-case insertion loss by 32% compared to ORNoC. By limiting the number of shortcuts and preventing the PDN waveguides from crossing any ring waveguide, XRing decreases the number of crossings, which benefits the reduction in both crossing loss and crosstalk

TABLE II
RESULTS OF ORNOC AND XRING FOR 8-, 16-, AND 32-NODE NETWORKS

	The setting for min. power for 8-node networks							
	$\#wl$	il_w^*	L	C	P	$\#s$	SNR_w	T
ORNoC	5	5.15	11.0	13	0.04	35	29.1	0.04
XRing	8	4.33	6.2	0	0.04	0	-	0.03
	The setting for max. SNR for 8-node networks							
	$\#wl$	il_w^*	L	C	P	$\#s$	SNR_w	T
ORNoC	8	5.11	10.6	12	0.06	29	29.3	0.03
XRing	8	4.33	6.2	0	0.04	0	-	0.03

	The setting for min. power and max. SNR for 16-node networks							
	$\#wl$	il_w^*	L	C	P	$\#s$	SNR_w	T
ORNoC	16	6.90	32.0	31	0.82	202	24.3	0.22
XRing	14	4.87	13.6	0	0.46	2	35.9	0.12

	The setting for min. power and max. SNR for 32-node networks							
	$\#wl$	il_w^*	L	C	P	$\#s$	SNR_w	T
ORNoC	32	12.52	116.0	60	79.24	871	16.2	4.45
XRing	31	8.53	137.6	0	29.75	10	35.6	0.98

il_w^*: the worst-case insertion loss without counting the loss happens in a PDN. P: total laser power denoted in W. $\#s$: the number of signals that suffer noise power. SNR_w : the worst-case SNR value among all signals denoted in dB.

TABLE III
RESULTS OF ORING AND XRING FOR A 16-NODE NETWORK

	The setting for min. power							
	$\#wl$	il_w^*	L	C	P	$\#s$	SNR_w	T
ORing	12	5.67	16.3	18	0.51	218	25.6	n/a
XRing	14	4.87	13.6	0	0.46	2	35.9	0.12
	The setting for max. SNR							
	$\#wl$	il_w^*	L	C	P	$\#s$	SNR_w	T
ORing	16	5.73	15.4	14	0.76	209	25.7	n/a
XRing	14	4.87	13.6	0	0.46	2	35.9	0.12

noise. Compared to ORNoC, XRing increases the worst-case SNR by more than a half for all test cases. In particular, more than 98% of signals in XRing do not suffer any first-order noise.

C. Comparison to ORing

In this section, we inclue the PDNs for ring routers and compare XRing to ORing [17]. ORing has manually synthesized the first 16-node ring router with a PDN. For comparison, we implement its 16-node ring router using the same node positions and the loss parameters proposed in [17]. For both ring routers, we determine their best settings of $\#wl$ for minimizing power and maximizing SNR.

According to the results shown in Table III, XRing shows superiority in both energy efficiency and signal quality over ORing. XRing reduces the laser power by 10% and increases the worst-case SNR by 29% compared to ORing. ORing suffers inevitable crossings when the PDN is applied, which implies a great increase in crossing loss and crosstalk noise. In ORing, 87% of signals suffer the first-order noise power, while only 1% of signals in XRing are affected by the first-order noise power. For optimization time, since we manually construct the ring waveguides for ORing, the optimization time depends on the experience of the designer. However, XRing automatically synthesizes the 16-node ring router within one second.

V. CONCLUSION

In this work, we propose XRing, which is the first design automation method to synthesize a ring-based router with a crossing-free PDN. In XRing, ring waveguides are automatically constructed by a mathematical model, which optimizes the length of ring waveguides and avoids the crossings formed by ring waveguides. XRing solves two problems in the current ring routers by constructing shortcuts and breaking ring waveguides. Compared to crossbar routers, XRing outperforms them in decreasing the insertion loss by reducing the number of crossings. Compared to ring routers with PDNs, XRing outperforms them in reducing laser power and enhancing the signal quality. The experimental results show that more than 98% of signals in XRing are not affected by the first-order noise. Moreover, XRing shows superiority in computational efficiency as it can synthesize a router including a PDN within one second.

ACKNOWLEDGMENT

This work is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Project Number 439798838.

REFERENCES

[1] S. Werner et al., "A Survey on Optical Network-on-Chip Architectures," *ACM Comput. Surv.*, vol. 50, no. 6, Dec. 2017.
[2] M. Li et al, "Maximizing the Communication Parallelism for Wavelength-Routed Optical Networks-On-Chips," in *2020 Asia and South Pacific Design Automation Conference*, pp. 109–114.
[3] Z. Zheng et al, "ToPro: A Topology Projector and Waveguide Router for Wavelength-Routed Optical Networks-on-Chip," in *2021 IEEE/ACM International Conference On Computer Aided Design*, pp. 1–9.
[4] Q. Cheng et al., "Recent advances in optical technologies for data centers: a review," *Optica*, vol. 5, no. 11, pp. 1354–1370, 2018.
[5] M. Li et al., "CustomTopo: A Topology Generation Method for Application-Specific Wavelength-Routed Optical NoCs," in *2018 IEEE/ACM International Conference on Computer-Aided Design*, pp. 1–8.
[6] M. Briere et al., "System Level Assessment of an Optical NoC in an MPSoC Platform," in *2007 Design, Automation Test in Europe Conference Exhibition*, pp. 1–6.
[7] X. Tan, et al., "On a Scalable, Non-Blocking Optical Router for Photonic Networks-on-Chip Designs," in *2011 Symposium on Photonics and Optoelectronics*, pp. 1–4.
[8] L. Ramini et al., "Contrasting Wavelength-Routed Optical NoC Topologies for Power-Efficient 3D-stacked Multicore Processors using Physical-Layer Snalysis," in *2013 Design, Automation Test in Europe Conference Exhibition*, pp. 1589–1594.
[9] Z. Zheng et al, "Light: A Scalable and Efficient Wavelength-Routed Optical Networks-On-Chip Topology," in *2021 Asia and South Pacific Design Automation Conference*, pp. 568–573.
[10] S. Le Beux et al., "Optical Ring Network-on-Chip (ORNoC): Architecture and Design Methodology," in *2011 Design, Automation Test in Europe Conference Exhibition*, pp. 1–6.
[11] M. Ortín-Obón et al., "A Tool for Synthesizing Power-Efficient and Custom-Tailored Wavelength-Routed Optical Rings," in *2017 Asia and South Pacific Design Automation Conference*, pp. 300–305.
[12] S. Le Beux et al., "Optical Crossbars on Chip: a comparative study based on worst-case losses," *Concurrency and Computation: Practice and Experience*, vol. 26, no. 15, pp. 2492–2503, 2014.
[13] T.-M. Tseng et al., "Wavelength-Routed Optical NoCs: Design and EDA State of the Art and Future Directions: Invited Paper," in *2019 IEEE/ACM International Conference on Computer-Aided Design*, pp. 1–6.
[14] M. Nikdast et al, "Crosstalk Noise in WDM-Based Optical Networks-on-Chip: A Formal Study and Comparison," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 23, no. 11, pp. 2552–2565, 2015.
[15] A. Beuningen et al, "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*, vol. 12, no. 4, 12 2015.
[16] Y. Chuang et al., "PlanarONoC: Concurrent Placement and Routing Considering Crossing Minimization for Optical Networks-on-Chip," in *2018 ACM/IEEE Design Automation Conference*, pp. 1–6.
[17] M. Ortín-Obón et al., "Contrasting Laser Power Requirements of Wavelength-Routed Optical NoC Topologies Subject to the Floorplanning, Placement, and Routing Constraints of a 3-D-Stacked System," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 25, no. 7, pp. 2081–2094, 2017.
[18] A. Truppel et al, "PSION 2: Optimizing Physical Layout of Wavelength-Routed ONoCs for Laser Power Reduction," in *2020 IEEE/ACM International Conference On Computer Aided Design*.
[19] Gurobi Optimization Inc, *Gurobi Optimizer Reference Manual*. http://www.gurobi.com, 2019.
[20] A. Truppel et al, "PSION+: Combining Logical Topology and Physical Layout Optimization for Wavelength-Routed ONoCs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 12, pp. 5197–5210, 2020.