

ToPro: A Topology Projector and Waveguide Router for Wavelength-Routed Optical Networks-on-Chip

Zhidan Zheng

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

Tsun-Ming Tseng

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

Mengchu Li

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

Ulf Schlichtmann

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

Abstract—To meet the ever-increasing requirements of on-chip communication, the trend is towards wavelength-routed optical networks-on-chip (WRONoCs), which support high-speed communication with low power. A typical WRONoC design flow consists of two consecutive steps: topological design and physical design. Current physical design tools interpret the input topology as a pure logic scheme and perform placement and routing for all network components from scratch. Due to the large design complexity and the layout constraints, additional waveguide crossings in the synthesized layouts are hardly avoidable, which results in an increase in insertion loss and crosstalk noise and thus degrades the network performance. In this work, we propose a physical design tool, *ToPro*, which retains the interconnection among the optical switching elements by projecting the structure of a WRONoC topology onto the physical plane, and focuses on the waveguide routing to the IP-cores. To avoid the increase in insertion loss and crosstalk noise, *ToPro* removes the extra crossings and long detours of waveguides by changing the routing order of nets. The experimental results demonstrate the superiority of *ToPro* in time- and energy-efficiency. For example, compared to a state-of-the-art design automation tool, *ToPro* synthesizes a network with 16 IP-cores with a 17% reduction on the worst-case insertion loss and decreases the synthesis time from more than six days to less than one second.

I. INTRODUCTION

As predicted by Cisco Report [1], the global IP traffic is expected to reach about 400 exabytes per month by 2022. The increased traffic in video services and machine-learning applications is underscoring the need for vast computing and storage resources [2]. The explosive growth is stressing the interconnections on chips. To accommodate these demands, the interconnection networks are required to carry more data with low latency and power consumption. Stimulated by the development of silicon photonics, the trend is towards optical networks-on-chip (ONoCs). Compared to the metallic interconnections in conventional networks-on-chip, ONoCs can support higher bandwidths with lower latency.

There are two categories of ONoCs: *control-networks-based* and *wavelength-routed* ONoCs (WRONoCs) [3]. On control-

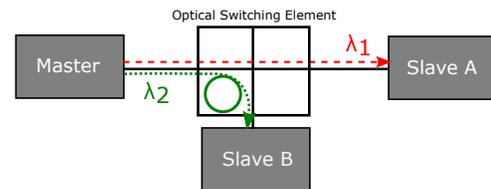


Fig. 1: *On-resonance* and *off-resonance* signals to optical switching elements

networks-based ONoCs, the signal paths for data transmission between a master (sender) and a slave (receiver) are set up dynamically one at a time by a control network. On the other hand, WRONoCs fix collision-free signal paths between all master-slave pairs at design time, and achieve an all-optical interconnection without the energy and latency overhead for arbitration [4].

Both categories of ONoCs apply wavelength-division-multiplexing (WDM), which allows a single waveguide to accommodate multiple optical signals on different wavelengths. Signals that are sent along the same waveguide can be guided to different destinations using optical switching elements (OSEs) formed by microring resonators (MRRs). An MRR consists of a looped optical waveguide and a coupling mechanism. To form an OSE, an MRR is configured to resonate with some specific wavelengths [5], [6]. When a signal gets close to an MRR along a nearby waveguide and the wavelength of the signal is *on-resonance* with the MRR, the signal will be coupled to the looped waveguide and then leave the MRR via another nearby waveguide, thus realizing a direction change in signal propagation.

Fig. 1 shows an OSE implemented using a pair of orthogonally placed waveguide sections and an MRR at the bottom left side. The MRR is configured to only resonate with wavelength λ_2 . When the signal on λ_2 approaches the MRR, it is coupled to the MRR and experiences a 90° direction change. On the

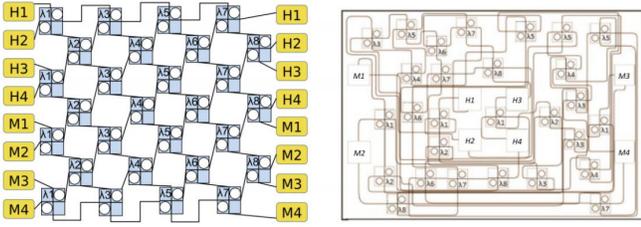


Fig. 2: A Comparison between an 8×8 λ -router topology and its physical layout

other hand, the signal on λ_1 , which is *off-resonance* with the MRR, will just go straight.

Currently, the design of WRONoCs is often carried out in two sequential steps: topological design and physical design. The topological design focuses on the configuration of the MRRs, the wavelength assignment among different signal paths, and the interconnections between network components; and the physical design focuses on the placement of the network components and the routing of waveguides. To date, many efficient WRONoC topologies have been proposed, such as λ -router [7], Snake [8], GWOR [9], and Light [10], and several design automation tools have been developed to automate the physical design process, such as the Proton family of tools [11], [12], PlanarONoC [13] and the PSION-family of tools [14]–[16].

Despite many remarkable progresses, the physical implementation of a WRONoC topology still faces performance and efficiency concerns. For example, Fig. 2 shows an 8×8 λ -router topology [7] and a corresponding layout proposed in [17] to implement the topology on a given physical plane. Compared to the original topology, the positions of the OSEs have been rearranged during physical design to adapt to the layout constraints. The resulting design contains 36 additional waveguide crossings, which generate a significant increase in crossing loss and crosstalk noise and thus degrade the signal quality [18], [19]. Moreover, due to the constantly increasing number of integrated cores, WRONoCs are required to support large-scale networks [20].

We notice that in most WRONoCs topologies, the waveguide interconnections have been optimized in such a manner that there are rarely long detours of waveguides or waveguide crossings outside the OSEs. But since the topological design is usually not concerned with physical constraints, there can be a mismatch in the master/slave positions between the input topology and the actual physical plane. In particular, some WRONoC topologies such as λ -router and Snake arrange masters and slaves in two distant sides of the network, while in realistic WRONoC applications, a network node usually both sends and receives data, and thus the master and the slave representing the same node are located in close spatial proximity on the photonic plane.

Due to the mismatch, state-of-the-art physical design tools interpret an input topology as a pure logic scheme, abandon the optimized waveguide interconnections, and perform placement and routing for all network components from scratch. Con-

sidering the quadratic increase in the number of OSEs and waveguides corresponding to the network size, the physical design complexity is huge even for the small networks [10]. As a result, current physical design methods can hardly approach an optimal solution in a reasonable time. Compared to the input topology, the synthesized physical layout usually contains many additional waveguide crossings or long detours of waveguides, which results in much insertion loss and crosstalk noise.

However, not all WRONoC topologies exhibit the physical mismatch. Some topologies such as GWOR and Light place a pair of master-slave nodes close to each other in the network, and are thus more adaptable to the physical constraints. In this paper, instead of decomposing a WRONoC topology into pure logic components, we propose to take advantage of the optimized internal interconnections in a physical-design-friendly WRONoC topology, so that the design complexity can be significantly reduced and the computational efforts for preventing waveguide detours and crossings can be saved.

To this end, we develop a physical design tool: *ToPro*, which projects an input topology directly onto the center of a photonic plane and focuses on waveguide routing between the OSEs and the network nodes such as IP-cores. To prevent long detours of waveguides and extra waveguide crossings outside the OSEs, we propose to optimize the routing order of nets with an efficient method that systematically explores different ordering options. We compare *ToPro* to three state-of-the-art physical design tools: Proton+ [12], PlanarONoC [13], and PSION+ [15]. The experimental results show that *ToPro* greatly accelerates the physical design process and shows superiority in energy-efficiency. For example, for an 8×7 network, *ToPro* reduces the worst-case insertion loss by 54% and 42% compared to Proton+ and PlanarONoC, respectively; and for a larger network with 16 IP-cores, *ToPro* decreases the synthesis time from more than six days to less than one second compared to PSION+.

II. BACKGROUND

A. Environmental Setting

Three-dimensional integrated circuits (3D ICs) using through silicon vias (TSVs) are widely considered a practical platform for WRONoCs [21], [22]. A typical setting of the 3D architecture consists of an optical layer stacked on top of an electronic layer, such as the processor-memory network shown in Fig. 3. Specifically, each cluster of processors on the electronic layer has a dedicated gateway to a hub on the optical layer, and each off-chip dual in-line memory module (DIMM) is connected to the optical layer with a memory controller (MC). The conversion between electronic and optical signals is achieved in the electronic-optical (E/O) and optical-electronic (O/E) interfaces, respectively.

Two features in the practical WRONoC settings are noteworthy for the sake of physical design: (1) the locations of some network nodes such as hubs are determined by the structure of the electronic layer and thus cannot be changed; (2) a network node usually both sends and receives data, i.e.

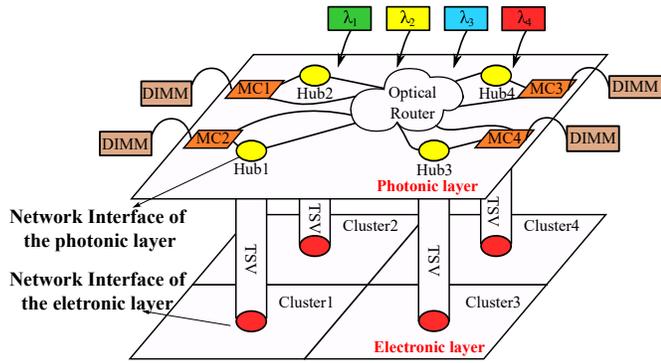


Fig. 3: 3D-stacked many-cores systems

serves as both master and slave. As a result, the node locations are usually given as input layout constraints to the physical design, and in particular, the master and the slave that represent the same node are located close to each other.

B. WRONoC Topologies

A WRONoC topology specifies the necessary network components and their interconnections. In particular, the topology decides the number, the wavelength configurations, and the connections of the OSEs.

On WRONoCs, all masters can communicate with all slaves at the same time without data arbitration. To reserve collision-free signal paths between all master-slave pairs at design time, the number of OSEs in an all-to-all WRONoC topology increases quadratically with respect to the number of master/slave nodes.

To date, several efficient and scalable crossbar-based WRONoC topologies have been proposed, including λ -router [7], Snake [8], GWOR [9], and Light [10]. Among these topologies, λ -router and Snake place masters and slaves at two distant ends of the network, which doesn't naturally match the physical layout constraints. Therefore, to implement such topologies onto the physical plane, additional waveguide crossings or long detours of waveguides are hardly avoidable, as shown in Fig. 2 and Fig. 5(a), respectively. On the other hand, some WRONoC topologies, such as GWOR and Light, place a pair of master and slave nodes close to each other, as shown in Fig. 4, which fits the layout constraints well, and

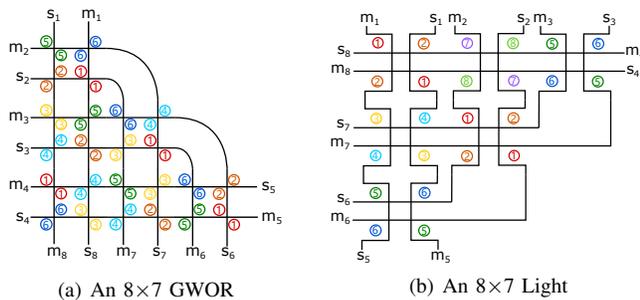


Fig. 4: GWOR and Light topologies

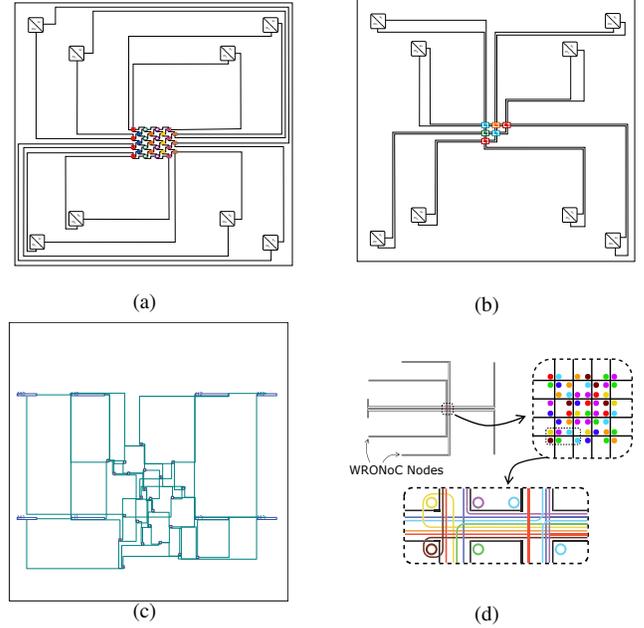


Fig. 5: (a) A manually optimized layout of an 8×8 λ -router topology, (b) A manually optimized layout of an 8×7 Light topology, (c) An automatically synthesized layout of an 8×7 GWOR using Proton+ (d) An automatically synthesized layout of an 8-IP-core network using PSION+

thus have the potential to be physically implemented without extra waveguide crossings and detours, as shown in Fig. 5(b).

C. Related Physical Design Approaches

Current physical design tools treat their input topologies as pure logic schemes. In particular, they interpret a topology as a set of logic components including nodes and OSEs, and a set of logic connections between the components. The physical design tasks thus include the placement of OSEs and the waveguide routing among the OSEs and between the OSEs and the master/slave nodes. Since the number of OSEs is usually much larger than the number of nodes, current physical design methods spend most of their computational efforts on optimizing the locations and the connections of OSEs. But due to the very large design space, existing methods can hardly approach an optimal layout in a reasonable time.

The Proton family of tools [11], [12] are the earliest physical design approaches for WRONoC. Given an input topology and the physical constraints, Proton+ uses a quadratic net model to synthesize the layout with adjustable optimization criteria concerning propagation loss and crossing loss. Fig. 5(c) shows the physical layout synthesized by PROTON+ for an 8×7 GWOR topology with a focus on crossing loss minimization. Since Proton+ allows additional waveguide crossings in the physical design, the number of crossings in the worst-case signal path increases from 10 in the original topology to 42 in the physical layout, which implies a drastic increase in insertion loss and crosstalk noise.

PlanarONoC [13] is another physical design tool that prevents waveguide crossings outside the OSEs but suffers long detours of waveguides. For example, in the layout synthesized by PlanarONoC for an 8×7 GWOR topology, the worst-case signal path is twice longer than the longest path in the layout synthesized by Proton+, which implies a significant increase in propagation loss.

The PSION family of tools [14]–[16] are among the newest design automation approaches for WRONoCs. They use predefined physical layout templates to reduce redundant exploration of the design space, and perform topology synthesis and physical design at the same time to generate a network that highly matches the physical constraints. For a network with 8 IP-cores, PSION+ outperforms PROTON+ and PlanarONoC by reducing 40% in worst-case insertion loss. However, PSION tools suffer exponential growth in design complexity as the network size increases. For example, the synthesis time for a network with 8 IP-cores in PSION+ is only several seconds, while the synthesis time for a network with 16 IP-cores increases to six days.

D. Performance Factors of WRONoCs

Insertion loss and crosstalk noise are two important performance factors to evaluate a WRONoC design.

Insertion loss is the loss of signal power during the transmission process [23]. On WRONoCs, insertion loss can be divided into five losses: *propagation loss* which depends on the lengths of passed waveguides, *crossing loss* which depends on the number of passed waveguide crossings, *bending loss* which depends on the number of passed waveguide bends, *drop loss* which happens when an optical signal is on-resonance with an MRR, and *through loss* which happens when an optical signal is off-resonance with an MRR [4]. Among the five losses, the drop loss and the through loss are determined by the logic schemes, which can not be optimized by physical design, and the bending loss is usually small compared to other losses. Thus, the focus of the physical design is to minimize the propagation loss and the crossing loss, which corresponds to the minimization of the waveguide lengths and waveguide crossings. In particular, the number of waveguide crossings has a large impact on the network performance as crossings generate not only crossing loss but also crosstalk noise.

As shown in Fig. 6(a), when an optical signal passes through a crossing, a portion of its power leaks to other ports as crosstalk noise [23]. The noise signals, especially the noise that has the same wavelength as the desired signals, degrade the signal quality severely [24]. The deterioration becomes even severer when the communication density is high. For example, in an 8×8 WRONoC network, eight signals on different wavelengths sent from IP_1 will generate 16 noise signals when they pass a crossing, as shown in Fig. 6(b). To enhance the signal quality and reduce the power consumption, extra waveguide crossings outside the OSEs should be prevented.

With insertion loss and crosstalk noise, we can calculate the signal-to-noise ratio (SNR), which is an important performance factor but lacks attention in WRONoC design

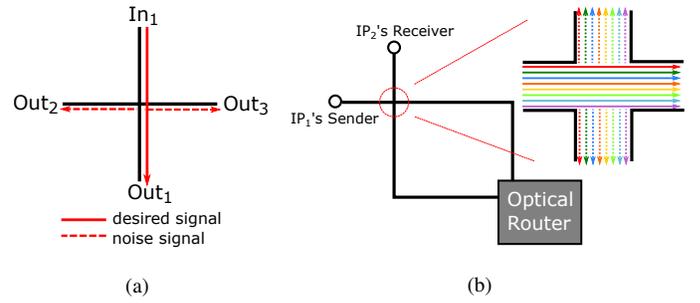


Fig. 6: (a) Crosstalk noise per crossing (b) crosstalk noise generated by waveguide crossing outside the optical router

automation works. In this work, we calculate and analyze the SNR values to evaluate the signal quality in the synthesized layouts.

III. TOPRO: A TOPOLOGY PROJECTOR AND WAVEGUIDE ROUTER

In this work, we propose a design automation tool: *Topro*. Instead of synthesizing the physical layout of a network from scratch based on a pure logic scheme, we propose to take advantage of the optimized interconnections in a WRONoC topology by directly projecting it onto the center of the photonic plane. Thus, the computational efforts for the placement of OSEs and the waveguide routing among the OSEs can be saved, and the focus of the physical design can be moved onto connecting the centralized topology to the network nodes.

First, we propose to select a WRONoC topology that fits the physical layout constraints as the starting point of the synthesis. Specifically, the positions of the master and the slave representing the same network node should be close to each other in the selected topology. Among current WRONoC topologies, GWOR and Light satisfy this requirement and are thus good options. Besides, both GWOR and Light are scalable, which means that we can easily extend their structure to synthesize the locations and the connections of the OSEs in an all-to-all connected WRONoC of an arbitrarily large size.

The selected topology will then be projected as a whole onto the physical plane. Specifically, we will place the OSEs in the topology onto the center of the photonic layer while retaining their relative positions and waveguide interconnections. Thus, the physical design complexity will not be affected by the quadratic increase in the number of OSEs, but remains linear to the number of network nodes. The centralized placement of the OSEs is commonly seen in the physical design templates proposed in [14] and in manually optimized layouts [25].

After the projection, *Topro* routes the waveguides from the centralized topology to the network nodes at predefined locations. For simplicity, we consider the centralized topology as a block and the locations of the masters and the slaves in the topology as *ports* on the block boundaries. Besides, we refer to the connection between a topology port and a network node as a *net*.

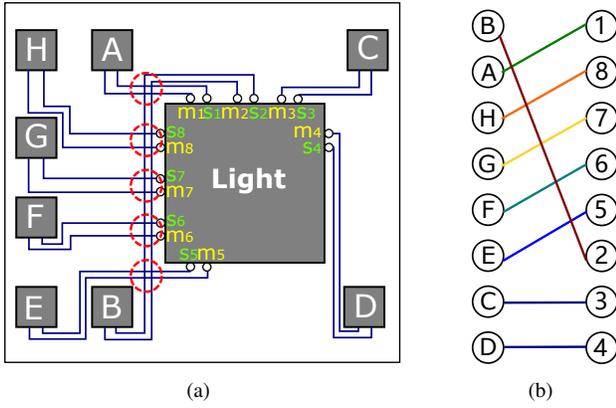


Fig. 7: (a) A WRONoC layout example. (b) The sets of nets that have and that don't have waveguide crossings in their shortest paths.

To determine the shortest paths of each net, we apply the shortest path search algorithm, *Lee Algorithm* [26]. In the ideal case, all nets are routed along the shortest paths without any crossing. However, considering realistic physical constraints, waveguide detours or crossings are inevitable in most cases. Fig. 7(a) shows a layout example, in which nodes A, B, \dots, H are connected to ports $m_1/s_1, m_2/s_2, \dots, m_8/s_8$ of a Light topology along the shortest paths. In particular, the net between node B and ports m_2/s_2 overlaps with five other nets, which results in five extra crossings outside the OSEs.

Since a waveguide crossing generates significant crossing loss and crosstalk noise, preventing waveguide crossings should be assigned a higher priority than minimizing the waveguide lengths, as long as the resulting waveguide detours are not severe. *ToPro* prevents waveguide crossings with a *dynamic pushing* algorithm [27]. Specifically, if the shortest paths of some nets cross one another, as shown in Fig. 8(a), we use the following method to resolve the conflicts:

- First, we pick a net from all conflicting nets and route it along its shortest path, such as net $(A1, A2)$ in Fig. 8(b).

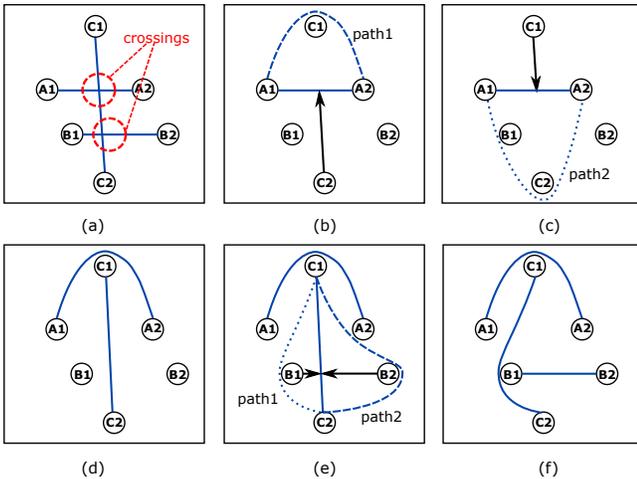


Fig. 8: The dynamic pushing mechanism

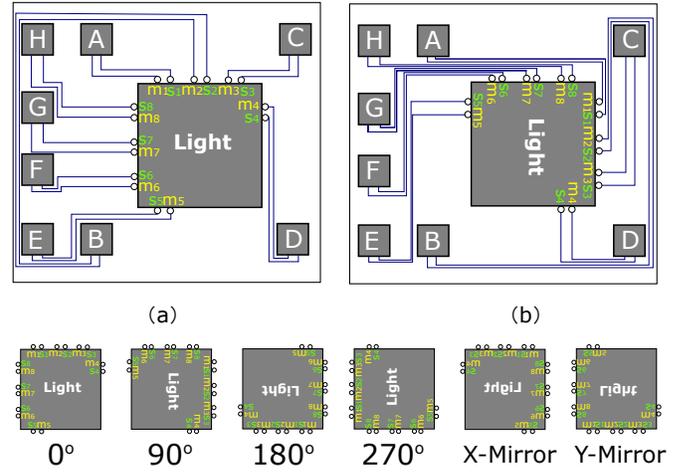


Fig. 9: The synthesized layouts for (a) the router turned by 0° (b) the router turned by 90°

- After that, we pick another unrouted net, such as net $(C1, C2)$, and find its shortest path. If its shortest path crosses a path routed earlier, we consider that the new path generates a force that can *push* the routed path towards one of two opposite directions to make the old path circumnavigate the new path, as shown in Fig. 8(b) and (c).
- We route the new net along its shortest path, and calculate the lengths of the two detouring options of the previous net. We then pick the shorter detouring option to route the net, as shown in Fig. 8(d).
- We repeat this process for the next unrouted path until all nets are routed, as shown in Fig. 8(e) and (f).

We can notice that using this method, the routing solutions are closely dependent on the order that we route the nets. In particular, the net that is routed at last can always be routed along its shortest path, while the nets that are routed earlier are more likely to be “pushed away” and make detours. In other words, to find the optimal routing solution, we need to systematically explore the ordering options for routing the nets.

We know that for a network consisting of n nets, the number of different net ordering options is $n!$. Besides, we notice that the rotation of the centralized topology may result in different waveguide routing performances, as shown in Fig. 9(a) and (b). Thus, *ToPro* will also explore six positioning options of the topology, including rotations of $0^\circ, 90^\circ, 180^\circ,$ and 270° , as well as reflections over the x and the y axis, as shown in Fig. 9. Thus, if we trivially explore all net ordering options, we need to repeat the whole routing process for $6 \times n!$ times, which implies significant computational loads for large networks.

To save the computational efforts on exploring redundant or insignificant routing options, we first perform Lee Algorithm [26] to achieve an initial routing solution for all nets. Specifically, each net is routed along its shortest path. If multiple shortest paths exist for a net, we choose the path that contains the fewest waveguide crossings and bends.

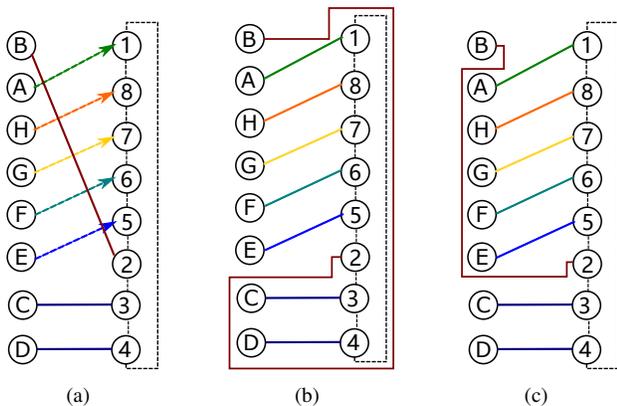


Fig. 10: Waveguide routing with dynamic pushing

Next, we divide the achieved initial routing paths into two disjoint sets based on whether a path overlaps with the others. For the set of paths that have no overlapping parts with the others, i.e. the paths that do not form any waveguide crossings, we fix them as the final routing solutions of the corresponding nets. In this manner, we confine the search space of the net ordering problem to only involve the nets that have waveguide crossings in their shortest paths.

For example, among the shortest paths in the layout that we showed in Fig. 7(a), the paths between node C and ports m_3/s_3 as well as the path between node D and ports m_4/s_4 do not overlap with other paths. Thus, we can exclude the nets between node C , D and the centralized topology from our net ordering problem and focus on the nets in the other set, as shown in Fig. 7(b).

Furthermore, we notice that if some of the nets that do not overlap each other are routed one by one without being interrupted, their ordering options will deliver the same routing results and are thus equivalent. In this case, we do not need to exhaustively explore all the equivalent ordering options but just need to select one of them.

For example, in the layout that we showed in Fig. 8, if we route the net $(C1, C2)$ at last, it makes no difference whether we start the routing with $(A1, A2)$ or $(B1, B2)$, since these two nets have no overlapping parts and will not involve the dynamic pushing process. In other words, the routing orders $(A1, A2) \rightarrow (B1, B2) \rightarrow (C1, C2)$ and $(B1, B2) \rightarrow (A1, A2) \rightarrow (C1, C2)$ deliver the same routing results, and thus we can only check one of them. Similarly, Fig. 7(b) shows that among six nets with overlapped shortest paths, five of them do not overlap each other. If we route the net between node B and ports m_2/s_2 at last, it makes no difference in which order we route the other 5 nets. In this case, among the $5! = 120$ ordering options for routing the 5 nets, we can choose one of them and exclude others from our search space.

After we collect all significant net ordering options, we perform waveguide routing based on each ordering option and resolve the waveguide crossings with dynamic pushing. During this process, the centralized topology as well as the

fixed shortest paths are considered unavailable for the routing.

For example, Fig. 10(a) shows a routing step, in which net $(B, 2)$ is supposed to be “pushed” by the other conflicting nets. Fig. 10(c) and 10(b) show the two detouring options of net $(B, 2)$. In particular, *ToPro* will select the option shown in Fig. 10(b) since it has smaller path length. To note is that the routing path needs to circumnavigate the area occupied by the topology ports $1, 2, \dots, 8$ and the fixed shortest paths of nets $(C, 3)$ and $(D, 4)$.

After exploring all net ordering options, we select the routing solutions with the best network performance to implement the final layout.

To evaluate the performance, we calculate the total insertion loss, the worst-case insertion loss, average SNR, and the worst-case SNR. Specifically, the insertion loss for each signal is the summation of the five losses introduced in Section II-D. The total insertion loss sums up each path’s insertion loss, and maximum insertion loss over all signals is the worst-case insertion loss.

For the SNR calculation, we only consider the noise generated by the signals, which have the same wavelength as the desired signals. With the definition of SNR [23], the SNR of a signal with wavelength λ_n is calculated as $10 \log \frac{P_{output}^{\lambda_n}}{P_{noise}^{\lambda_n}}$, where P_{output} denotes the output power of the desired signals and P_{noise} is the power of all intrachannel noise. The minimum SNR value over all signal paths is the worst-case SNR.

IV. EXPERIMENTAL RESULTS

ToPro is implemented in C++, and all experiments discussed in this paper were carried out on a 2.6 GHz CPU. For each test case, we synthesized two layouts with *ToPro* based on GWOR and Light topologies, respectively. We compare *ToPro* to three state-of-the-art physical design tools. In Section IV-A, we compare *ToPro* to the classical physical design tools, Proton+ [12] and PlanarONoC [13], for an 8-node processor-memory network in terms of the worst-case insertion loss, the length of critical path, the number of crossings passed by the critical path, and program runtime. Furthermore, we tested *ToPro* on four different locations of memory-controllers proposed in [12] and compare our results with the results of Proton+ [12]. In Section IV-B, we compare *ToPro* against PSION+ [15] for networks with 8 and 16 nodes in terms of the worst-case insertion loss, MRR usage, wavelength usage, and program runtime. We synthesized both GWOR and Light for different test cases. Besides, we present the average SNR and the worst-case SNR of the synthesized results of GWOR and Light for different test cases in Section IV-C. As current physical design tools have not presented the SNR results, we compare our results to the SNR values in their logic schemes reported in [10].

A. *ToPro* versus Classical Physical Design Tools

We synthesized two layouts implementing GWOR and Light with *ToPro* for an 8-node processor-memory network, which contains four hubs and four memory controllers, with the same

TABLE I: Results for an 8-node processor-memory network on different node positions

		pos (a)				pos (b)				pos (c)				pos (d)			
		il_w	L	C	T	il_w	L	C	T	il_w	L	C	T	il_w	L	C	T
Proton+	GWOR	8.4	13.0	38	88.5	9.1	14.7	41	81.5	8.1	11.0	38	88.5	8.1	13.8	35	79
PlanarONoC	GWOR	6.4	28.6	10	0.1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
ToPro	GWOR	3.8	14.2	8	0.19	5.0	22.2	8	0.15	4.5	18.4	8	0.14	4.0	13.5	10	0.17
	Light	5.5	21	12	0.19	6.4	33.3	6	0.2	5.2	19	12	0.15	4.3	13.5	12	0.07

il_w : the maximum 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 (including the crossings in the OSEs) passed by the signal with maximum insertion loss. T: the program runtime denoted in seconds.

node locations, die dimension, size of OSEs, and loss parameters as applied in Proton+ and PlanarONoC. We compare our synthesis results with the layouts synthesized by Proton+ and PlanarONoC for an 8×7 GWOR topology. Different sets of the coefficients in the objective function of Proton+ result in different layouts. Here, we use the best results of Proton+ for comparison. In particular, we tested **ToPro** with four different positions of memory-controllers proposed in [12], namely: memory-controllers located (a) pairwise at the periphery, (b) at the corner, (c) on the four sides, (d) at the leftmost side of the photonic layer. PlanarONoC has only published the synthesis results considering the first position setting, i.e. pos (a). The synthesis results for other position settings are thus not available.

Table I shows the results of the comparisons. In general, **ToPro** outperforms Proton+ and PlanarONoC in reducing insertion loss. For the synthesized layouts of an 8×7 GWOR, **ToPro** greatly reduces the worst-case insertion loss by 50% on average compared to Proton+, which is mainly driven by the significant reduction in the number of waveguide crossings. The removal of extra crossings in **ToPro** has two benefits. First, fewer crossings indicate less crossing loss. Second, the reduction of crossings benefits the enhancement of signal quality considering that crossings are important sources of crosstalk noise. Both Proton+ and **ToPro** have their worst results on pos (b), as the memory controllers are placed a bit far away from the center, which results in a longer path length. Other positions of memory controllers are closer to the center, which benefits the reduction of the path length. In addition, due to the deterministic methodology and the reduction of design complexity by retaining the interconnections in topologies, **ToPro** solves all test cases within 1 second.

Comparing to PlanarONoC, **ToPro** decreases the worst-case insertion loss by 42%, which is driven by the significant reduction in the length of the waveguide detours. For the synthesized layout of an 8×7 GWOR on pos (a), **ToPro** decreases the length of the critical path by 53% compared to PlanarONoC and thereby greatly reduces the propagation loss.

 TABLE II: Results of PSION+ and **ToPro** for 8- and 16-node networks

		il_w	#MRR	#wl	T	
8-node	PSION+	CGT-e0	3.1	52	8	13
		CGT-e6	3.7	52	7	75
		DGT	3.6	48	8	3
		Ring	3.1	88	7	31347
		Custom	4.1	40	7	<1
	ToPro	GWOR	3.7	48	7	0.17
		Light	4.3	24	8	0.07
16-node	PSION+	CGT-e10	4.2	320	17	561600
	ToPro	GWOR	4.3	224	15	0.72
		Light	3.5	112	16	0.96

il_w : the maximum insertion loss value denoted in *dB*. #MRR: the number of MRRs. #wl: the number of wavelengths. T: the program runtime denoted in seconds.

B. ToPro versus PSION+

We compare **ToPro** to PSION+ for two different applications: an 8-node network and a 16-node network in terms of the worst-case insertion loss, MRR usage, wavelength usage, and program runtime. To note is that these two applications do not require all-to-all communications between all network nodes. Specifically, the 8-node network requires 44 communication paths, among which $4 \times 7 = 28$ paths are used by four hubs to communicate with the other seven nodes and $4 \times 4 = 16$ paths are used by four memory controllers to communicate with four hubs. On the other hand, the 16-node network requires $16 \times 15 = 240$ communication paths excluding self communications of nodes. PSION+ takes advantage of the reduced communication requirements and does not synthesize the unnecessary signal paths. Nevertheless, since GWOR and Light naturally support all-to-all communication, **ToPro** keeps all the signal paths in the synthesized layouts.

Different from Proton+ and PlanarONoC, which perform physical design from scratch, PSION+ applies physical templates and produces router designs reduced from the templates. Here, for the 8-node network, we consider the centralized grid template (CGT-e0), expanded centralized grid template (CGT-e6), distributed grid template (DGT), ring template (Ring), and custom template (Custom) introduced in [14], [15] with pos

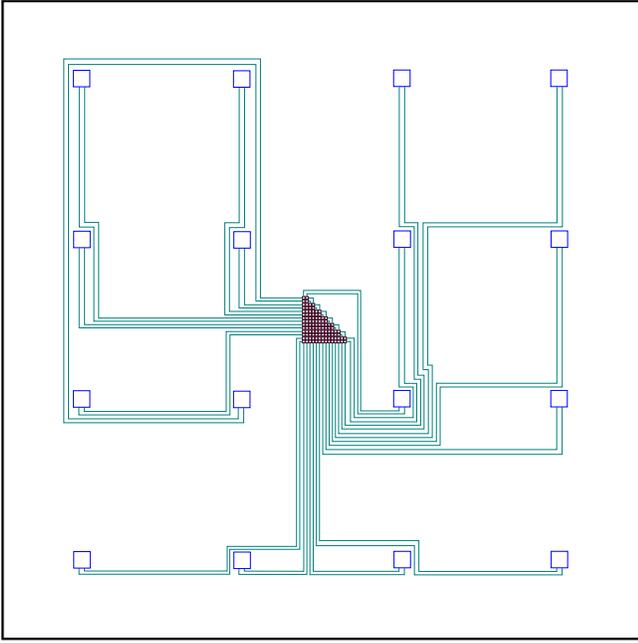


Fig. 11: The synthesized layout of a 16×15 Light

(d) shown in Table I as the node positions. For the 16-node network, we consider the expanded centralized grid template (CGT-e10), as applied in PSION+. One thing to be noticed is that PSION+ applied the loss parameters in [12] for the 8-node network and the loss parameters in [28] for the 16-node network. For a fair comparison, *ToPro* applies the same node positions, die dimension, size of OSEs, and loss parameters as PSION+.

Table II shows the results of PSION+ and *ToPro* for the two applications. For the 8-node network, PSION+ has comparable insertion loss with *ToPro* when the templates CGT-e6, DGT, and Custom are applied and less insertion loss than *ToPro* when the templates CGT-e0 and Ring are applied. The better performance of PSION+ comes from its co-optimization of topological and physical design, whereas the topology taken by *ToPro* as its input is not optimized for the targeting application. The co-optimization of PSION+, however, also causes a computational burden. While *ToPro* is always able to finish the physical design within 1 second regardless of the network size, the program runtime of PSION+ drastically increases when the network size becomes larger. In particular, PSION+ needs 561600 seconds (about 6.5 days) to synthesize the design for the 16-node network. Fig. 11 shows the layout for a 16×15 Light. The topology is placed on the center of the die and turned by 270° to minimize the path length. In this case, only one net in Light makes a long detour to avoid the crossings and suffers relatively more propagation loss. Still, *ToPro* decreases the worst-case insertion loss by 17% compared to PSION+. Besides the insertion loss, *ToPro* also requires fewer MRRs and wavelengths compared to PSION+. The reduction is achieved by retaining the optimized resource usage in the topologies.

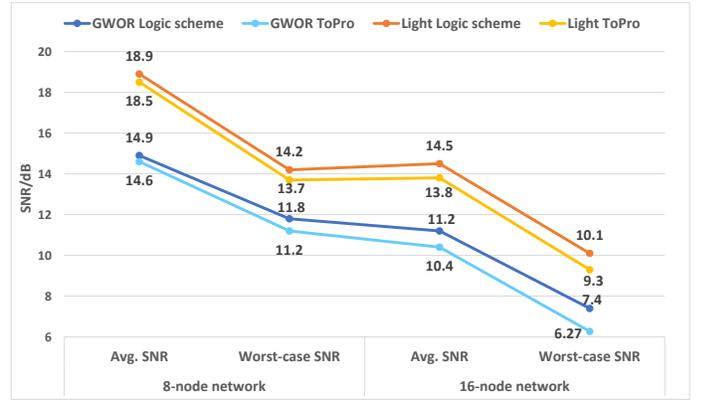


Fig. 12: Average and worst-case SNR values in the physical layouts and the topologies for GWOR and Light

C. Analysis on SNR

To evaluate the signal quality of our synthesized results, we calculate and compare the average and worst-case SNR values of the layouts synthesized by *ToPro* implementing GWOR and Light. It is worth mentioning that the current physical design tools have not presented the SNR results of their layouts.

Fig. 12 shows the SNR values. In general, for both GWOR and Light, the SNR values of the synthesized layouts are only slightly worse than the SNR values of the original topologies. The SNR decrease is caused by the additional propagation loss and bending loss in their synthesized layouts. Since *ToPro* prevents extra waveguide crossings and thus the resulting crosstalk noise, the network does not suffer a significant performance decline after the physical implementation.

V. CONCLUSION

In this work, we propose a physical design tool: *ToPro*. Taking advantage of the optimized internal interconnections of physical-design-friendly WRONoC topologies, *ToPro* projects the input topologies directly onto the photonic plane and focuses on waveguide routing. It prevents the formation of extra waveguide crossings and long waveguide detours by searching for the optimal routing order. Compared with the classical physical design tools, *ToPro* outperforms them in decreasing the insertion loss by significantly reducing the number of crossings and the path lengths. Compared to a new design automation tool, PSION+, which combines topology synthesis with physical design, *ToPro* shows the superiority in energy- and computational efficiency for large networks. Based on the SNR analysis, we demonstrate that the physical designs produced by *ToPro* can well maintain the signal quality which has been optimized by the input topologies.

ACKNOWLEDGMENT

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

REFERENCES

- [1] C. V. Networking, "Cisco visual networking index (2017-2022)," 2019.
- [2] Q. Cheng, M. Bahadori, M. Glick, S. Rumley, and K. Bergman, "Recent advances in optical technologies for data centers: a review," *Optica*, vol. 5, no. 11, pp. 1354–1370, Nov 2018.
- [3] S. Werner, J. Navaridas, and M. Luján, "A Survey on Optical Network-on-Chip Architectures," *ACM Comput. Surv.*, vol. 50, no. 6, Dec. 2017.
- [4] M. Li, T.-M. Tseng, D. Bertozzi, M. Tala, and U. Schlichtmann, "CustomTopo: A Topology Generation Method for Application-Specific Wavelength-Routed Optical NoCs," in *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, ser. ICCAD '18. New York, NY, USA: Association for Computing Machinery, 2018.
- [5] W. Bogaerts, P. De Heyn, T. Van Vaerenbergh, K. De Vos, S. Kumar Selvaraja, T. Claes, P. Dumon, P. Bienstman, D. Van Thourhout, and R. Baets, "Silicon microring resonators," *Laser & Photonics Reviews*, vol. 6, no. 1, pp. 47–73, 2012.
- [6] M. Li, T.-M. Tseng, M. Tala, and U. Schlichtmann, "Maximizing the Communication Parallelism for Wavelength-Routed Optical Networks-On-Chips," in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2020, pp. 109–114.
- [7] M. Briere, B. Girodias, Y. Bouchebaba, G. Nicolescu, F. Mieyeville, F. Gaffiot, and I. O'Connor, "System Level Assessment of an Optical NoC in an MPSoC Platform," in *2007 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2007, pp. 1–6.
- [8] L. Ramini, P. Grani, S. Bartolini, and D. Bertozzi, "Contrasting wavelength-routed optical NoC topologies for power-efficient 3d-stacked multicore processors using physical-layer analysis," in *2013 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2013, pp. 1589–1594.
- [9] X. Tan, M. Yang, L. Zhang, Y. Jiang, and J. Yang, "On a Scalable, Non-Blocking Optical Router for Photonic Networks-on-Chip Designs," in *2011 Symposium on Photonics and Optoelectronics (SOP0)*, 2011, pp. 1–4.
- [10] Z. Zheng, M. Li, T.-M. Tseng, and U. Schlichtmann, "Light: A Scalable and Efficient Wavelength-Routed Optical Networks-On-Chip Topology," in *Proceedings of the 26th Asia and South Pacific Design Automation Conference (ASP-DAC)*, ser. ASPDAC '21. Association for Computing Machinery, 2021, p. 568573.
- [11] A. Boos, L. Ramini, U. Schlichtmann, and D. Bertozzi, "PROTON: An automatic place-and-route tool for optical Networks-on-Chip," in *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 11 2013, pp. 138–145.
- [12] A. V. Beuningen, L. Ramini, D. Bertozzi, and U. Schlichtmann, "PROTON+: A Placement and Routing Tool for 3D Optical Networks-on-Chip with a Single Optical Layer," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 12, no. 4, 12 2015.
- [13] Y.-K. Chuang, K.-J. Chen, K.-L. Lin, S.-Y. Fang, B. Li, and U. Schlichtmann, "PlanarONoC: Concurrent Placement and Routing Considering Crossing Minimization for Optical Networks-on-Chip *," in *Proceedings of the 55th Annual Design Automation Conference (DAC)*, ser. DAC '18. Association for Computing Machinery, 2018.
- [14] A. Truppel, T.-M. Tseng, D. Bertozzi, J. C. Alves, and U. Schlichtmann, "PSION: Combining Logical Topology and Physical Layout Optimization for Wavelength-Routed ONoCs," in *Proceedings of the 2019 International Symposium on Physical Design (ISPD)*, ser. ISPD '19. Association for Computing Machinery, 2019, p. 4956.
- [15] A. Truppel, T.-M. Tseng, D. Bertozzi, J. Alves, and U. Schlichtmann, "PSION+: Combining Logical Topology and Physical Layout Optimization for Wavelength-Routed ONoCs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. PP, pp. 1–1, 02 2020.
- [16] A. Truppel, T.-M. Tseng, and U. Schlichtmann, "PSION 2: Optimizing Physical Layout of Wavelength-Routed ONoCs for Laser Power Reduction," in *Proceedings of the 39th International Conference on Computer-Aided Design (ICCAD)*, ser. ICCAD '20. Association for Computing Machinery, 2020.
- [17] L. Ramini, D. Bertozzi, and L. Carloni, "Engineering a Bandwidth-Scalable Optical Layer for a 3D Multi-core Processor with Awareness of Layout Constraints," in *2012 IEEE/ACM Sixth International Symposium on Networks-on-Chip*, 05 2012, pp. 185–192.
- [18] Y. Xie, M. Nikdast, J. Xu, W. Zhang, Q. Li, X. Wu, Y. Ye, X. Wang, and W. Liu, "Crosstalk Noise and Bit Error Rate Analysis for Optical Network-on-Chip," in *Proceedings of the 47th Design Automation Conference (DAC)*, ser. DAC '10. Association for Computing Machinery, 2010, p. 657660.
- [19] J. Chan, G. Hendry, A. Biberman, and K. Bergman, "Architectural Design Exploration of Chip-scale Photonic Interconnection Networks Using Physical-layer Analysis," in *2010 Conference on Optical Fiber Communication (OFC/NFOEC), collocated National Fiber Optic Engineers Conference*, 2010, pp. 1–3.
- [20] S. Le Beux, J. Trajkovic, I. O'Connor, G. Nicolescu, G. Bois, and P. Paulin, "Multi-Optical Network-on-Chip for Large Scale MPSoC," *IEEE Embedded Systems Letters*, vol. 2, no. 3, pp. 77–80, 2010.
- [21] Y. Ye, L. Duan, J. Xu, J. Ouyang, M. K. Hung, and Y. Xie, "3D optical networks-on-chip (NoC) for multiprocessor systems-on-chip (MPSoC)," in *2009 IEEE International Conference on 3D System Integration*. IEEE, 2009, pp. 1–6.
- [22] T. Tseng, A. Truppel, M. Li, M. Nikdast, and U. Schlichtmann, "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 (ICCAD)*, 2019, pp. 1–6.
- [23] M. Nikdast, J. Xu, L. H. K. Duong, X. Wu, X. Wang, Z. Wang, Z. Wang, P. Yang, Y. Ye, and Q. Hao, "Crosstalk Noise in WDM-Based Optical Networks-on-Chip: A Formal Study and Comparison," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 11, pp. 2552–2565, 2015.
- [24] L. H. K. Duong, Z. Wang, M. Nikdast, J. Xu, P. Yang, Z. Wang, Z. Wang, R. K. V. Maeda, H. Li, X. Wang, S. Le Beux, and Y. Thonnart, "Coherent and Incoherent Crosstalk Noise Analyses in Interchip/Intrachip Optical Interconnection Networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 7, pp. 2475–2487, 2016.
- [25] S. Le Beux, H. Li, G. Nicolescu, J. Trajkovic, and I. O'Connor, "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.
- [26] C. Y. Lee, "An Algorithm for Path Connections and Its Applications," *IRE Transactions on Electronic Computers*, vol. EC-10, no. 3, pp. 346–365, 1961.
- [27] S. Liu, G. Chen, T. T. Jing, L. He, T. Zhang, R. Dutta, and X.-L. Hong, "Topological Routing to Maximize Routability for Package Substrate," in *Proceedings of the 45th Annual Design Automation Conference (DAC)*, ser. DAC '08. Association for Computing Machinery, 2008, p. 566569.
- [28] M. OrtnObn, M. Tala, L. Ramini, V. Vials-Yufera, and D. Bertozzi, "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 (VLSI) Systems*, vol. PP, pp. 1–14, 03 2017.