FAST: A Fast Automatic Sweeping Topology Customization Method for Application-Specific Wavelength-Routed Optical NoCs

Xiao Moyuan
Chair of Electronic Design Automation
Technical University of Munich
Munich, Germany
moyuan.xiao@tum.de

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

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

***Abstract*—Optical network-on-chip (ONoC) is an emerging upgrade for electronic network-on-chip (ENoC). As a kind of ONoC, wavelength-routed optical network-on-chip (WRONoC) shows ultra-high bandwidth and ultra-low latency in data communication. Manually designed WRONoC topologies typically reserve all to all communications. Topologies customized for application-specific networks can save resources, but require automation for their efficient design. The state-of-the-art design automation method proposes an integer-linear-programming (ILP) model. The runtime for solving the ILP model increases exponentially with the growth of communication density. Besides, the locations of the physical ports are not taken into consideration in the model. This causes unavoidable detours and crossings in physical layout. In this work, we present FAST: an automatic topology customization and optimization method combining ILP and a sweeping technique. FAST overcomes the runtime problem and provides multiple topology variations with different port orders for physical layout. Experimental results show that FAST is thousands times faster when tackling dense communications and ten to thousands times faster when tackling sparse communications while providing multiple better or equivalent topologies regarding resource usage and the worst-case insertion loss.**

***Index Terms*—WRONoC, Micro Ring Resonators, Topology Generation, ILP, Optimization, Computer–Aided Design**

I. Introduction

Multiprocessor systems-on-chip (MPSoCs) is one of the most promising solutions for dense computation. Due to the demands of high-quality communication in MPSoCs, a novel data transmission approach with high bandwidth and low latency is urgently required. In recent years, optical network-on-chip (ONoC) has emerged and has become a promising next-generation data transmission platform. Instead of using electronic signals, ONoC uses optical signals to transmit data and thus acquires ultra-high bandwidth and ultra-low latency. ONoC can be classified into two kinds: 1) active networks in which a control system is applied to control the routing behavior in real time during the communication. 2) passive networks in which all routing paths are predefined, also named as wavelength-routed optical networks-on-chip (WRONoC) [1] [2]. WRONoC provides even lower latency than other ONoCs because no control is required during transmission.

WRONoC is enabled by the rapid development of silicon photonics and CMOS fabrication technology. There are two core components in WRONoC: 1) Optical waveguide. It is the medium where light passes through, like conductor for electrons. Optical signals modulated to different wavelengths are allowed to travel along the same waveguide. This is known as wavelength-division multiplexing (WDM) [3]. 2) Silicon Microring Resonator (MRR). An MRR is a ring-formed waveguide. When the optical path length of an MRR can be exactly divided by the wavelength of an optical signal, we say this signal is resonant with the MRR, otherwise nonresonant [4]. As shown in Fig. 1, if an optical signal is resonant with an MRR, it changes its direction when passing by the MRR. If nonresonant, it ignores the MRR and goes straight. Each signal suffers power loss when it passes through waveguides, crossings, MRRs or makes turns. These power losses are generally called insertion loss [5]. The worst-case insertion loss among all signals is an important property because it determines the required laser power.

Typical manually designed WRONoC topologies such as folded crossbar [6], lambda router [7] or GWOR [8] assume full connectivity, i.e. each sender (master) sends messages to all receivers (slaves) and each receiver receives messages from all senders. This assumption is not required for application-specific networks [9]. If topologies supporting full connectivity are directly used without tailoring, this leads to the waste of resources and power. The tailoring work on the other hand is tedious, especially for large-scale communication networks. Moreover, manually tailored topology does not guarantee the optimal solution.

So far, there is only one fully automatic topology-synthesis tool realizing topology customization, optimization and wavelength assignment for WRONoCs. It is named as CustomTopo [9]. CustomTopo includes the WRONoC topology structure and its communications into an integer-linear-programming (ILP)

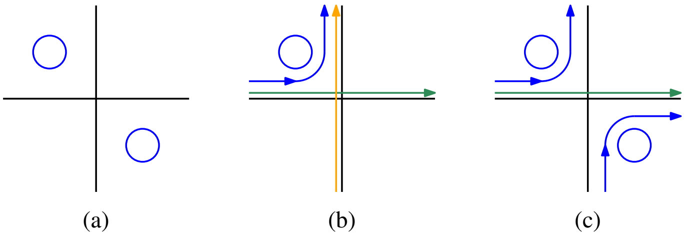
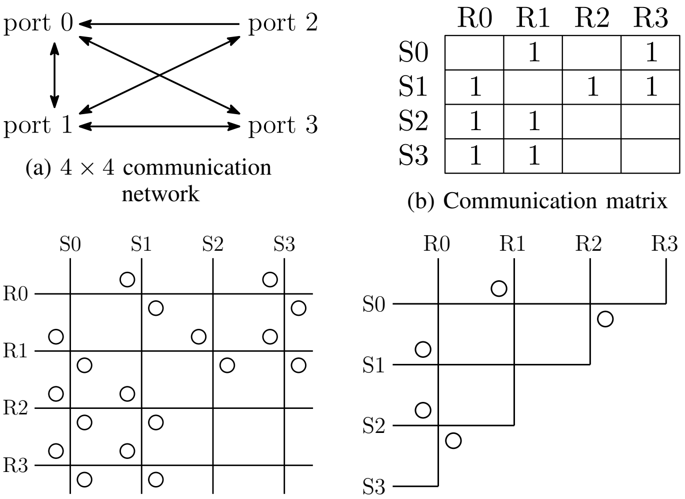


Fig. 1: Basic components and routing behavior of WRONoC. (a) A waveguide crossing with 2 MRRs. (b)(c) Blue signals are resonant with the MRRs. Orange and green signals are nonresonant with the MRRs.

DATE 2021, DOI: 10.23919/DATE51398.2021.9474159
https://ieeexplore.ieee.org/document/9474159



(a) 4 × 4 communication network

	R0	R1	R2	R3
S0		1		1
S1	1		1	1
S2	1	1		
S3	1	1		

(b) Communication matrix

(c) Initial topology in CustomTopo [9]

(d) Initial topology in FAST

Fig. 2: A 4×4 communication network and its initial topologies in CustomTopo and FAST. Each port in (a) has a sender (S) and a receiver (R), e.g. port 0 has S0 and R0. If a sender communicates with a receiver, we call it a **communication**, e.g. **communication (S0, R1)**. In (b), each communication is marked with 1.

model. The optimization targets are MRR usage, wavelength usage and the worst-case insertion loss. The aspects that can be improved in CustomTopo are: (1) The initial topology of CustomTopo is a complete matrix. An example is shown in Fig. 2c. This leads to numbers of empty crossings (crossings with no MRRs), especially for sparse communication networks. (2) In CustomTopo, the basic communication unit is add-drop filter (ADF). It is a waveguide crossing with two MRRs as shown in Fig. 1a. One of the two MRRs is usually redundant and cannot be removed. (3) The computational complexity of the ILP model increases exponentially with the growth of communication density and network size. (4) CustomTopo does not consider the physical port locations of the network. This leads to extra waveguide detours and crossings in physical layout.

In this work, we present FAST, a fast automatic sweeping topology customization and optimization method for application-specific WRONoCs. FAST solves the four problems of CustomTopo by four features: (1) A half-matrix initial topology with fewer empty crossings (shown in Fig. 2d) is proposed. (2) FAST addresses each MRR and ensures that no MRR is redundant. (3) FAST combines a fast sweeping technique and an ILP model, which makes it ten to thousands times faster than CustomTopo despite running on a much weaker computer. (4) Multiple topology variations with different port orders are generated. The variation matching the physical port locations the best can be selected as the final topology for layout.

II. Initial Topology and General Optimization Idea

A. Logic Scheme of the Initial Topology

We modify Snake, a WRONoC topology proposed in [6], and use the modified version as the logic scheme. As shown in Fig. 3a, we place senders on the left side and receivers on the top. With this arrangement, the topology can directly be represented as a matrix. Rings with different colors represent MRRs resonant with different wavelengths. The numbers inside each MRR represent a communication. For example, (3, 1) in the ring means S3 sends a message to R1 and the optical signal changes direction by MRR (3, 1). Not all crossings are associated with two MRRs because the topology does not support full connectivity. Some communications like (S2, R1), (S3, R0) don't rely on MRRs. These communications are called **default communications** in this work. The path of a default communication is called **default path**.

A comprehensive comparison of the layout efficiency of different ONoC logic topologies under practical physical constraints has demonstrated the superiority of Snake. Based on the experimental results in [6], Snake outperforms other logic topologies including folded crossbar, lambda router, GWOR and ORNoC [10] in the physical layout regarding the worst-case insertion loss and power consumption. Therefore, we use it as the logic scheme of the initial topology.

B. General Optimization Idea

We optimize the topology by changing the sequence of senders and receivers. For example: In application-specific WRONoCs, not every port simultaneously sends and receives messages. Some of the senders and receivers are redundant. In Fig. 3a, S1 doesn't send any signal and R3 doesn't receive any signal. We set S1 and R3 as the terminals of a default path (Fig. 3b) and remove that entire default path (Fig. 3c).

The general optimization idea is: For a communication network, different topology variations can be generated based on different sender/receiver orders. We sweep through these variations to find the best ones.

To realize this idea, we need to describe the logic topologies shown in Fig. 3 by matrices, assign a wavelength to each MRR and design an optimization algorithm. In Section III, we propose three methods to solve these problems and provide three proofs to support the validity of the methods.

III. Three Proofs and Three Methods

The following three proofs are the backbone of this work. The first proof theoretically supports the validity of a fast initial topology generation method. This method will be introduced following the first proof. The second proof carries out the wavelength assignment rule. After this proof, an integer-linear-programming (ILP) model is presented to minimize the wavelength usage and assign a wavelength to each MRR. The third

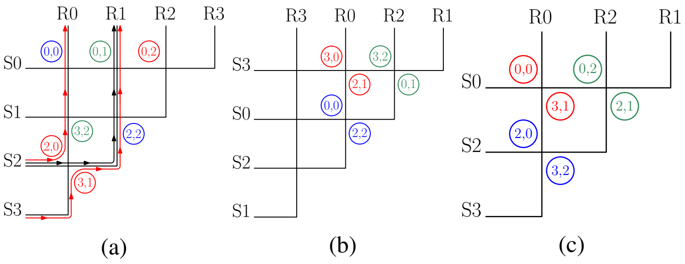


Fig. 3: Logic scheme and an optimization example.

proof allows us to quickly pick out those topologies with the minimal wavelength usage.

A. Proof 1 and Method 1: Generate initial topology

Fig. 4a is a half-matrix initial topology of FAST supporting full connectivity. Fig. 4b is a complete-matrix topology. We notice MRRs with the same colors in Fig. 4a are symmetric with respect to the antidiagonal in Fig. 4b. If this is always true, we can easily generate a half matrix through folding the complete matrix along the antidiagonal.

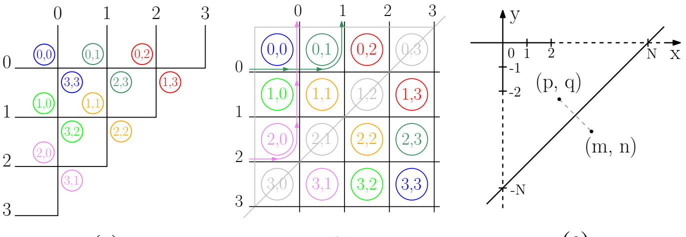
Now, we prove: If a complete matrix like Fig. 4b is folded along the antidiagonal, the two MRRs which overlap are exactly the two MRRs associated to a crossing in Fig. 4a. To prove this, we have to prove: 1) The two MRRs associated to a crossing in Fig. 4a (like MRR (0, 0) and (3, 3)) must be symmetric with respect to the antidiagonal, when they are in a complete matrix. This is proved in Proof 1. 2) Each crossing in a complete-matrix topology is only associated with one MRR. This argument does not need to be proved. It is directly shown in Fig. 4b.

Proof 1: We call the size of the communication matrix "*degree*". For example, in Fig. 4a, the *degree* of the communication matrix is 4. In general, the default communications of the initial topology can be expressed with $(a, N-a)$ $(N = degree - 1, a = 0, 1, 2, ..., N)$. In the 4×4 initial topology, the default communications are $(S0, R3)$, $(S1, R2)$, $(S2, R1)$, $(S3, R0)$, i.e. $(0, 3)$, $(1, 2)$, $(2, 1)$, $(3, 0)$. In Fig. 4a, every two default paths have a crossing. If the communications in the upper-left corner and the lower-right corner of a crossing are (p, q) and (m, n), according to the default communication expression, there are always:

$$\begin{cases} m = N - q, \\ n = N - p, \quad N, p, q, m, n \in \mathbb{N} \end{cases} \tag{1}$$

Now, we prove: in a complete matrix like Fig. 4b, if two communications (p, q), (m, n) satisfy (1), their MRRs must be symmetric with respect to the antidiagonal. This proof is done in Fig. 4c by proving: 1) The dotted line segment connecting (p, q) and (m, n) is perpendicular to the antidiagonal (the solid line connecting $(0, -N)$ and $(N, 0)$). 2) The midpoint of the dotted line segment between (p, q) and (m, n) is on the antidiagonal. In Fig 4c, the gradient of the solid line is 1. (p, q) and (m, n) satisfy (1). As they are now placed in the quadrant IV of a cartesian coordinate system, we modify (1) into:

$$\begin{cases} m = N - |q|, \\ |n| = N - p, \quad N, p, m \in \mathbb{N}, q, n \in \mathbb{Z}, q, n \leq 0 \end{cases} \tag{2}$$



(a) (b) (c)

Fig. 4: (a) Half-matrix topology. (b)(c) Complete-matrix topology and its coordinate system expression.

First, using (2), we determine the gradient of the dotted line segment is -1. This proves that the line segment between (p, q) and (m, n) is perpendicular to the antidiagonal. Second, the equation for the antidiagonal in the coordinate system is $y = x - N$. It is easy to prove the midpoint of the line segment between (p, q) and (m, n) is on the antidiagonal. These two arguments prove that any (p, q) and (m, n) satisfying (1) are symmetric with respect to the antidiagonal in a complete matrix.

Method 1: Now, a fast topology customization method based on this proof is proposed using the 4×4 communication network in Fig. 2 as an example:

1) First of all, we generate a communication dictionary shown in (3), to clarify which sender or receiver is represented by which index number in matrix:

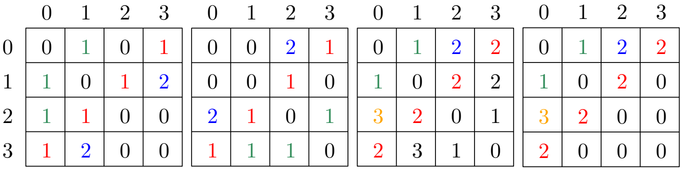
$$\begin{cases} \textit{sender dict} : (0 : S0, 1 : S1, 2 : S2, 3 : S3) \\ \textit{receiver dict} : (0 : R0, 1 : R1, 2 : R2, 3 : R3) \end{cases} \tag{3}$$

2) As shown in Fig. 5a, based on (3), we generate a complete matrix similar to Fig. 2b. Communication nodes on the left side of the antidiagonal and on the antidiagonal are marked with 1, but communication nodes on the right side of the antidiagonal are marked with 2.
3) As shown in Fig. 5b, mirror this complete matrix generated in step 2 with respect to the antidiagonal.
4) As shown in Fig. 5c, add the two matrices shown in Fig. 5a and Fig. 5b. This step overlaps those MRRs which are symmetric with respect to the antidiagonal in a complete matrix.
5) As shown in Fig. 5d, remove non-zero values under the antidiagonal. In Fig. 5d, each 1 represents an MRR in the upper-left corner of a crossing; 2s on the antidiagonal represent default communications; each 2 off the antidiagonal represents an MRR in the lower-right corner of a crossing; 3 means both MRRs are required. The coordinates of these values are called **non-zero coordinates**. Fig. 5d is called **initial matrix**. It can be directly transferred to the initial topology (Fig. 2d).

B. Proof 2 and Method 2: Wavelength assignment

A sender must use different wavelengths to send signals to different receivers. A receiver needs to receive different wavelengths from different senders [9]. This rule has to be fulfilled for conflict-free communications in WRONoCs. **Now, we discuss and prove** the communication rule of FAST.

Proof 2: As shown in Fig. 6, there are two kinds of situations. 1) For communications requiring MRRs, e.g. communication (1, 1) and (2, 2), a signal must not pass by an MRR whose color



	0	1	2	3
0	0	1	0	1
1	1	0	1	2
2	1	1	0	0
3	1	2	0	0

(a)

	0	1	2	3
0	0	0	2	1
1	0	0	1	0
2	2	1	0	1
3	1	1	1	0

(b)

	0	1	2	3
0	0	1	2	2
1	1	0	2	2
2	3	2	0	1
3	2	3	1	0

(c)

	0	1	2	3
0	0	1	2	2
1	1	0	2	0
2	3	2	0	0
3	2	0	0	0

(d)

Fig. 5: Initial matrix generation process.

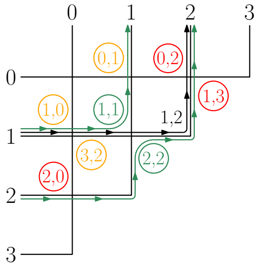


Fig. 6: Wavelength assignment for FAST.

is the same as the signal's own resonant MRR. For example, on the path of communication (2, 2), there must be no other green MRRs other than MRR (2, 2). Otherwise the green signal will be led to the wrong terminal. 2) Similarly, for default communications requiring no MRR, e.g. communication (1, 2), there must be no black MRRs on the default path.

Because every crossing in FAST is the intersection of two default paths, the communication rule can be simply formulated as follows: Wavelengths assigned to each non-zero coordinate on a default path should be different from each other.

Method 2: Based on the communication rule, we present an ILP model to determine the minimal wavelength usage and assign a wavelength to each MRR. The **input** of this ILP model is an initial matrix shown in Fig. 5d. The **outputs** are the minimal wavelength usage and the assignment of a wavelength to each MRR. The **optimization objective** is to minimize the wavelength usage.

The ILP model includes three groups of variables and four constraints. **The first variable group** is $indicator_{(m,n),w}$. They are binary variables. They indicate whether non-zero coordinate (m,n) in the initial matrix takes wavelength w. If (m,n) is assigned with w, $indicator_{(m,n),w} = 1$. If (m,n) is not assigned with w, $indicator_{(m,n),w} = 0$. **The second variable group** is $W_{(m,n)}$. They are integer variables with a lower bound 1 and an upper bound $degree$. They indicate the wavelength type of (m,n). **The third variable** is W_{max}. It's also an integer variable with a lower bound 1 and an upper bound $degree$. Constraints are listed in the following:

1) Each non-zero coordinate in initial matrix is assigned with exactly one wavelength:

$$\forall (m,n) \in C: \sum_{w=1}^{degree} indicator_{(m,n),w} = 1 \qquad (4)$$

C is the set of all non-zero coordinates in an initial matrix.

2) In each default path, a wavelength type must not appear more than once:

$$\forall p_{default} \in P_{default}\ \forall w \in [1,2,...,degree]: \sum_{(m,n)\in nzc} indicator_{(m,n),w} \leq 1 \qquad (5)$$

$p_{default}$ is one of the default paths in the initial matrix. $P_{default}$ is the set of all default paths in the initial matrix. nzc is the set of all non-zero coordinates on a default path. This constraint describes the communication rule.

3) If $indicator_{(m,n),w} = 1$, $W_{(m,n)}$ must be equal to w:

$$\forall (m,n) \in C\ \forall w \in [1,2,...,degree]: indicator_{(m,n),w} = 1 \rightarrow W_{(m,n)} = w \qquad (6)$$

This constraint assigns a wavelength to each non-zero coordinate.

4) Finally, we introduce the following constraint:

$$\forall (m,n) \in C: W_{max} \geq W_{(m,n)} \qquad (7)$$

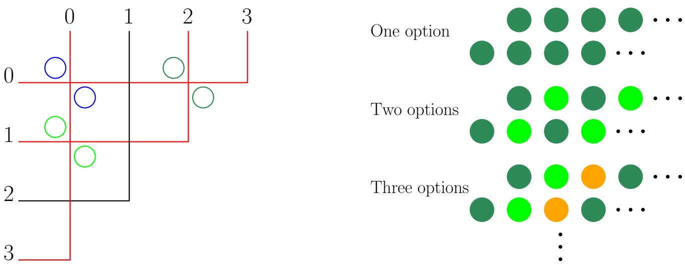
To minimize wavelength usage, we just have to minimize the biggest wavelength type number, which is W_{max}.

C. Proof 3 and Method 3: The indication of the minimal wavelength usage

We try to directly recognize the topologies requiring the minimal wavelengths without running the ILP model. To do this, we need to find an indication of the wavelength usage. In Proof 2, we have verified that the communication rule for FAST is each non-zero coordinate on a default path should be assigned with a different wavelength. If $N_{p_{default}}$ ($p_{default} \in P_{default}$) represents the number of non-zero coordinates on a default path, the minimal wavelength usage is at least $max(N_{p_{default}})$ ($p_{default} \in P_{default}$). We call this number N_{max}. **Now, we analyze** the worst case of wavelength assignment, to prove that a smaller N_{max} indicates less wavelength usage. We don't rely on N_{max} to determine the minimal wavelength usage. So it is not necessary to strictly prove that N_{max} is the minimal number of required wavelengths.

Proof 3: Fig. 7a shows the worst case (no default communication). N_{max} equals to 2 but the minimal wavelength usage is 3. This is because every two non-empty crossings are located on the same default path and thus all three non-empty crossings must not use the same wavelength (color). Essentially, if there are only two different wavelengths, when crossing $(0,0)$ (the crossing with two blue MRRs) occupies one of the two, it is impossible for crossing $(1,0)$ and crossing $(0,2)$ to have different wavelengths because there is only one option left. This can be illustrated in Fig. 7b "One option". Once $N_{max} > 2$, N_{max} is always enough to shift the colors. This analysis at least shows that N_{max} is a strong indication for the minimal wavelength usage. It is enough for our implementation.

Method 3: In the optimization process, we select the topologies with the smallest N_{max}. Then use the ILP model to



(a) The worst case

(b) Assignment illustration

Fig. 7: Wavelength assignment analysis. Different colors in (a) and (b) represent different wavelengths.

determine the wavelength usage and assign a wavelength to each MRR only for the selected topologies. This method vastly accelerates the algorithm.

IV. Optimization Process

In this section, we introduce the fully automatic optimization process of FAST.

Input: (1) A communication network e.g. Fig. 2a and insertion loss values [5] [9].

Output: Multiple optimized topologies with their communication dictionaries, MRR usage, the worst-case insertion loss, minimal wavelength usage and wavelength assignment information.

Step 1: Find redundant senders and receivers. Clear out all empty default paths in the topology as shown in Fig. 3b. Then we randomly order the rest senders/receivers, make communication dictionaries for senders and receivers based on this port order. Then we generate an initial matrix based on the communication dictionaries.

Next, reorder the senders and the receivers to generate new communication dictionaries and new matrices. To reduce variation numbers, we propose three reduction techniques:

- Fix empty default paths as shown in Fig. 3b. Empty default paths can be directly removed in physical layout.
- Fix all crossings with two MRRs. If this structure is broken down, the two MRRs could occupy two crossings. We want more empty crossings because they are removable in physical layout.
- Fix the paths of all default communications. If default communications become communications requiring MRRs, extra MRRs have to be added in the topology.

With these reduction techniques, the search space is significantly reduced. Variations which are worse than the initial topology are ignored. For dense communication networks, only one or few topologies are generated. For sparse networks, thousands of topologies can be generated within one second. In the code, we restrict the topology generation time to one second. It is enough to find multiple topologies which are equivalent or better than the state of the art.

Step 2: Sweep through all generated topologies, select the topologies with the smallest N_{max}.

Step 3: Launch the ILP model to determine the minimal wavelength usage and assign a wavelength to each MRR for every selected topologies.

Step 4: Sweep through the selected topologies again. Collect topologies which simultaneously have the minimal MRR usage, the smallest worst-case insertion loss and the minimal wavelength usage. Finally, multiple optimized topologies with their communication dictionaries, MRR usage, the worst-case insertion loss, minimal wavelength usage and wavelength assignment information are printed as outputs.

V. Experimental Results

We use Python to implement FAST. The ILP model is solved by Gurobi [11], a mixed integer linear programming solver. To compare FAST and CustomTopo comprehensively, we test all the cases tested in [9]. The results are shown in Table I.

A. General Comparison

CustomTopo runs on a computer with dual Xeon processors under 2.67GHz base frequency [9]. FAST runs on a Core i5-8265U single processor computer under 1.6GHz base frequency. Despite running on a much weaker computer, FAST is still much faster and provides multiple competitive topologies. Due to changeable port orders, FAST has a direct connection with physical layout, making it perform even better when considering physical constraints e.g. physical port locations.

- Case 1 and 5 represent dense communication networks. FAST outputs one optimized result for each case due to the reduction techniques. When the information of physical port locations is given to FAST, FAST can directly generate the topology with matched port orders. For case 1, FAST is 1233 times faster and provides better results in MRR usage, the worst-case insertion loss and equivalent result in wavelength usage. For case 5, FAST is 3450 times faster and provides results as good as CustomTopo.
- Case 2, 3, 4, 6, 7 represent sparse networks. FAST is 8.5 to 2333 times faster and provides multiple optimized variations with the same properties regarding MRR usage, wavelength usage and the worst-case insertion loss. For case 2, FAST is better in all aspects and 147 times faster. In case 3, 4, 6, the worst-case insertion loss in FAST is slightly bigger than in CustomTopo, but the half-matrix structure and multiple variations help FAST perform better in physical layout.

In next subsection, we introduce two exclusive features in FAST which can eliminate empty crossings and waveguide detours in physical layout.

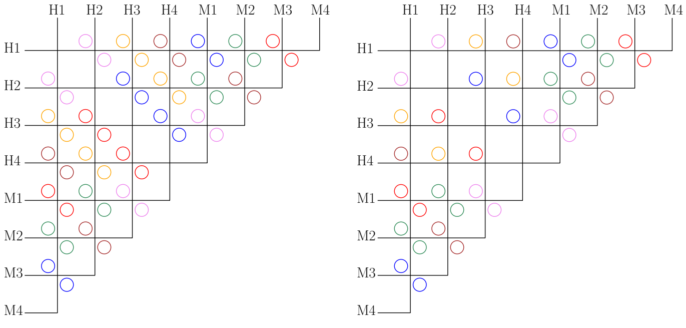
B. Discussion: physical layout

Fig. 8a and Fig. 8b show the optimized topologies of CustomTopo and FAST for case 1, respectively. Two topologies

TABLE I: Comparison between CustomTopo and FAST

Idx	d	N	Method	MRR	W	$\mathbf{I_{worst}}$	V	Time
1	8	44	CustomTopo	48	7	0.85	1	53s
			FAST	36	7	0.835	1	0.04s
2	12	26	CustomTopo	26	8	0.8	1	184s
			FAST	24	7	0.77	4	1.25s
3	12	20	CustomTopo	18	5	0.6	1	14s
			FAST	14	5	0.64	7	1.65s
4	16	22	CustomTopo	20	7	0.7	1	13s
			FAST	19	7	0.73	5	1.50s
5	8	48	CustomTopo	40	6	0.9	1	138s
			FAST	40	6	0.9	1	0.04s
6	8	24	CustomTopo	24	7	0.8	1	3s
			FAST	20	6	0.82	10	0.31s
7	8	24	CustomTopo	24	7	0.8	1	63s
			FAST	24	6	0.8	1	0.03s

Idx: index of test cases; **d**: *degree* (size of communication matrix, 8 means 8 × 8 communication matrix.); **N**: total number of communications in the network; **MRR**: total number of MRRs; **W**: total number of wavelengths. **(In CustomTopo [9], only wavelengths assigned to ADFs are counted, one more wavelength has to be added for default communications.)**; $\mathbf{I_{worst}}$: the worst-case insertion loss in dB; **V**: number of variations; **Time**: the program runtime in seconds.



(a)

(b)

Fig. 8: (a) Optimized topology of CustomTopo for case 1. (b) Optimized topology of FAST for case 1.

have exactly the same structure, signal paths and port orders while CustomTopo has 12 more MRRs which are redundant. FAST can directly use the physical layout for case 1 implemented in CustomTopo [9] and cut 12 redundant MRRs. This guarantees a better performance.

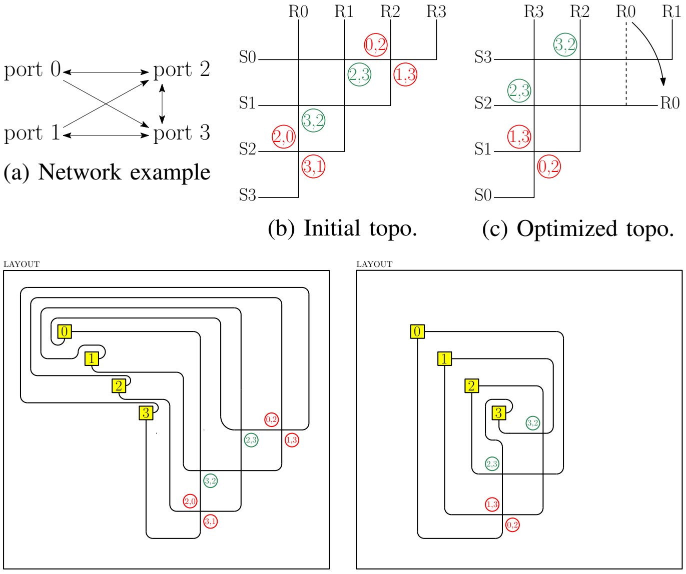
Moreover, in physical layout, the inconsistent sender/receiver orders in FAST help eliminate empty crossings inside the topology while not adding crossings outside the topology. Fig. 9c shows an optimized topology with inconsistent sender/receiver orders for the network in Fig. 9a. Sender order is S3, S2, S1, S0, receiver order is R3, R2, R0, R1 (note the order difference between Fig. 9b and Fig. 9c). This inconsistency enables the elimination of empty crossings both inside and outside the topology in physical layout.

Most importantly, the state of the art ignores the physical position of communication ports. In physical layout, if the orders of senders/receivers in topology do not match with the physical port locations, waveguide detours have to be introduced. For example, the layouts in Fig. 9d and Fig. 9e have equivalent chip areas and physical port locations. Fig. 9e has significantly shorter waveguides due to matched port orders. In FAST, the sequence of senders and receivers are changeable. If the physical information is given to FAST, FAST can generate topologies with matched port orders. This is especially beneficial to dense networks because different port orders won't worsen the topology but will significantly improve the layout. For sparse networks, FAST always provides multiple topology variations for the physical layout to find the best tradeoff between topology and layout.

Based on the half-matrix topology, FAST proposes inconsistent port orders and multiple optimized variations to eliminate empty crossings and avoid waveguide detours. These two features give FAST a direct connection with physical layout and make FAST not only an efficient topology customization algorithm, but also a promising layout platform.

VI. Conclusion

In this work, we propose FAST, a general WRONoC topology customization and optimization method for application-specific designs. The combination of an ILP model and a special sweeping technique makes FAST ten to thousands times faster than the state of the art while providing multiple better or equivalent topologies. Moreover, inconsistent port orders and different variations help FAST avoid empty crossings and waveguide detours in physical layout. FAST is also an efficient platform to optimize other properties like crosstalk and signal-to-noise ratio (SNR). This will be addressed in future work.



(a) Network example

(b) Initial topo.

(c) Optimized topo.

(d) Layout using (b)

(e) Layout using (c)

Fig. 9: Physical layout comparison. (b) The initial topology for (a). (c) One of the optimized topologies for (a).

References

[1] T.-M. Tseng, A. Truppel, M. Li, M. Nikdast, and U. Schlichtmann, "WavelengthRouted Optical NoCs: Design and EDA - State of the Art and Future Directions," in Int. Conf. Comput.-Aided Des., pp. 1-6, 2019.

[2] 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 IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, Iss. 12, pp. 5197-5210, 2020.

[3] M. Li, T.-M. Tseng, M. Tala, and U. Schlichtmann, " Maximizing the Communication Parallelism for Wavelength-Routed Optical NetworksOn-Chips," in 25th Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 109-114, 2020.

[4] W. Bogaerts et al., "Silicon microring resonators," in Laser Photonics Rev., pp. 47-73, 2012.

[5] M. Nikdast et al., "Crosstalk noise in WDM-based optical networks-on-chips: A formal study and comparison," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 11, pp. 2552-2565, 2015.

[6] L. Ramini et al., "Contrasting wavelength-routed optical noc topologies for power-efficient 3D-stacked multicore processors using physical-layer analysis," in Proc. Design, Automation, and Test Europe Conf., pp. 1589-1594, 2013.

[7] M. Brière et al., "System level assessment of an optical noc in an mpsoc platform," in Proc. Design, Automation, and Test Europe Conf., pp. 1084-1089, 2007.

[8] X. Tan et al., "On a scalable, non-blocking optical router for photonic networks-on-chip designs," in Symp. Photonics and Optoelectronics (SOPO), 2011.

[9] 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 Int. Conf. Comput.-Aided Des., 2018.

[10] S. L. Beux et al., "Optical ring network-on-chip (ORNoC): Architecture and design methodology," in Design, Automation, and Test Europe Conf., pp. 788-793, 2011.

[11] Gurobi Optimization, Inc., Gurobi Optimizer Reference Manual. http://www.gurobi.com.