Technische Universität München
Department of Electrical and Computer Engineering
Chair of Electronic Design Automation

Crosstalk-Aware Automatic Topology Customization and Optimization for Wavelength-Routed Optical NoCs

Master Thesis

Moyuan Xiao

Technische Universität München
Department of Electrical and Computer Engineering
Chair of Electronic Design Automation

Crosstalk-Aware Automatic Topology Customization and Optimization for Wavelength-Routed Optical NoCs

Master Thesis

Moyuan Xiao

Supervisor :	Dr.-Ing. Tsun-Ming Tseng
Supervising Professor :	Prof. Dr.-Ing. Ulf Schlichtmann
Topic issued :	09.01.2020
Date of submission :	14.10.2020

Moyuan Xiao
Arcisstraße 21
D-80333 Munich Germany

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 the 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.

Crosstalk is another important aspect for WRONoC topologies. It determines the most important property of an optical signal, signal-to-noise ratio (SNR), together with the insertion loss. Crosstalk analysis on the router level is complicated. Trying to optimize crosstalk only through building ILP models will suffer serious runtime problem. This runtime problem makes the ILP model or algorithm impractical. Based on the runtime advantage of FAST, we propose a comprehensive crosstalk analysis method. This method can be implemented on any WRONoC topologies based on the parallel switching element (PSE) and the crossing switching element (CSE). In this work, we implement this method on FAST and determine the crosstalk and signal-to-noise ratio (SNR) for every optical signal in a topology.

This work is a platform for WRONoC topology optimization, tackling the optimization of MRR usage, wavelength usage, the insertion loss and crosstalk. Moreover, this work proposes a set of solutions to eliminate empty crossings and waveguide detours in the physical layout. FAST can be expanded to a comprehensive topology optimization and physical layout optimization tool, being very competitive in both the optimization results and the runtime.

Acknowledgements

I would like to thank my supervisor Dr. Tsun-Ming Tseng for giving me such challenging and meaningful topic as my Master's Thesis. His trust in my ability is the best motivation for me to conquer encountered difficulties during the past half year. In this work, I have learned mathematical modelling and the principle of optical network-on-chip. Moreover, I have developed the interest in programming, data structure and database. It is especially meaningful to have the chance to submit a paper to one of the top conferences in electronic design automation. During this process, I have experienced all the steps of paper writing from draft writing, revising to submission. Dr. Tsun-Ming Tseng has been helping me improve the paper carefully and patiently. With his constructive advices, the presentation of this work has been significantly improved.

During the revision, Prof. Schlichtmann pointed out mistakes regarding labels of equations and helped me avoid ambiguities. His comments were professional and constructive. Before the submission deadline, Prof. Schlichtmann further improved the presentation in person. His efforts make the paper smooth and professional.

Contents

List of Figures

List of Tables

1. 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 (Jiang et al. 2013) (Tseng et al. 2019). 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). 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) (Vantrease et al. 2008) (Li et al. 2020). 2) Silicon Microring Resonantor (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 (Bogaerts et al. 2012). As shown in Fig. 1.1, if an optical signal is resonant with an MRR, it changes its direction (also called drop) 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 (Nikdast et al. 2015). As shown in Fig. 1.2a, an optical signal loses power when it travels along a waveguide. This power loss is called propagation loss. Propagation loss strictly relies on the physical layout. It can not be tackled in topology design. As shown in Fig. 1.2b, an optical signal suffers crossing loss when it passes through a waveguide crossing. In Fig. 1.2c, when a nonresonant optical signal passes by an MRR, it suffers a passing loss. Last but not least, when a resonant signal drops by a resonant MRR, it suffers a drop loss (Fig. 1.2d). The exact values of different insertion losses are shown in Table 1.1. A drop loss is equal to 11% of the total input power. The propagation loss that an optical signal suffers after it travels through one centimeter of waveguide is equal to 6% of the total input power. A crossing loss costs 0.6% of the input power and a passing loss costs 0.115% of the input power. From these values, it is clear that drop loss is the biggest insertion loss source. In WRONoCs, the worst-case insertion loss among all signals is an important property because it determines the required laser power. To minimize the worst-case insertion loss, we only allow each signal drops maximal once in this work. Beside the insertion losses shown in Table 1.1, there is also a kind of insertion loss called bending loss. In physical implementation of the logic topologies, optical signals suffer bending loss when the waveguides bend. The value of bending loss is $-0.005\ dB/90°$. In this work, we focus on topology generation, and thus the propagation loss and bending loss are not taken into consideration.

Table 1.1.: Insertion loss values.

Insertion loss types	**Value**
drop loss	$-0.5\ dB$
propagation loss	$-0.274\ dB/cm$
crossing loss	$-0.04\ dB$
passing loss	$-0.005\ dB$

Typical manually-designed WRONoC topologies such as Folded Crossbar (Ramini et al. 2013), Lambda Router (Bri‘ere et al. 2007) or GWOR (Tan et al. 2011) 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 (Li et al. 2018). 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.

To realize automatic topology customization and optimization, (Li et al. 2018) presents CustomTopo. CustomTopo includes the WRONoC topology structure and its communications into an integer-linear-programming (ILP) 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. 1.3c. This leads to numbers of empty crossings (crossings with no MRRs),

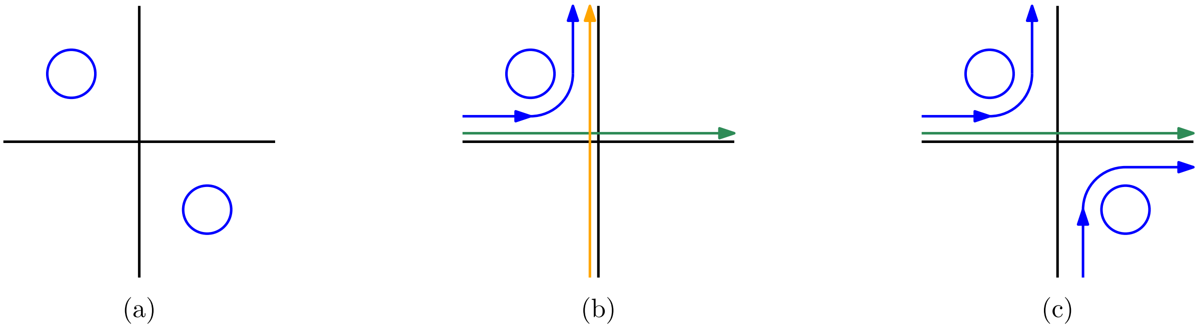
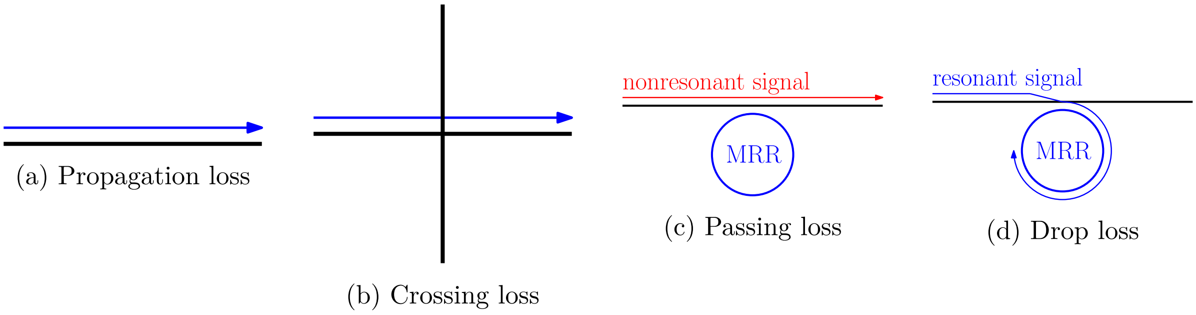


Figure 1.1.: Basic components and routing behavior of WRONoC. (a) A waveguide crossing with 2 MRRs. (b)(c) Blue signals are resonant with the MRRs. Orange and green signals are nonresonant with the MRRs.

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. 1.1a. 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 the physical layout.



(a) Propagation loss

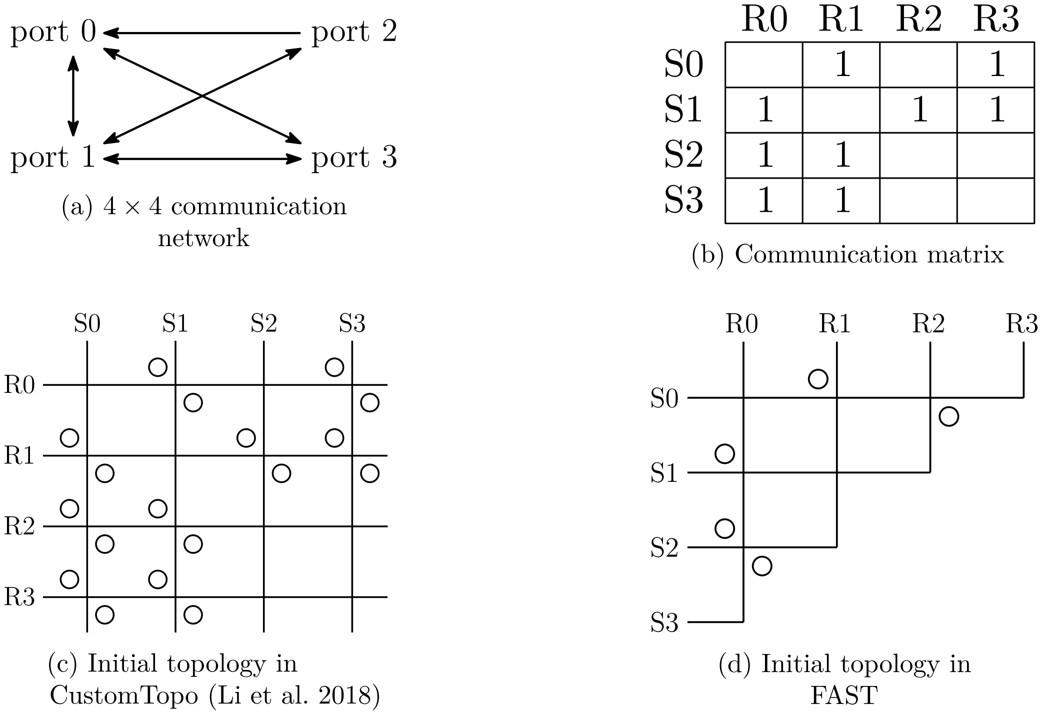
(b) Crossing loss

(c) Passing loss

(d) Drop loss

Figure 1.2.: Different kinds of insertion losses.

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. 1.3d) 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.



(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 (Li et al. 2018)

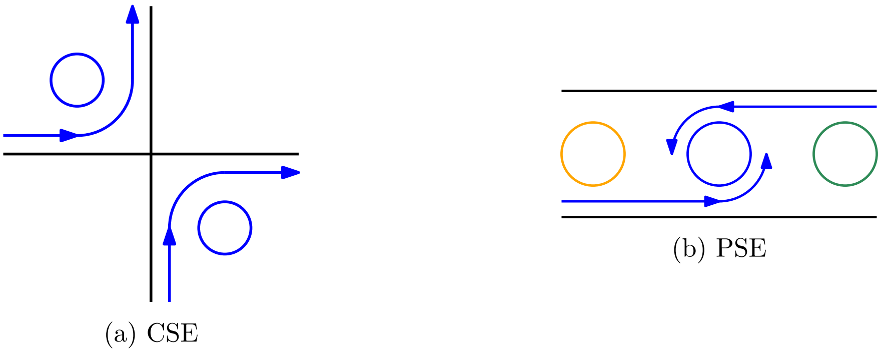
(d) Initial topology in FAST

Figure 1.3.: 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.

CustomTopo is implemented on a computer with 2 × Xeon processors under 2.67GHz base frequency. We implemented all test cases presented in (Li et al. 2018) with FAST on an Intel Core i5-8265U single processor computer under 1.6GHz base frequency. For dense communication networks, FAST is thousands times faster than CustomTopo while providing better or equivalent solutions despite running on a much weaker computational platform. For sparse communications, FAST is ten to thousands times faster while being able to provide multiple topologies which are all equivalent or better than CustomTopo regarding resources usage and the worst-cast insertion loss. Moreover, FAST introduces a set of solutions to eliminate redundant crossings and waveguide detours in physical layout. These features connect FAST with the physical layout strongly.

Based on the runtime advantage of FAST, we are able to analyze and optimize another important feature in WRONoC, crosstalk. Crosstalk determines the signal-to-noise ratio (SNR) of a signal together with the insertion loss. Crosstalk is the power leakage during signal transmission when optical signals passing by optical routing elements e.g. crossings and MRRs.

Due to the nonideality of routing elements, small portions of optical signals always go to the wrong track, disturbing the valid signals. These small portions of signals become noises in the communication network, also called crosstalk. In this work, we propose a comprehensive crosstalk analysis method which can be applied on any WRONoC topologies built based on the crossing switching element (CSE) or the parallel switching element (PSE) shown in Fig. 1.4. This algorithm can be used to compare the SNR of representative WRONoC topologies, or be integrated into the optimization process of FAST to further optimize crosstalk and SNR.


(a) CSE

(b) PSE

Figure 1.4.: Basic routing units.

The rest of this thesis is organized as follows. In Chapter 2, we introduce the logic structure of the topology used in FAST and explain the basic optimization idea. In Chapter 3, we propose three solutions to generate initial topologies, assign wavelengths and find the minimal wavelength usage. Three proofs are presented to theoretically support the validity of these three solutions, respectively. In Chapter 4, the optimization process of FAST is introduced step by step. To make the algorithm more efficient, we propose three reduction techniques in this chapter. In Chapter 5, we demonstrate the optimization results of FAST regarding MRR usage, wavelength usage, the worst-case insertion loss and the runtime, using the test cases used in (Li et al. 2018). In this chapter, FAST is compared with the state-of-the-art topology customization method: CustomTopo (Li et al. 2018). In Chapter 6, a comprehensive router-level crosstalk analysis method is proposed. We implement this method on the logic topology of FAST, calculating the worst case SNR for every test case used in (Li et al. 2018). In the last part of this chapter, we discuss interchannel crosstalk and intrachannel crosstalk and show the superiority of FAST in avoiding interchannel crosstalk. In Chapter 7, some important thoughts about how to improve or expand the current work are proposed. These future works can further significantly improve FAST and build a complete topology customization and layout optimization tool. This chapter shows the promising future of the platform built in this master's thesis. In the last chapter, Chapter 8, we briefly conclude this work.

2. Initial Topology and General Optimization Idea

2.1. Logic Scheme of the Initial Topology

We modify Snake, a WRONoC topology proposed in (Ramini et al. 2013), and use the modified version as the logic scheme of the initial topology. As shown in Fig. 2.1a, 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**.

The feasibility of the initial topology under physical constraints is another important concern. A comprehensive comparison of the layout efficiency of different ONoC logic topologies under practical physical constraints has demonstrated the superiority of the Snake topology (Ramini et al. 2013). Snake is the most competitive logic topology regarding the worst-case insertion

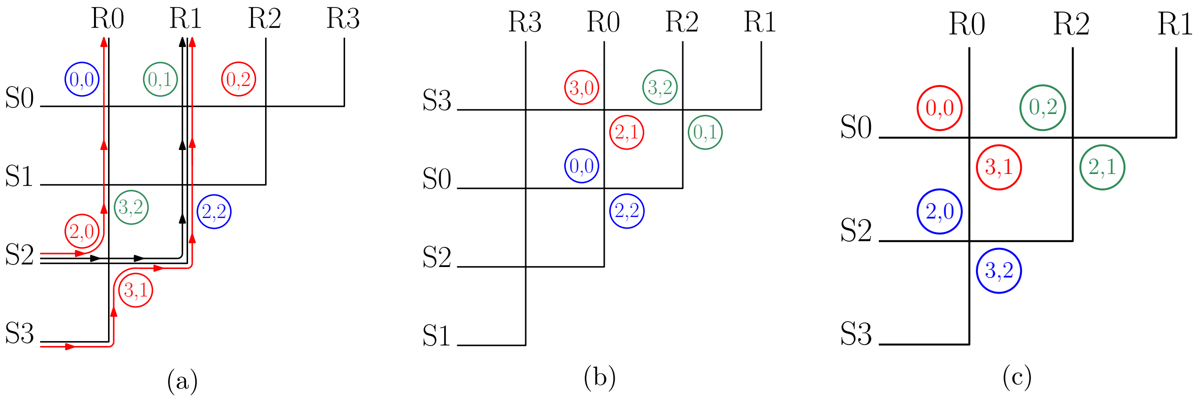


Figure 2.1.: Logic schema of the initial topology and an optimization example.

loss and power consumption compared to all other logic topologies including Folded Crossbar, Lambda Router, GWOR and ORNoC (Beux et al. 2011). Using Snake as the logic scheme shows superiority in physical layout, therefore we use it as the logic scheme of the initial topology.

2.2. 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. 2.1a, 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. 2.1b) and remove that entire default path (Fig. 2.1c).

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. 2.1 by matrices, assign a wavelength to each MRR and design an optimization algorithm. In Section 3, we propose three methods to solve these problems and provide three proofs to support the validity of the methods.

3. 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 proof allows us to quickly recognize the topologies with the minimal wavelength usage.

3.1. Proof 1 and Method 1: Generate initial topology

Fig. 3.1a is a half-matrix initial topology of FAST supporting full connectivity. Fig. 3.1b is a complete-matrix topology. We notice MRRs with the same colors in Fig. 3.1a are symmetric with respect to the antidiagonal in Fig. 3.1b. 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. 3.1b is folded along the antidiagonal, the two MRRs which overlap are exactly the two MRRs associated to a crossing in Fig. 3.1a. To prove this, we have to prove: 1) The two MRRs associated to a crossing in Fig. 3.1a (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. 3.1b.

Proof 1: We call the size of the communication matrix "*degree*". For example, in Fig. 3.1a, 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. 3.1a, every two default paths have a crossing. If the

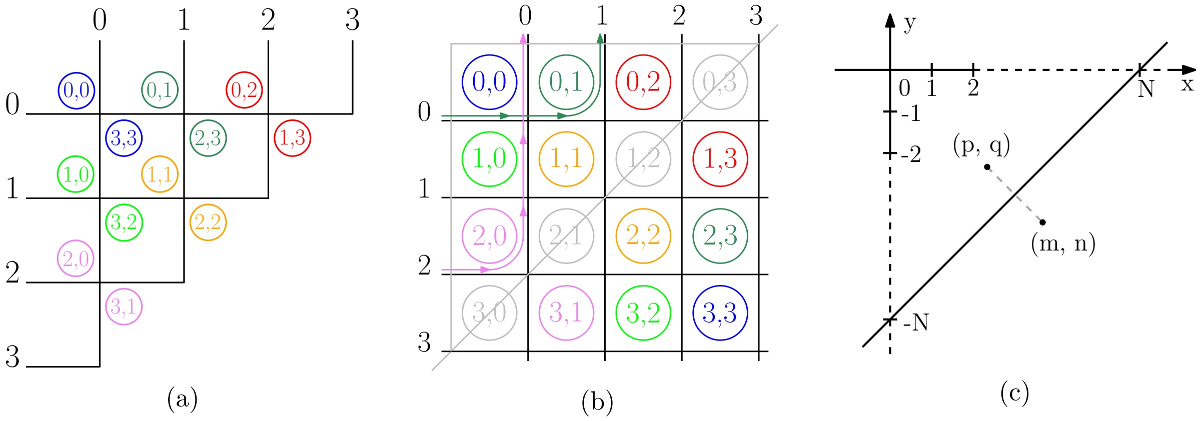


Figure 3.1.: (a) Half-matrix topology. (b)(c) Complete-matrix topology and its coordinate system expression.

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:

$$\begin{cases} m = N - q, \\ n = N - p, \quad N, p, q, m, n \in \mathbb{N} \end{cases} \tag{3.1}$$

Now, we prove: in a complete matrix like Fig. 3.1b, 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. 3.1c 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 3.1c, 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 (3.1) into:

$$\begin{cases} m = N - |q|, \\ |n| = N - p, \quad N, p, m \in \mathbb{N}, q, n \in \mathbb{Z}, q, n \leq 0 \end{cases} \tag{3.2}$$

First, using (3.2), 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. 1.3 as an example:

1. First of all, we generate a communication dictionary shown in (3.3), to clarify which sender or receiver is represented by which index number in matrix:

$$\begin{cases} \textit{sender dict} : (0 : S0, 1 : S1, 2 : S2, 3 : S3) \\ \textit{receiver dict} : (0 : R0, 1 : R1, 2 : R2, 3 : R3) \end{cases} \tag{3.3}$$

2. As shown in Fig. 3.2a, based on (3.3), we generate a complete matrix similar to Fig. 1.3b. 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. 3.2b, mirror this complete matrix with respect to the antidiagonal using (3.4):

$$A_{n\times n}^{-T} = J_{n\times n} \cdot A_{n\times n}^{T} \cdot J_{n\times n} \tag{3.4}$$

   In (3.4), A is the complete communication matrix generated in step 2, J is exchange matrix, n is the size of the square matrix, $A_{n\times n}^{-T}$ is the mirrored matrix.

4. As shown in Fig. 3.2c, add the two matrices shown in Fig. 3.2a and Fig. 3.2b. This step overlaps those MRRs which are symmetric with respect to the antidiagonal in a complete matrix.

5. As shown in Fig. 3.2d, remove non-zero values under the antidiagonal. In this matrix, each 1 represents an MRR in the upper left corner of a crossing; each 2 on the antidiagonal represents a default communication; 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**. This matrix is called **initial matrix**. It can be directly transferred to the initial topology shown in Fig. 1.3d .

3.2. Proof 2 and Method 2: Wavelength assignment

A sender must use different wavelengths for different receivers. A receiver needs to receive different wavelengths from different senders (Li et al. 2018). 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. 3.3, 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 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. 3.2d. The **outputs** are the minimal wavelength usage and the assignment of wavelengths 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 vari-**

	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	2	1
0	0	1	0
2	1	0	1
1	1	1	0

(b)

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

(c)

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

(d)

Figure 3.2.: Initial topology generation process.

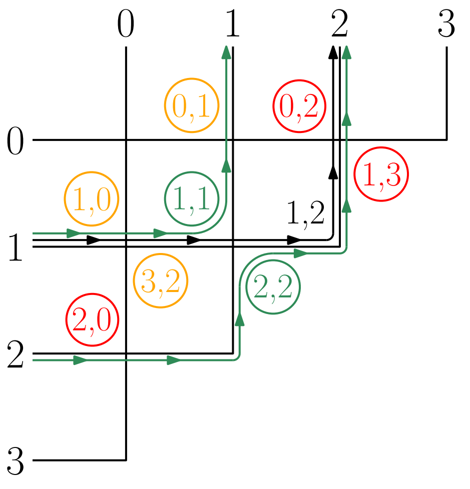


Figure 3.3.: Wavelength assignment for FAST.

able 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) isn't 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:

$$\forall (m,n) \in C : \sum_{w=1}^{degree} indicator_{(m,n),w} = 1 \tag{3.5}$$

   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:

$$\begin{gathered}\forall p_{default} \in P_{default} \; \forall w \in [1, 2, ..., degree] : \\ \sum_{(m,n) \in nzc} indicator_{(m,n),w} \leq 1\end{gathered} \tag{3.6}$$

   $p_{default}$ 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:

$$\begin{aligned} &\forall (m,n) \in C \; \forall w \in [1,2,...,degree]: \\ &indicator_{(m,n),w} = 1 \rightarrow W_{(m,n)} = w \end{aligned} \tag{3.7}$$

   This constraint assigns wavelength to each non-zero coordinate.

4. Finally, we introduce the following constraint:

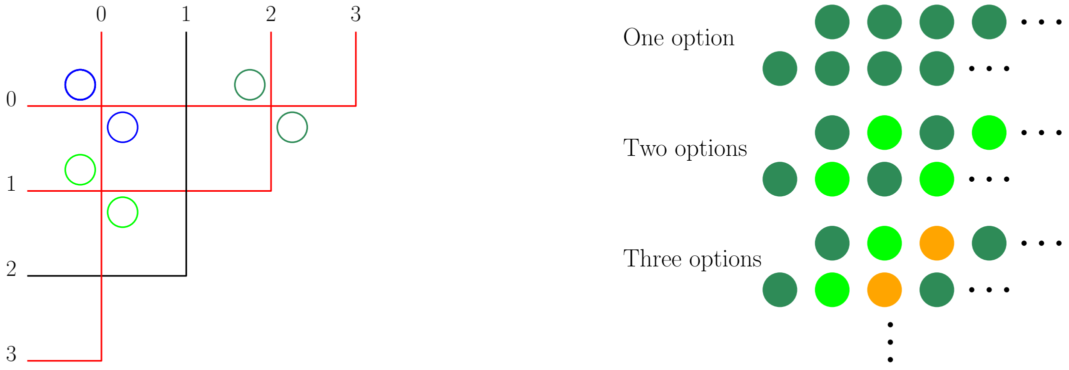
$$\forall (m,n) \in C : W_{max} \geq W_{(m,n)} \tag{3.8}$$

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

3.3. 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 have 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 $max(N_{p_{default}})$ ($p_{default} \in P_{default}$). In the rest of this work, we call this number N_{max}. **Now, we analyze** the worst case for wavelength assignment, to prove that a smaller N_{max} indicates fewer wavelengths. 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. 3.4a 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. 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. 3.4b "One option". Once $N_{max} > 2$, N_{max} is sufficient for wavelength assignment. This is shown in the following examples.

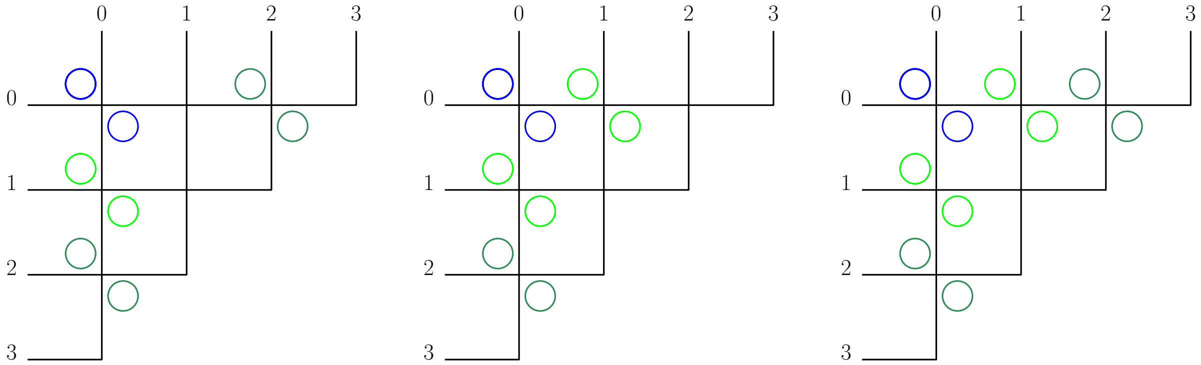


(a) The worst case

(b) Assignment illustration

Figure 3.4.: Wavelengths assignment analysis.

As shown in Fig. 3.5a, when $N_{max} = 3$, crossing (1, 0) and crossing (0, 2) can have different colors because there are two options left (green and light green). This is explained in Fig. 3.4b "Two options". The reason behind it is: When there are more than one option left, colors can always shift to avoid the same color appearing on the same default path. As shown in Fig. 3.5b and Fig. 3.5c, two options (green and light green) are always enough in different situations.



(a)

(b)

(c)

Figure 3.5.: Wavelengths assignment examples.

As shown in Fig. 3.6, when there are 3 options (green, light green, yellow), different colors can also shift to satisfy the communication rule. And there are more than one way to assign wavelengths. Fig. 3.7 shows two wavelength assignment options for a 5×5 full-communication network (with default communications). This analysis at least shows that N_{max} is a strong indication for the minimal wavelength usage. It is enough for our implementation.

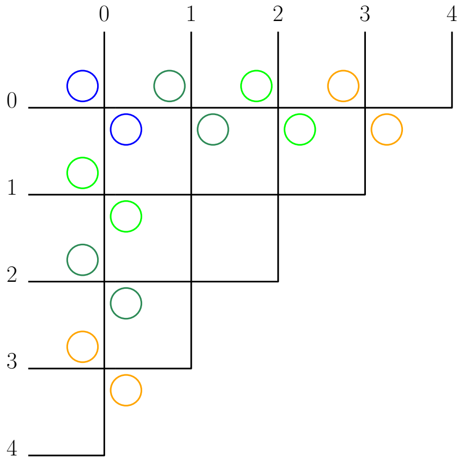
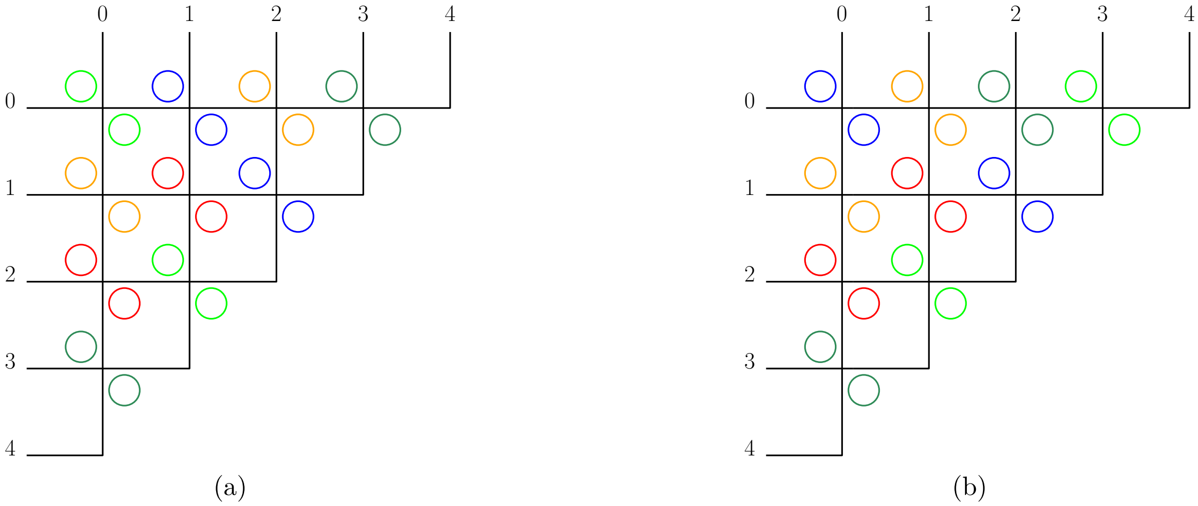


Figure 3.6.: 5 × 5 topology example.

Method 3: In the optimization process, we select the topologies with the smallest N_{max}. Then use the ILP model to determine the wavelength usage and assign a wavelength to each MRR only for the selected topologies. This method vastly accelerates the algorithm.



(a)

(b)

Figure 3.7.: 5 × 5 full-communication network wavelength assignment.

4. Optimization Process

Input: (1) A communication network e.g. Fig. 1.3a and the insertion loss values (Li et al. 2018).

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

Now, we introduce the optimization process step by step:

Step1: Find redundant senders and receivers. Clear out all empty default paths in the topology as shown in Fig. 2.1b. 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. 2.1b. Empty default paths can be directly removed in physical layout. Once they are cleared out, they should be fixed at their positions.
- 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 searching space is significantly reduced. Variations which are worse than the initial topology are ignored. Thousands of topologies which are at least as good as the initial topology 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.

Step2: Select the topologies which simultaneously have the minimal MRR usage, the smallest worst-case insertion loss and the minimal N_{max} (indication of wavelengths usage) among all generated topologies. After that, the topologies with the least amount of non-empty crossings are again selected from these optimized topologies. Less non-empty crossings means more crossings with two MRRs and more default communications. This step finds the sparsest topologies, benefiting the physical layout.

Step3: Run the ILP model to formally determine the minimal wavelengths usage and assign wavelengths to each MRR for the chosen topologies. 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.

5. Experimental Results

We use Python to implement FAST. The ILP model is solved by Gurobi (Gurobi Optimization, Inc. 2012), a mixed integer linear programming solver. To compare FAST and CustomTopo comprehensively, we test all the cases tested in (Li et al. 2018).

5.1. General Comparison

CustomTopo runs on a computer with dual Xeon processors under 2.67GHz base frequency (Li et al. 2018). 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 thousands times faster than CustomTopo for dense communication networks. For sparse networks, FAST is ten to thousands times faster and provides multiple better or equivalent 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. In Table 5.1, we calculate all the insertion loss values without considering the crossing losses generated by empty crossings in the topology. The is because in reference (Li et al. 2018), crossing losses generated by empty crossings are not considered.

- 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

Table 5.1.: 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

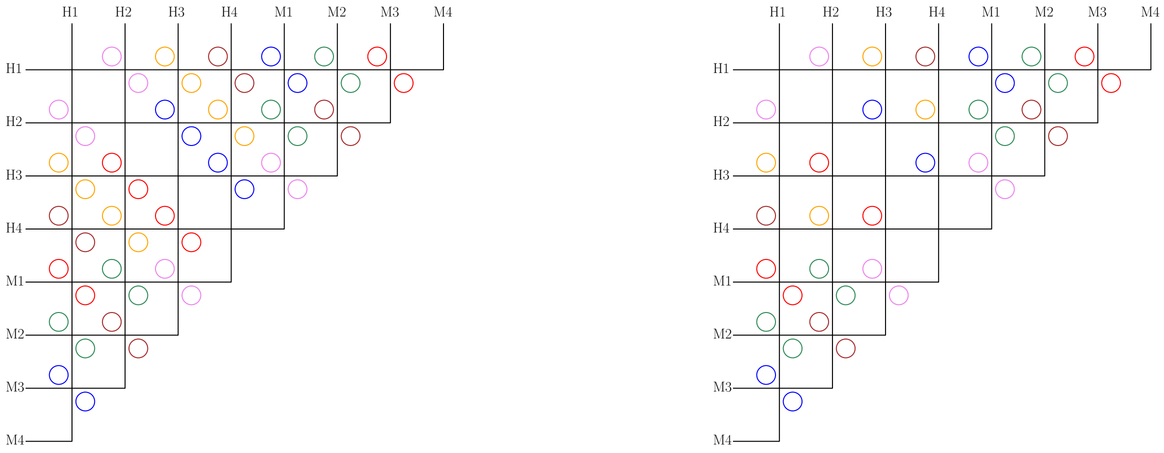
Idx: index of test cases; **d**: *degree* (size of communication matrix, 8 means 8×8 communication matrix.); **N**: total number of communications in the network; **MRR**: total number of MRRs; **W**: total number of wavelengths. (In CustomTopo (Li et al. 2018), only wavelengths assigned to ADFs are 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.

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.

5.2. Discussion: physical layout

Fig. 5.1a and Fig. 5.1b show the optimized topologies of CustomTopo and FAST for case 1, respectively. Two topologies 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 (Li et al. 2018) and cut 12 redundant MRRs. This guarantees a better performance in MRR usage and signal insertion losses.



(a)

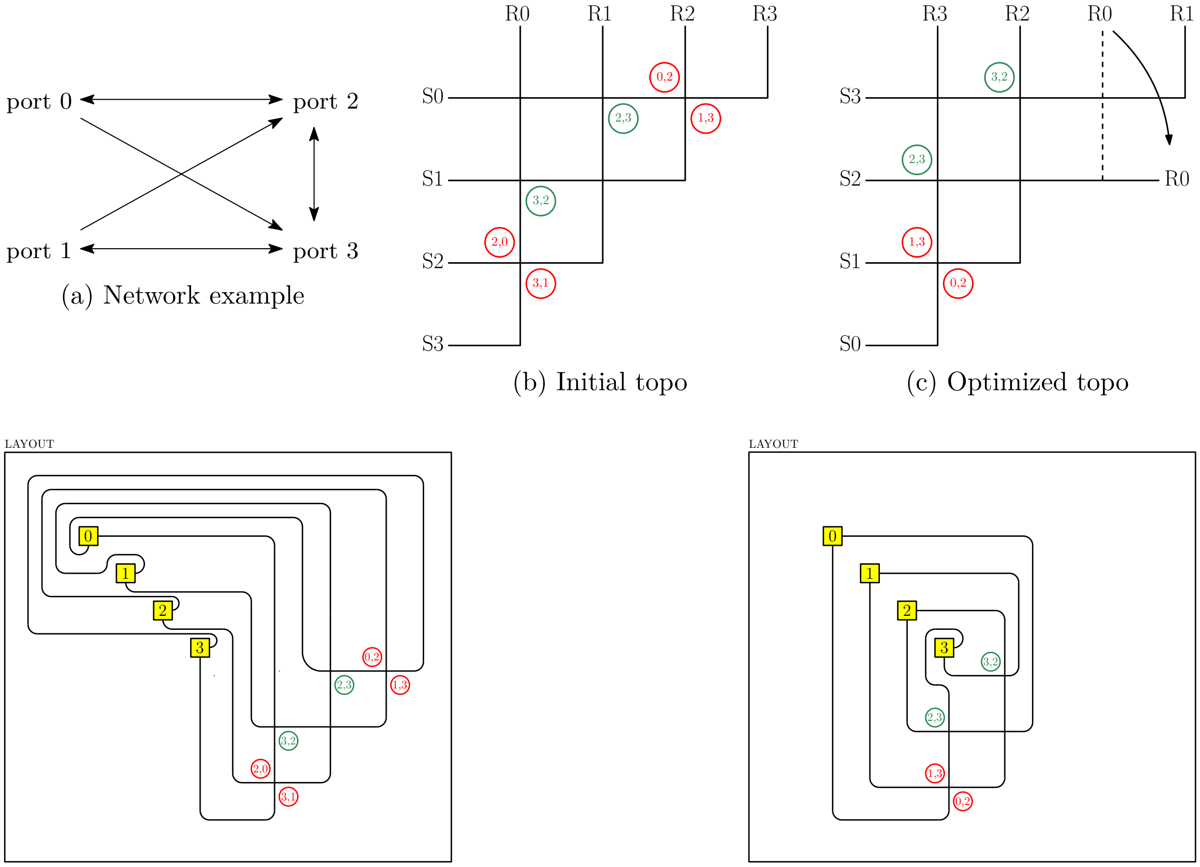
(b)

Figure 5.1.: (a) Optimized topology of CustomTopo for case 1. (b) Optimized topology of FAST for case 1.

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. 5.2c shows an optimized topology with inconsistent sender/receiver orders for the network in Fig. 5.2a. Sender order is S3, S2, S1, S0, receiver order is R3, R2, R0, R1 (note the order difference between Fig. 5.2b and Fig. 5.2c). 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. 5.2d and Fig. 5.2e have equivalent chip areas and physical port locations. Fig. 5.2e 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.



(a) Network example

(b) Initial topo

(c) Optimized topo

(d) Layout using (b)

(e) Layout using (c)

Figure 5.2.: Physical layout comparison.

6. Comprehensive Crosstalk Analysis and Verification

In wavelength-routed optical network-on-chip (WRONoC), multiple optical signals with different wavelengths are allowed to travel on the same waveguide. This feature is named as wavelength-division multiplexing (WDM) (Nikdast et al. 2015). WDM gives WRONoC ultra-high bandwidth because data flows with high throughput carried by different wavelengths can be transmitted on a single waveguide. On the other side, WDM brings WRONoCs crosstalk. Crosstalk is a common problem in data communication. In ONoCs, crosstalk is the unwanted signals or noises caused by leakage or coupling of optical signals and non-ideal routing components. It is similar to the background noise of the radio. When background noise is too loud, the valid signal is hard to be distinguished. In WRONoCs, crosstalk is even a severer problem because it seriously limits the scalability of WRONoC topologies.

In WRONoCs, optical signals suffer insertion loss during transmission. This means the power of the wanted signal decreases when passing through waveguides, MRRs or crossings. Insertion loss and crosstalk together cause the low quality of the optical signals on the receiver side. The quality of the signals on the receiver side can be quantified as signal-to-noise ratio (SNR). SNR and SNR in dB are expressed by equation 6.1 and equation 6.2, respectively.

$$SNR^{\lambda_n} = \frac{P_S^{\lambda_n}}{P_N^{\lambda_n}} \tag{6.1}$$

$$SNR_{dB}^{\lambda_n} = 10 \log \frac{P_S^{\lambda_n}}{P_N^{\lambda_n}} \tag{6.2}$$

In equation 6.1 and 6.2, λ_n represents a certain wavelength i.e. a certain optical signal. $P_S^{\lambda_n}$ is the valid signal power of signal λ_n, $P_N^{\lambda_n}$ is the total noise power for signal λ_n. From equation 6.1 and 6.2, we find: To achieve high SNR, big $P_S^{\lambda_n}$ and small $P_N^{\lambda_n}$ are simultaneously required. In early sections, we have designed a redundancy-free initial topology and a special sweeping technique to optimize the worst case insertion loss. These approaches help us achieve low insertion loss, namely a big $P_S^{\lambda_n}$. In this section, we analyze crosstalk and discuss the optimization approaches. A general crosstalk calculation algorithm is presented to quantify crosstalk and SNR for WRONoC topologies. We will also discuss the advantage of the logic topology of FAST in avoiding crosstalk systematically.

6.1. Classification for crosstalk

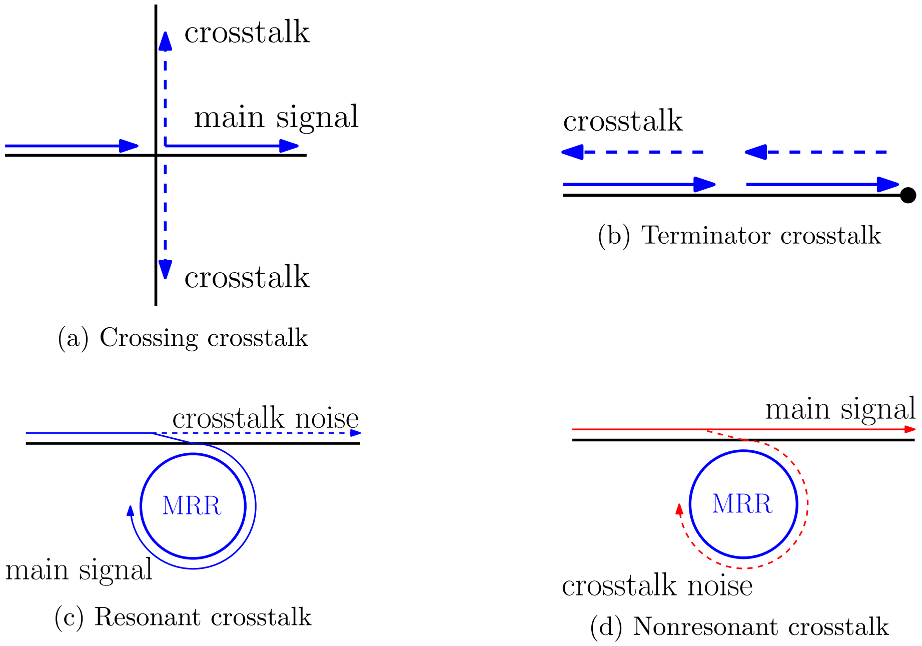
In this section, we introduce different types of crosstalk and clarify which types of crosstalk are going to be take into consideration in the formal analysis.

Due to the nonideality of the routing components in the WRONoC router, crosstalk is created when an optical signal passing by a crossing or an MRR. Besides, when a signal is terminated by an optical terminator, a small portion of crosstalk will be reflected back. Different types of crosstalk are shown in Fig. 6.1. (1) Fig. 6.1a shows crossing crosstalk. When an optical signal travels through a waveguide crossing, two portions of crosstalk will be generated to the perpendicular waveguide. (2) Fig. 6.1b shows terminator crosstalk. It is the reflection of the terminated signal by an optical terminator. (3) Fig. 6.1c shows resonant crosstalk. When an optical signal is resonant with a certain MRR, the main signal drops at the MRR while a small portion of power escapes from the MRR and becomes resonant crosstalk. (4) Fig. 6.1d shows nonresonant crosstalk. When an optical signal does not resonant with an MRR, it should ignore the MRR and go straight. But due to nonideality, a small portion of power still drops by the MRR and becomes crosstalk noise.

Table 6.1.: Values of different crosstalk.

Crosstalk Types	**Value**
crossing crosstalk	$-40\ dB$
terminator crosstalk	$-50\ dB$
resonant crosstalk	$-25\ dB$
nonresonant crosstalk	$-35\ dB$

The values of different kinds of crosstalk are shown in Table. 6.1. The values of crossing crosstalk, terminator crosstalk and resonant crosstalk are directly from reference (Nikdast et al. 2015). Now, we explain how to set the value of the nonresonant crosstalk as $-35\ dB$. To do this, some concepts have to be introduced in the first place:



(a) Crossing crosstalk

(b) Terminator crosstalk

(c) Resonant crosstalk

(d) Nonresonant crosstalk

Figure 6.1.: Crosstalk types.

- **Quality factor**: Quality factor (Q factor) is a dimensionless parameter that describes the resonance behavior of an underdamped harmonic oscillator (resonator) (Wikipedia 2020b). In WRONoCs, high Q means the ideality of the MRR is good. MRRs with high Q generate less resonant crosstalk and nonresonant crosstalk (Preston et al. 2011).

- **Free Spectral Range (FSR)** describes the range or bandwidth of the wavelengths which can be used in optical communications (Nikdast et al. 2015).

- **Interchannel crosstalk** is the crosstalk signal which has a sufficiently different wavelength from the desired signal.

- **Intrachannel crosstalk** is the crosstalk signal whose wavelength is the same with or close to the desired signal.

- **Incoherent crosstalk**: In physics, two wave sources are coherent if they have identical frequency and waveform and constant phase difference (Wikipedia 2020a). If any of these properties is different, the waves are incoherent. Incoherent crosstalk usually has different wavelengths or comes from different sources (Duong et al. 2016).

- **Coherent crosstalk**: In WRONoCs, coherent crosstalk is generated by the same optical signal and thus has identical frequency and waveform and constant phase difference. If a coherent crosstalk joins its source signal again, we don't consider it as crosstalk any more.

- **First-order crosstalk** is the crosstalk which is directly generated by valid optical signals (Duong et al. 2016).

- **Second-/third-...order crosstalk** is the crosstalk generated by other crosstalk signals. In this work, we only consider first-order crosstalk.

Reference (Preston et al. 2011) does a comprehensive analysis on FSR, wavelengths usage and crosstalk of WDM based optical interconnection. In (Preston et al. 2011), experimental results show that $r = 1.7\ \mu m$ is the minimum radius of the MRRs to achieve Q = 10000. For $r > 1.7\ \mu m$, it is found the maximum FSR covering the C-band is 50 nm (6.3 THz). This means: to keep the MRRs at a reasonable size and a high quality factor, FSR cannot exceed 50 nm. In this work, we use 50 nm as the FSR. As shown in Table 5.1, among all the test cases tested in FAST, maximal 7 wavelengths are required. This means: the distance between wavelengths in this work is at least 8 nm. Fig. 6.2 shows the crosstalk value from the 1550 nm wavelength channel on its three nearest neighbors. In Fig. 6.2, the first vertical line connects three neighbors with a distance of 0.6 nm from each other. This means: the nearest neighbor (the orange line) is 0.6 nm away from the 1550 nm wavelength; the second neighbor (the green line) is 1.2 nm away from the 1550 nm wavelength; the third neighbor (the purple line) is 1.8 nm away from the 1550 nm wavelength. Similarly, the second vertical line connects three neighbors with a distance of 1.4 nm from each other. So the purple line at the bottom represents a neighbor which is 4.2 nm away from the 1550 nm wavelength. This is the farthest neighbor shown in Fig. 6.2 and the corresponding crosstalk is $-35\ dB$. Because the minimal distance between wavelengths from the test cases in this work is 8 nm, to consider the worst case crosstalk, we use $-35\ dB$ as **the nonresonant crosstalk value**. This crosstalk value is already much higher than the actual value. So only the nearest neighbor of the resonant signal generates $-35\ dB$ of nonresonant crosstalk. The nonresonant crosstalk generated by further neighbors are negligible.

The crosstalk value of signal λ_n in dB is derived from the following equation:

$$Crosstalk_{dB}^{\lambda_n} = 10 \log \frac{P_{out}^{\lambda_n}}{P_{in}^{\lambda_n}} \tag{6.3}$$

In equation 6.3, $P_{in}^{\lambda_n}$ is the input signal or the desired signal. $P_{out}^{\lambda_n}$ is the crosstalk signal generated from $P_{in}^{\lambda_n}$. Crosstalk values derived from equation 6.3 are always negative. It represents the proportion of crosstalk signals to the source signals. In some work like (Li et al. 2018), equation 6.3 is changed to:

$$Crosstalk_{dB}^{\lambda_n} = 10 \log \frac{P_{in}^{\lambda_n}}{P_{out}^{\lambda_n}} \tag{6.4}$$

Values derived from equation 6.4 are positive. But essentially these two equations express the same meaning. In this work, we perform the computation based on equation 6.4, because it is more intuitive in the calculation process.

The value of resonant crosstalk is 25 dB. This crosstalk value means when an optical signal drops by its matched MRR, $\frac{1}{316}$ of the signal power becomes crosstalk and goes straight (shown in Fig. 6.1c). The value of nonresonant crosstalk is 35 dB. This means when an optical signal passes by a nonresonsnt MRR, $\frac{1}{3160}$ of the signal power drops by the MRR and becomes crosstalk (shown in Fig. 6.1d). Crossing crosstalk is 40 dB. When an optical signal goes through a waveguide crossing, two portions of crossing crosstalk are generated. This is shown in Fig. 6.1a. Each portion of crossing crosstalk is $\frac{1}{10000}$ of the input signal power. Terminator crosstalk (shown in Fig. 6.1b) is the smallest crosstalk which is only 50 dB. It is only $\frac{1}{100000}$ of the input signal power.

In this work, we only analyze crossing crosstalk, resonant crosstalk and nonresonant crosstalk, because optical terminator is not included in the current topology. For a given WRONoC topology, all the crosstalk can be added together to calculate the total noise power for a certain optical signal and the corresponding SNR (Chan et al. 2010). In general, we analyze **first-order incoherent intra/interchannel crosstalk**.

6.2. Formal crosstalk analysis

In this section, we introduce a general crosstalk calculation algorithm for all WRONoC topologies with a basic crossing switching element (CSE) structure. Then an implementation example on the logic schema of FAST will be presented. The computation of crosstalk on the router level is extremely complicated. It is impossible to manually verify the results. Thus, a verification

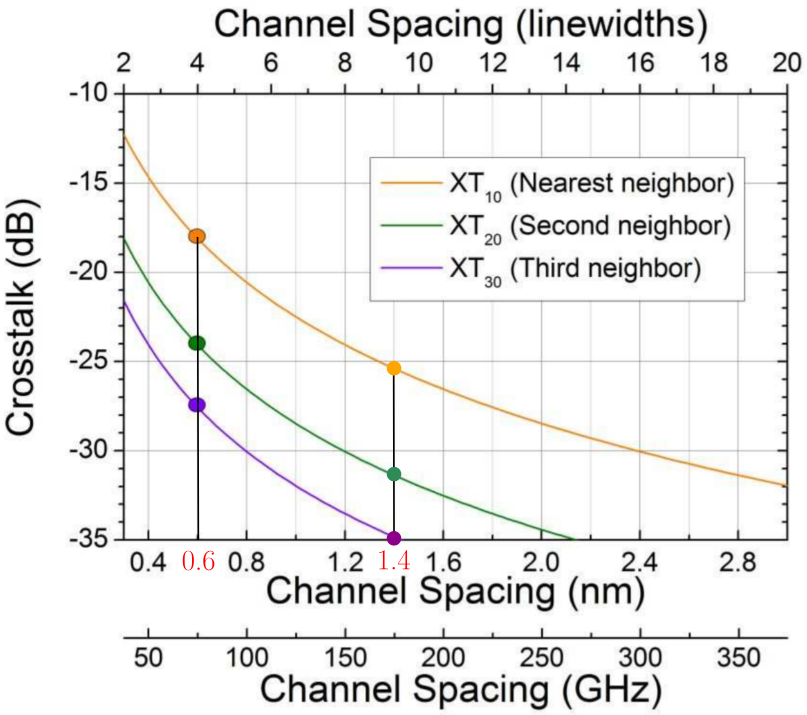
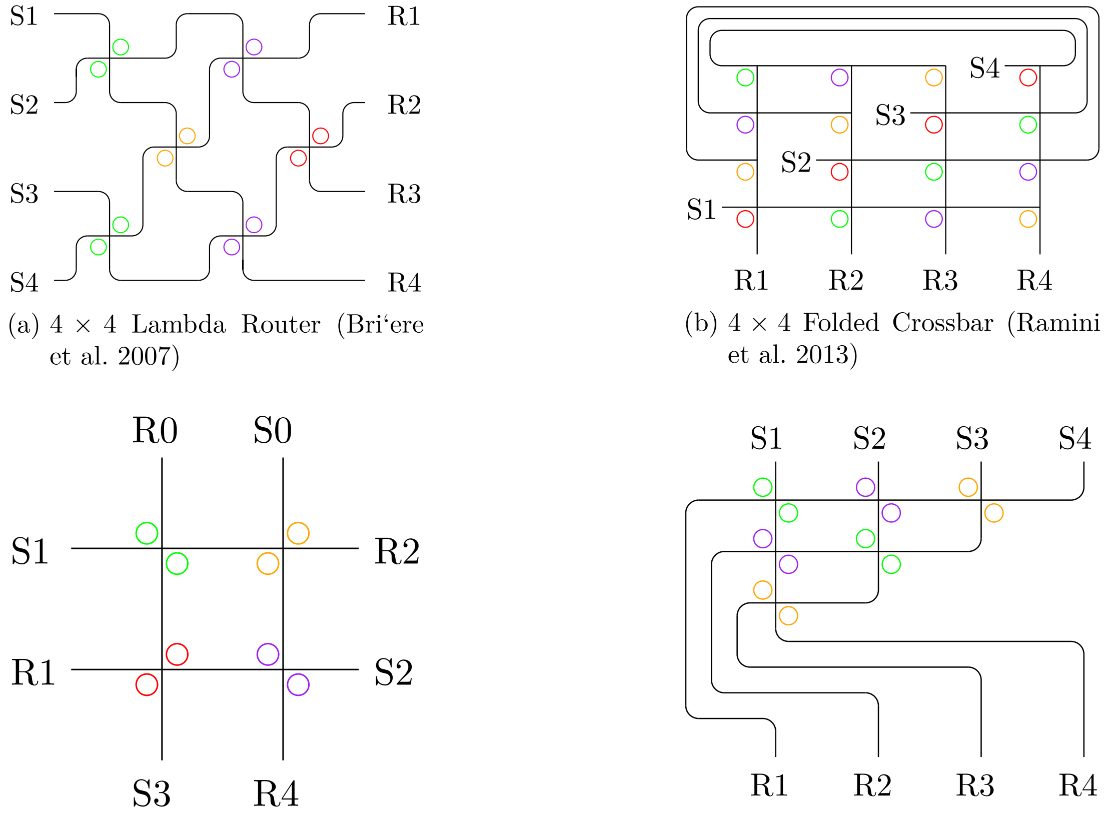


Figure 6.2.: Crosstalk from the 1550 *nm* wavelength channel on the three nearest neighbors (Preston et al. 2011).

idea will be proposed to guarantee the correctness of the crosstalk calculation algorithm.

As shown in Fig. 6.3, the state-of-the-are representative WRONoC topologies all have the basic CSE structure. It is built by a waveguide crossing and one or two MRRs. Whenever an optical signal passes through a crossing, crosstalk will be generated. Crosstalk will travel like a normal signal in waveguides (Nikdast et al. 2015). This makes the analysis of crosstalk very complicated. To verify the correctness of crosstalk calculation, we treat crosstalk exactly like a small portion of optical signal and calculate insertion losses and crosstalk all in one algorithm. In this case, if the insertion loss values are always correct, we are confident to say crosstalk values are also correct. Insertion loss has been analyzed earlier in this work. It is easier to manually verify insertion loss because the insertion loss of each signal is only impacted by its own routing path. Moreover, the state-of-the-art topology generation tool like CustomTopo (Li et al. 2018) also provides router level insertion loss analysis method. We test the same test cases used in CustomTopo (Li et al. 2018). So we can compare the insertion loss values between CustomTopo and this work. If the results from both works are the same, it guarantees the correctness of insertion loss values calculated in this work. In the crosstalk



(a) 4 × 4 Lambda Router (Bri‘ere et al. 2007)

(b) 4 × 4 Folded Crossbar (Ramini et al. 2013)

(c) 4 × 4 GWOR (Tan et al. 2011)

(d) 4 × 4 Snake (Ramini et al. 2013)

Figure 6.3.: Representative WRONoC topologies. Circles in different colors represent MRRs resonant with different wavelengths.

calculation algorithm, if the insertion loss results are always correct, we can safely say that the crosstalk values are also correct, because they go through the same computation process.

Now, based on the CSE structure, we propose a general crosstalk analysis algorithm and implement it on FAST. As shown in Fig. 6.4, CSE structure includes 4 different cases i.e. empty crossing, crossing with an MRR in the upper-left corner, crossing with an MRR in the lower-right corner and crossing with two MRRs. In a WRONoC router, these 4 variations all have **two input ports** and **two output ports**. Now, we use the logic schema of FAST as an example to propose a special data structure and the details of this crosstalk calculation algorithm.

Fig. 6.5 shows the basic structure of the topology in FAST. Senders are placed on the left side. Receivers are placed on top. Because of this arrangement, the ports on the left and downside are input ports, the ports on the right and upside are output ports. We separate

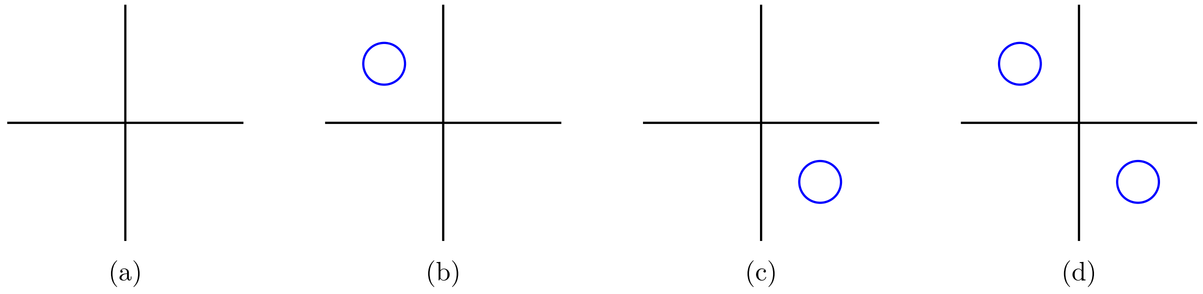


Figure 6.4.: CSE variations.

each waveguide crossing in the topology as a communication block with two input ports and two output ports. As shown in Fig. 6.6b, for a 4×4 communication network, there are 6 communication blocks. To calculate insertion loss and crosstalk all in one step, we need to connect each communication block and let data flow in the topology. Next, we design a special data structure to store the information of signals and crosstalk.

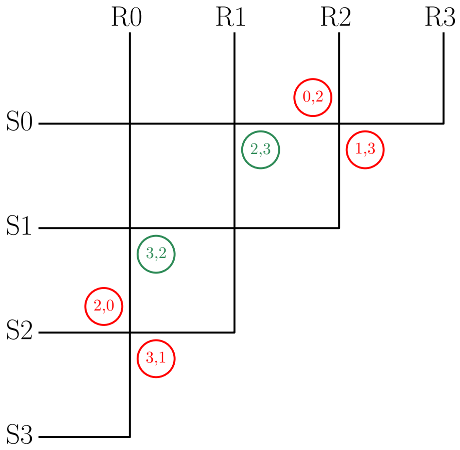
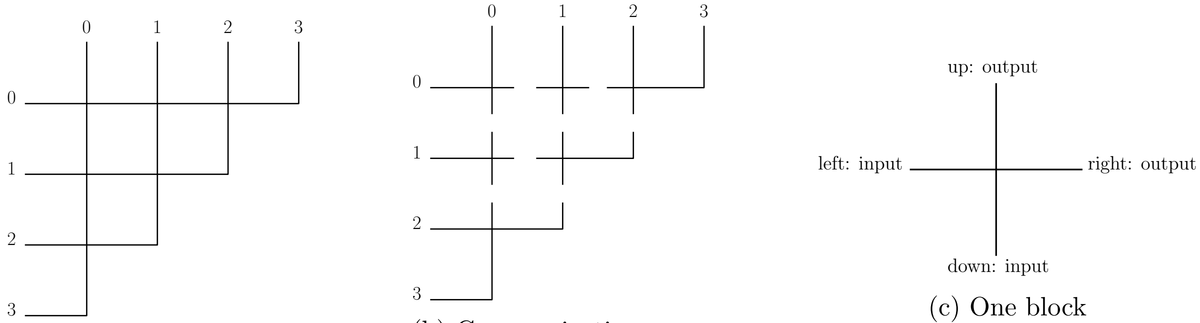


Figure 6.5.: Logic scheme of FAST.

We identify each communication block (a waveguide crossing) by its matrix index (m, n). Each communication block is a data computation center. The desired signals and crosstalk are treated equally. They get into a communication block as inputs, and then go through the internal structure of different communication blocks. Finally, the signals, crosstalk and newly generated crosstalk are stored in the output ports.

The data structure of a communication block is shown in the following:

Inside of a communication block (m, n)**:**



(a) Logic structure

(b) Communication blocks

(c) One block

Figure 6.6.: Communication blocks separation.

- Inputs:
  - Left:
    - Signals: Signal 1, Signal 2...
    - Crosstalk: Crosstalk 1, Crosstalk 2...
  - Down:
    - Signals: Signal 1, Signal 2...
    - Crosstalk: Crosstalk 1, Crosstalk 2...
- Outputs:
  - Right:
    - Signals: Signal 1, Signal 2...
    - Crosstalk: Crosstalk 1, Crosstalk 2...
  - Up:
    - Signals: Signal 1, Signal 2...
    - Crosstalk: Crosstalk 1, Crosstalk 2...

Different signals and crosstalk are identified with their wavelengths. Different crossings are identified with their coordinates in a matrix. (Here we should notice the difference between the identification of a communication and a crossing. A communication is identified with its sender index and its receiver index.) For example: $(2,0)[down][signals][3]$ is one of the signals on the down side of crossing $(2,0)$. Its wavelength is 3. $(3,1)[right][crosstalk][4]$ is one of the crosstalk on the right side of crossing $(3,1)$. Its wavelength is 4. Now, we should connect the communication blocks shown in Fig. 6.6b. We classify all the communication blocks into four groups. Fig. 6.7 shows a 4×4 matrix example, we call the size of the matrix *degree*. The *degree* of a 4×4 matrix is 4.

- The red crossing is the first group. Its left side and down side directly connect with external communication ports. This crossing is the starting point of the whole algorithm.

- The blue crossings are the second group. Their left inputs directly connect with external communication ports. Their down inputs connect with the up outputs of the crossings below them. This can be expressed with:

$$(m,0)[down] = (m+1,0)[up],\ m \in [0,1,...,degree-3] \tag{6.5}$$

- The orange crossings are the third group. Their left inputs connect with the right outputs of the crossings to the left side of them. Their down inputs connect with the right outputs of the crossings to the lower left side of them. This can be expressed with:

$$\begin{cases} (m,n)[left] = (m,n-1)[right], \\ (m,n)[down] = (m+1,n-1)[right], \quad m \in [0,1,...,degree-3],\ n \in [1,2,...,degree-2] \end{cases} \tag{6.6}$$

- The green crossings are the fourth group. Their left inputs connect with the right outputs of the crossings to the left side of them. Their down inputs connect with the up outputs of the crossings below them. This can be expressed with:

$$\begin{cases} (m,n)[left] = (m,n-1)[right], \\ (m,n)[down] = (m+1,n)[up], \quad m \in [0,1,...,degree-4],\ n \in [1,2,...,degree-3] \end{cases} \tag{6.7}$$

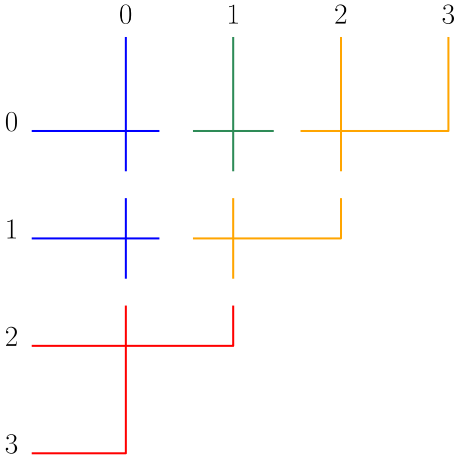


Figure 6.7.: Different block groups.

With Equation 6.5, 6.6, 6.7, all communication blocks are connected. The signals and crosstalk can flow inside the topology. In the crosstalk calculation algorithm, sequence is very important. This is because the output of one block could be the input of the other block. In a 4×4 matrix like Fig. 6.7, the calculation sequence must be:

$$(2,0) \rightarrow (1,0) \rightarrow (0,0) \rightarrow (1,1) \rightarrow (0,1) \rightarrow (0,2) \tag{6.8}$$

Fig. 6.8 shows the communication matrix which has redundant senders/receivers i.e. empty default paths. In FAST, empty default paths are systematically cleared out and placed on the left side of the topology. In physical layout, these empty default paths can be directly ignored. If a communication network has empty default paths, Equation 6.5, 6.6, 6.7 have to be changed according to the number of empty default paths.

Next, we analyze the computation process inside each communication block. Before that, we summarize all the insertion loss values and crosstalk values that are going to be used in this algorithm. This is shown in Table 6.2. There are always that 4 different cases shown in Fig. 6.4

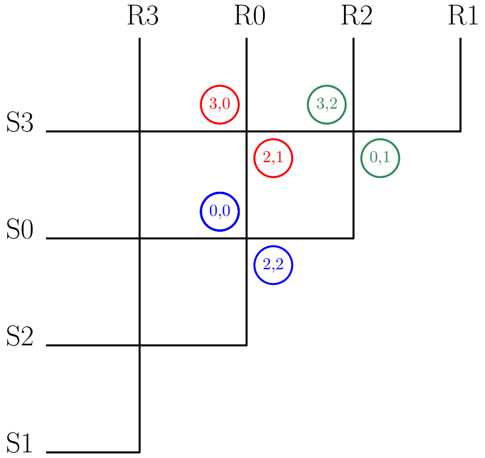


Figure 6.8.: Communication matrix with empty default path.

for no matter which kind of communication block. We go through every case step by step.

Table 6.2.: Insertion loss and crosstalk values.

Crosstalk/Insertion loss Types	**Value**
crossing crosstalk	40 dB
terminator crosstalk	50 dB
resonant crosstalk	25 dB
nonresonant crosstalk	35 dB
drop loss	0.5 dB
crossing loss	0.04 dB
passing loss	0.005 dB

For the red crossing and blue crossings shown in Fig. 6.7, we clarify the initial input values. All inputs directly connecting external communication ports have signals values 0 dB. This means the initial signals have 100% power. On the other hand, all inputs directly connecting external communication ports have no crosstalk, so the space reserved for crosstalk is empty. They are shown in the following:

Inputs of the red crossing:

- Left:
  - Signals: Signal 1 = 0 dB, Signal 2 = 0 dB,...
  - Crosstalk: empty

- Down:
  - Signals: Signal 1 = 0 dB, Signal 2 = 0 dB,...
  - Crosstalk: empty

Inputs of the blue crossings:

- Left:
  - Signals: Signal 1 = 0 dB, Signal 2 = 0 dB,...
  - Crosstalk: empty
- Down:
  - Signals: Signal 1, Signal 2...
  - Crosstalk: Crosstalk 1, Crosstalk 2...

In this algorithm, we analyze insertion loss and crosstalk for all four kinds of crossings shown in Fig. 6.7. For each kind of crossing, there are four different cases which are shown in Fig. 6.4. So totally 16 different scenarios need to be analyzed. We propose a general analysis process for all 4 kinds of crossings. As explained before, not all input ports have crosstalk, so the following process needs to be tailored for different kinds of crossings. The index of the crossing is always (m, n). For crossing crosstalk, we only consider the portions to the right direction and the up direction, because only crosstalk to these two directions will directly arrive the receivers. In the following figures, solid lines represent signals, dashed lines represent crosstalk. If a signal or a crosstalk is resonant with an MRR, they have the same color, otherwise nonresonant. Nonresonant signals are classified into two kinds: the nearest nonresonant signals generate nonresonant crosstalk when passing by a nonresonant MRR; the other nonresonant signals do not generate nonresonant crosstalk when passing by a nonresonant MRR.

1. **Empty crossing:**

   Fig. 6.9a shows that an optical signal from the left input port travels through a crossing. The signal itself suffers a crossing loss. A crossing crosstalk is generated to the up of the

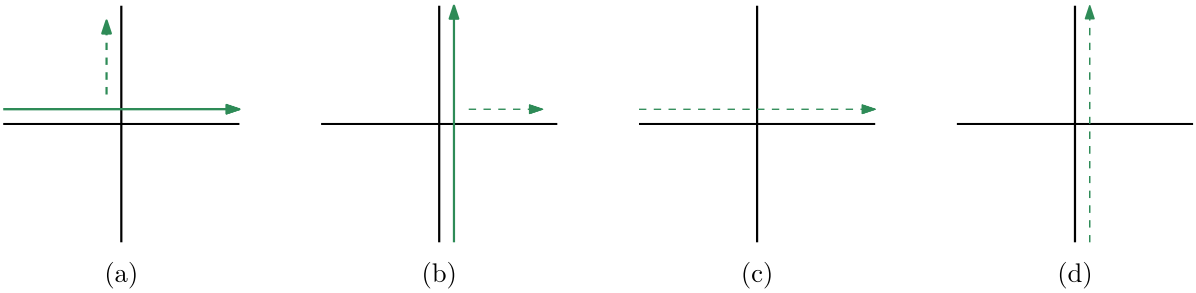


Figure 6.9.: Empty crossing. (a) Signal from left travels to right and generates a crossing crosstalk to up. (b) Signal from down travels to up and generates a crossing crosstalk to right. (c) Crosstalk from left travels to right. (d) Crosstalk from down travels to up.

crossing. These can be expressed with the following equations:

$$\begin{cases} (m,n)[right][signal] = (m,n)[left][signal] - crossing\ loss, \\ (m,n)[up][crosstalk] = (m,n)[left][signal] - crossing\ crosstalk \end{cases} \tag{6.9}$$

Fig. 6.9b shows that an optical signal from the down input port travels through a crossing. The signal itself suffers a crossing loss. A crossing crosstalk is generated to the right of the crossing. These can be expressed with the following equations:

$$\begin{cases} (m,n)[up][signal] = (m,n)[down][signal] - crossing\ loss, \\ (m,n)[right][crosstalk] = (m,n)[down][signal] - crossing\ crosstalk \end{cases} \tag{6.10}$$

Fig. 6.9c shows that a crosstalk from the left input port travels through a crossing. The crosstalk suffers a crossing loss. No crossing crosstalk is generated because we only consider first-order crosstalk. This can be expressed with the following equation:

$$((m,n)[right][crosstalk] = (m,n)[left][crosstalk] - crossing\ loss \tag{6.11}$$

Fig. 6.9d shows that a crosstalk from the down input port travels through a crossing.

The crosstalk suffers a crossing loss. No crossing crosstalk is generated because we only consider first-order crosstalk. This can be expressed with the following equation:

$$((m,n)[up][crosstalk] = (m,n)[down][crosstalk] - crossing\ loss \tag{6.12}$$

2. **Crossing with an MRR in the upper-left corner:**

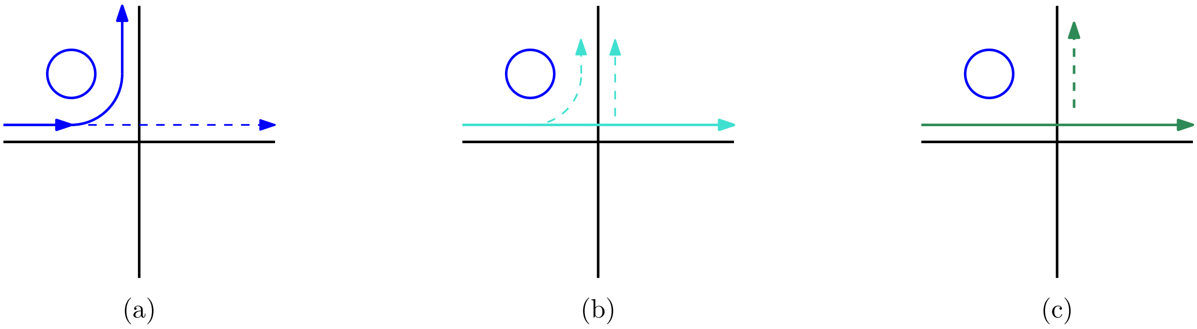


Figure 6.10.: Signals from the left. (a) Resonant signal from left drops to up and generates a resonant crosstalk to right. (b) Nearest nonresonant signals from left travel to right and generate a nonresonant crosstalk and a crossing crosstalk to up. (c) Other nonresonant signals from left travel to right and generate a crossing crosstalk to up.

Fig. 6.10a shows a resonant signal from the left input port drops on the MRR and goes to up. The signal itself suffers a drop loss. A resonant crosstalk is generated to the right of the crossing and suffers a crossing loss. These can be expressed with the following equations:

$$\begin{cases} (m,n)[up][signal] = (m,n)[left][signal] - drop\ loss, \\ (m,n)[right][crosstalk] = (m,n)[left][signal] - resonant\ crosstalk - crossing\ loss \end{cases} \tag{6.13}$$

Fig. 6.10b shows a nearest nonresonant signal from the left input port travels to right. The signal itself suffers a passing loss and a crossing loss. This can be expressed with the following equation:

$$(m, n)[right][signal] = (m, n)[left][signal] - passing\ loss - crossing\ loss \tag{6.14}$$

In the meantime, a nonresonant crosstalk and a crossing crosstalk are generated to the up of the crossing. These two crosstalks are from the same source and thus are coherent signals. We add them and get one crosstalk signal:

$$\begin{cases} (m, n)[up][crosstalk] = 10 \log(10^a + 10^b), \\ a = \frac{(m,n)[left][signal] - nonresonant\ crosstalk}{10}, \\ b = \frac{(m,n)[left][signal] - passing\ loss - crossing\ crosstalk}{10} \end{cases} \tag{6.15}$$

Fig. 6.10c shows a nonresonant signal from the left input port travels to right. The signal itself suffers a passing loss and a crossing loss. It also generates a crossing crosstalk to up. This can be expressed with the following equations:

$$\begin{cases} (m, n)[right][signal] = (m, n)[left][signal] - passing\ loss - crossing\ loss, \\ (m, n)[up][crosstalk] = (m, n)[left][signal] - passing\ loss - crossing\ crosstalk \end{cases} \tag{6.16}$$

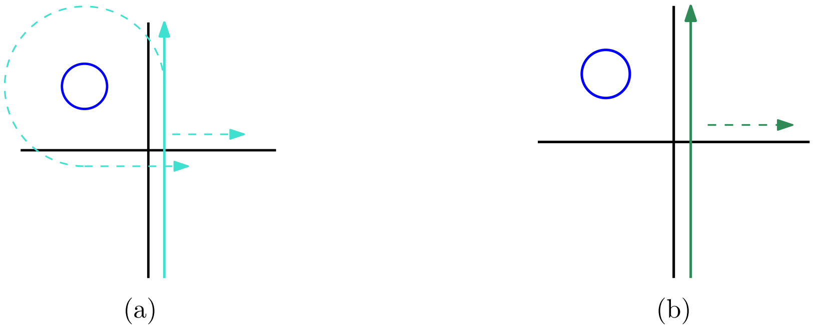


Figure 6.11.: Signals from down. (a) Nearest nonresonant signals from down travel to up and generate a crossing crosstalk and a nonresonant crosstalk to right. (b) Other nonresonant signals from down travel to up and generate a crossing crosstalk to right.

Fig. 6.11a shows the nearest nonresonant signal from the down input port travels to up. The signal itself suffers a crossing loss and a passing loss. It also generates a crossing crosstalk to right and a nonresonant crosstalk to right. The nonresonant crosstalk suffers a crossing loss. These can be expressed with the following equations:

$$(m,n)[up][signal] = (m,n)[down][signal] - crossing\ loss - passing\ loss \tag{6.17}$$

$$\begin{cases} (m,n)[right][crosstalk] = 10\log\left(10^{a} + 10^{b}\right), \\ a = \frac{(m,n)[down][signal] - crossing\ crosstalk}{10}, \\ b = \frac{(m,n)[down][signal] - crossing\ loss - nonresonant\ crosstalk - crossing\ loss}{10} \end{cases} \tag{6.18}$$

Fig. 6.11b shows other nonresonant signals from the down input port travels to up. The signal itself suffers a crossing loss and a passing loss. It also generates a crossing crosstalk to right. No nonresonant crosstalk is generated because the distance between the wavelength of the signal and the resonant wavelength of the MRR is too far. These can be expressed with the following equations:

$$\begin{cases} (m,n)[up][signal] = (m,n)[down][signal] - crossing\ loss - passing\ loss, \\ (m,n)[right][crosstalk] = (m,n)[down][signal] - crossing\ crosstalk \end{cases} \tag{6.19}$$

For a crossing with an MRR in its upper-left corner, it is impossible that a resonant signal appears in its down input port. This is because in the logic topology of FAST, one MRR is only responsible for one optical signal. If the lower-right corner of a crossing is empty, it means no resonant signal from down needs to be dropped. In a sharing structure, this situation changes because one MRR is responsible for two signals. Topologies using MRR sharing structure is not analyzed in this algorithm.

Fig. 6.12a shows a resonant crosstalk from the left input port drops by an MRR and goes to up. The crosstalk suffers a drop loss. This can be expressed with the following

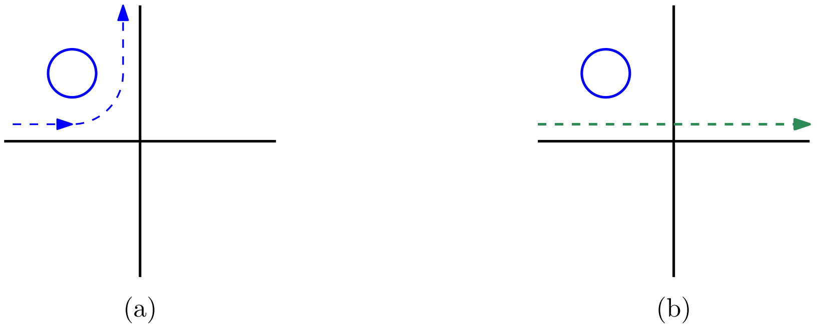


Figure 6.12.: Crosstalk from left. (a) A resonant crosstalk from left drops to up, suffering a drop loss. (b) A nonresonant crosstalk from left travels to right, suffering a passing loss and a crossing loss.

equation:

$$((m,n)[up][crosstalk] = (m,n)[left][crosstalk] - drop\ loss \tag{6.20}$$

Fig. 6.12b shows a nonresonant crosstalk from the left input port travels through a crossing to right. The crosstalk suffers a passing loss and a crossing loss. No crossing crosstalk is generated because we only consider first-order crosstalk. This can be expressed with the following equation:

$$((m,n)[right][crosstalk] = (m,n)[left][crosstalk] - passing\ loss - crossing\ loss \tag{6.21}$$

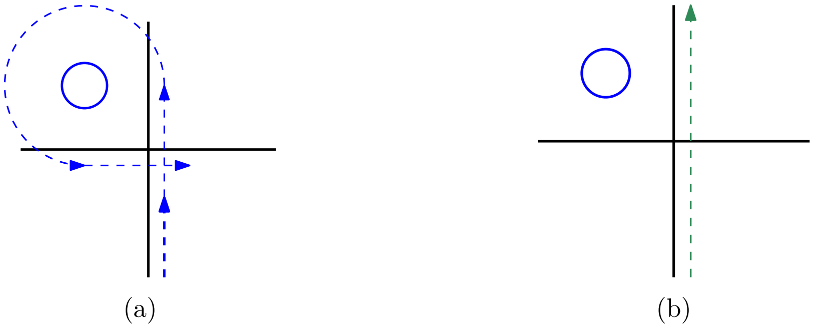


Figure 6.13.: Crosstalk from down. (a) A resonant crosstalk from down drops to right, suffering a crossing loss, a drop loss and a crossing loss. (b) A nonresonant crosstalk from down travels to up, suffering a crossing loss and a passing loss.

Fig. 6.13a shows a resonant crosstalk from the down input port drops by an MRR and goes to right. The crosstalk suffers a crossing loss, a drop loss and a crossing loss. This can be expressed with the following equation:

$$((m,n)[right][crosstalk] = (m,n)[down][crosstalk] - crossing\,loss - drop\,loss - crossing\,loss \tag{6.22}$$

Fig. 6.13b shows a nonresonant crosstalk from the down input port travels to up. The crosstalk suffers a crossing loss and a passing loss. This can be expressed with the following equation:

$$((m,n)[up][crosstalk] = (m,n)[down][crosstalk] - crossing\,loss - passing\,loss \tag{6.23}$$

3. **Crossing with an MRR in the lower-right corner:**

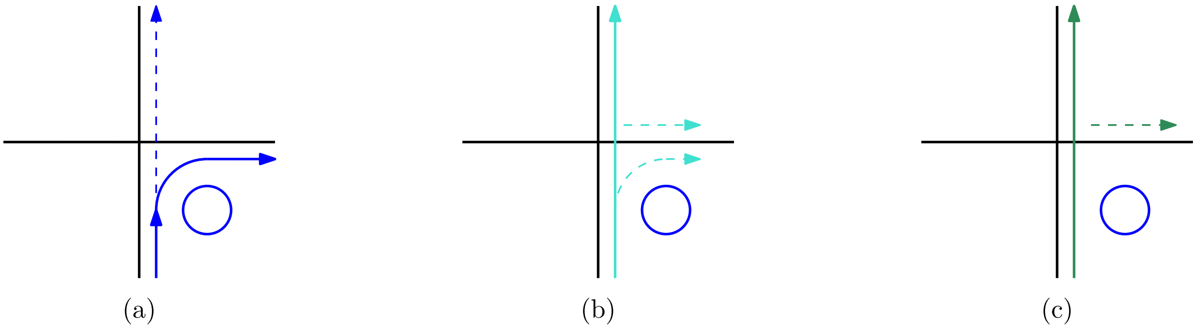


Figure 6.14.: Signals from down. (a) Resonant signal from down drops to right and generates a resonant crosstalk to right. (b) Nearest nonresonant signals from down travel to up and generate a nonresonant crosstalk and a crossing crosstalk to right. (c) Other nonresonant signals from down travel to up and generate a crossing crosstalk to right.

Fig. 6.14a shows a resonant signal from the down input port drops on the MRR and goes to right. The signal itself suffers a drop loss. A resonant crosstalk is generated to the up of the crossing and suffers a crossing loss. These can be expressed with the following equations:

$$\begin{cases} (m,n)[right][signal] = (m,n)[down][signal] - drop\ loss, \\ (m,n)[up][crosstalk] = (m,n)[down][signal] - resonant\ crosstalk - crossing\ loss \end{cases} \tag{6.24}$$

Fig. 6.14b shows a nearest nonresonant signal from the down input port travels to up. The signal itself suffers a passing loss and a crossing loss. This can be expressed with the following equation:

$$(m,n)[up][signal] = (m,n)[down][signal] - passing\ loss - crossing\ loss \tag{6.25}$$

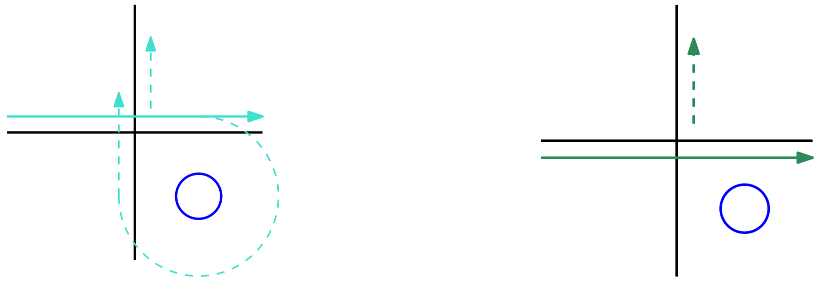
In the meantime, a nonresonant crosstalk and a crossing crosstalk are generated to the right of the crossing. These two crosstalk are from the same source and thus are coherent signals. We add them too just like case 2 and get one crosstalk signal:

$$\begin{cases} (m,n)[right][crosstalk] = 10\log\left(10^a + 10^b\right), \\ a = \frac{(m,n)[down][signal] - nonresonant\ crosstalk}{10}, \\ b = \frac{(m,n)[down][signal] - passing\ loss - crossing\ crosstalk}{10} \end{cases} \tag{6.26}$$

Fig. 6.14c shows a nonresonant signal from the down input port travels to up. The signal itself suffers a passing loss and a crossing loss. It also generates a crossing crosstalk to right. These can be expressed with the following equations:

$$\begin{cases} (m,n)[up][signal] = (m,n)[down][signal] - passing\ loss - crossing\ loss, \\ (m,n)[right][crosstalk] = (m,n)[down][signal] - passing\ loss - crossing\ crosstalk \end{cases} \tag{6.27}$$

Fig. 6.15a shows the nearest nonresonant signal from the left input port travels to right. The signal itself suffers a crossing loss and a passing loss. It also generates a crossing



(a)

(b)

Figure 6.15.: Signals from left. (a) Nearest nonresonant signals from left travel to right and generate a crossing crosstalk and a nonresonant crosstalk to up. (b) Other nonresonant signals from left travel to right and generate a crossing crosstalk to up.

crosstalk to up and a nonresonant crosstalk to up. The nonresonant crosstalk suffers a crossing loss. These can be expressed with the following equations:

$$(m,n)[right][signal] = (m,n)[left][signal] - crossing\ loss - passing\ loss \tag{6.28}$$

$$\begin{cases} (m,n)[up][crosstalk] = 10\log\left(10^{a} + 10^{b}\right), \\ a = \frac{(m,n)[left][signal] - crossing\ crosstalk}{10}, \\ b = \frac{(m,n)[left][signal] - crossing\ loss - nonresonant\ crosstalk - crossing\ loss}{10} \end{cases} \tag{6.29}$$

Fig. 6.15b shows other nonresonant signals from the left input port travels to right. The signal itself suffers a crossing loss and a passing loss. It also generates a crossing crosstalk to up. No nonresonant crosstalk is generated because the distance between the wavelength of the signal and the resonant wavelength of the MRR is too far. These can be expressed with the following equations:

$$\begin{cases} (m,n)[right][signal] = (m,n)[left][signal] - crossing\ loss - passing\ loss, \\ (m,n)[up][crosstalk] = (m,n)[left][signal] - crossing\ crosstalk \end{cases} \tag{6.30}$$

For a crossing with an MRR in its lower-right corner, it is impossible that a resonant signal appears in its left input port. This is because in the logic topology of FAST, one MRR is only responsible for one optical signal. If the upper-left corner of a crossing is empty, it means there is no resonant signal from left needs to be dropped.

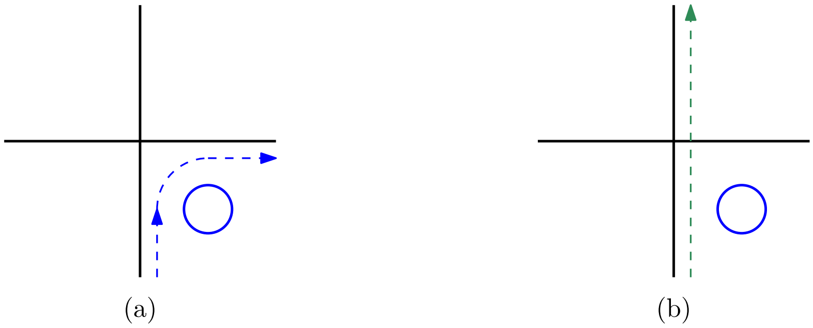


Figure 6.16.: Crosstalk from down. (a) A resonant crosstalk from down drops to right, suffering a drop loss. (b) A nonresonant crosstalk from down travels to up, suffering a passing loss and a crossing loss.

Fig. 6.16a shows a resonant crosstalk from the down input port drops by an MRR and goes to right. The crosstalk suffers a drop loss. This can be expressed with the following equation:

$$((m,n)[right][crosstalk] = (m,n)[down][crosstalk] - drop\ loss \tag{6.31}$$

Fig. 6.16b shows a nonresonant crosstalk from the down input port travels through a crossing to up. The crosstalk suffers a passing loss and a crossing loss. No crossing crosstalk is generated because we only consider first-order crosstalk. This can be expressed with the following equation:

$$((m,n)[up][crosstalk] = (m,n)[down][crosstalk] - passing\ loss - crossing\ loss \tag{6.32}$$

Fig. 6.17a shows a resonant crosstalk from the left input port drops by an MRR and goes to up. The crosstalk suffers a crossing loss, a drop loss and a crossing loss. This can

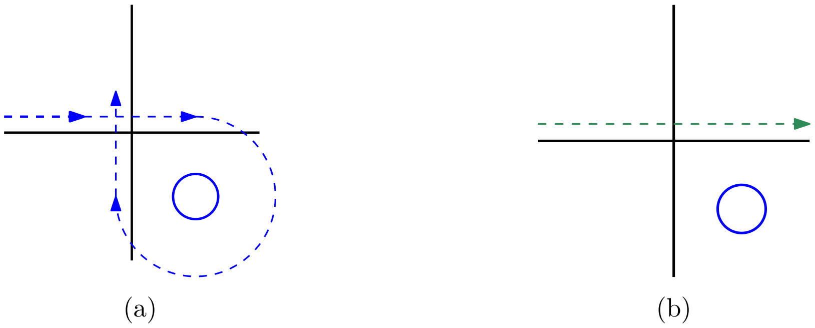


Figure 6.17.: Crosstalk from left. (a) A resonant crosstalk from left drops to up, suffering a crossing loss, a drop loss and a crossing loss. (b) A nonresonant crosstalk from left travels to right, suffering a crossing loss and a passing loss.

be expressed with the following equation:

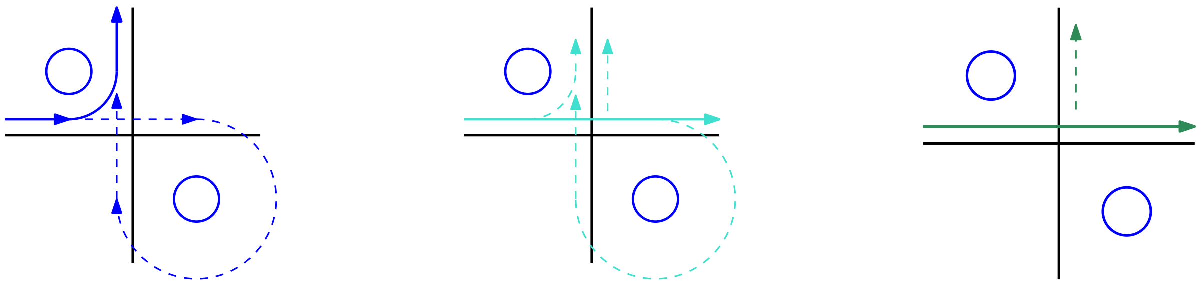
$$((m,n)[up][crosstalk] = (m,n)[left][crosstalk] - crossing\,loss - drop\,loss - crossing\,loss \tag{6.33}$$

Fig. 6.17b shows a nonresonant crosstalk from the left input port travels to right. The crosstalk suffers a crossing loss and a passing loss. This can be expressed with the following equation:

$$((m,n)[right][crosstalk] = (m,n)[left][crosstalk] - crossing\,loss - passing\,loss \tag{6.34}$$

4. **Crossing with two MRRs:**

Fig. 6.18a shows a resonant signal from the left input port drops on the MRR and goes to up. The signal itself suffers a drop loss. A resonant crosstalk is generated to the right. It is then dropped by the other MRR and finally goes up. This crosstalk is coherent with the desired signal, so we add it back into the signal. This resonant crosstalk suffers a crossing loss, a drop loss, a crossing loss and a passing loss. These can be expressed with



(a)

(b)

(c)

Figure 6.18.: Signals from the left. (a) Resonant signal from left drops to up and generates a resonant crosstalk to up. (b) Nearest nonresonant signals from left travel to right and generate a nonresonant crosstalk, a crossing crosstalk and a nonresonant crosstalk to up. (c) Other nonresonant signals from left travel to right and generate a crossing crosstalk to up.

the following equations:

$$\begin{cases} m, n)[up][signal] = 10\log\left(10^{a} + 10^{b}\right), \\ a = \frac{(m,n)[left][signal] - drop\ loss}{10}, \\ b = \frac{(m,n)[left][signal] - resonant\ crosstalk - crossing\ loss - drop\ loss - crossing\ loss - passing\ loss}{10} \end{cases} \tag{6.35}$$

Fig. 6.18b shows the nearest nonresonant signal from the left input port travels to right. The signal itself suffers a passing loss, a crossing loss and a passing loss. This can be expressed with the following equation:

$$(m, n)[right][signal] = (m, n)[left][signal] - passing\ loss - crossing\ loss - passing\ loss \tag{6.36}$$

In the meantime, a nonresonant crosstalk, a crossing crosstalk and a nonresonant crosstalk are generated to the up of the crossing. These three crosstalk are from the same source and thus are coherent signals. We add them and get one crosstalk signal:

$$\begin{cases} (m,n)[up][crosstalk] = 10\log\left(10^{a} + 10^{b} + 10^{c}\right), \\ a = \frac{(m,n)[left][signal] - nonresonant\ crosstalk}{10}, \\ b = \frac{(m,n)[left][signal] - passing\ loss - crossing\ crosstalk}{10}, \\ c = \frac{(m,n)[left][signal] - passing\ loss - crossing\ loss - nonresonant\ crosstalk - crossing\ loss - passing\ loss}{10} \end{cases} \tag{6.37}$$

Fig. 6.18c shows other nonresonant signals from the left input port travel to right. The signal itself suffers a passing loss, a crossing loss and a passing loss. It also generates a crossing crosstalk to up. This can be expressed with the following equations:

$$\begin{cases} (m,n)[right][signal] = (m,n)[left][signal] - passing\ loss - crossing\ loss - passing\ loss, \\ (m,n)[up][crosstalk] = (m,n)[left][signal] - passing\ loss - crossing\ crosstalk \end{cases} \tag{6.38}$$

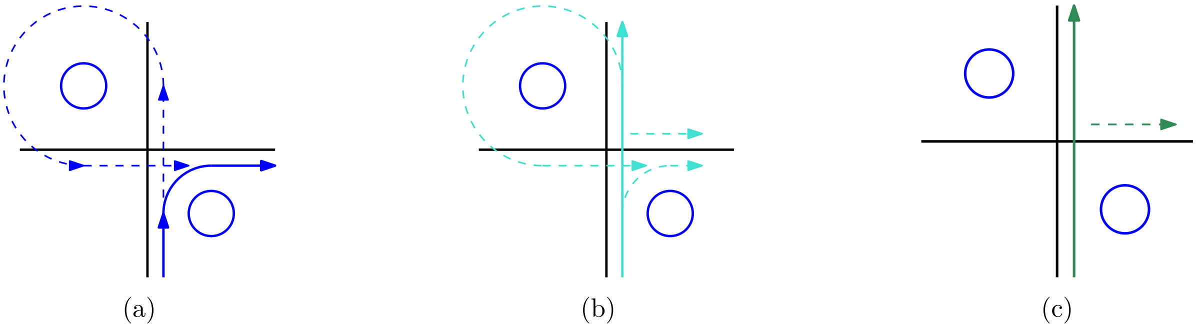


Figure 6.19.: Signals from down. (a) Resonant signal from down drops to right and generates a resonant crosstalk to right. (b) Nearest nonresonant signals from down travel to up and generate a nonresonant crosstalk, a crossing crosstalk and a nonresonant crosstalk to right. (c) Other nonresonant signals from down travel to up and generate a crossing crosstalk to right.

Fig. 6.19a shows a resonant signal from the down input port drops on the MRR and goes to right. The signal itself suffers a drop loss. A resonant crosstalk is generated to the up. It is then dropped by the other MRR and finally goes right. This crosstalk is coherent with the desired signal, so we add it back into the signal. This resonant crosstalk suffers a crossing loss, a drop loss, a crossing loss and a passing loss. These can be expressed with the following equations:

$$\begin{cases} m,n)[right][signal] = 10\log\left(10^{a} + 10^{b}\right), \\ a = \frac{(m,n)[down][signal] - drop\ loss}{10}, \\ b = \frac{(m,n)[down][signal] - resonant\ crosstalk - crossing\ loss - drop\ loss - crossing\ loss - passing\ loss}{10} \end{cases} \tag{6.39}$$

Fig. 6.19b shows the nearest nonresonant signal from the down input port travels to up. The signal itself suffers a passing loss, a crossing loss and a passing loss. This can be expressed with the following equation:

$$(m,n)[up][signal] = (m,n)[down][signal] - passing\ loss - crossing\ loss - passing\ loss \tag{6.40}$$

In the meantime, a nonresonant crosstalk, a crossing crosstalk and a nonresonant crosstalk are generated to the right of the crossing. These three crosstalk are from the same source and thus are coherent signals. We add them and get one crosstalk signal:

$$\begin{cases} (m,n)[right][crosstalk] = 10\log\left(10^{a} + 10^{b} + 10^{c}\right), \\ a = \frac{(m,n)[down][signal] - nonresonant\ crosstalk}{10}, \\ b = \frac{(m,n)[down][signal] - passing\ loss - crossing\ crosstalk}{10}, \\ c = \frac{(m,n)[down][signal] - passing\ loss - crossing\ loss - nonresonant\ crosstalk - crossing\ loss - passing\ loss}{10} \end{cases} \tag{6.41}$$

Fig. 6.19c shows other nonresonant signals from the down input port travel to up. The signal itself suffers a passing loss, a crossing loss and a passing loss. It also generates a crossing crosstalk to right. This can be expressed with the following equations:

$$\begin{cases} (m,n)[up][signal] = (m,n)[down][signal] - passing\ loss - crossing\ loss - passing\ loss, \\ (m,n)[right][crosstalk] = (m,n)[down][signal] - passing\ loss - crossing\ crosstalk \end{cases} \tag{6.42}$$

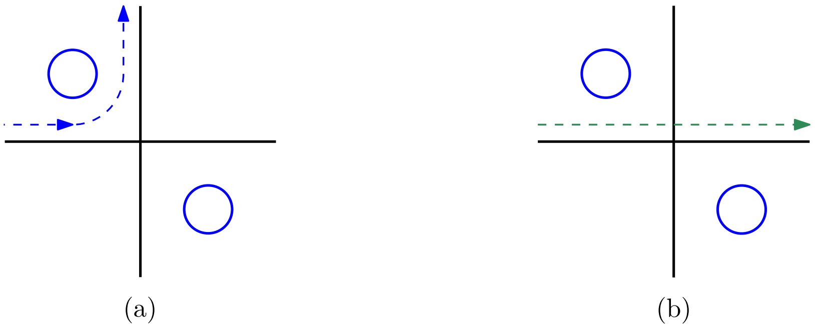

(a) (b)

Figure 6.20.: Crosstalk from left. (a) A resonant crosstalk from left drops to up, suffering a drop loss. (b) A nonresonant crosstalk from left travels to right, suffering a passing loss, a crossing loss and a passing loss.

Fig. 6.20a shows a resonant crosstalk from the left input port drops by an MRR and goes to up. The crosstalk suffers a drop loss. This can be expressed with the following equation:

$$((m,n)[up][crosstalk] = (m,n)[left][crosstalk] - drop\ loss \tag{6.43}$$

Fig. 6.20b shows a nonresonant crosstalk from the left input port travels through a crossing to right. The crosstalk suffers a passing loss, a crossing loss and a passing loss. No crossing crosstalk is generated because we only consider first-order crosstalk. This can be expressed with the following equation:

$$((m,n)[right][crosstalk] = (m,n)[left][crosstalk] - passing\ loss - crossing\ loss - passing\ loss \tag{6.44}$$

Fig. 6.21a shows a resonant crosstalk from the down input port drops by an MRR and

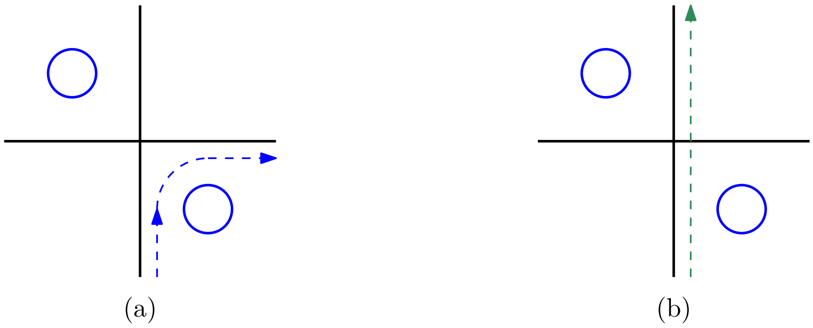


Figure 6.21.: Crosstalk from down. (a) A resonant crosstalk from down drops to right, suffering a drop loss. (b) A nonresonant crosstalk from down travels to up, suffering a passing loss, a crossing loss and a passing loss.

goes to right. The crosstalk suffers a drop loss. This can be expressed with the following equation:

$$((m,n)[right][crosstalk] = (m,n)[down][crosstalk] - drop\ loss \tag{6.45}$$

Fig. 6.21b shows a nonresonant crosstalk from the down input port travels through a crossing to up. The crosstalk suffers a passing loss, a crossing loss and a passing loss. No crossing crosstalk is generated because we only consider first-order crosstalk. This can be expressed with the following equation:

$$((m,n)[up][crosstalk] = (m,n)[down][crosstalk] - passing\ loss - crossing\ loss - passing\ loss \tag{6.46}$$

6.3. Results demonstration

Table 6.3 shows the computation results of all the test cases presented in (Li et al. 2018). We calculate the average SNR and the worst case SNR both in normal values and dB values.

- Case 1 has the lowest average SNR among all tested cases. The power of valid signals is 124.30 times higher than the noise power. This shows the logic scheme of FAST performs well when crosstalk is taken into consideration.

- Case 2 has the lowest worst-case SNR among all tested cases. Signal power is 38.79 times higher than the noise power.

- Generally, the average SNR differs from the worst-case SNR a lot. This indicates that the optimization space is big. In practice, we want higher worst-case SNR, because it determines the quality of the communication topology.

Table 6.3.: SNR

Idx	**d**	**N**	**ave_SNR**	**ave_SNR_dB**	**worst_SNR**	**worst_SNR_dB**
1	8	44	124.30	20.94 dB	59.33	17.73 dB
2	12	26	174.02	22.41 dB	38.79	15.89 dB
3	12	20	263.38	25.60 dB	74.08	18.70 dB
4	16	22	196.31	22.93 dB	53.09	17.25 dB
5	8	48	149.48	21.75 dB	128.21	21.08 dB
6	8	24	270.24	24.32 dB	60.40	17.81 dB
7	8	24	427.63	26.31 dB	234.61	23.70 dB

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; **ave_SNR**: the average SNR for all communications in the topology; **ave_SNR_dB**: the average SNR for all communications in the topology in dB; **worst_SNR**: the worst-case SNR for all communications in the topology; **worst_SNR_dB**: the worst-case SNR for all communications in the topology in dB.

Table 6.4 shows the insertion loss and crosstalk of the signal which has the worst-case SNR in a topology. In practice, we find: to achieve high SNR, crosstalk is more important than insertion loss. For example, in Case 1, communication (3, 7) has -0.86 dB insertion loss, -18.59 dB crosstalk. Its SNR is 59.33. Communication (1, 7) has -0.76 dB insertion loss, -18.59 dB crosstalk. Its SNR is 60.71. These two signals have the same crosstalk noise and almost the same SNR. But their insertion loss differs for 0.1 dB. In the comparison performed in Chapter 5, the biggest insertion loss difference between CustomTopo and FAST is only 0.04 dB, so 0.1 dB is quilt significant. But it does not lead to significant SNR difference.

On the other hand, also in Case 1, communication (1, 2) has -0.63 dB insertion loss, -22.94 dB crosstalk. Its SNR is 170.26. Communication (0, 3) has -0.63 dB insertion loss, -22.55 dB crosstalk. Its SNR is 155.66. Two signals suffers the same insertion loss. But 0.39 dB crosstalk difference leads to a SNR difference of 15. In the same test case, the crosstalk of different signals can differs for 4 dB. This means, optimizing crosstalk is more helpful for optimizing SNR. This conclusion gives us an instruction of how to optimize SNR.

Table 6.4.: Insertion loss and crossstalk of the signal with the worst-case SNR

Idx	d	N	Insertion loss/dB	Crosstalk/dB
1	8	44	−0.86	−18.59
2	12	26	−0.96	−16.84
3	12	20	−1.17	−19.86
4	16	22	−0.84	−18.09
5	8	48	−0.87	−21.95
6	8	24	−0.75	−18.56
7	8	24	−0.89	−24.59

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.

6.4. Interchannel crosstalk and intrachannel crosstalk

In Section 6.3, we add all the noise sources together to calculate the crosstalk and SNR (Preston et al. 2011). These are two different noise sources for a signal: 1) Interchannel crosstalk. 2)Intrachannel crosstalk. **Interchannel crosstalk** is the crosstalk signal which has a sufficiently different wavelength from the desired signal. **Intrachannel crosstalk** is the crosstalk signal whose wavelength is the same with or close to the desired signal.

The effects of intrachannel crosstalk can be much more severe than those of interchannel crosstalk since they cannot be removed by filtering (Nikdast et al. 2015)(Ramaswami et al. 2009). To optimize crosstalk, one of the most important measures is to minimize the portion of intrachannel crosstalk. Here we present the values of different sources of crosstalk again to quantify the effects of crosstalk (Table 6.5).

Table 6.5.: Values of different crosstalk 2.

Crosstalk Types	Value
crossing crosstalk	$-40\ dB$
terminator crosstalk	$-50\ dB$
resonant crosstalk	$-25\ dB$
nonresonant crosstalk	$-35\ dB$

Resonant crosstalk is $\frac{1}{316}$ of the valid optical signal. Nonresonant crosstalk is $\frac{1}{3160}$ of the valid optical signal, which is 10 times smaller than resonant crosstalk . Crossing crosstalk is 40 dB, which is only $\frac{1}{10000}$ of the input signal power. Terminator crosstalk is $\frac{1}{100000}$ of the input signal power. From this comparison, we clearly see that resonant crosstalk is the biggest crosstalk source and much more severe than the other sources. In FAST, one of the biggest advantage is resonant crosstalk will never become the intrachannel crosstalk of any signal, which means the biggest source of crosstalk can be completely filtered out in the logic topology of FAST. This is explained in the following.

Fig. 6.22 shows how resonant crosstalk is generated. Due to the nonideality of MRRs, a portion of resonant signal will not drop by the resonant MRR but go straight. In the logic scheme of FAST (shown in Fig. 6.23), there are three kinds of signals.

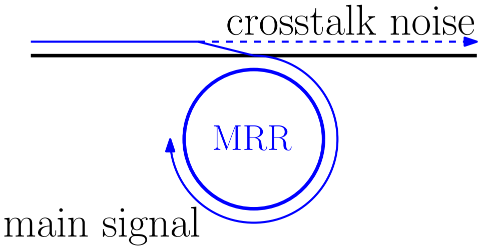


Figure 6.22.: Resonant crosstalk.

- Signals dropping by MRRs in the upper-left corner of a crossing, e.g. communication (1, 1).
- Signals dropping by MRRs in the lower-right corner of a crossing, e.g. communication (2, 2).
- Signals requiring no MRRs, e.g. communication (1, 2).

For default communciations e.g. communication (1, 2), no resonant crosstalk is generated because there is no resonant MRR. For optical signals which requires MRR, there are two situations: 1) Crossings associated with one MRR. 2) Crossings associated with two MRRs. For crossings associated with one MRR (shown in Fig. 6.24a) and Fig. 6.24b), resonant signal only comes in from one direction. If MRR is located in the upper left corner, resonant signal comes from the left side. On the other hand, If MRR is located in the lower right corner, resonant signal only comes from the down side. For these two cases, resonant crosstalk always goes to different direction from the main signal. The communication rule of FAST is: Crossings on the same default path must have different colors. So on the rest path of the resonant crosstalk, it will not meet another blue MRR, that means the blue crosstalk will not become an intrachannel crosstalk of any signal. For crossings associated with two MRRs (shown in Fig. 6.24c) and Fig. 6.24d), resonant crosstalk will always drop by the blue MRR in the other corner of the same crossing and joins the main signal again.

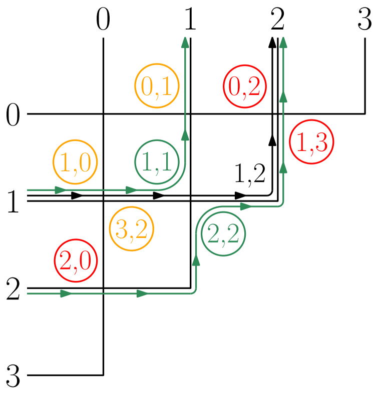


Figure 6.23.: Three kinds of signals.

The arguments above show that the biggest crosstalk source i.e. resonant crosstalk is always interchannel crosstalk for an optical signal, which means resonant crosstalk never has the same wavelength as the main signal on any receiver and thus can be filtered out.

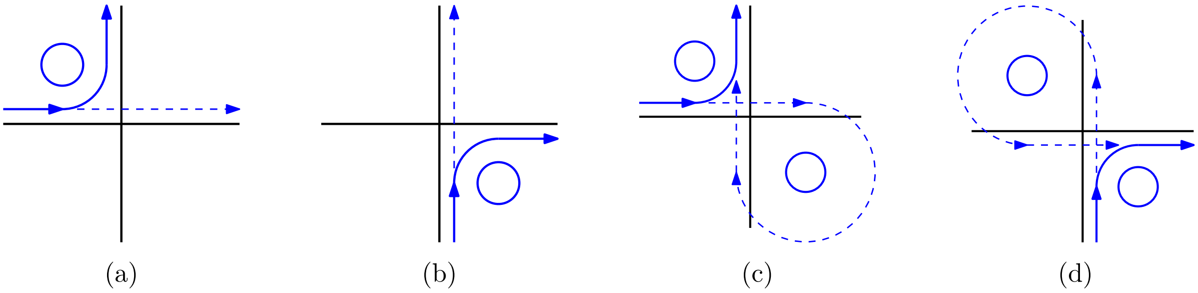
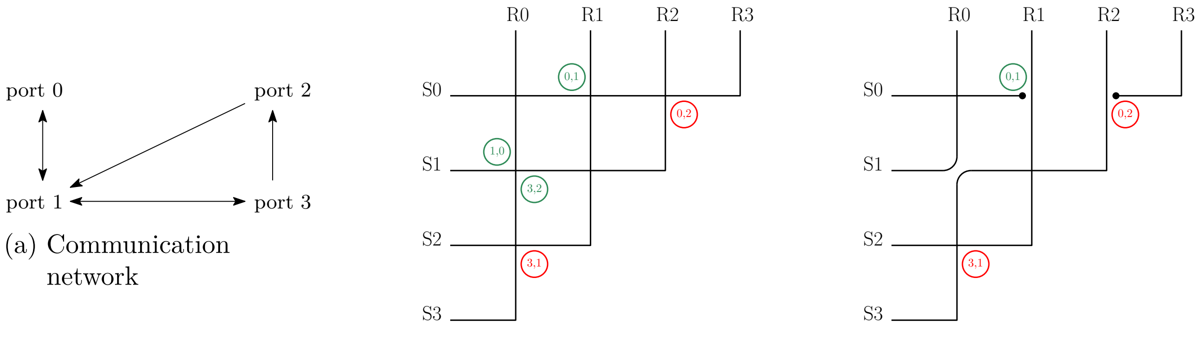


Figure 6.24.: Resonant crosstalk analysis.

7. Future Work

7.1. Eliminating redundant waveguides

In WRONoC topology, not only MRRs, wavelengths, crossings but also waveguides could be redundant (Truppel et al. 2019). Fig. 7.1 demonstrates the possibility of removing redundant waveguides. Fig. 7.1b shows the initial topology of FAST for the communication network shown in Fig. 7.1a. Fig. 7.1c shows the optimized topology after removing the redundant waveguides.



(a) Communication network

(b) Initial topology

(c) Optimized topology

Figure 7.1.: Removing redundant waveguides, crossings and MRRs.

In the optimized topology, we actually just introduce one new type of element: optical terminators which have directions. Optical terminator is a common optical routing element which terminates an optical signal. In FAST, we can assign optical terminators with directions, e.g. positive terminators and negative terminators. A positive terminator on horizontal waveguides means the waveguide on its right side can be removed. A negative terminator on horizontal waveguides means the waveguide on its left side can be removed. This is illustrated in Fig. 7.2. Similarly, a positive terminator on vertical waveguides means the waveguide on its down side can be removed. A negative terminator on vertical waveguides means the waveguide on its upper side can be removed. When four terminators get together, we can directly remove two MRRs and form two default communications. This is illustrated in Fig. 7.3.

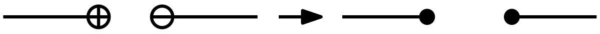


Figure 7.2.: Optical terminators on a horizontal waveguide.

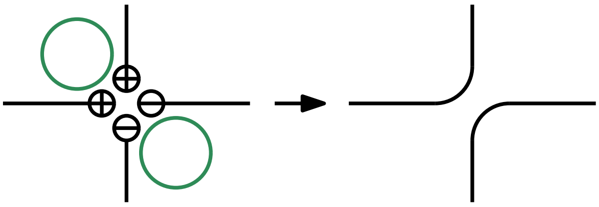
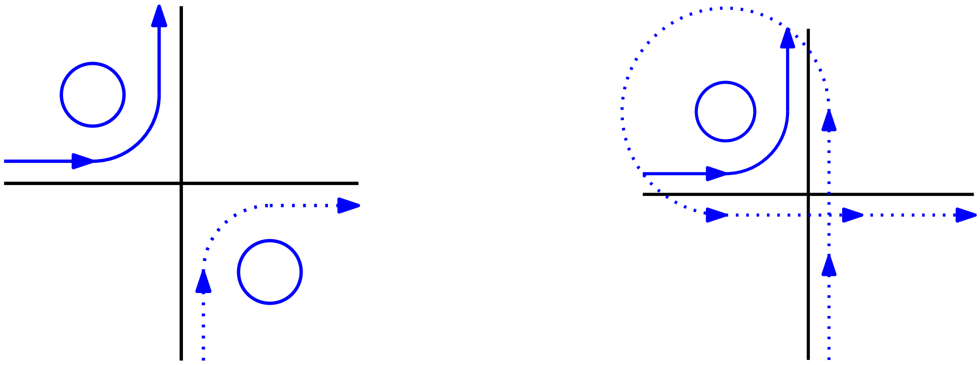


Figure 7.3.: Four optical terminators congregate.

Introducing optical terminators into FAST can greatly reduce the overall crosstalk. Because of the fast sweeping technique in FAST, the algorithm won't slow down when new features are introduced.

7.2. MRR sharing structure

Reference (Lin & Lea 2012) demonstrates the MRR sharing structure. As illustrated in Fig. 7.4, solid line with arrow represents an optical signal coming from left, dotted line with arrow represents an optical signal coming from down. In Fig. 7.4a, two signals both drop 90° on their own MRRs. In Fig. 7.4b, two signals share one MRR. The solid line drops 90° and the dotted line drops 270°.



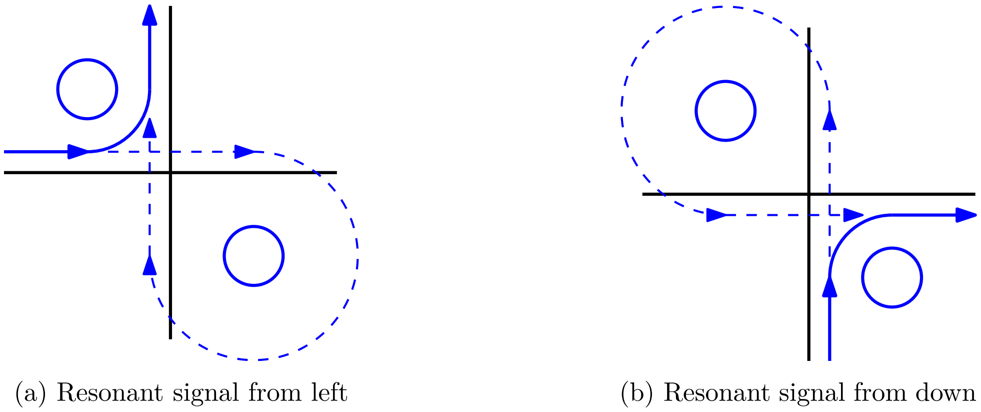
(a) Two MRRs for two signals

(b) One MRR for two signals

Figure 7.4.: MRR sharing structure.

The advantage of MRR sharing structure is less MRR usage. The disadvantage is that the optical signal represented by the dotted line in Fig. 7.4b has to suffer two more crossing losses.

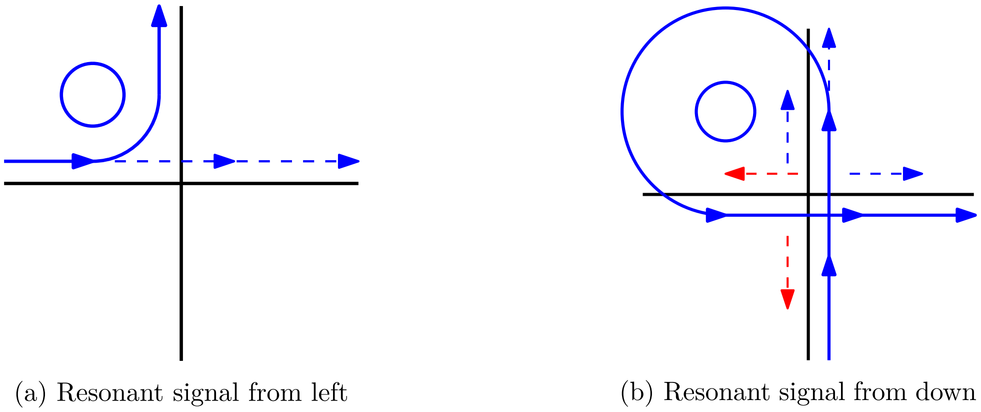
Moreover, as shown in Fig. 7.5, when there are two MRRs in a crossing, the resonant crosstalk from both signals will drop by the other MRR again and becomes coherent crosstalk of the signal itself. The coherent crosstalk joins the source signals again and becomes a part of the desired signals again. Thus, two MRRs can help the signal eliminate resonant crosstalk.


(a) Resonant signal from left

(b) Resonant signal from down

Figure 7.5.: Two MRRs eliminate resonant crossing. The dashed line represent resonant crosstalk.

However, when MRR sharing is used, two resonant crosstalk will be released. This is shown in Fig. 7.6. More seriously, as shown in Fig. 7.6b, when signal comes from down side, the signal must go through the crossing two times. This not only causes two more crossing losses, but also causes four portions of crossing crosstalk. Two of the four crosstalk (the red ones shown in Fig. 7.6b) are against the main signal. This could disturb the signal and could cause serious problem. Currently, there is no research addressing how to handle this situation.


(a) Resonant signal from left

(b) Resonant signal from down

Figure 7.6.: MRR sharing structure releases two resonant crossing. The dashed line represent resonant crosstalk.

Although MRR sharing structure causes more crossing losses and more crosstalk for resonant signal, it reduces the passing loss and nonresonant crosstalk for nonresonant signal because of one less MRR. Crossing loss and resonant crosstalk are both way bigger than passing loss and nonresonant crosstalk. So the average performance or overall performance of a router using sharing structure is for sure not as good as a router without sharing structure. But worse overall/average SNR doesn't necessarily mean worse the worst case SNR. Thus, a comprehensive SNR analysis needs to be done to formally analyze the advantage and disadvantage of MRR sharing structure.

In the applications in which overall/average SNR is not so important but MRR usage is very important, we can use sharing structure on those signal paths that are not critical to further reduce MRR usage. Moreover, a comprehensive SNR comparison between using sharing structure and not using sharing structure on representative WRONoC topologies e.g. Lambda Router, GWOR, Folded Crossbar and Snake can be implemented using the algorithm introduced in Section 6.2.

7.3. Building a competitive layout optimization algorithm

Half-matrix topology, inconsistent sender/receiver orders and multiple topology variations give FAST a natural connection with physical layout. FAST should naturally be a physical layout tool, not only a topology customization tool. When FAST is expanded to a physical layout tool, it naturally combines the optimization of topology customization and physical layout in a novel and efficient way. FAST can produce layouts very competitively in both runtime and the quality.

8. 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. These two features give FAST a natural connection with the physical layout, making it not only an efficient topology customization algorithm but also a promising layout platform. In the second part of this work, we propose a general crosstalk and SNR analysis method for all WRONoC topologies with basic PSE or CSE structures. This algorithm can be integrated into the optimization flow of FAST to further comprehensively optimize the crosstalk and SNR of the topologies. FAST is naturally a physical layout platform. We aim at developing a competitive automatic physical layout tool in the future work.

Bibliography

Beux, S. L., Trajkovic, J., O'Connor, I., Nicolescu, G., Bois, G. & Paulin, P. (2011): Optical ring network-on-chip (ornoc): Architecture and design methodology, Design, Automation, and Test Europe Conf. S. 788–793.

Bogaerts, W., Heyn, P. D., Vaerenbergh, T. V., Vos, K. D., Selvaraja, S. K. & T. Claes, et al (2012): Silicon microring resonators, Laser Photonics Rev., S. 47–73.

Bri'ere, M., Girodias, B., Bouchebaba, Y., Nicolescu, G., Mieyeville, F., Gaffiot, F. & O'Connor, I. (2007): System level assessment of an optical noc in an mpsoc platform, Proc. Design, Automation, and Test Europe Conf., S. 1084–1089.

Chan, J., Hendry, G., Biberman, A. & Bergman, K. (2010): Architectural Exploration of Chip-Scale Photonic Interconnection Network Designs Using Physical-Layer Analysis, J. Lightwave Technol., S. 1305–1315.

Duong, L. H. K., Wang, Z., Nikdast, M., Xu, J., Yang, P., Wang, Z., Wang, Zh., Maeda, R. K. V., Li, H., Wang, X., Beux, S. L. & Thonnart, Y. (2016): Coherent and incoherent crosstalk noise analyses in interchip/intrachip optical interconnection networks, IEEE Transactions on Very Large Scale Integration (VLSI) Systems **24**: 2475–2487.

Gurobi Optimization, Inc. (2012): Gurobi optimizer reference manual.
`http://www.gurobi.com`

Jiang, N., Becker, D. U., Michelogiannakis, G., Balfour, J., Towles, B., Shaw, D. E., Kim, J. & Dally, W. J. (2013): A detailed and flexible cycle-accurate network-on-chip simulator, IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS) S. 86–96.

Li, M., Tseng, T.-M., Bertozzi, D., Tala, M., & Schlichtmann, U. (2018): CustomTopo: A topology generation method for application-specific wavelengthrouted optical nocs, Int. Conf. Comput.-Aided Des., S. 100.

Li, M., Tseng, T. M., Tala, M. & Schlichtmann, U. (2020): Maximizing the Communication Parallelism for Wavelength-Routed Optical Networks-on-Chips, IEEE/ACM Asia and South Pacific Design Automation Conference (ASP-DAC).

Lin, B. & Lea, C. (2012): Crosstalk analysis for microring based optical interconnection networks, JOURNAL OF LIGHTWAVE TECHNOLOGY **30**: 2415–2420.

Nikdast, M., Xu, J., Duong, L. H. K., Wu, X., Wang, X., Wang, Z., Wang, Z., Yang, P., Ye, Y. & Hao, Q. (2015): Crosstalk noise in wdm-based optical networkson-chip: A formal study and comparison, IEEE Transactions on Very Large Scale Integration (VLSI) Systems **23**: 2552–2565.

Preston, K., Sherwood-Droz, N., Levy, J. S. & Lipson, M. (2011): Performance guidelines for wdm interconnects based on silicon microring resonators, IEEE, Laser Science to Photonic Applications .

Ramaswami, R., Sivarajan, K. & Sasaki, G. (2009): Optical networks: A practical perspective, San Mateo, CA, USA: Morgan Kaufmann .

Ramini, L., Grani, P., Bartolini, S. & Bertozzi, D. (2013): Contrasting wavelength-routed optical noc topologies for power-efficient 3D-stacked multicore processors using physical-layer analysis, Proc. Design, Automation, and Test Europe Conf., S. 1589–1594.

Tan, X., Yang, M., Zhang, L., Jiang, Y., & Yang, J. (2011): On a scalable, nonblocking optical router for photonic networks-on-chip designs, Symp. Photonics and Optoelectronics (SOPO).

Truppel, A., Tseng, T.-M., Bertozzi, D., Alves, J. C. & Schlichtmann, U. (2019): PSION: Combining logical topology and physical layout optimization for Wavelength-Routed ONoCs, ACM/SIGDA International Symposium on Physical Design (ISPD), S. 49–56.

Tseng, T.-M., Truppel, A., Li, M., Nikdast, M. & Schlichtmann, U. (2019): Wavelength-Routed Optical NoCs: Design and EDA — State of the Art and Future Directions, IEEE/ACM International Conference on Computer-Aided Design (ICCAD).

Vantrease, D., Schreiber, R., Monchiero, M., McLaren, M., Jouppi, N. P. & Fiorentino, M. (2008): Corona: System implications of emerging nanophotonic technology, ACM SIGARCH Computer Architecture News **36**: 153–164.

Wikipedia (2020a): Coherence (physics), `https://en.wikipedia.org/wiki/Coherence_(physics)`: [Online; accessed 05-September-2020].

Wikipedia (2020b): Q factor, `https://en.wikipedia.org/wiki/Q_factor`: [Online; accessed 05-September-2020].