Technische Universität München
Department of Electrical Engineering and Information Technology
Institute for Electronic Design Automation

Wavelength-Routed Optical Network-on-Chip Router Design Using Parallel Switching Elements

Master Thesis

Zhidan Zheng

Technische Universität München
Department of Electrical Engineering and Information Technology
Institute for Electronic Design Automation

Wavelength-Routed Optical Network-on-Chip Router Design Using Parallel Switching Elements

Master Thesis

Zhidan Zheng

Supervisor :	Dr.-Ing. Tsun-Ming Tseng
Supervising Professor :	Prof. Dr.-Ing. Ulf Schlichtmann
Topic issued :	17.03.2020
Date of submission :	12.07.2020

Zhidan Zheng
Arcisstraße 21
D-80333 Munich Germany

Abstract

Traditional electronic interconnects can not satisfy the high requirements of MPSoCs due to their large noise power and extremely high propagation delay. On the other hand, the optical networks-on-chips (ONoCs), especially wavelength-routed optical networks-on-chips (WRONoCs), are attracting worldwide interest, because they provide higher bandwidth, lower latency and lower power consumption.

However, many WRONoC topologies have been suffering increasing MRR usage and the problem of incongruity between the logic scheme and physical layout. When the size of the network is increasing, these problems become more severe. The typical WRONoC topologies, like λ-router and Snake, use the crossing switch elements (CSEs); and each CSE has two identical microrings (MRRs). In this case, for large networks, a great deal of MRRs are needed in these topologies. The large MRR usage brings about non-negligible insertion loss and crosstalk noise, which seriously degrades the system performance. However, the parallel switch elements (PSEs), overlooked by many researchers, can reduce MRR usage due to the special working mechanism and the structure. Comparing to the CSEs with two identical MRRs in the typical WRONoC topologies, PSEs only have half of MRRs. Furthermore, removing the crossings in PSEs can reduce both insertion loss and crosstalk noise. Thus, PSEs become an appealing option for reducing MRR usage, while promising high performance. Another challenge of state-of-the-art WRONoC topologies is that their physical layouts are inconsistent with their logic schemes since logic synthesis and physical design are separated into two independent steps in the design flow. Without taking the physical features into consideration during the logic synthesis causes extra crossings or detours in realistic implementation.

In this thesis, I propose a novel 4×3 structure excluding self-communication: *Hash*. Based on this basic building block *Hash*, I propose a scalable $N \times N$-1 topology *Light* which provides an efficient way to reduce MRR usage; furthermore, the physical layout of *Light* matches its logic scheme perfectly without generating any additional crossings and detours. According to the comparison to λ-router, *Light* has a better performance in reducing insertion loss and improving SNR on average.

Acknowledgements

Foremost, I would like to express my special appreciation and thanks to my supervisor Dr.-Ing. Tsun-Ming Tseng for giving me such an interesting and exciting topic as my Master's thesis. As my teacher and mentor, he has provided me extensive personal and professional guidance and taught me a great deal about scientific research. I would also like to thank Mengchu Li, whom I have had the pleasure to work with during this project. She has shown me, by her example, what a good researcher should be.

Most importantly, I want to thank my parents and my brother whose unfailing love and great support are with me in whatever I pursue.

In this special time, I am grateful to all medical staff and researchers for all of the sacrifices that they have made to defeat the COVID-19. Their great efforts and strong beliefs convince me that this is a battle that we could win. Wir schaffen das!

Contents

List of Figures

List of Tables

1. Introduction

Nowadays, multiprocessor systems-on-chips (MPSoCs) are attracting widespread interest, because MPSoCs support better scalability and higher bandwidth than traditional communication infrastructures (Scandurra & O'Connor 2011). MPSoCs become a widely accepted solution for reaching performance requirements of compute-intensive applications. The interconnection in MPSoCs is required to support high-speed communication with low latency and power (O'Connor et al. 2005). Unfortunately, the metallic interconnects have unacceptable noise and propagation delay caused by the capacitive and inducting coupling between wires (Beux et al. 2014). The noise and propagation delay seriously damage the system performance, and hence there remains a need for an innovative method that can overcome these drawbacks.

With the development of silicon photonics, optical networks-on-chips (ONoCs) are generating considerable recent research interest for supporting high-speed data transmission in MPSoCs (Preston et al. 2011). In ONoCs, multiple wavelengths can be transmitted at the same time without collision through the use of wavelength-division multiplexing (WDM) and silicon microring resonators (MRRs) (Beux et al. 2013). Comparing to traditional electronic interconnects, ONoCs provide higher bandwidth with lower power.

There are two categories of ONoCs : *active ONoCs* and *passive ONoCs* (Manolatou & Haus 2002). In *active ONoCs*, some control units are required to dynamically tune switching elements to preclude data collision (Khouzani et al. 2012). Although the number of switching elements in this kind of network is slightly less than the number of switching elements in *passive ONoCs*, the extra overhead of energy and latency is inevitable for arbitration. The *passive ONoCs*, also known as wavelength-routed ONoCs (WRONoCs), avoid this overhead by using some routing elements with fixed wavelength assignment (Manolatou & Haus 2002). The signal paths are previously fixed during the design time, and hence no energy and time are wasted for arbitration in WRONoCs (Ramini et al. 2012).

Some state-of-the-art WRONoC topologies have been developed, including the Matrix (Bianco et al. 2012), λ-router (Brière et al. 2007), GWOR (Tan et al. 2011) and Snake (Ramini et al.

2013). According to the state-of-the-art flow, logic synthesis and physical design are separated into two independent steps (Tseng et al. 2019). Most WRONoC topologies have focused on logic synthesis, while overlooked the physical features in this step. As a result, the performances of their realistic layouts fall behind the performances of their logic schemes.

Consider the typical processor-memory communication infrastructure as an example, the structure is shown by Figure 1.1 (Ramini et al. 2012). According to the 3D-Architecture, memory controllers are distributed about the periphery of the photonic layer, and hubs are located in the middle of this layer. Each IP-core, either a memory controller or a processor, can send or receive data through a *master* port or a *slave* port. It is reasonably inferred that the master and the slave from the same IP-core should be very close rather than far away from each other. Without considering these physical features, in most WRONoC logic schemes, researchers place the master and slave from the same IP-cores to two opposite sides. In this case, some unexpected waveguides crossings or detours are generated in physical layout. For example, in (Ramini et al. 2012), the physical layout of an 8×8 λ-router shows that the number of crossings increases from 28 in the logic scheme to 64 in its physical layout, the great effort taken in reducing the insertion loss of λ-router may be offset by these extra crossings. The separation between the logic scheme and physical design is challenging the performance of ONoCs. To match its physical constraints, a logic scheme should be designed with conceiving these constraints.

Another challenge of WRONoC topologies is the large MRR usage. Most WRONoC topologies have focused on the crossing switch elements (CSEs) with two identical MRRs, such as λ-router (Brière et al. 2007), GWOR (Tan et al. 2011) and Snake (Ramini et al. 2013). The MRR usage dramatically increases with the increase in the size of networks. For example, for the network with 64 IP-cores, 4032 MRRs are used in λ-router, and 3968 MRRs are used in GWOR. The huge consumption of MRRs results in severe insertion loss and crosstalk noise, which damages the quality of signals. Thus, it is urgently required to reduce MRR usage without degrading the system performance.

Although the MRRs in CSEs support both the 90-degree turns and the 270 degree turns of the signals carried by the resonant wavelengths, the insertion loss caused by the 270-degree turns is larger than the insertion loss caused by the 90-degree turns (Lin & Lea 2012, Zhang et al. 2014). Because of the additional crossing loss in 270-degree turns, most researchers prefer the CSEs with two identical MRRs nearby the crossings of two orthogonal waveguides. Both MRRs support the 90-degree turns to avoid the additional crossing loss.

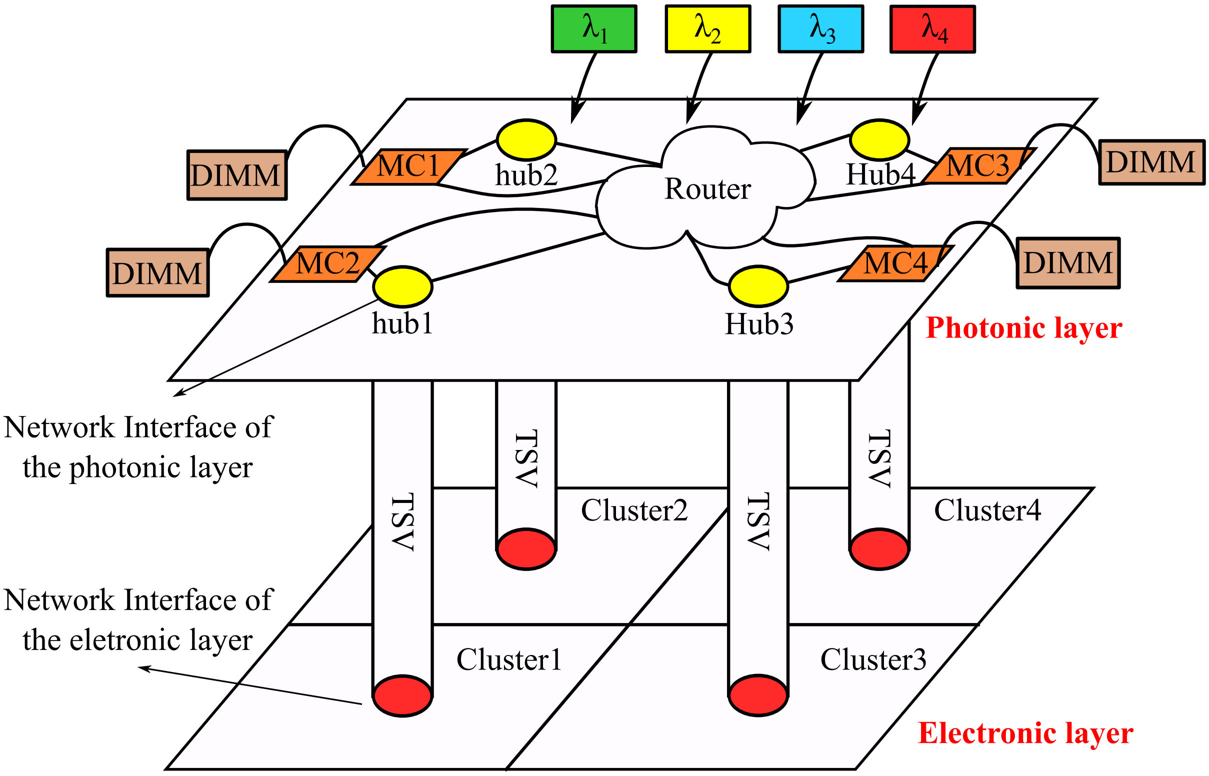

Figure 1.1.: 3D-Sturcture

Few researchers have paid much attention to parallel switch elements (PSEs), which have a different working mechanism from CSEs. The MRRs in PSEs only support 180-degree turn of signals. In each PSE, an MRR is placed between two parallel waveguides. Hence, PSEs can not only avoid the crossings of two orthogonal waveguides but also reduce insertion loss and crosstalk noise (Tseng et al. 2019).

Few topologies have used PSEs, but one *active ONoC* topology called *Crux* proposed in (Xie et al. 2010) combines PSEs with CSEs in order to achieve higher signal-to-noise ratio (SNR). Since the Crus belongs to *active ONoCs*, it requires extra switching fabrics and control units. Based on what I know, a WRONoC topology based on PSEs has not been developed yet.

In this thesis, I propose a novel 4×3 structure: *Hash* which uses 4 PSEs to support 12 signal paths concurrently without collision. Based on *Hash*, I propose an $N \times N$-1 scalable topology: *Light* and a simple way to assign the resonant wavelength to each MRR. For the same size of networks, *Light* can reduce the number of MRRs by half compared to the typical WRONoC topology, λ-router. Besides that, the *Light* logic scheme matches its physical layout by placing the master and the slave from the same IP-cores close to each other. As a result, extra crossings and detours would not be generated in the physical layout of *Light* for any sizes of networks.

This thesis is organised as follows: Section 2 gives a brief background about the optical switching elements (OSEs) and an introduction of some typical WRONoCs topologies. In Section 3, I present the structure of *Hash* and three types of signal paths. The *Light* topology and the rule of wavelength assignment are demonstrated in Section 4. The introduction and analysis of the crosstalk noise and insertion loss are given in Section 5. The experimental results about the comparison between *Light* and λ-router in terms of the MRR usage, insertion loss, and SNR are provided in Section 6.

2. Background

2.1. Working Mechanisms of Optical Switch Elements

Optical switch elements (OSEs) consist of several waveguides and MRRs, which can change the directions of signals on certain wavelengths. If a signal is coupled with an MRR and shifted to another waveguide, then this signal is regarded as an *on-resonance* signal. If a signal ignores the MRR when it passes through the waveguide, then this signal is referred to as an *off-resonance* signal to the MRR (Tseng et al. 2019). The wavelengths of the *on-resonance* signals are determined by the radius of the MRR (Bogaerts et al. 2012).

According to the different structures of OSEs, there are two categories of OSEs:

1) A crossing switch element (CSE) consists of a pair of orthogonal waveguides and an MRR nearby the crossing waveguides. The MRRs in CSEs support not only 90-degree switching but also 270-degree switching to *on-resonance* signals (Lin & Lea 2012). Figure 2.1(a) illustrates the working mechanism of a 90-degree turn in CSEs. The *on-resonance* signal, represented by the blue dash line (λ_n) in Figure 2.1(a), is coupled with the MRR (MR_n) and switched to the vertical waveguide, while the *off-resonance* signal, represented by the red dash line (λ_m) in Figure 2.1(a), ignores the MRR and passes through the waveguide without being changed.

2) A parallel switch element (PSE) consists of a pair of parallel waveguides and an MRR located between these waveguides. The MRRs in PSEs only support 180-degree switching to *on-resonance* signals as shown in Figure 2.1(b). The *on-resonance* signal, represented by the blue dash line (λ_n) in Figure 2.1(b), is coupled by the MRR (MR_n) and switched to the opposite waveguide, while the *off-resonance* signal represented by the red dash line (λ_m) in Figure 2.1(b) ignores the MRR and directly passes through the waveguide.

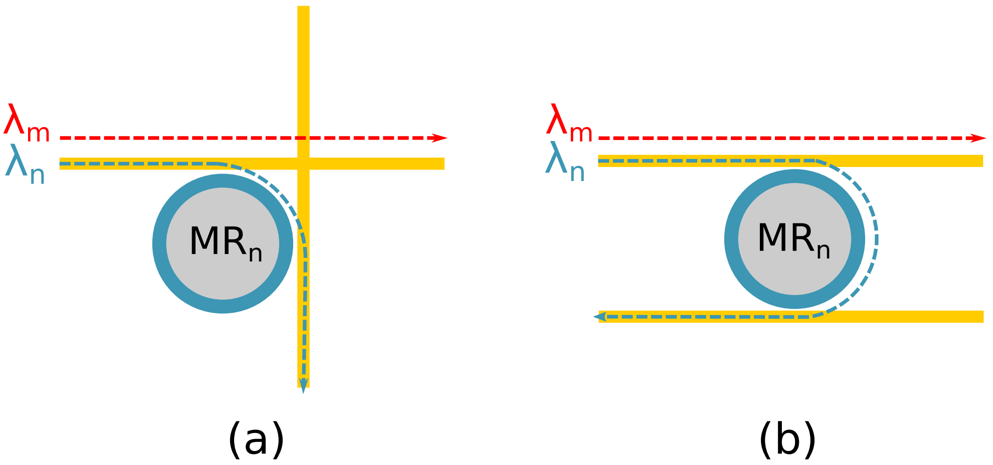

Figure 2.1.: (a) The working mechanism of CSEs (b) The working mechanism of PSEs

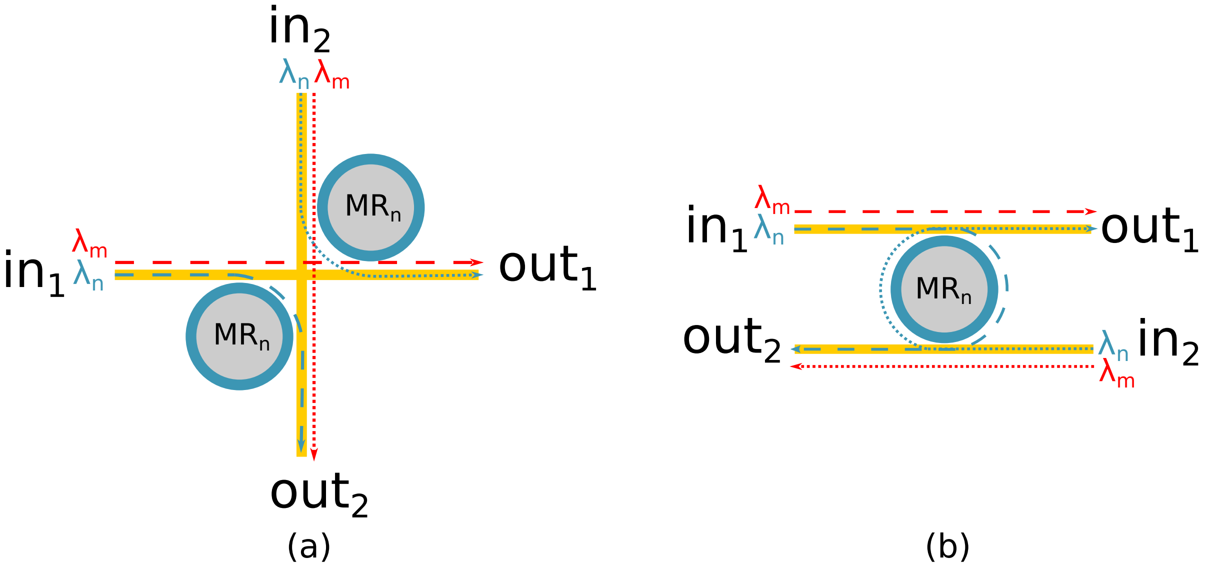

Figure 2.2.: (a) A 2×2 CSE (b) A 2×2 PSE

2.2. 2 × 2 Optical Switch Elements

In WRONoCs, wavelength assignment and communication paths are determined during the design time. With the use of WDM and OSEs in WRONoCs, several signals on different wavelengths can transmit at the same time without causing data collision (Beux et al. 2013). Comparing to the OSEs only with one input and one output shown in Figure 2.1, the OSEs with two inputs and two outputs can simultaneously support more signal paths to minimize the MRR usage.

Many WRONoC topologies, such as λ-router, Snake and GWOR, have focused on CSEs, especially the CSEs with two identical MRRs nearby the intersection of two waveguides (Tseng et al. 2019). The MRRs in a 2×2 CSE change the directions of the *on-resonance* signals in 90-degree to avoid the additional crossing loss in 270-degree truns. As shown in Figure 2.2(a), a 2×2 CSE can simultaneously support up to 4 different signal paths. When the *on-resonance* signal (λ_n) enters from in_1 or in_2, it is coupled with the first MRR that it meets, and switched to another waveguide as shown in Figure 2.2(a). The *off-resonance* signal (λ_m) directly passes the waveguide without being affected by the MRR.

A 2×2 PSE is presented by Figure 2.2(b). Comparing with a 2×2 CSE, a 2×2 PSE has only one MRR located between two parallel waveguides. In a 2×2 PSE, two inputs are diagonally placed to support 4 different signal paths simultaneously. The *on-resonance* signals from in_1 and in_2, which are represented by blue dash lines in Figure 2.2(b), are coupled with the MRR and finally received by out_2 and out_1, respectively. The *off-resonance* signals pass through the waveguide without being changed. Due to the special working mechanism of PSEs, the two *on-resonance* signals on the path $in_1 \to out_2$ and $in_2 \to out_1$ have equal insertion loss values. The detailed introduction of insertion loss and crosstalk noise in OSEs is given in Section 5.1.

Compared with 2×2 CSEs, 2×2 PSEs reduce the number of MRRs by half while promising high bandwidth. The wavelengths of two *on-resonance* signals in PSEs should be incoherent. Otherwise, the switching mechanism may not be achieved as described above (Lin & Lea 2012).

2.3. State-of-the-art WRONoC topologies

Some WRONoC topologies have been proposed based on CSEs with two identical MRRs, such as λ-router (Brière et al. 2007), Snake (Ramini et al. 2013) and GWOR (Tan et al. 2011).

Figure 2.3 shows the logic schemes of a 4×4 λ-router and a 4×4 Snake. Both topologies can achieve the full-bandwidth communication among 4 masters and 4 slaves. For example, in the 4×4 λ-router, master m_1 sends data to four slaves (s_1, s_2, s_3 and s_4) at the same time by using four different wavelengths (λ_2, λ_3, λ_1 and λ_4, respectively). The four signal paths are represented by the red, blue, green, yellow lines in Figure 2.3(a).

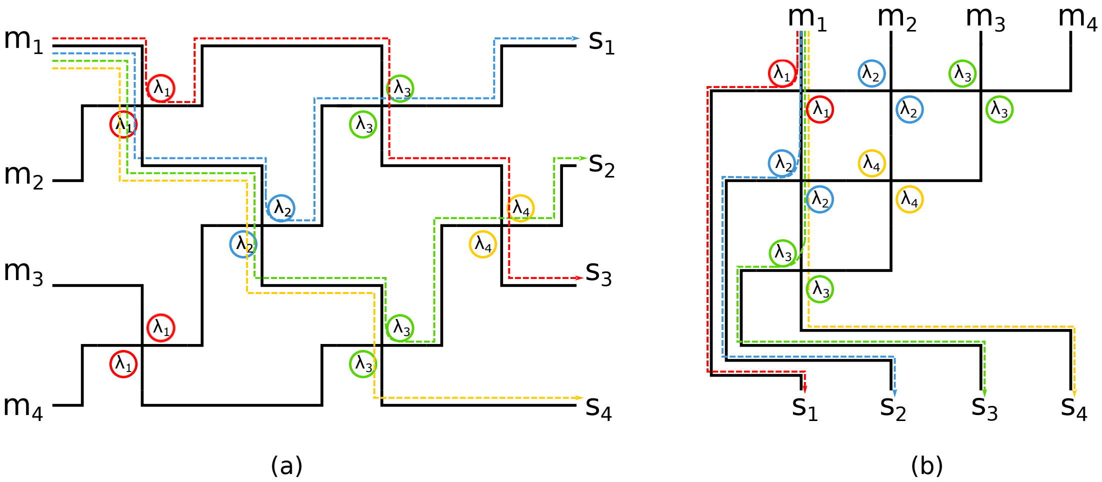

Figure 2.3.: (a) A 4×4 λ-router (b) A 4×4 Snake

The masters and the slaves in both topologies are placed to two opposite sides of their logic schemes. The unrealistic assumption in both λ-router and Snake leads to extra crossings or long detours in their physical layouts. For example, (Ramini et al. 2013) manually designed the physical layout of a 4×4 λ-router, and they found that the number of crossings increases from 6 in the logic scheme to 15 in physical layout. It is inefficient to manually design a physical layout, especially for some large-scale networks. Thus, some tools have been developed to automatically design a physical layout according to its logic scheme, such as PROTON+ (Beuningen et al. 2015). By using PROTON+, the number of crossings for an 8×8 λ-router increases from 28 in the logic scheme to 90 in physical layout (Li et al. 2018). Although Snake can avoid the additional crossings, the length of waveguides for detouring is relatively high (Ramini et al. 2013). Long detours results in large propagation loss, which is why many researchers allow additional crossings in physical design to trade off the propagation loss against crossing loss.

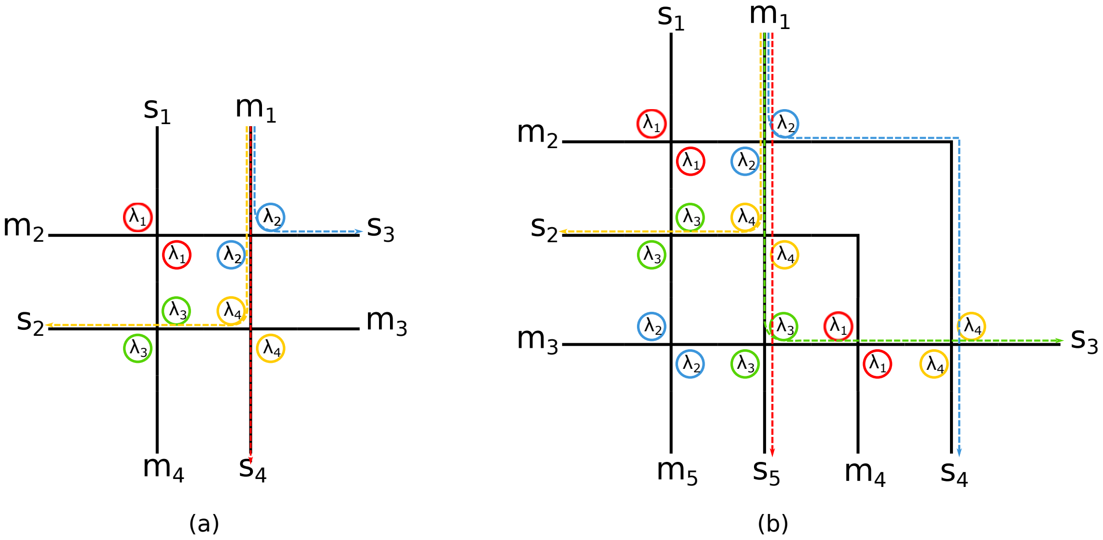

Figure 2.4.: (a) A 4×3 GWOR (b) A 5×4 GWOR

Different from λ-router and Snake, GWOR removes all self-communications. It is power efficient to connect a master with a slave from the same IP-core by electronic interconnects because two ports are placed very close to each other and electronic interconnects inside an IP-core do not require any E/O or O/E converters. For the sake of saving power, self-communication can be removed in the logic scheme but achieved by electronic interconnects in the physical implementation.

Figure 2.4(a) shows that, in a 4×3 GWOR, the masters and the slaves from the same IP-cores are close to each other. When the 4 IP-cores are connected to this router, the extra crossings and long detours may be avoided in physical layout. Unfortunately, when the number of IP-cores is odd, a pair of master and slave are separated to two opposite sides in GWOR. For example, in 5×4 GWOR, m_3 and s_3 are placed apart from each other as shown in Figure 2.4(b). In this case, the extra crossings or long detours are inevitable in the physical layout.

All the logic topologies mentioned above are built with 2×2 CSEs. There remains a need for an efficient method that can decrease the number of MRRs and flexibly extent itself for any sizes of networks without degrading system performance. After being motivated, I propose a 4×3 *Hash* and a scalable topology *Light*.

3. Hash: A 4×3 Basic Building Block for Light

Hash is a novel 4×3 optical router built by 4 PSEs. In this section, I present the structure of *Hash* structure and its logic scheme. For the same size of networks, *Hash* requires the least MRRs comparing with the typical state-of-the-art WRONoC topologies built with 2×2 CSEs; and I prove this statement in this section. Furthermore, *Hash* can be used as a basic building block to construct the $N \times N$-1 *Light* topology.

3.1. Structure of Hash

Hash consists of 2 pairs of parallel waveguides and 4 MRRs. Each MRR is placed between two parallel waveguides to form a PSE. Four masters and four slaves are connected to the waveguides as shown in Figure 3.1.

As explained in Section 2.3, self-communications can be removed in the logic scheme. In *Hash*, a master would not send signals carried by the wavelengths to the slave from the same IP-core. Except for the self-communications, a master can communicate with the slaves from different IP-cores. For example, m_1 communicates with s_2, s_3 and s_4 with signals on λ_2, λ_3 and λ_1, respectively. The signal carried by λ_1 is coupled with the PSE at the upper left and switched to s_4, while the signal carried by λ_2 is coupled with the PSE at the lower left then received by s_2. The signal on λ_3 is not affected by any MRRs, and it goes to s_3 directly. The three signal paths, represented by the blue, green, and red line in Figure 3.1, can be transmitted at the same time to achieve the full-bandwidth communication among m_1 and the three slaves from different IP-cores.

The wavelengths used by all signal paths are presented by the communication matrix in Figure 3.2. For example, m_2 sends signals carried by λ_2, λ_1 and λ_3 to s_1, s_3 and s_4, respectively. By taking good advantage of the removal of all self-communications in logic scheme, *Hash* only uses 4 MRRs to support all 12 paths simultaneously. It can be proved that the least number of MRRs used to support 12 different signal paths is 4.

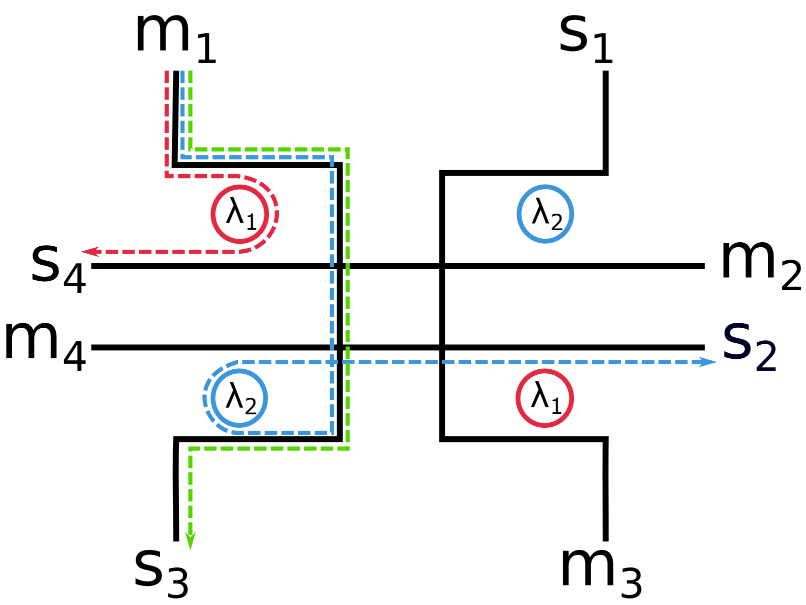

Figure 3.1.: The structure of Hash

$$
\begin{array}{c|cccc|}
 & m_1 & m_2 & m_3 & m_4 \\
\hline
s_1 & \times & \lambda_2 & \lambda_3 & \lambda_1 \\
s_2 & \lambda_2 & \times & \lambda_1 & \lambda_3 \\
s_3 & \lambda_3 & \lambda_1 & \times & \lambda_2 \\
s_4 & \lambda_1 & \lambda_3 & \lambda_2 & \times
\end{array}
$$

Figure 3.2.: Communication matrix for Hash

3.2. Proof of Minimization in MRR Usage

For a network with 4 IP-cores, after removing all self-communications, totally 12 signal paths are left, which are $m_1{\rightarrow}s_2$, $m_1{\rightarrow}s_3$, $m_1{\rightarrow}s_4$, $m_2{\rightarrow}s_1$, $m_2{\rightarrow}s_3$, $m_2{\rightarrow}s_4$, $m_3{\rightarrow}s_1$, $m_3{\rightarrow}s_2$, $m_3{\rightarrow}s_4$, $m_4{\rightarrow}s_1$, $m_4{\rightarrow}s_2$ and $m_4{\rightarrow}s_3$.

As explained in Section 2.2, in a 2×2 CSE or a 2×2 PSE, maximum 4 different signal paths can travel simultaneously shown in Figure 2.2. Consequently, at least three 2×2 CSEs or 2×2 PSEs are required to achieve all 12 paths, without considering any limitations.

As I remove all self-connectivities, for a 2×2 CSE or a 2×2 PSE, two constraints should be added to prevent self-communications:

$$in_1 \neq out_1, in_1 \neq out_2 \tag{3.1}$$

$$in_2 \neq out_1, in_2 \neq out_2 \tag{3.2}$$

where in_1 and in_2 are the inputs of the 2×2 CSEs or the 2×2 PSEs, which can be connected only to master ports. On the other hand, out_1 and out_2 are the outputs of the 2×2 CSEs or the 2×2 PSEs, which can be connected only to slave ports.

An additional constraint should be added to preclude that a port, either a master or a slave, is connected to two different ends of the router, which is expressed as

$$in_1 \neq in_2, out_1 \neq out_2 \tag{3.3}$$

Taking constraints (3.1), (3.2) and (3.3) into consideration, for each 2×2 CSE or 2×2 PSE, two outputs are determined, once two inputs are confirmed. There are 6 possible combinations of ports in the 2×2 CSE or the 2×2 PSE:

①The 2×2 CSE$_1$ or 2×2 PSE$_1$ with ports m_1, m_2, s_3 and s_4 supports 4 signal paths, which are $m_1{\rightarrow}s_3$, $m_1{\rightarrow}s_4$, $m_2{\rightarrow}s_3$ and $m_2{\rightarrow}s_4$.

②The 2×2 CSE$_2$ or 2×2 PSE$_2$ with ports m_1, m_3, s_2 and s_4 supports 4 signal paths, which are $m_1{\rightarrow}s_2$, $m_1{\rightarrow}s_4$, $m_3{\rightarrow}s_2$ and $m_3{\rightarrow}s_4$.

③The 2×2 CSE$_3$ or 2×2 PSE$_3$ with ports m_1, m_4, s_2 and s_3 supports 4 signal paths, which are $m_1 \rightarrow s_2$, $m_1 \rightarrow s_3$, $m_4 \rightarrow s_2$ and $m_4 \rightarrow s_3$.

④The 2×2 CSE$_4$ or 2×2 PSE$_4$ with ports m_2, m_3, s_1 and s_4 supports 4 signal paths, which are $m_2 \rightarrow s_1$, $m_2 \rightarrow s_4$, $m_3 \rightarrow s_1$ and $m_3 \rightarrow s_4$.

⑤The 2×2 CSE$_5$ or 2×2 PSE$_5$ with ports m_2, m_4, s_1 and s_3 supports 4 signal paths, which are $m_2 \rightarrow s_1$, $m_2 \rightarrow s_3$, $m_4 \rightarrow s_1$ and $m_4 \rightarrow s_3$.

⑥The 2×2 CSE$_6$ or 2×2 PSE$_6$ with ports m_3, m_4, s_1 and s_2 supports 4 signal paths, which are $m_3 \rightarrow s_1$, $m_3 \rightarrow s_2$, $m_4 \rightarrow s_1$ and $m_4 \rightarrow s_2$.

To build up 12 paths, the chosen 2×2 CSEs or 2×2 PSEs should support all signal paths. Unfortunately, it's impossible to find out three 2×2 CSEs or 2×2 PSEs that can support all signal paths in the network with 4 IP-cores. However, four 2×2 CSEs or 2×2 PSEs, such as CSE$_1$, CSE$_2$, CSE$_4$ and CSE$_6$, can easily support these paths.

For a network with 4 IP-cores, the WRONoC topologies built by the 2×2 CSEs with two identical MRRs require more MRRs than *Hash*. Comparing to 12 MRRs in λ-router and 8 MRRs in GWOR, *Hash* only requires 4 MRRs by using the 2×2 PSEs, which provides a way to reduce MRR usage.

3.3. Three Types of Signal Paths in Hash

From Figure 3.1 it can be observed that the signal paths in *Hash* can be divided into three groups:

1) Signals on Type-I paths directly pass through the waveguides and reach the slave ports without being coupled by any MRRs. In other words, the signals on Type-I paths are the *off-resonance* signals to all MRRs. For instance, the signal from m_1 to s_3, represented by the green line in Figure 3.1, is not coupled with any MRRs along this waveguide. In this case, the wavelength of the signal path $m_1 \rightarrow s_3$ can be assigned to λ_3 which is not the resonant wavelengths of the MRRs in this *Hash*. Therefore, the wavelengths of Type-I signal paths can be configured as any wavelengths except for the resonant wavelengths of MRRs.

2) Signals on Type-II paths are coupled with the first MRR that they meet along the waveguide. For example, the signal from m_1 to s_4 carried by λ_1, represented by the red line in Figure 3.1, is coupled with the MRRs at the upper left, then switched to the opposite waveguide connected to s_4. In this *Hash*, this signal from m_1 to s_4 do not pass any crossings or other MRRs in this *Hash*. Hence, one property of the signals on Type-II paths is that they would not pass any crossings inside the *Hash* that can changes their direction.

3) Signals on Type-III paths are coupled with the second MRR that they meet along the waveguide. After that, signals pass through some crossings and another MRR but without being coupled. For example, the signal from m_1 to s_2 carried by λ_2, represented by the blue line in Figure 3.1, passes the MRR at the upper left; then it is coupled with the MRR at the lower left. After passing the MRR at the lower right, the signal is received by s_2. The signals on Type-III paths have to pass more MRRs and crossings than the signals on Type-II paths. Hence, they would generate more insertion loss and crosstalk noise. The analysis of crosstalk noise and insertion loss in *Hash* is given in Section 5.2.

Table 3.1 lists all paths of each type in *Hash*.

Table 3.1.: Three types of signal paths in *Hash*

Types of signal paths	Signal paths
Type-I	$m_1 \rightarrow s_3$, $m_2 \rightarrow s_4$, $m_3 \rightarrow s_1$, $m_4 \rightarrow s_2$
Type-II	$m_1 \rightarrow s_4$, $m_2 \rightarrow s_1$, $m_3 \rightarrow s_2$, $m_4 \rightarrow s_3$
Type-III	$m_1 \rightarrow s_2$, $m_2 \rightarrow s_3$, $m_3 \rightarrow s_4$, $m_4 \rightarrow s_1$

4. Light: An N×N-1 Scalable WRONoC Topology

As introduced before, *Hash* is used as the basic building block for an $N \times N$-1 scalable WRONoC topology: *Light*. In this section, I present the general structure of *Light* and a rule of assigning wavelengths to MRRs in *Light*. To illustrate the way to construct *Light* for any sizes of networks, two examples of an 8×7 *Light* topology and a 7×6 *Light* topology are displayed in Section 4.3.

4.1. General Structure of Light

The structure of an $N \times N$-1 *Light* is shown by Figure 4.1. In an $N \times N$-1 *Light*, $\lceil \frac{N}{2} \rceil (\lceil \frac{N}{2} \rceil - 1)/2$ Hashes are needed; and this structure can be expanded to any sizes. The steps are followed to form the structure:

1) Place $\lceil \frac{N}{2} \rceil - 1 - (k-1)$ Hashes horizontally in k-th row with $1 \leq k \leq \lceil \frac{N}{2} \rceil - 1$. Connect the left ports of each Hash with its left neighbor.

2) Connect the bottom ports of Hash to its bottom neighbor except for the Hash at the rightmost of each row. Connect the bottom ports of Hash at the rightmost of each row to its bottom left neighbor.

3) Connect the upper ports of Hashes in the first row to the ports m_1, s_1, m_2, s_2, m_3, s_3, $\ldots$, $m_{\lceil \frac{N}{2} \rceil - 1}$ and $s_{\lceil \frac{N}{2} \rceil - 1}$, sequentially. If the number of IP-cores is even, then connect the right input and output ports of the Hash at the rightmost in the first row to $m_{\lceil \frac{N}{2} \rceil}$ and $s_{\lceil \frac{N}{2} \rceil}$.

4) Connect the left ports of Hashes in the first column to ports m_N, s_N, m_{N-1}, s_{N-1}, $\ldots$, $m_{\lceil \frac{N+1}{2} \rceil + 2}$, $s_{\lceil \frac{N+1}{2} \rceil + 2}$, $m_{\lceil \frac{N+1}{2} \rceil + 1}$, $s_{\lceil \frac{N+1}{2} \rceil + 1}$, sequentially. Connect the bottom input and output of Hash in the last row to $m_{\lceil \frac{N+1}{2} \rceil}$ and $s_{\lceil \frac{N+1}{2} \rceil}$.

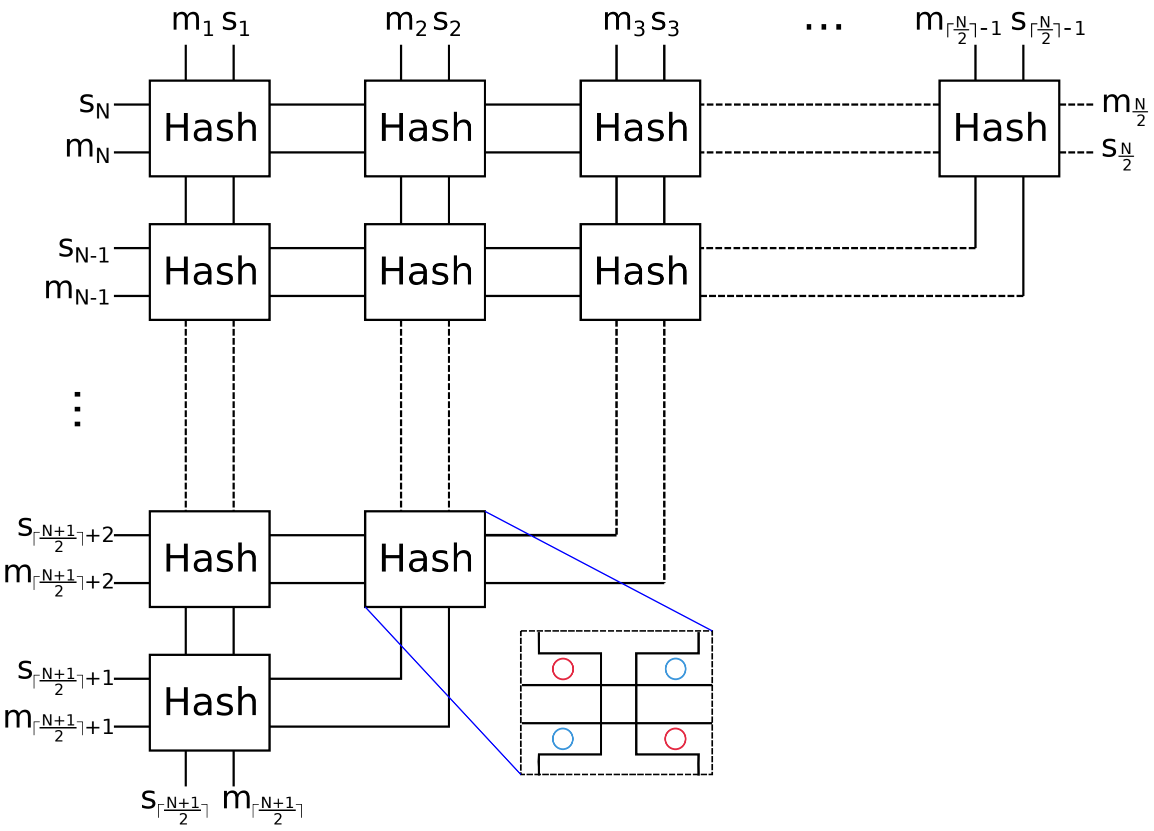
m_1 s_1
m_2 s_2
m_3 s_3
...
$m_{\lceil\frac{N}{2}\rceil\text{-}1}$ $s_{\lceil\frac{N}{2}\rceil\text{-}1}$
$m_{\frac{N}{2}}$
$s_{\frac{N}{2}}$
s_N
m_N
$s_{N\text{-}1}$
$m_{N\text{-}1}$
$s_{\lceil\frac{N+1}{2}\rceil+2}$
$m_{\lceil\frac{N+1}{2}\rceil+2}$
$s_{\lceil\frac{N+1}{2}\rceil+1}$
$m_{\lceil\frac{N+1}{2}\rceil+1}$
$s_{\lceil\frac{N+1}{2}\rceil}$ $m_{\lceil\frac{N+1}{2}\rceil}$
Hash

Figure 4.1.: The $N \times N - 1$ *Light* structure

4.2. Hash Matrix

According to the *Hash* logic scheme in Section 3.1, each *Hash* consists of 4 MRRs. In the *Hash* shown in Figure 3.1, the wavelength of the MRR at the upper left and the wavelength of the MRR at the lower right are identical, while the wavelength of the MRR at the upper right is same as the wavelength of the MRR at the lower left. The two different wavelengths can be regarded as a wavelength-set (Λ). For example, in this *Hash*, the wavelength-set Λ_1 contains λ_1 and λ_2 ($\Lambda_1=\{\lambda_1,\lambda_2\}$). Because of the structure of *Hash*, each *Hash* uses a wavelength-set with two different wavelengths. In this case, the question to assign wavelengths to MRRs is converted into the question to assign wavelength-sets to Hashes.

In order to represent the position of each Hash in *Light*, I build a *Hash Matrix* by labelling each *Hash* as shown in Figure 4.2. After that, a $(\lceil \frac{N}{2} \rceil - 1)\times(\lceil \frac{N}{2} \rceil - 1)$ *Hash Matrix* can be set up as shown in Figure 4.3. For a network with N IP-cores, $\lceil \frac{N}{2} \rceil(\lceil \frac{N}{2} \rceil - 1)/2$ Hashes are required to construct *Light*, and hence the *Hash Matrix* is an oblique upper triangular matrix. If no Hashes in this position, a 0 is filled in this place. Section 4.3 gives an introduction of the way to assign wavelength-sets to Hashes.

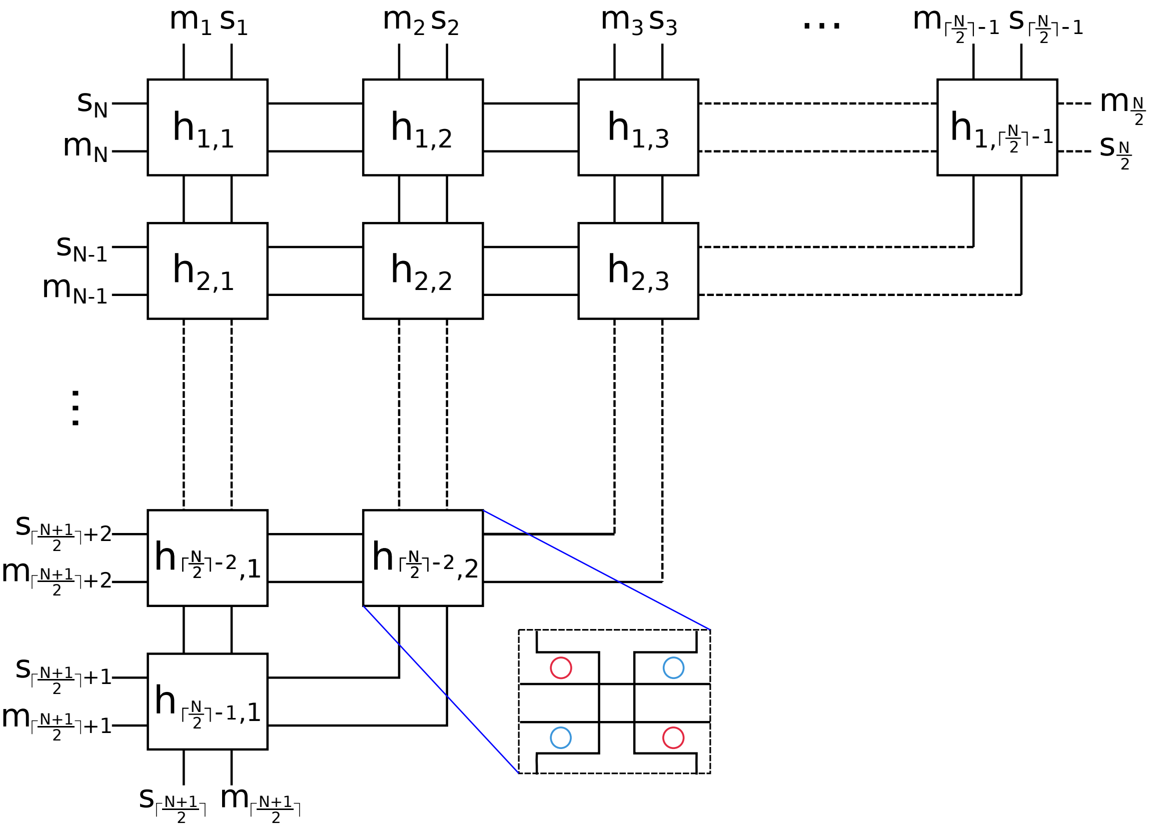

Figure 4.2.: The $N \times N - 1$ *Light* topology with labelling each Hash

$$\begin{bmatrix}
h_{1,1} & h_{1,2} & h_{1,3} & \dots & h_{1,\lceil\frac{N}{2}\rceil-2} & h_{1,\lceil\frac{N}{2}\rceil-1} \\
h_{2,1} & h_{2,2} & h_{2,3} & \dots & h_{2,\lceil\frac{N}{2}\rceil-2} & 0 \\
h_{3,1} & h_{3,2} & h_{3,3} & \dots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
h_{\lceil\frac{N}{2}\rceil-2,1}, & h_{\lceil\frac{N}{2}\rceil-2,2} & 0 & \dots & 0 & 0 \\
h_{\lceil\frac{N}{2}\rceil-1,1} & 0 & 0 & \dots & 0 & 0
\end{bmatrix}$$

Figure 4.3.: $(\lceil\frac{N}{2}\rceil - 1)\times(\lceil\frac{N}{2}\rceil - 1)$ Hash Matrix

4.3. Wavelength Assignment

There are two important principles need to be declared to avoid data-collision: (Tseng et al. 2019)

1) Wavelengths assigned to the signal paths between the same *masters* and different *slaves* must be different;

2) Wavelengths assigned to the signal paths between the different *masters* and same *slaves* must be different.

For example, in Figure 3.1, the wavelength of the signal from m_1 to s_2 is not same as the wavelength of the signal from m_2 to s_2. If m_1 and m_2 send signals on the same wavelength to s_2, then s_2 can not distinguish which signal is from m_1 and which signal is from m_2.

Based on the structure of *Light*, if two Hashes have shared waveguides, then they have the same masters or slaves. In Figure 4.2, the bottom input and output of $h_{1,1}$ are connected to its bottom neighbor $h_{2,1}$. They share the same vertical waveguides; and they have the same masters or slaves in their signal paths, such as $m_N \to s_1$ and $m_{N-1} \to s_1$. In this case, the wavelength-sets for $h_{1,1}$ and $h_{2,1}$ can not be identical.

If no shared waveguides between two Hashes, then the wavelength-sets of two Hashes can be identical. For example, in Figure 4.2, the ports of $h_{1,1}$ are not connected to the ports of $h_{2,2}$. In other words, there are no common masters or slaves in their signal paths. Thus, these Hashes can be configured to the same wavelength-sets.

Each entry of the *Hash Matrix* shown in Figure 4.3 represents a Hash in *Light*. The signal paths of Hashes in the same row or the same column have common masters or slaves. Therefore, the entries of *Hash Matrix* in the same row or column should be configured with different wavelength-sets. Besides that, The Hash in the first row and k-th column has shared waveguides with the Hash in the $(\lceil \frac{N}{2} \rceil - k - 2)$-th row and $(k-1)$-th column with $2 \leq k \leq \lceil \frac{N}{2} \rceil - 1$. Hence the wavelength-sets of these Hashes can not be identical.

The way to assign wavelength-sets to Hashes in *Hash Matrix* is given as follows:

1) For the Hashes in the first column of *Hash Matrix*, assign the wavelength-set Λ_k to $h_{k,1}$ with $1 \leq k \leq \lceil \frac{N}{2} \rceil - 1$ successively.

2) For the Hashes in the first row, assign $\Lambda_{\lceil \frac{N}{2} \rceil-(k-2)}$ to $h_{1,k}$ with $2 \leq k \leq \lceil \frac{N}{2} \rceil - 1$.

3) Assign the wavelength-set of $h_{i-1,j-1}$ to the $h_{i,j}$ with $2 \leq i \leq \lceil \frac{N}{2} \rceil - 1$ and $2 \leq j \leq \lceil \frac{N}{2} \rceil - 1$. If the entry of *Hash Matrix* is 0, then no wavelength-sets need to be assigned to this entry.

Figure 4.4 presents the general assignment of wavelength-sets to the $\lceil \frac{N}{2} \rceil - 1 \times \lceil \frac{N}{2} \rceil - 1$ *Hash Matrix.*

$$\begin{bmatrix}
\Lambda_1 & \Lambda_{\lceil \frac{N}{2} \rceil} & \Lambda_{\lceil \frac{N}{2} \rceil-1} & \Lambda_{\lceil \frac{N}{2} \rceil-2} & \dots & \Lambda_3 \\
\Lambda_2 & \Lambda_1 & \Lambda_{\lceil \frac{N}{2} \rceil} & \Lambda_{\lceil \frac{N}{2} \rceil-1} & \dots & 0 \\
\Lambda_3 & \Lambda_2 & \Lambda_1 & \Lambda_{\lceil \frac{N}{2} \rceil} & \dots & 0 \\
\vdots & \Lambda_3 & \Lambda_2 & \Lambda_1 & \dots & \vdots \\
\vdots & \vdots & \Lambda_3 & \Lambda_2 & \dots & \vdots \\
\vdots & \vdots & \vdots & \Lambda_3 & \dots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\Lambda_{\lceil \frac{N}{2} \rceil-3} & \Lambda_{\lceil \frac{N}{2} \rceil-4} & \Lambda_{\lceil \frac{N}{2} \rceil-5} & 0 & \dots & \vdots \\
\Lambda_{\lceil \frac{N}{2} \rceil-2} & \Lambda_{\lceil \frac{N}{2} \rceil-3} & 0 & 0 & \dots & \vdots \\
\Lambda_{\lceil \frac{N}{2} \rceil-1} & 0 & 0 & 0 & \dots & 0
\end{bmatrix}$$

Figure 4.4.: Wavelength-set assignment to the $(\lceil \frac{N}{2} \rceil - 1)\times(\lceil \frac{N}{2} \rceil - 1)$ Hash Matrix

The wavelength-sets assigned to an 8×7 and a 16×15 *Light* are displayed in Figure 4.5 and Figure 4.6.

$$\begin{bmatrix}
\Lambda_1 & \Lambda_4 & \Lambda_3 \\
\Lambda_2 & \Lambda_1 & 0 \\
\Lambda_3 & 0 & 0
\end{bmatrix}$$

Figure 4.5.: Wavelength-set assignment to a 3×3 *Hash Matrix* of an 8×7 *Light* topology

$$\begin{bmatrix}
\Lambda_1 & \Lambda_8 & \Lambda_7 & \Lambda_6 & \Lambda_5 & \Lambda_4 & \Lambda_3 \\
\Lambda_2 & \Lambda_1 & \Lambda_8 & \Lambda_7 & \Lambda_6 & \Lambda_5 & 0 \\
\Lambda_3 & \Lambda_2 & \Lambda_1 & \Lambda_8 & \Lambda_7 & 0 & 0 \\
\Lambda_4 & \Lambda_3 & \Lambda_2 & \Lambda_1 & 0 & 0 & 0 \\
\Lambda_5 & \Lambda_4 & \Lambda_3 & 0 & 0 & 0 & 0 \\
\Lambda_6 & \Lambda_5 & 0 & 0 & 0 & 0 & 0 \\
\Lambda_7 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Figure 4.6.: Wavelength-set assignment to a 7×7 *Hash Matrix* of a 16×15 *Light* topology

To illustrate the way to construct the *Light* topology when the number of IP-cores is even, the example of an 8×7 *Light* topology is presented. An 8×7 *Light* consists of 6 Hashes, and the wavelength-set assigned to each Hash is given by Figure 4.5. Assume that Λ_1 = (λ_1,λ_2), Λ_2 = (λ_3,λ_4), Λ_3 = (λ_5,λ_6), Λ_4 = (λ_7,λ_8), in $h_{1,1}$ the MRR at the upper left and the MRR at the lower right are configured with λ_1, represented by the red circles in Figure 4.7, while the MRR at the lower left and the MRR at the upper right are configured with λ_2. After assigning wavelengths to all MRRs, the 8×7 *Light* logic scheme is displayed in Figure 4.7.

The communication matrix of the 8×7 *Light* in Figure 4.8 indicates which wavelength is used by a signal path. For example, m_1 communicates with s_8 by the signals carried by λ_1, while m_2 sends signals carried by λ_7 to s_8.

If a master is directly connected to a slave by a waveguide, the wavelength of this signal path can be configured with any wavelengths except for the resonant wavelengths of MRRs along the waveguide. For example, the 6 MRRs from top to bottom along the waveguide between m_1 and s_5 are configured with $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6$, respectively. Thus, the wavelength of the signal path between m_1 and s_5 is assigned to $\lambda_7 \neq \{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6\}$.

If the number of IP-cores is odd, then the right ports of the rightmost Hash in the first row do not connect to any IP-cores. Take a 7×6 *Light* topology as an example, its structure shown in Figure 4.9 which is almost the same as the structure of the 8×7 *Light* topology. Furthermore, the 7×6 *Light* topology has the same wavelength-set assignment of Hashes as the 8×7 *Light* has, but their communication matrices are partially different. The communication matrix of the 7×6 *Light* topology is presented in Figure 4.10. Comparing Figure 4.10 with Figure 4.8, the signal paths between m_1 and s_4 have different wavelength assignments in these two sizes of topologies (λ_7 in 7×6 *Light* and λ_2 in 8×7 *Light*), which ascribes to the placement of ports. For example m_4 and s_4 are placed at the rightmost in the 8×7 *Light*, whereas they are placed at the bottommost in the 7×6 *Light.* In spite of different placements of ports, the masters and slaves from same IP-cores are always placed close to each other in any sizes of networks.

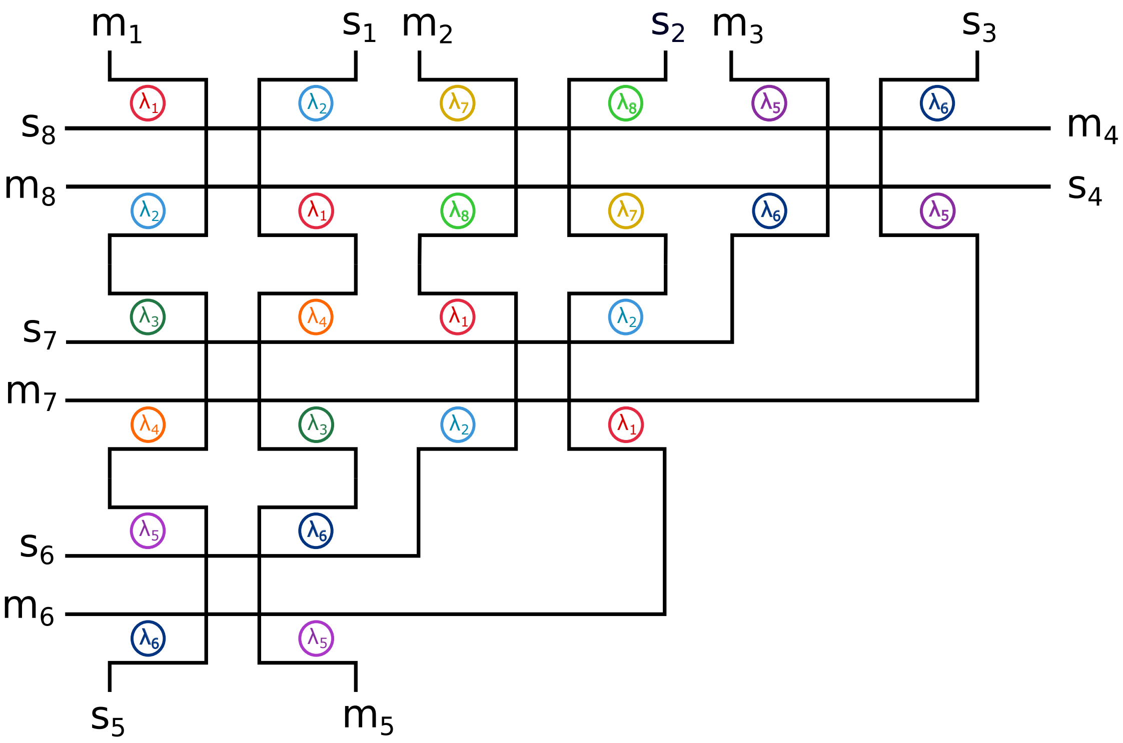

Figure 4.7.: A 8 × 7 *Light* topology

$$
\begin{array}{c|cccccccc}
 & m_1 & m_2 & m_3 & m_4 & m_5 & m_6 & m_7 & m_8 \\
s_1 & \times & \lambda_6 & \lambda_4 & \lambda_2 & \lambda_7 & \lambda_5 & \lambda_3 & \lambda_1 \\
s_2 & \lambda_6 & \times & \lambda_2 & \lambda_8 & \lambda_5 & \lambda_3 & \lambda_1 & \lambda_7 \\
s_3 & \lambda_4 & \lambda_2 & \times & \lambda_6 & \lambda_3 & \lambda_1 & \lambda_7 & \lambda_5 \\
s_4 & \lambda_2 & \lambda_8 & \lambda_6 & \times & \lambda_1 & \lambda_7 & \lambda_5 & \lambda_3 \\
s_5 & \lambda_7 & \lambda_5 & \lambda_3 & \lambda_1 & \times & \lambda_6 & \lambda_4 & \lambda_2 \\
s_6 & \lambda_5 & \lambda_3 & \lambda_1 & \lambda_7 & \lambda_6 & \times & \lambda_2 & \lambda_8 \\
s_7 & \lambda_3 & \lambda_1 & \lambda_7 & \lambda_5 & \lambda_4 & \lambda_2 & \times & \lambda_6 \\
s_8 & \lambda_1 & \lambda_7 & \lambda_5 & \lambda_3 & \lambda_2 & \lambda_8 & \lambda_6 & \times
\end{array}
$$

Figure 4.8.: Communication matrix for a 8×7 *Light* topology

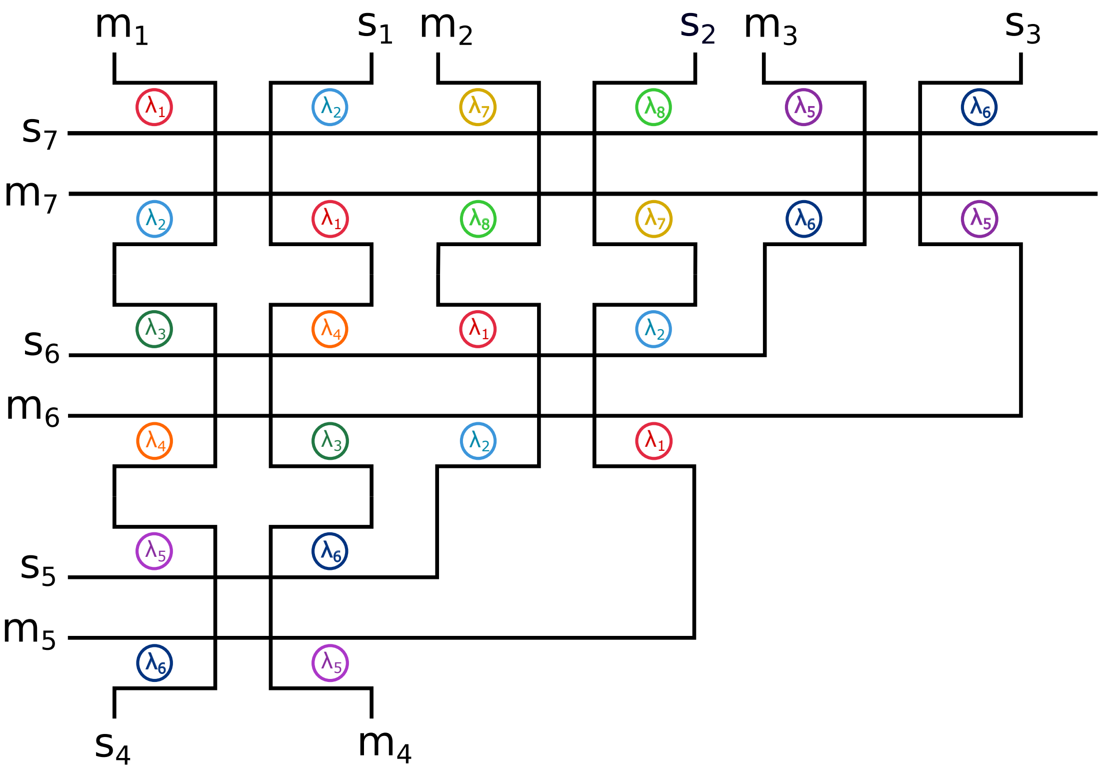
m_1
s_1
m_2
s_2
m_3
s_3
λ_1 λ_2 λ_7 λ_8 λ_5 λ_6
s_7
m_7
λ_2 λ_1 λ_8 λ_7 λ_6 λ_5
λ_3 λ_4 λ_1 λ_2
s_6
m_6
λ_4 λ_3 λ_2 λ_1
λ_5 λ_6
s_5
m_5
λ_6 λ_5
s_4
m_4

Figure 4.9.: A 7×6 *Light* topology

$$
\begin{array}{c c}
 & \begin{array}{ccccccc} m_1 & m_2 & m_3 & m_4 & m_5 & m_6 & m_7 \end{array} \\
\begin{array}{c} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{array} &
\left[\begin{array}{ccccccc}
\times & \lambda_6 & \lambda_4 & \lambda_7 & \lambda_5 & \lambda_3 & \lambda_1 \\
\lambda_6 & \times & \lambda_2 & \lambda_5 & \lambda_3 & \lambda_1 & \lambda_7 \\
\lambda_4 & \lambda_2 & \times & \lambda_3 & \lambda_1 & \lambda_7 & \lambda_5 \\
\lambda_7 & \lambda_5 & \lambda_3 & \times & \lambda_6 & \lambda_4 & \lambda_2 \\
\lambda_5 & \lambda_3 & \lambda_1 & \lambda_6 & \times & \lambda_2 & \lambda_8 \\
\lambda_3 & \lambda_1 & \lambda_7 & \lambda_4 & \lambda_2 & \times & \lambda_6 \\
\lambda_1 & \lambda_7 & \lambda_5 & \lambda_2 & \lambda_8 & \lambda_6 & \times
\end{array}\right]
\end{array}
$$

Figure 4.10.: Communication matrix for a 8×7 *Light* topology

5. Analysis of Crosstalk Noise and Insertion Loss

The optical switch elements like PSEs and CSEs have inevitable crosstalk noise and insertion loss, which decrease the signal-to-noise ratio (SNR) and cause additional power penalty (Xie et al. 2010). In this section, I analyzed the insertion loss and SNR in the *Hash* and in the *Light*. Before that, Section 5.1 presents a brief introduction of the crosstalk noise and insertion loss in OSEs. Section 5.2 summarizes the insertion loss and SNR in the *Hash* and Section 5.3 demonstrates the insertion loss and SNR in the *Light* topology.

5.1. Crosstalk Noise and Insertion Loss in Optical Switch Elements

The insertion loss contains *propagation loss* (l_p), which is related to the lengths of waveguides, *bending loss* (l_b), which is related to the number of bending waveguides, *through loss* (l_t), which is generated when an *off-resonance* signal passes an MRR, *drop loss* (l_d), which is generated when an *on-resonance* signal passes an MRR and *crossing loss* (l_c), which is generated when signals pass a crossing (Truppel et al. 2020).

The *bending loss* and *propagation loss* are hardly evaluated with a logic scheme, because the actual length of waveguides and the number of bending waveguides remain unknown without physical implementation. Thus, the analysis of the insertion loss and crosstalk noise analyzed here are mainly about *crossing loss*, *through loss* and *drop loss.*

Figure 5.1 displays the *crossing loss*, *through loss* and *drop loss* in different structures. To calculate the output signal power and noise power in a component, the coefficients of the insertion loss and crosstalk noise are denoted by L_t, L_d, L_c, K_r and K_c, where L_t denotes the through loss coefficient, L_d denotes the drop loss coefficient, L_c denotes the crossing loss coefficient, K_r denotes the crosstalk coefficient per MRR, and K_c denotes the crosstalk coefficient per crossing. These values multiplied with the input signal power are negative to indicate the power of output signals and crosstalk noise signals.

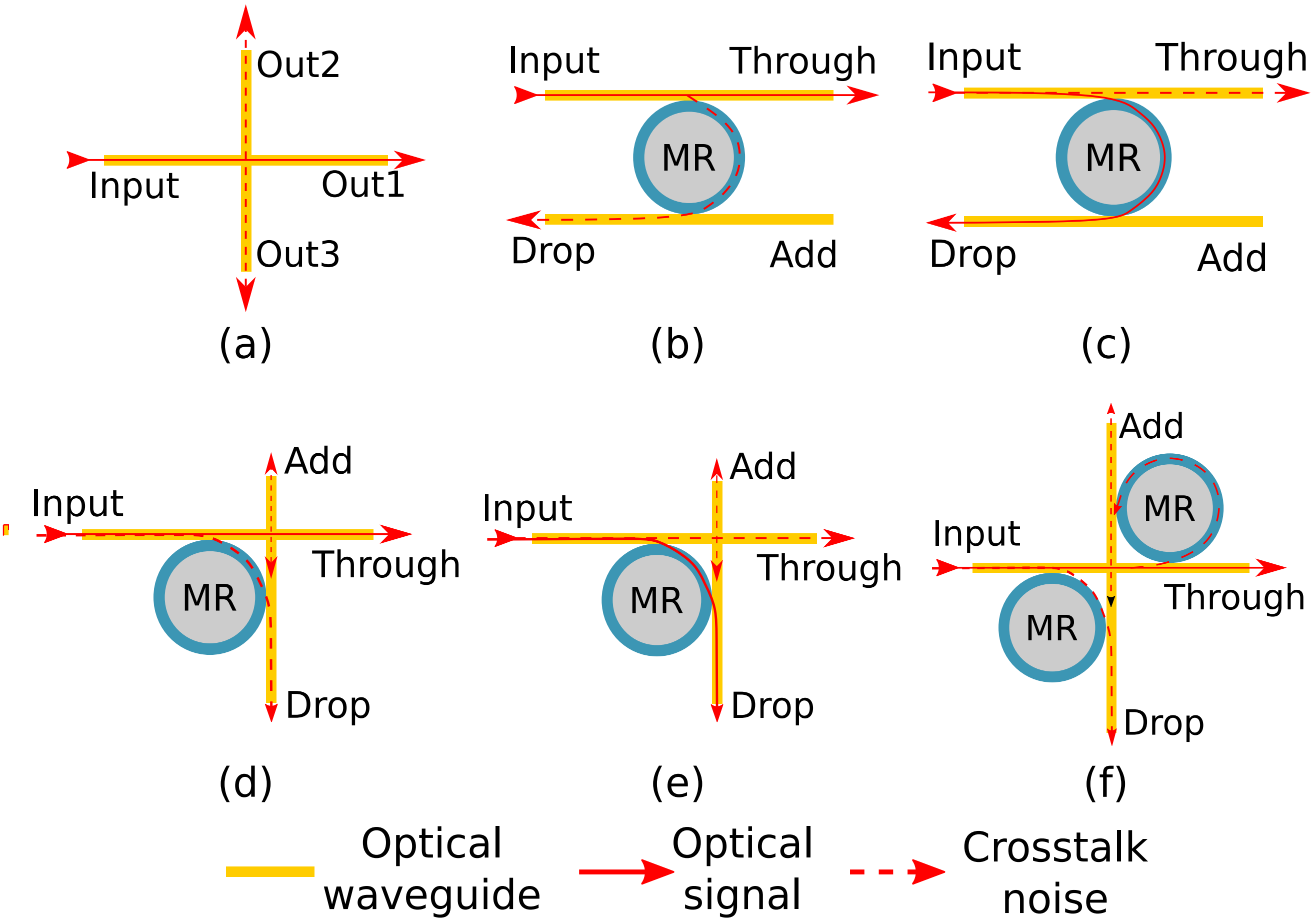

Figure 5.1.: (a) Crosstalk in the crossing of two orthogonal waveguides
(b) The crosstalk when *off-resonance* signals passing PSEs
(c) The crosstalk when *on-resonance* signals passing PSEs
(d) The crosstalk when *off-resonance* signals passing CSEs with 1 MRR
(e) The crosstalk when *on-resonance* signals passing CSEs with 1 MRR
(f) The crosstalk when *off-resonance* signals passing CSEs with 2 MRRs

For a crossing shown in Figure 5.1(a), the input signal passes the crossing with non-negligible *crossing loss*, while a portion of signal power goes to *out2* and *out3* as noise represented by red dash line in Figure 5.1(a). The output signal power at *out1* and the noise power at *out2* and at *out3* can be calculated as

$$P_{O1,signal} = L_c P_I \tag{5.1}$$

$$P_{O2,noise} = P_{O3,noise} = K_c P_I \tag{5.2}$$

where P_I is the input signal power.

Secondly, when an *off-resonance* signal passes through an MRR in PSE with *through loss*, the output signal power at the *through* port ($P_{T,signal,pse}$) can be calculated as

$$P_{T,signal,pse} = L_t P_I \tag{5.3}$$

and the noise power at the *drop* port in PSEs as shown by Figure 5.1(b) can be calculated as

$$P_{D,noise,pse} = K_r P_I \tag{5.4}$$

On the other hand, when an *on-resonance* signal is coupled with the MRR in a PSE, the output signal power at *drop* port ($P_{D,signal,pse}$) and noise power at *through* port ($P_{T,noise,pse}$) are

$$P_{D,signal,pse} = L_d P_I \tag{5.5}$$

$$P_{T,noise,pse} = K_r P_I \tag{5.6}$$

When the *off-resonance* signal passes through the CSE shown in Figure 5.1(d) with *through loss*, the noise signals go to *drop* port and *add* port, respectively. The output signal power at *through* port ($P_{T,signal,cse}$) is

$$P_{T,signal,cse} = L_t L_c P_I \tag{5.7}$$

and noise power ($P_{D,noise,cse}$) and ($P_{A,noise,cse}$) can be expressed as

$$P_{D,noise,cse} = (K_r + L_t^2 K_c) P_I \tag{5.8}$$

$$P_{A,noise,cse} = K_r L_t P_I \tag{5.9}$$

For the *on-resonance* signal shown by Figure 5.1(e), the output power and noise power can be calculated as

$$P_{D,signal,cse} = L_d P_I \tag{5.10}$$

$$P_{T,noise,cse} = L_c K_r P_I \tag{5.11}$$

$$P_{A,noise,cse} = K_c K_r P_I \tag{5.12}$$

In Figure 5.1(f), the *off-resonance* signal pass the 2×2 CSE. In this case, the output power and noise power can be calculated as $P_{T,signal,cse2} = L_c L_t^2 P_I$, $P_{D,noise,cse2} = (K_r + K_c L_t^2 + K_r L_c^2 L_t^2)P_I$ and $P_{A,noise,cse2} = (K_r L_t^2)P_I$. For an *on-resonance* signal, the output power and noise power in a 2×2 CSE can be calculated as $P_{D,signal,cse2} = L_d P_I$, $P_{T,noise,cse2} = L_c K_r L_t$ and $P_{A,noise,cse2} = K_c K_r L_t P_I$.

The values of the insertion loss coefficients and crosstalk coefficients are shown in Table 5.1 and Table 5.2 (Nikdast et al. 2015).

Table 5.1.: Insertion loss coefficients

Through loss coefficient (L_t)	Drop loss coefficient (L_d)	Crossing loss coefficient (L_c)
-0.0005dB	-0.5dB	-0.04dB

Table 5.2.: Crosstalk coefficients

Crosstalk coefficient per MRR (K_r)	Crosstalk coefficient per crossing (K_c)
-25dB	-40dB

With these values, Table 5.3 presents the power of insertion loss and crosstalk noise in a 2×2 PSE and a 2×2 CSE. For an *on-resonance* signal, the loss power values and noise power values in both elements are almost identical. However, for an *off-resonance* signal, a 2×2 PSE has less insertion loss and crosstalk noise than a 2×2 CSE. As a result, PSEs have great potential to reduce insertion loss and enhance the SNR.

Table 5.3.: Loss power and noise power in a 2×2 PSE and a 2×2 CSE

	For an *off-resonance* signal		For an *on-resonance* signal	
	in a 2×2 PSE	in a 2×2 CSE	in a 2×2 PSE	in a 2×2 CSE
Loss power/dB	-0.0005dB×P_I	-0.041dB×P_I	-0.5dB×P_I	-0.5dB×P_I
Noise power/dB	-25dB×P_I	-24.61dB×P_I	-25dB×P_I	-25.04dB×P_I

5.2. Crosstalk Noise and Insertion Loss in Hash

In *Hash*, there are three types of signal paths introduced in Section 3.3. The crossings and MRRs on the signal paths lead to different crosstalk and insertion loss in *Hash*. Additionally, I analyzed the SNR to indicate the signal's quality. In this analysis, only the first-order noise generated by signals is taken into consideration; and the second-order noise and the higher-order noise generated by the first-order noise signals or other noise signals are ignored.

5.2.1. Type-I Signal Paths

The signals on Type-I paths are not coupled by any MRRs. Each signal of this types passes two crossings and two MRRs on its path. Taking signal path $m_1 \rightarrow s_3$, represented by the green line in Figure 5.2, as an example, when the signal from m_1 passes the MRR at the upper left, the *through loss* is generated. And the power of the *through loss* can be calculated as $P_{through,loss} = (1 - L_t)P_I$. Then the signal passes a crossing near this MRR, and the *crossing loss* is generated with $P_{crossing,loss} = (1 - L_c)L_tP_I$. After that, the signal passes another MRR and a crossing to reach s_3. The output signal power at s_3 can be calculated as

$$P_{signal,hash,typ1} = L_{signal,typ1}P_I = {L_c}^2{L_t}^2P_I \tag{5.13}$$

and the loss power on Type-I signal paths can be calculated as

$$P_{loss,hash,typ1} = P_I - P_{signal,hash,typ1} = (1 - L_{signal,typ1})P_I \tag{5.14}$$

To simplify the calculation, the insertion loss of Type-I signal paths can be easily calculated as $l_{hash,ty1} = 2l_t + 2l_c = 2(-L_c - L_t) = 0.09dB$. The negative sign should be added to indicate the insertion loss value rather than the coefficient of insertion loss.

The crosstalk noise generated by the signals on Type-I paths will degrade the signal's quality and decrease the SNR values at other slave ports. For example, the signal from m_1 to s_3 generates 4 noise signals, represented by n_1, n_2, n_3 and n_4 in Figure 5.2, to s_2 and s_4. As shown in Figure 5.2, the noise signals n_1 and n_2 go to s_4, while n_3 and n_4 go to s_2. The noise

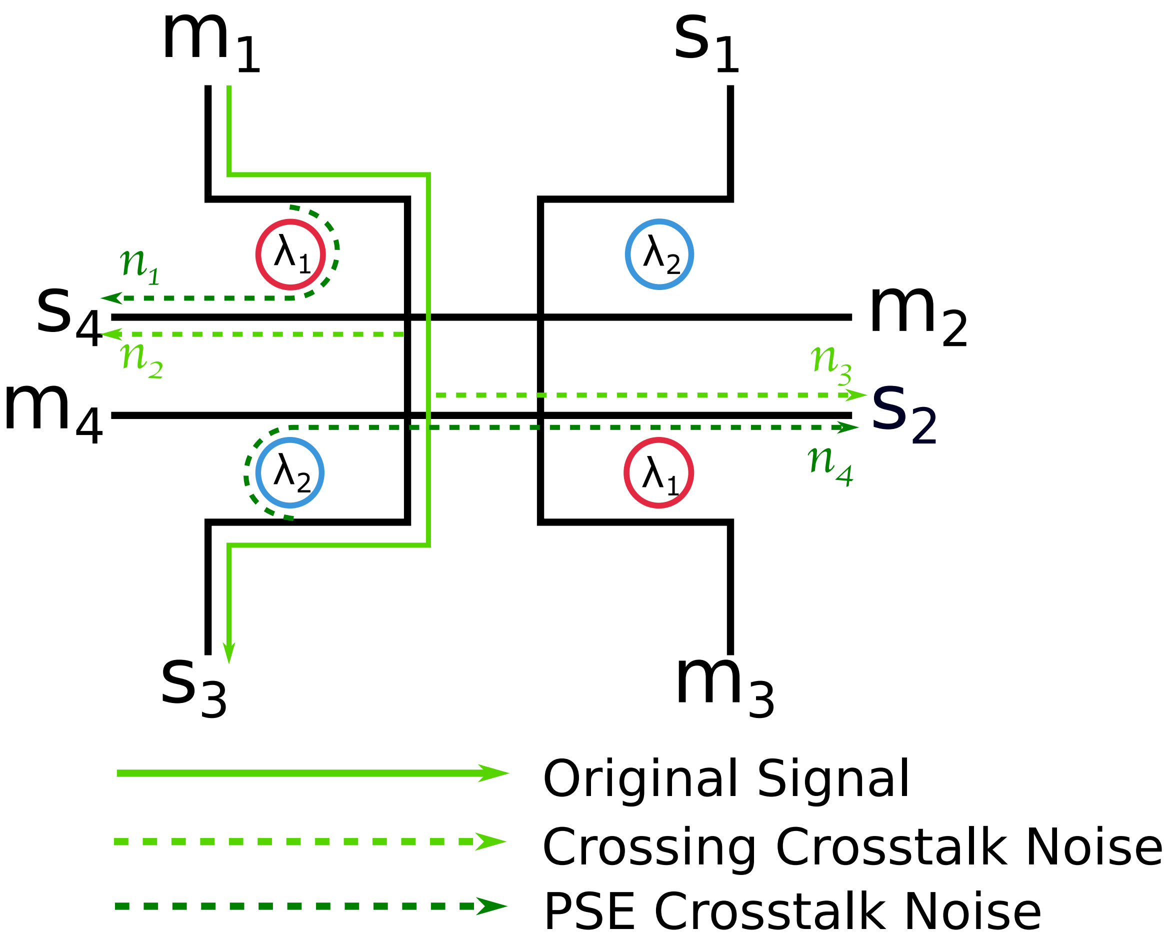

Figure 5.2.: Crosstalk noise and insertion loss of Type-I signal paths

signal power can be expressed as

$$
\begin{aligned}
P_{noise,hash,typ1} &= (L_{noise,typ1,1} + L_{noise,typ1,2})P_I \\
&= (\underbrace{K_r}_{n_1} + \underbrace{K_c{L_t}^2}_{n_2} + \underbrace{K_c{L_t}^2{L_c}^2}_{n_3} + \underbrace{K_r{L_t}^2{L_c}^4}_{n_4})P_I
\end{aligned}
\tag{5.15}
$$

and the $L_{noise,tp1,1}$ denotes the noise coefficient for s_4 and $L_{noise,typ1,2}$ denotes the noise coefficient for s_2.

As concluded in Table 3.1, the paths $m_1 \rightarrow s_3$, $m_2 \rightarrow s_4$, $m_3 \rightarrow s_1$, $m_4 \rightarrow s_2$ are the Type-I signal paths. Assume that the input signals of all masters have identical power, because of the symmetric structure of the *Hash*, the noise power generated by the Type-I signal paths at each slave is $(L_{noise,typ1,1} + L_{noise,typ1,2})P_I$. With the values in Table 5.1 and Table 5.2, the noise signal power at each slave port is -21.90dB×P_I.

5.2.2. Type-II Signal Paths

The signals on Type-II paths are coupled by the first MRR they meet in a *Hash*. The representative example is the path $m_1 \rightarrow s_4$. The signal from m_1 is coupled with the MRR at the upper left and turned to the waveguide connected to s_4 as shown in Figure 5.3. In its travel, only the *drop loss* is generated, and the loss power is $P_{drop,loss} = (1 - L_d)P_I$. In this case, The output signal power at s_4 can be calculated as

$$
P_{signal,hash,typ2} = L_{signal,typ2}P_I = L_dP_I \tag{5.16}
$$

and the power of the *drop loss* is:

$$
P_{loss,hash,typ2} = P_I - P_{signal,hash,typ2} = (1 - L_{signal,typ2})P_I \tag{5.17}
$$

The insertion loss of this kind of signal paths is expressed as $l_{hash,typ2} = l_d = -L_d = 0.5dB$.

The first-order noise signal caused by path $m_1 \rightarrow s_4$ is represented by the red dash line in Figure 5.3. The noise power can be calculated as

$$
P_{noise,hash,typ2} = L_{noise,typ2} = (K_rL_t{L_c}^2)P_I \tag{5.18}
$$

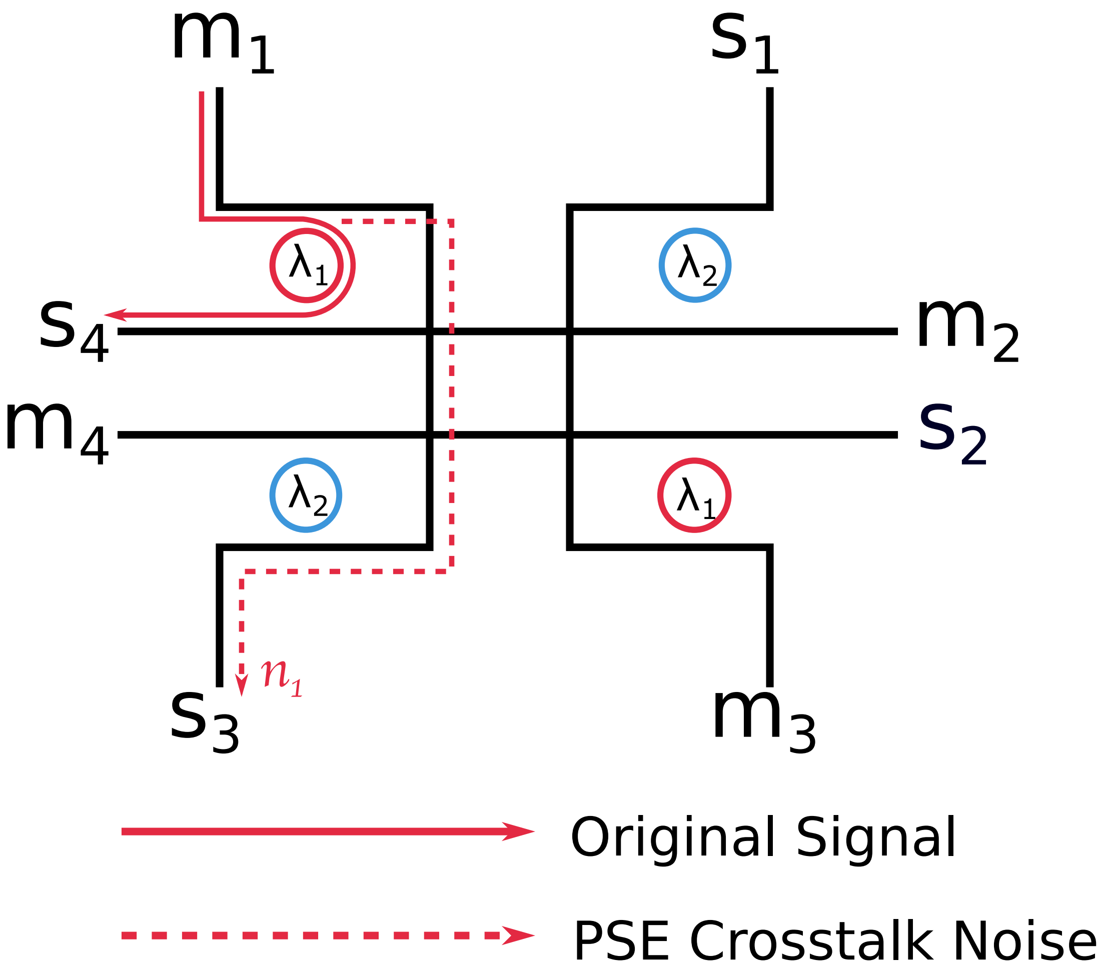

Figure 5.3.: Crosstalk noise and insertion loss of Type-II signal paths

where $L_{noise,typ2}$ denotes the noise coefficient for signals on Type-II paths. For all Type-II paths in Table 3.1, the noise power generated by those signal paths is $L_{noise,typ2} = (K_r L_t {L_c}^2)P_I = -25.1605dB \times P_I$, with the values in Table 5.2 and Table 5.3.

5.2.3. Type-III Signal Paths

The signals on Type-III paths are coupled with the second MRR they meet in the *Hash*. Taking the path $m_1 \to s_2$ as an example shown in Figure 5.4, the signal sent by m_1 ignores the MRR at the upper left, then it is coupled with the second MRR at the lower left in the *Hash* and switched to the waveguide connected to s_2. In this case, the output power at s_2 is

$$P_{signal,hash,typ3} = L_{signal,typ3}P_I = (L_d {L_t}^2 {L_c}^4)P_I \tag{5.19}$$

and the power loss of this kind of path is

$$P_{loss,hash,typ3} = P_I - P_{signal,hash,typ3} = (1 - L_{signal,typ3})P_I \tag{5.20}$$

The insertion loss of Type-III paths can also be expressed as $l_{hash,typ3} = 2l_t + l_d + 4l_c = -2L_t - L_d - 4L_c = 0.67dB$.

The Type-III path $m_1 \to s_2$ generates 5 noise signals shown by Figure 5.4. Among them, n_1, n_2, n_4 and n_5 go to s_4, while n_3 goes to s_3. In this case, the noise power can be calculates as

$$\begin{aligned} P_{noise,block,p2} &= (L_{noise,typ3,1} + L_{noise,typ3,2})P_I \\ &= (\underbrace{K_r}_{n_1} + \underbrace{K_c {L_t}^2}_{n_2} + \underbrace{K_c {L_t}^2 {L_d}^2 {L_c}^6}_{n_4} + \underbrace{K_r {L_t}^2 {L_d}^2 {L_c}^8}_{n_5} + \underbrace{K_r L_t {L_c}^2}_{n_3})P_I \end{aligned} \tag{5.21}$$

where $L_{noise,typ3,1}$ denotes the noise coefficient of the noise signals at s_4 and $L_{noise,p2,3}$ denotes the noise coefficient of the noise signal at s_3. With the values in Table 5.2 and Table 5.3, the noise power at s_4 is $L_{noise,typ3,1}P_I = -24.6710dB \times P_I$ and the noise power at s_3 is $L_{noise,typ3,2}P_I = -25.0805dB \times P_I$.

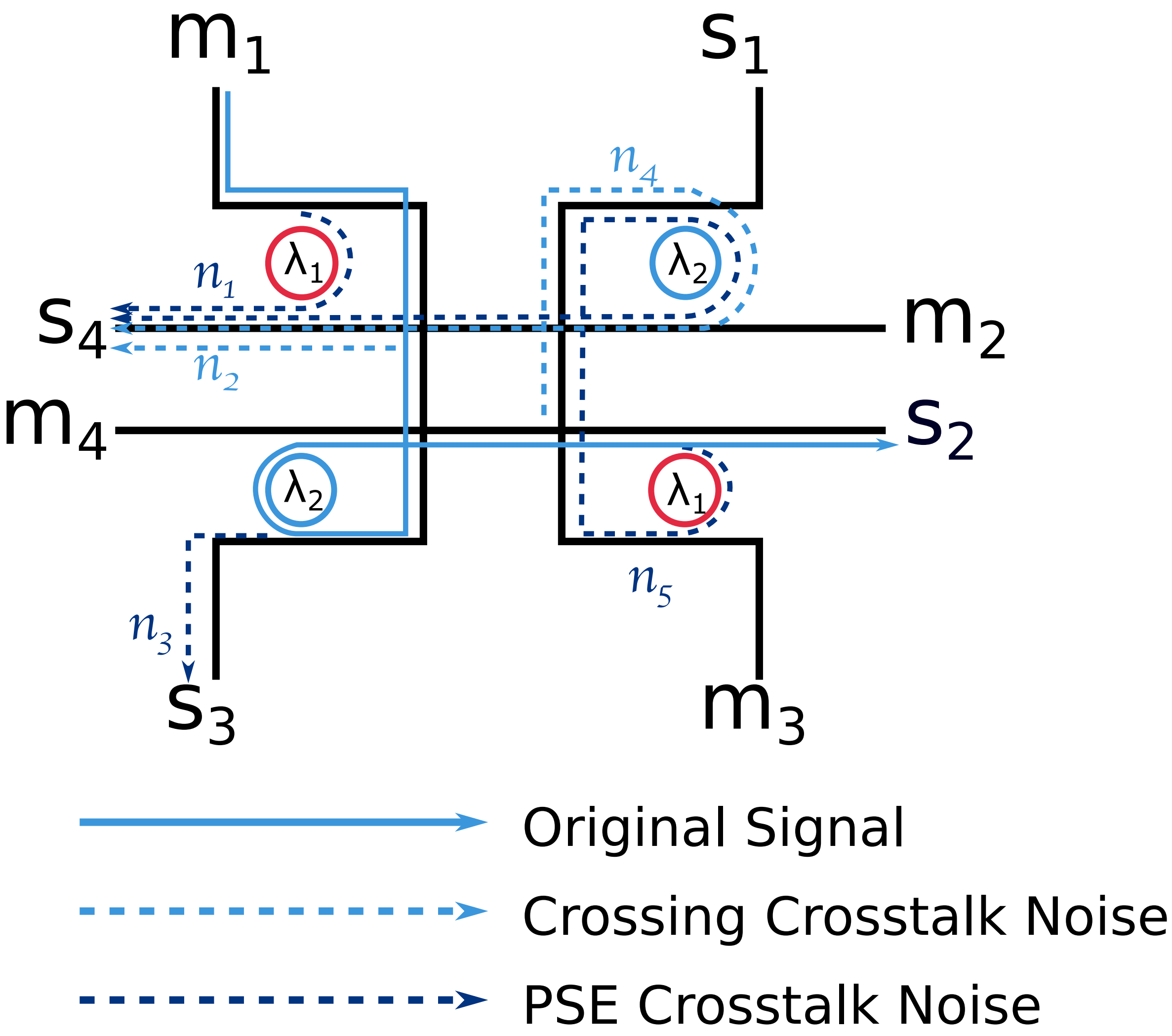

Figure 5.4.: Crosstalk noise and insertion loss of Type-III signal paths

5.2.4. Summary of Crosstalk Noise and Insertion Loss in Hash

Due to the symmetric structure of the *Hash*, the insertion loss values of the signals on the same type of paths are identical. With the definition of SNR, the SNR can be expressed as $10log\frac{P^{\lambda n}_{output}}{P^{\lambda n}_{noise}}$. Similar to the insertion loss value, the SNR value of the signals on the same type of paths are identical.

Table 5.4 summarizes the insertion loss values and SNR values of the three types of signal paths. It can be observed that the signals on Type-I paths have the least insertion loss than the signals on the other two types of paths. Although the insertion loss of Type-II signal paths is nearly equal to the insertion loss of Type-III signal paths, the SNR value of Type-II signal paths is 25% larger than the SNR value of Type-III signal paths. Furthermore, the signals on Type-II paths have the largest SNR value than the signals on other two types of signal paths. Among the three types, Type-III signal paths have the least SNR value because the signals on these paths pass more crossings and MRRs. It is reasonable to infer that the insertion loss and the SNR of a signal are strongly relevant to the number of MRRs and crossings on the signal's tour, which is why the MRR usage needs to be cut down, especially for the large-scale networks.

Comparing the *Hash* to a 4×3 λ-router, the *Hash* has 10% higher average SNR value (22.11dB in *Hash* and 20.12dB in λ-router) and 16% larger worst-case SNR value (19.90dB in *Hash* and 17.14dB in λ-router) than the λ-router. The detailed comparison between *Light* and λ-router in terms of the crosstalk noise and insertion loss is given in Section 6.

Table 5.4.: Insertion loss and SNR in Hash

	Type-I signal paths	Type-II signal paths	Type-III signal paths
Insertion loss/dB	0.09dB	0.5dB	0.67dB
SNR/dB	21.8476dB	24.585dB	19.9019dB

5.3. Crosstalk Noise and Insertion Loss in Light

The crosstalk noise and insertion loss in *Light* are related with the number of Hashes. When the signals on Type-II paths or Type-III paths pass the Hashes, which do not change their directions, they always generate *through loss* twice and *crossing loss* twice in each *Hash*. In order to know how many Hashes in their tours, I label a Hash by a coordinate $h_{X,Y}$. Taking the *Hash* $h_{\lceil \frac{N}{2} \rceil -2,2}$ as an example, represented by the yellow block in Figure 5.5, I demonstrate how many Hashes would be passed by the signals from all directions to the *Hash* $h_{\lceil \frac{N}{2} \rceil -2,2}$.

In the Figure 5.5, the orange line represents the signal from m_2 to $h_{\lceil \frac{N}{2} \rceil -2,2}$. This signal passes $\lceil \frac{N}{2} \rceil - 3$ Hashes before reaching the $h_{\lceil \frac{N}{2} \rceil -2,2}$. Therefore, the signals, received or sent by the top ports of the $h_{X,Y}$, will pass $X-1$ Hashes to reach their destinations.

The signal from $m_{\lceil \frac{N+1}{2} \rceil +2}$ to $h_{\lceil \frac{N}{2} \rceil -2,2}$, represented by the red line in Figure 5.5, passes only 1 Hash, and hence the signals received or sent by the left ports of $h_{X,Y}$, have $Y-1$ Hashes in their tours.

By passing $\lceil \frac{N}{2} \rceil - 3$ hashes, the signal from m_3, represented by the green line in Figure 5.5, reaches the $h_{\lceil \frac{N}{2} \rceil -2,2}$. In total $(\frac{N}{2} - 1) - Y$ Hashes are passes by the signals received by the right ports of $h_{X,Y}$.

The signal from $m_{\lceil \frac{N+1}{2} \rceil +1}$, represented by the blue line in Figure 5.5, passes 1 Hash before arriving at the $h_{\lceil \frac{N}{2} \rceil -2,2}$. Thus, the signals pass $(\frac{N}{2} - 1) - X$ Hashes before being received by the bottom ports of $h_{X,Y}$.

Tab.5.5 summarizes the number of Hashes passed by the signals from the senders in all directions to the *Hash* $h_{X,Y}$ or from the *Hash* $h_{X,Y}$ to receivers in all directions.

Table 5.5.: The number of Hashes on different signal's tours

	the number of Hashes
Signals from/to top ports of $h_{X,Y}$	$n_t = X - 1$
Signals from/to bottom ports of $h_{X,Y}$	$n_b = (\lceil \frac{N}{2} \rceil - 1) - X$
Signals from/to left ports of $h_{X,Y}$	$n_l = Y - 1$
Signals from/to right ports of $h_{X,Y}$	$n_r = (\lceil \frac{N}{2} \rceil - 1) - Y$

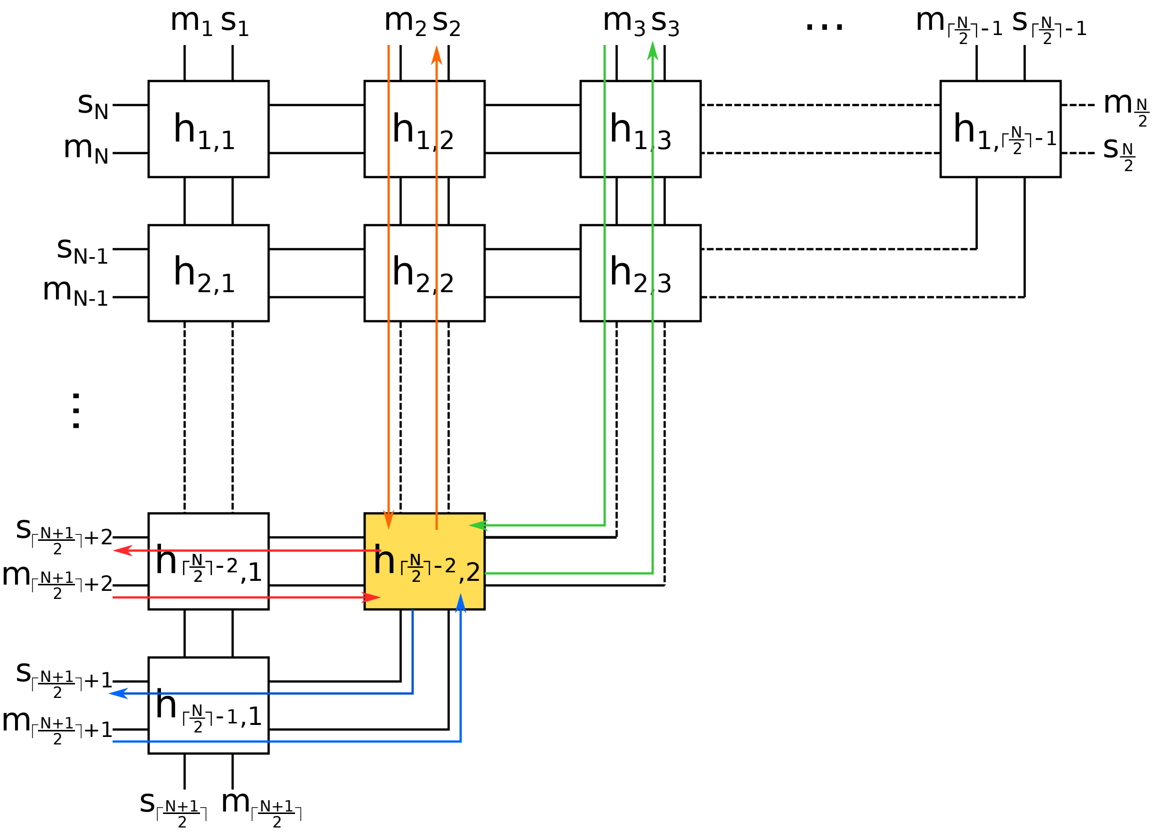
m_1 s_1
m_2 s_2
m_3 s_3
...
$m_{\lceil \frac{N}{2} \rceil -1}$ $s_{\lceil \frac{N}{2} \rceil -1}$
s_N
m_N
$h_{1,1}$
$h_{1,2}$
$h_{1,3}$
$h_{1,\lceil \frac{N}{2} \rceil -1}$
$m_{\frac{N}{2}}$
$s_{\frac{N}{2}}$
s_{N-1}
m_{N-1}
$h_{2,1}$
$h_{2,2}$
$h_{2,3}$
⋮
$s_{\lceil \frac{N+1}{2} \rceil +2}$
$m_{\lceil \frac{N+1}{2} \rceil +2}$
$h_{\lceil \frac{N}{2} \rceil -2,1}$
$h_{\lceil \frac{N}{2} \rceil -2,2}$
$s_{\lceil \frac{N+1}{2} \rceil +1}$
$m_{\lceil \frac{N+1}{2} \rceil +1}$
$h_{\lceil \frac{N}{2} \rceil -1,1}$
$s_{\lceil \frac{N+1}{2} \rceil}$ $m_{\lceil \frac{N+1}{2} \rceil}$

Figure 5.5.: The $N \times N - 1$ *Light* structure with an *default Hash* ($h_{\lceil \frac{N}{2} \rceil -2,2}$)

5.3.1. Type-I Signal Paths

The signals on Type-I paths are coupled with any MRRs, and they pass $\lceil \frac{N}{2} \rceil - 1$ Hashes in total on their tours. Thus, the output signal power is calculated as

$$P_{signal,light,typ1} = L_{signal,typ1}^{\lceil \frac{N}{2} \rceil - 1} P_I = [{L_c}^2 {L_{p1}}^2]^{\lceil \frac{N}{2} \rceil - 1} P_I \tag{5.22}$$

and the insertion loss is $l_{light,typ1} = (\lceil \frac{N}{2} \rceil - 1) \times (2l_t + 2l_c) = 0.09(\lceil \frac{N}{2} \rceil - 1)dB$.

As introduced in Section 5.2.1, the noise signals generated by Type-I signal paths are consisted of two parts, which are denoted by the noise power coefficients $L_{noise,typ1,1}$ and $L_{noise,typ1,2}$. In this example, for the Type-I path from m_2 to $s_{\lceil \frac{N+1}{2} \rceil}$ shown in Figure 5.5, a part of noise signals goes to $s_{\lceil \frac{N+1}{2} \rceil + 2}$ and another part of noise signal goes to s_3. Thus, the noise power at these two slave ports can be expressed as

$$P_{noise\_s_{\lceil \frac{N+1}{2} \rceil + 2},light,typ1} = (L_{signal,typ1}^{n_r} L_{noise,typ1,1}) L_{signal,typ1}^{n_t} P_I \tag{5.23}$$

and

$$P_{noise\_s_3,light,typ1} = (L_{signal,typ1}^{n_l} L_{noise,typ1,2}) L_{signal,typ1}^{n_t} P_I \tag{5.24}$$

where the n_t, n_l, and n_r can be found in Table 5.5. In this example, $n_t = \lceil \frac{N}{2} \rceil - 3$, $n_l = 1$ and $n_r = \lceil \frac{N}{2} \rceil - 3$.

Likewise, the noise signal caused by other Type-I signal paths can be calculated in this way. Another example is the noise signal power of the path $m_3 \to s_{\lceil \frac{N+1}{2} \rceil + 2}$, the noise signal power at s_2 is $L_{signal,typ1}^{n_t} L_{noise,typ1,1} L_{signal,typ1}^{n_l}$ and the noise signal power at $s_{\lceil \frac{N+1}{2} \rceil} + 1$ can be calculated as $L_{signal,typ1}^{n_b} L_{noise,typ1,2} L_{signal,typ1}^{n_l}$, where $n_l = 1$, $n_b = 1$, and $n_t = \lceil \frac{N}{2} \rceil - 3$.

5.3.2. Type-II Signal Paths

Signals on the Type-II paths can be regarded as the *off-resonance* signals to the Hashes that are not coupled with them. In this case, the output power of signals on Type-II paths is

$$P_{signal,light,typ2} = L_{signal,typ1}^{n} L_{signal,typ2} P_I = [{L_c}^2 {L_{p1}}^2]^n L_d P_I \tag{5.25}$$

where n is the number of Hashes on the signal's tour except for the one *Hash* that changes its direction. For example, the signal from $m_{\lceil\frac{N+1}{2}\rceil+2}$ to $s_{\lceil\frac{N+1}{2}\rceil+1}$ passes $n_l + n_b = 2$ Hashes, namely $h_{\lceil\frac{N}{2}\rceil-2,1}$ and $h_{\lceil\frac{N}{2}\rceil-1,1}$. In this example, n is the $n_l + n_b$. Taking the path $m_3 \rightarrow s_2$ as another example, the $n = n_r + n_t = 2\lceil\frac{N}{2}\rceil - 6$. Therefore, n can be expressed by the parameters on the Table 5.5.

The insertion loss of this type of signal path can be calculated in the same way, which is $l_{light,typ2} = n \times (2l_t + 2l_c) + l_{hash,typ2} = (0.09n + 0.5)dB$.

Each signal on Type-II paths only generates one noise signal and the noise power coefficient is $L_{noise,typ2}$. For example, the noise signal generated by the path $m_{\lceil\frac{N+1}{2}\rceil+2} \rightarrow s_{\lceil\frac{N+1}{2}\rceil+1}$ will reach s_3, and the its power can be calculated as

$$P_{noise,light,typ2} = L_{signal,typ1}^{n_l} L_{noise,typ2} L_{signal,typ1}^{n_r} P_I \tag{5.26}$$

where $n_l = 1$ and $n_r = \lceil\frac{N}{2}\rceil - 2$ in this example.

5.3.3. Type-III Signal Paths

Similar to the signals on Type-II paths, the output signal power on Type-III paths is

$$P_{signal,light,typ3} = L_{signal,typ1}^{n} L_{signal,typ3} P_I = [{L_c}^2 {L_{p1}}^2]^n (L_d L_t^2 L_c^4) P_I \tag{5.27}$$

and the insertion loss can be calculated as $l_{light,typ3} = n \times (2l_t + 2l_c) + l_{hash,typ3} = (0.09n + 0.67)dB$, where n is the number of Hashes on the signal's path except for the one that changes its direction.

The noise signals of Type-III signal paths have 5 components as illustrated in Section 5.2.3. Take the path $m_{\lceil\frac{N+1}{2}\rceil+2} \rightarrow s_2$ as an example, a part of noise signal goes to $s_{\lceil\frac{N+1}{2}\rceil+1}$, while the rest part goes to s_3. The noise power can be calculated as

$$P_{noise,light,typ3} = (L_{signal,typ1}^{n_b} L_{noise,typ3,1} + L_{signal,typ1}^{n_r} L_{noise,typ3,2}) L_{signal,typ1}^{n_l} P_I \tag{5.28}$$

where $n_b = 1$, $n_l = 1$ and $n_r = \lceil\frac{N}{2}\rceil - 3$ in this example.

6. Comparison and Discussion

The signal's quality is strongly related to the insertion loss and crosstalk noise; furthermore, these factors are influenced by the MRR usage. To present the performacne of *Light*, I compare *Light* with λ-router in terms of MRR usage, insertion loss, and SNR.

6.1. MRR Usage

In an $N{\times}N$-1 *Light*, there are $\lceil\frac{N}{2}\rceil(\lceil\frac{N}{2}\rceil-1)/2$ Hashes, namely $2\lceil\frac{N}{2}\rceil(\lceil\frac{N}{2}\rceil-1)$ MRRs, since each Hash has 4 MRRs. An $N{\times}N$ λ-router consists of $N\lceil\frac{N}{2}\rceil{+}(N-1)\lfloor\frac{N}{2}\rfloor$ CSEs, namely $2(N\lceil\frac{N}{2}\rceil{+}(N-1)\lfloor\frac{N}{2}\rfloor)$ MRRs, since each CSE has 2 MRRs. I compare the number of MRRs between the $N{\times}N$-1 *Light* with the $N{\times}N$ λ-router with N= 4, 8, 16, 32 and 64.

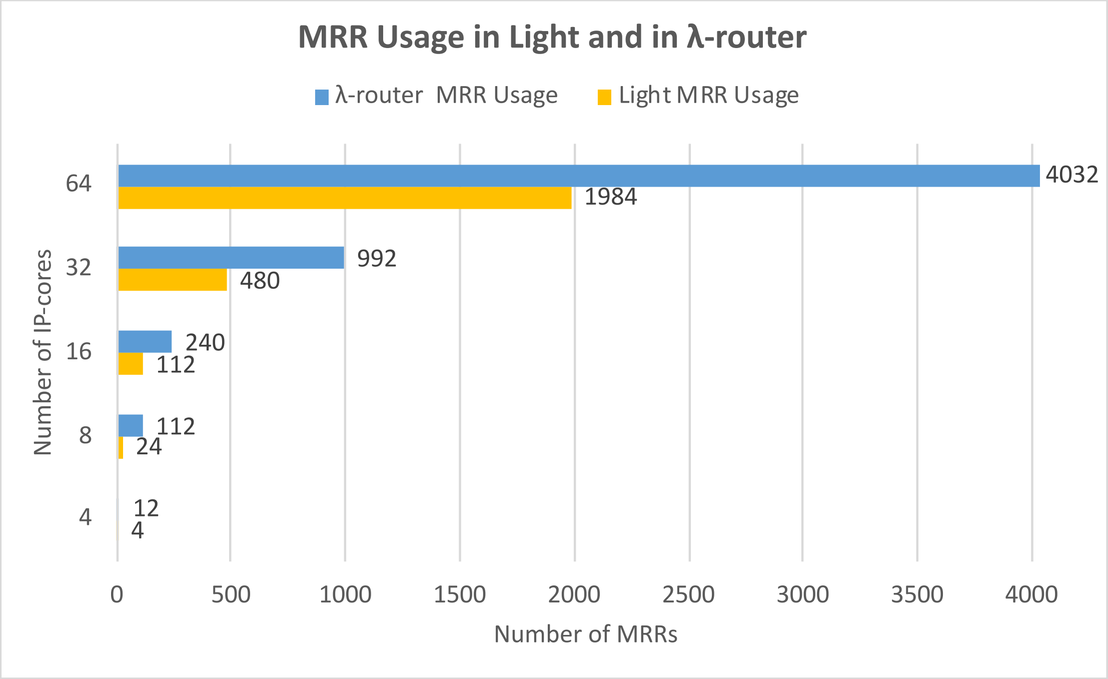

Figure 6.1.: MRR Usage between an $N{\times}N$-1 *Light* and an $N{\times}N$ λ-router when N= 4, 8, 16, 32, 64

Figure 6.1 presents the MRR usage in *Light* and λ-router for different sizes of networks. In general, it is apparent that the number of MRRs in *Light* is considerably less than the number of MRRs in λ-router. Comparing with λ-router, *Light* reduces more than 50% MRRs in each case. For example, in the 64×63 network, the MRRs in *Light* is 51% less than the MRRs in λ-router.

The reduction of MRRs ascribes to the use of PSEs. According to the structures of both topologies, in λ-router, the 2×2 CSE is composed of two identical MRRs, whereas the 2×2 PSE in *Light* has only one MRRs. Taking advantage of PSEs helps in reducing both the MRR usage and the router's area.

The large MRR usage also contributes to the significant insertion loss and crosstalk noise, which degrades the signal's quality. The analysis of insertion loss and SNR of both topologies are given in Section 6.3 and Section 6.4, respectively.

6.2. Physical Implementation

As introduced before, in the typical processor-memory network, the memory controllers are placed on the periphery of the optical layer, and hubs are located along with a rectangle in the middle of this layer. Each IP-core has one master to send data and one slave to receive data, and hence these two ports are close to each other.

Based on such physical constraints, the physical implementations of an 8×8 λ-router and an 8×7 *Light* topology are manually designed, which are shown by Figure 6.2 and Figure 6.3, respectively. To avoid additional crossings in λ-router, the long detours are inevitable. The long detours in the physical layout of the λ-router result in a great deal of propagation loss to seriously damage the system's performance. In (Beux et al. 2014), the way to avoid additional crossings brings about even more insertion loss than the way to allow extra crossings in the physical implementation of λ-router. Therefore, in some physical design tools, extra crossings are allowed to minimized insertion loss.

On the contrary, in the physical layout of the 8×8 *Light*, no extra crossings and detours are generated. It can be observed from Figure 6.3 that the propagation loss in the physical layout of *Light* is appreciably less comparing to the λ-router. Besides that, because of the reduction of MRRs, the 8×7 *Light* occupies less area than the 8×8 λ-router.

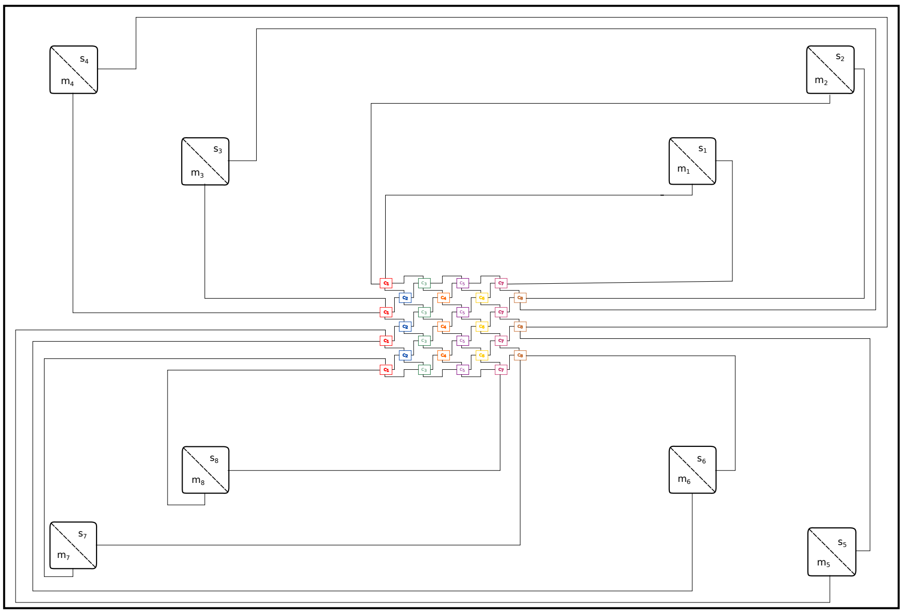

Figure 6.2.: Physical layout of a 8×8 λ-router

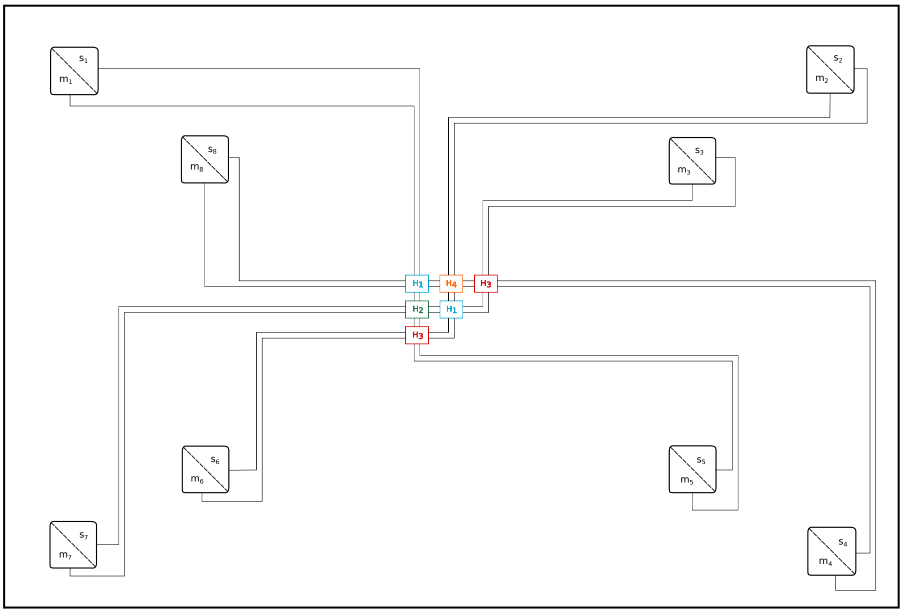

Figure 6.3.: Physical layout of a 8×7 *Light* topology

6.3. Insertion Loss

To evaluate the insertion loss of the *Light* and the λ-router, a tool in C++ was developed to calculate the *drop loss*, *through loss* and *crossing loss* with parameter value given by Table 5.1 and Table 5.2; and I removed the self-communications in the λ-router for a fair comparison.

The analysis of insertion loss consists of two parts: the average insertion loss and the worst-case insertion loss. Based on the logic schemes, *Light* outperforms λ-router in the average insertion loss, but in some cases, *Light* has more worst-case insertion loss than λ-router.

Figure 6.4 shows that in the small size of networks, such as the 4×3 network or the 8×7 network, the average insertion loss values of both topologies are almost identical. However, in the large size of networks, *Light* has appreciably less average loss values than λ-router. For example, in the network with 64 IP-cores, the average insertion loss of *Light* is 8.8% less than the average loss of λ-router.

Moreover, the difference of the average insertion loss values between *Light* and λ-router is growing as the size of networks is increasing. For the network with more IP-cores, the *Light* has less insertion loss than λ-router on average.

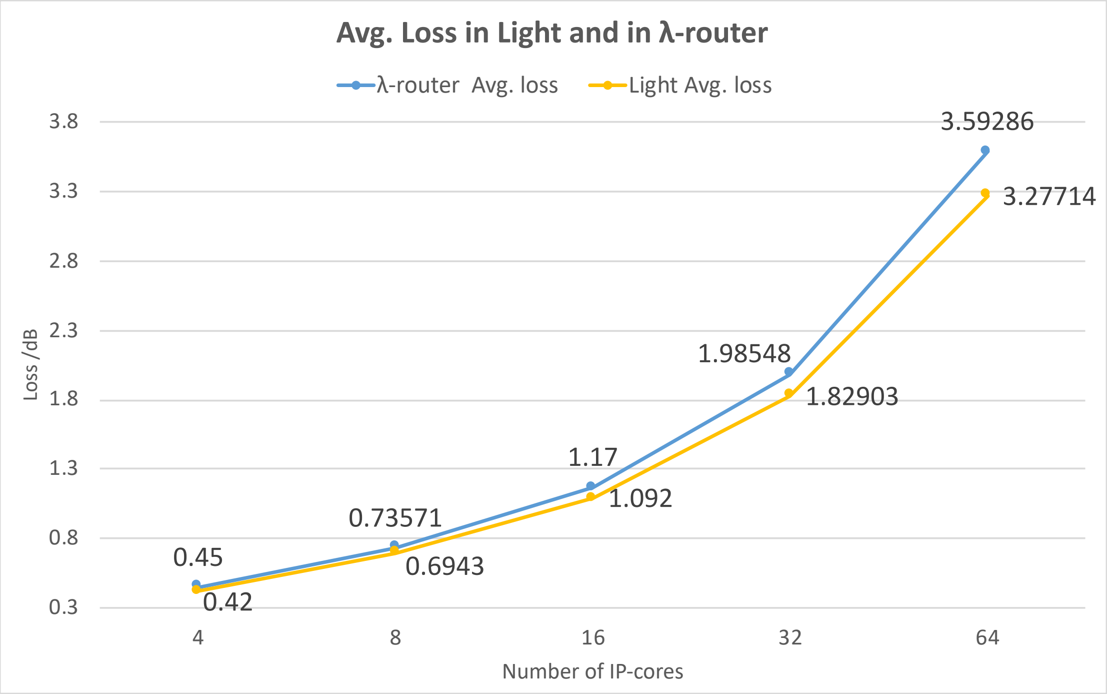

Figure 6.4.: Average loss in *Light* and λ-router

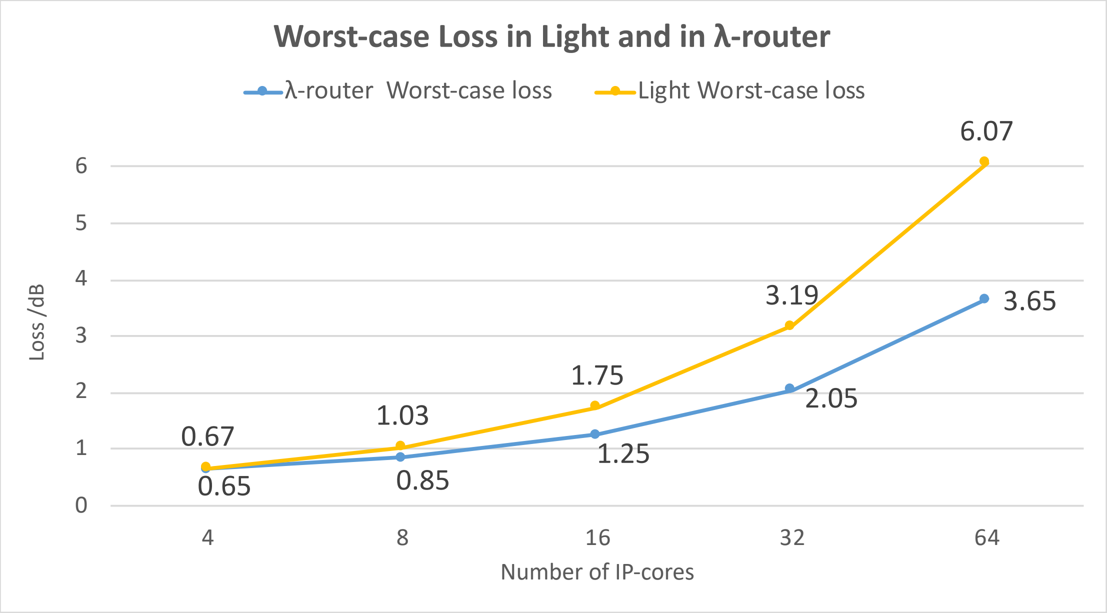

Figure 6.5.: Worst-case loss in *Light* and λ-router

Figure 6.5 presents the worst-case insertion loss values in *Light* and in λ-router for different sizes of networks. *Light* and λ-router have similar worst-case insertion loss values in the small size of networks, but for large networks with more than 8 IP-cores, the worst-case insertion loss values of *Light* are relatively greater than the worst-case insertion loss values of λ-router.

I analyzed the insertion loss of each signal path in *Light* and λ-router for a network with 32 IP-cores. Figure 6.6 lists the number of paths according to the insertion loss in the 32×31 *Light* and λ-router. It can be found that the insertion loss values of the paths in *Light* are widely distributed from 0.5dB to 3.2dB; by contrast the insertion loss values of the paths in λ-router are intensively distributed around the worst-case insertion loss value (2.05dB).

In λ-router, among all 992 paths, 960 (95%) paths have larger loss values than the average value of λ-router (1.985dB) and 240 (24%) paths suffer the worst-case insertion loss value (2.05dB). Although λ-router has less worst-case insertion loss values than *Light*, the number of paths, that suffers large loss values, is exceptionally large.

On the contrary, Only 3 (0.3%) paths in *Light* have the worst-case insertion loss value (3.2dB) and 379 (38%) paths have greater loss values than the average value of λ-router (1.985dB). Comparing to the 960 paths in λ-router, 581 (59%) paths of *Light* have less insertion loss values. Although 233 (20%) paths in *Light* have greater insertion loss values than the worst-case loss

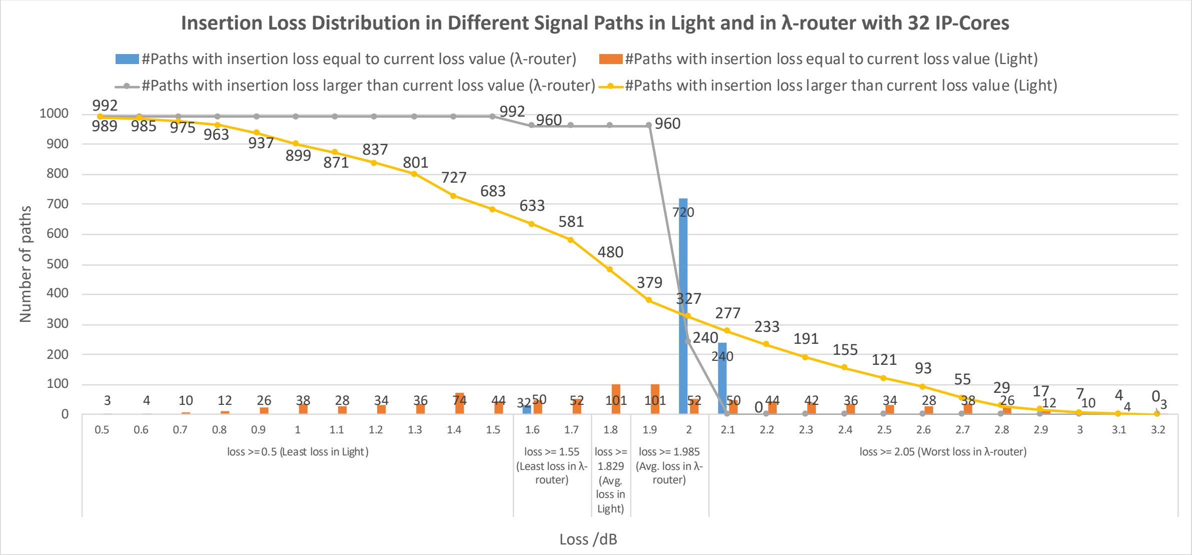

Figure 6.6.: Path distribution according to loss of *Light* and λ-router in 32×32 full-bandwidth communication

of λ-router (2.05dB), 613 (61%) paths in *Light* have less insertion loss than 960 (95%) paths in λ-router.

Since the calculation and analysis are based on the logic schemes, the *propagation loss* and *bending loss* are not considered. If the calculation and analysis are carried out according to their physical layouts, *Light* would perform much better than λ-router, since the propagation loss would be larger in λ-router than in *Light* and additional crossings in λ-router would also increase the crossing loss.

6.4. Signal-to-Noise Ratio (SNR)

It is not sufficient to give a complete view of performance of *Light* only with the insertion loss. The SNR is also an essential factor to reflect the signal's quality. For this propose, first I calculated the SNR of both topologies based on their logic schemes with the definition of SNR, $SNR^{\lambda_n} = 10log(P_S^{\lambda_n}/P_N^{\lambda_n})$. Afterwards, I compared the $N \times N$-1 *Light* with the λ-router in terms of average SNR values and worst-case SNR values when N= 4, 8, 16, 32 and 64.

In general, *Light* performs better in average SNR than λ-router for all cases as shown in Figure 6.7. It is shown that the average SNR values of *Light* are much greater than the average SNR

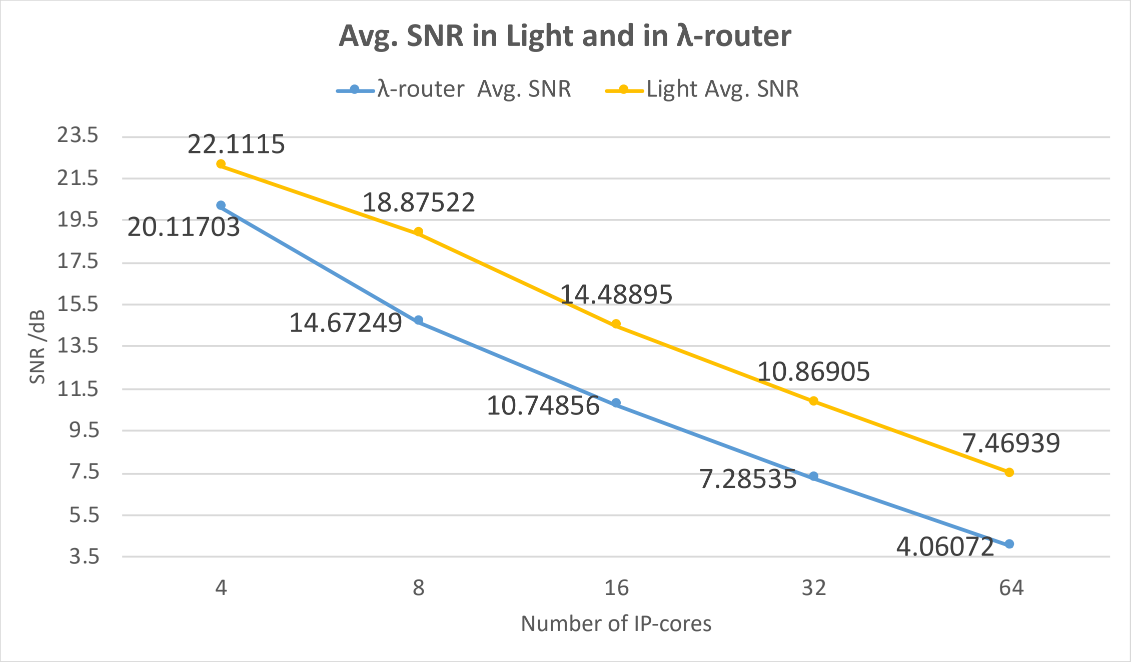

Figure 6.7.: Average SNR in *Light* and λ-router

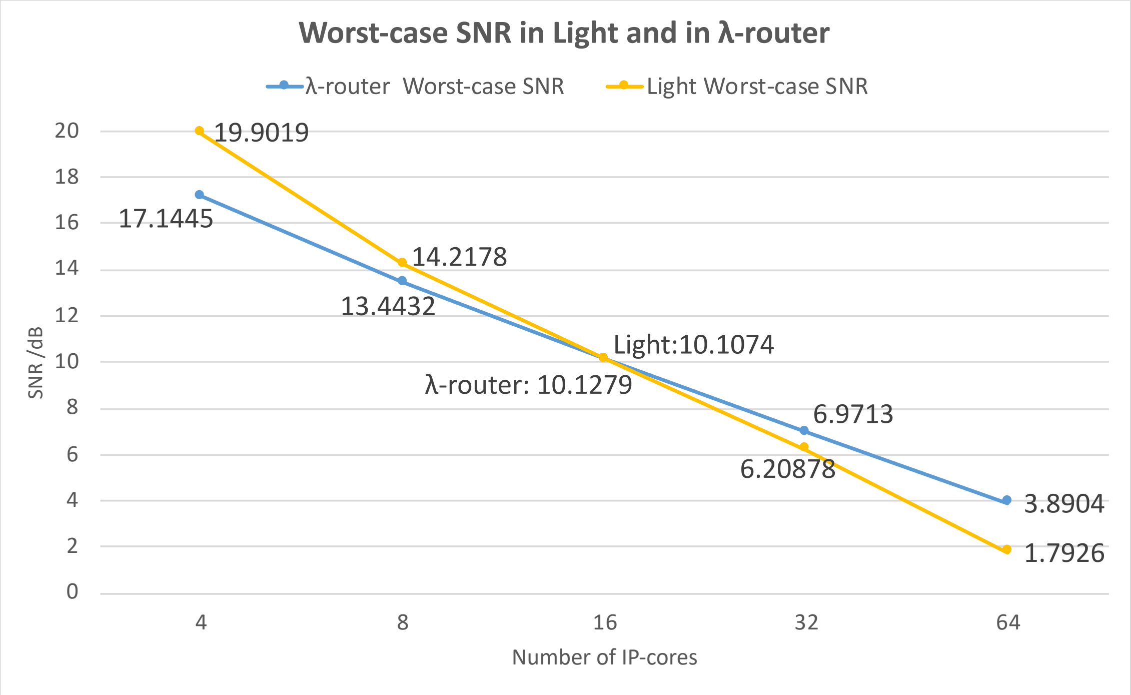

Figure 6.8.: Worst case SNR in *Light* and λ-router

values of λ-router, particularly in the large size of networks. For example, in the 64×63 *Light*, the average SNR value (7.46939dB) is 45.6% greater than the average SNR value of the 64×63 λ-router (4.06072dB). The difference of average SNR values between *Light* and λ-router is growing up with the increasing size of networks, and λ-router behaves worse in average SNR than *Light* for all cases.

The worst-case SNR values in *Light* and in λ-router are shown by Figure 6.8. For the networks with IP-cores fewer than 16, *Light* has larger worst-case SNR values than λ-router. For example, in the network with 4 IP-cores, *Light* increases the worst-case SNR value by 16% compared to the 4×3 λ-router. On the other hand, for the large networks with IP-cores more than 16, λ-router outperforms *Light* in worst-case SNR.

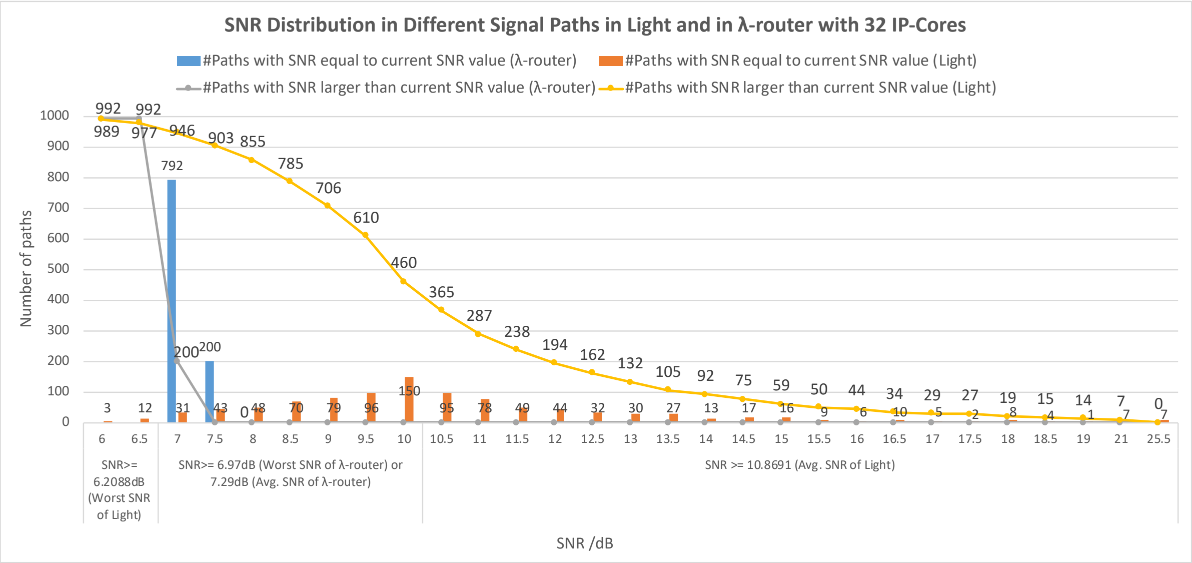

Figure 6.9.: SNR distribution in different signal path in *Light* and in λ-router with 32 IP-cores

Similar to the analysis of insertion loss distribution, I analyzed the SNR of each path in *Light* and λ-router. Figure 6.9 displays the SNR distribution in *Light* and λ-router for the network with 32 IP-cores. As detailed in Figure 6.9, the SNR values of 79.8% paths in λ-router are distributed around the worst-case SNR (6.9713dB), and 903 (91%) paths in *Light* have higher SNR values than the average SNR value of λ-router (7.29dB). Comparing to the 79.8% paths in λ-router with the worst-case SNR value (6.9713dB), only 15 (2%) paths in *Light* have worser SNR values. Therefore, most paths in *Light* outperform the paths in λ-router and only a few of paths in *Light* have the worser SNR than λ-router's paths.

6.5. Discussion

As explained in Section 4.1, it is found that *Light* is a scalable topology. Although λ-router is also widely recognized by its scalability, the exceptionally high MRR usage contributes to a great deal of insertion loss and crosstalk noise, which degrades the system's performance and signal's quality. Furthermore, the masters and the slaves from the same IP-cores are connected to two distant ends in λ-router. As displayed in Section 6.2, λ-router can not avoid the unexpected crossings or long detours in realistic implementation. Therefore, the insertion loss values calculated from the actual physical layout of the λ-router maybe quite larger than the results calculated from its logic scheme. The great efforts, which have been taken in reducing the insertion loss and enhancing SNR in λ-router, might be offset by the unexpected crossings or long detours.

By contrast, *Light* can improve the signal's quality and the system performance by reducing the MRR usage. It is evident that *Light* outperforms λ-router on average insertion loss and SNR, especially for the large networks with IP-cores more than 16. As mentioned in Section 6.2, the masters and the slaved from the same IP-cores are put close to each other, as a result, *Light* can avoid the extra crossings or long detours in physical implementation. Benefiting from the reduction of the MRRs and removal of all self-communications, *Light* can improve the system performance and match the physical constraints without generating any extra crossings or long detours.

As displayed by Figure 6.5 and Figure 6.8, λ-router performs better than *Light* in the worst-case insertion loss and the worst-case SNR. Fortunately, the number of paths with these values in *Light* is much fewer than in λ-router; and the SNR values of most paths in *Light* are larger than the SNR values of most paths in λ-router. In the 32×31 λ-router, nearly 80% paths have the worst-case SNR value. However, more than 90% paths of *Light* have the SNR values more than the worst-case SNR value of λ-router.

As introduced in Section 5.1, the insertion loss and crosstalk noise are strongly related to the number of MRRs and crossings. By counting the number of MRRs and crossings passed by the signals which have the worst-case insertion loss. The statistics in Figure 6.10 illustrate that the number of crossings on the signal tours is the main reason for larger worst-case insertion loss values in *Light*. However, in reality, λ-router would add more crossings to avoid long detours. Thus, it is hard to tell that the number of crossings passed by the signals with the worst-case

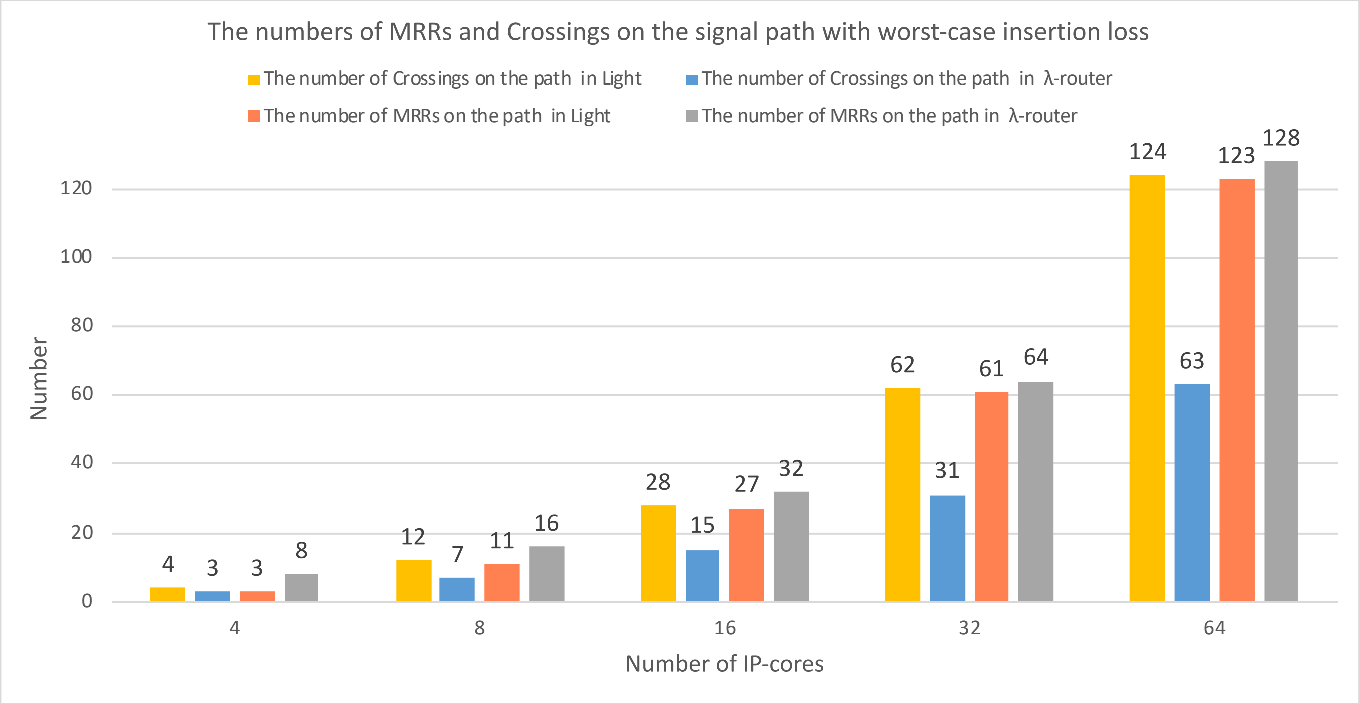

Figure 6.10.: The number of MRRs and crossings passed by signals on the path which has worst-case insertion loss in *Light* and in λ-router

insertion loss is definitely greater in *Light* than in λ-router. In the future, care should be taken to reduce the crossings in the physical layout of *Light* to optimize the worst-case insertion loss and SNR, particularly in large sizes of the networks.

7. Conclusion

ONoCs, especially WRONoCs, are attracting widespread interest for providing higher bandwidth, lower latency and lower power consumption. Many WRONoC topologies have been developed, but few researchers have addressed the problems raised by the huge MRR usage and the mismatch between the logic schemes and the realistic layouts.

In this thesis, I first propose a novel 4×3 structure: *Hash* which uses 4 PSEs. Based on *Hash*, I propose a $N \times N$-1 scalable topology: *Light* and the rule of configuring the resonant wavelength to each MRR.

Light provides an efficient way to reduce MRR usage; furthermore, the *Light* logic scheme matches its physical layout perfectly. Taking advantage of removing self-communication in the logic scheme, *Light* reduces its MRR usage by half comparing to representative WRONoC topologies, such as λ-router, Snake and GWOR. The reduction of MRRs is beneficial to lower insertion loss and improve SNR.

In an attempt to satisfy the physical constraints of the typical processor-memory communication infrastructure, *Light* places the masters and the slaves from the same IP-cores close to each other. In this way, no additional crossings and detours are generated in the physical implementation of *Light* for any sizes of the networks.

According to the comparison between *Light* and λ-router, it is evident that *Light* outperforms λ-router in average insertion loss value and average SNR value. Future work will focus on improving the worst-case insertion loss and SNR, and adapting *Light* to the specific routing applications.

Bibliography

Beuningen, Anja Von, Ramini, Luca, Bertozzi, Davide & Schlichtmann, Ulf (2015): PROTON+: A placement and routing tool for 3d optical networks-on-chip with a single optical layer, J. Emerg. Technol. Comput. Syst. **12**(4): 44:1–44:28.

Beux, Sebastien Le, Li, Hui, Nicolescu, Gabriela, Trajkovic, Jelena & O'Connor, Ian (2014): Optical crossbars on chip, a comparative study based on worst-case losses, Concurrency and Computation: Practice and Experience .

Beux, Sébastien Le, O'Connor, Ian, Nicolescu, Gabriela, Bois, Guy & Paulin, Pierre G. (2013): Reduction methods for adapting optical network on chip topologies to 3d architectures, Microprocessors and Microsystems: Embedded Hardware Design **37**(1): 87–98.

Bianco, A., Cuda, D., Garrich, M., Castillo, G. G., Gaudino, R. & Giaccone, P. (2012): Optical interconnection networks based on microring resonators, IEEE/OSA Journal of Optical Communications and Networking **4**(7): 546–556.

Bogaerts, W., De Heyn, P., Van Vaerenbergh, T., De Vos, K., Kumar Selvaraja, S., Claes, T., Dumon, P., Bienstman, P., Van Thourhout, D. & Baets, R. (2012): Silicon microring resonators, Laser & Photonics Reviews **6**(1): 47–73.

Brière, Matthieu, Girodias, Bruno, Bouchebaba, Youcef, Nicolescu, Gabriela, Mieyeville, Fabien, Gaffiot, Frédéric & O'Connor, Ian (2007): System level assessment of an optical noc in an mpsoc platform, Proc. Design, Autom., and Test Europe Conf., S. 1084–1089.

Khouzani, H. A., Koohi, S. & Hessabi, S. (2012): Fully contention-free optical noc based on wavelenght routing, The 16th CSI International Symposium on Computer Architecture and Digital Systems (CADS 2012), S. 81–86.

Li, Mengchu, Tseng, Tsun-Ming, Bertozzi, Davide, Tala, Mahdi & Schlichtmann, Ulf (2018): Customtopo: A topology generation method for application-specific wavelength-routed optical nocs, Proc. Int. Conf. Comput.-Aided Des.

Lin, B. & Lea, C. (2012): Crosstalk analysis for microring based optical interconnection networks, Journal of Lightwave Technology **30**(15): 2415–2420.

Manolatou, Christina & Haus, Hermann A. (2002): Passive Components for Dense Optical Integration, Springer.

Nikdast, Mahdi, Xu, Jiang, Duong, Luan Huu Kinh, Wu, Xiaowen, Wang, Xuan, Wang, Zhehui, Wang, Zhe, Yang, Peng, Ye, Yaoyao & Hao, Qinfen (2015): Crosstalk noise in wdm-based optical networks-on-chips: A formal study and comparison, IEEE Transactions on Very Large Scale Integration (VLSI) Systems **23**(11): 2552–2565.

O'Connor, Ian, Briere, Matthieu, Drouard, Emmanuel, Kazmierczak, Art, Tissafi-Drissi, Faress, Navarro, David, Mieyeville, Fabien, Dambre, Joni, Stroobandt, Dirk, Fedeli, Jean-Marc, Lisik, Zbigniew & Gaffiot, Frédéric (2005): Towards reconfigurable optical networks on chip, ReCoSoC.

Preston, Kyle, Scherwood-Droz, Nicolas, Levy, Jacob S. & Lipson, Michal (2011): Performance guidelines for wdm interconnects based on silicon microring resonators, CLEO: Science and Innovations .

Ramini, Luca, Bertozzi, Davide & Carloni, Luca P. (2012): Engineering a bandwidth-scalable optical layer for a 3d multi-core processor with awareness of layout constraints, IEEE/ACM International Symposium on Networks-on-Chip (NoCS), S. 185–192.

Ramini, Luca, Grani, Paolo, Bartolini, Sandro & Bertozzi, Davide (2013): Contrasting wavelength-routed optical noc topologies for power-efficient 3d-stacked multicore processors using physical-layer analysis, Proc. Design, Autom., and Test Europe Conf., S. 1589–1594.

Scandurra, Alberto & O'Connor, Ian (2011): Scalable cmos-compatible photonic routing topologies for versatile networks on chip.

Tan, Xianfang, Yang, Mei, Zhang, Lei, Jiang, Yingtao & Yang, Jianyi (2011): On a scalable, non-blocking optical router for photonic networks-on-chip designs, Symp. Photonics and Optoelectronics (SOPO).

Truppel, A., Tseng, T., Bertozzi, D., Alves, J. C. & Schlichtmann, U. (2020): PSION+: Combining logical topology and physical layout optimization for wavelength-routed onocs, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems S. 1–1.

Tseng, T., Truppel, A., Li, M., Nikdast, M. & Schlichtmann, U. (2019): Wavelength-routed optical nocs: Design and EDA - state of the art and future directions: Invited paper, 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), S. 1–6.

Xie, Y., Nikdast, M., Xu, J., Zhang, W., Li, Q., Wu, X., Ye, Y., Wang, X. & Liu, W. (2010): Crosstalk noise and bit error rate analysis for optical network-on-chip, Proc. Design Autom. Conf., S. 657–660.

Zhang, L., Man, Y., Tan, X., Yang, M., Hu, T., Yang, J. & Jiang, Y. (2014): On reducing insertion loss in wavelength-routed optical network-on-chip architecture, IEEE/OSA Journal of Optical Communications and Networking **6**(10): 879–889.