TUM School of Computation, Information and Technology
Technische Universität München

Design Automation for Wavelength-Routed Optical Networks-on-Chip

Alexandre Carvalho Truppel

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz:

Prof. Dr.-Ing. Wolfgang Kellerer

Prüfende der Dissertation:

1. Prof. Dr.-Ing. Ulf Schlichtmann
2. Assoc. Prof. José Carlos Alves, Ph.D.
3. Prof. Yao-Wen Chang, Ph.D.

Die Dissertation wurde am 14.09.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 23.01.2025 angenommen.

Abstract

Optical Networks-on-Chip (ONoCs) are a potential solution for the rising requirements of Networks on Chip (NoCs). Compared to traditional Electrical NoCs (ENoCs), ONoCs should provide higher bandwidth and lower latency while consuming less power. Wavelength-Routed ONoCs (WRONoCs) are a sub-type of ONoCs whose design and mode of operation eliminate contention in the network and thus guarantee minimal transmission latency.

The WRONoC field is still in its infancy. Early WRONoC designs were created manually and over time tools that perform some Electronic Design Automation (EDA) tasks have been developed, but the state of the art has yet to take the next significant leap: the development of a tool that performs complete, no-compromise and turn-key WRONoC design and optimisation.

This work starts by providing a detailed description of the WRONoC design problem. It then proposes and justifies a plan to incrementally develop such a complete tool and performs the first three steps in that plan: combine logical topology and physical layout for insertion loss optimisation, optimise physical layout for laser power and calculate accurate infinite-order crosstalk. Each step is evaluated against the state of the art and found to outperform existing methods.

Acknowledgements

Firstly, I would like to thank my supervisor, Dr. Tsun-Ming Tseng, for being unequivocally the best supervisor one can ask for. This is the same sentence I wrote back when I had just finished my master thesis, and after four more years of work under Dr. Tseng I can confidently say it has never been more accurate. During these past years, he always encouraged me to be independent and develop my own ideas, and yet was always available to give his insightful (and crucial) feedback.

Secondly, I must thank Prof. Ulf Schlichtmann for welcoming me to his Electronic Design Automation research group, for his availability to discuss any necessary matter over the last years, and for our smooth collaboration in teaching the Numerical Methods in Electrical Engineering course.

My most sincere thanks to my family, especially to my mother, Margarida Silva, for always being here for me, no matter what. Her unwavering care, kindness, loyalty, assistance and encouragement were crucial. Absolutely none of this would have been possible without her.

Finally, I would like to thank my very dear friends Luís Abreu and Francisca Braga da Cruz, who put up with me in the same house for many years. And the Rust language, who saved me from the absolutely prehistoric, incoherent, hazardous and just straight up unpleasant excuse for a programming language that is C++.

Alexandre Carvalho Truppel

"Vincit qui se vincit"
(He conquers who conquers himself)

Dedicated to my mother,
who has helped me more
than what mere words
could ever hope to explain

Contents

List of Figures

List of Tables

Acronyms

BER	Bit Error Rate.
BiCGSTAB	Biconjugate Gradient Stabilised.
CG	Conjugated Gradient.
CGT	Centralised Grid Template.
CM	Communication Matrix.
DGT	Distributed Grid Template.
DMA	Demodulator Array.
EDA	Electronic Design Automation.
ENoC	Electrical Network-on-Chip.
GMRES	Generalised Minimal Residual.
GRU	Generic Routing Unit.
GT	Grid Template.
GWOR	Generic Wavelength-routed Optical Router.
IC	Integrated Circuit.
LP	Linear Programming.
MA	Modulator Array.
MILP	Mixed-Integer Linear Programming.
MRR	Micro-Ring Resonator.
NoC	Network on Chip.
ONoC	Optical Network-on-Chip.
OOK	On-Off Keying.
P&R	Place & Route.
PDN	Power Distribution Network.

PLT	Physical Layout Template.
PSE	Photonic Switching Element.
SNR	Signal-to-Noise Ratio.
SoC	System on Chip.
TSV	Through Silicon Via.
WDM	Wavelength-Division Multiplexing.
WL	Wavelength.
WRONoC	Wavelength-Routed Optical Network-on-Chip.

1 Introduction

The remarkable progress in Integrated Circuit (IC) fabrication technologies over the past few decades has resulted in a substantial decrease in IC size while simultaneously increasing transistor count and expanding IC capability. This surge in development has been driven by the need for greater processing power and has paved the way for the production of highly multifaceted ICs.

These versatile ICs integrate various components to perform a series of tasks. Components include, but are not limited to, one or more CPU and GPU cores, memory banks and controllers, neural accelerators, cryptography engines, video encoders and decoders, signal processing blocks, analog/radio-frequency blocks and networking interfaces. Such complex systems are referred to as Systems on Chip (SoCs).

Naturally, SoCs require a framework to bring all of their components together so that they may work in unison. This framework necessarily includes some form of inter-component communication. While in smaller cases, simple point-to-point or shared communication buses may be enough, more complex ICs benefit from full communication networks. A Network on Chip (NoC) serves as a high-performance communication network that links all components of an SoC together [3].

As IC processing power increases, so do the bandwidth, throughput and latency demands placed on the NoC, which must be met while minimising power consumption. Unfortunately, conventional Electrical NoCs (ENoCs), which rely on electrical connections, have started to fall short of meeting these rising demands, especially under reasonable power budgets [3].

More recently, silicon photonics has been proposed as a way to address these limitations. By using optical waveguides and optical signals to replace longer metal connections (such as those in an IC-wide ENoC), this technology can be leveraged to create entire Optical NoCs (ONoCs). These offer several advantages over traditional ENoCs, including significantly higher bandwidth, extremely low latency and lower dynamic power consumption, which can potentially result in lower overall power consumption [4, 5]. These advantages make ONoCs a promising approach to meet the rising demands of modern ICs.

ONoCs can be active or passive depending on their mode of operation. Passive ONoCs are also called Wavelength-Routed ONoCs (WRONoCs) and are designed in such a way that all contention in the network is eliminated and thus minimal latency is guaranteed. This work focuses on WRONoCs and its objective is to advance the

state of the art in WRONoC design automation by developing tools that can create optimised WRONoC designs.

Following this concise overview and rationale for ONoCs, chapter 2 starts with an introduction to ONoC working principles, then explains all relevant ONoC elements and ends with the distinction between active and passive ONoCs. Chapter 3 details the WRONoC design problem along with the current state of the art approaches and their shortcomings. It ends with a multi-step plan on how to incrementally improve the state of the art. Chapters 4 to 6 perform the first three steps of this plan. Finally, chapter 7 concludes this work with a well-rounded assessment of its success and suggests potential avenues for future work.

2 Background on ONoCs[1]

This chapter introduces the working principles of ONoCs. First, their physical structure is outlined. Then, the primary ONoC elements relevant to this work are enumerated and described. Finally, the two major classes of ONoCs (active and passive) are explained, with a focus on the similarities and differences of their operating mechanisms and resulting network architectures.

As stated in chapter 1, a NoC is often necessary to facilitate communication between the various components of a SoC. All components that participate in a NoC as senders, receivers or both are referred to as the nodes of the NoC. It is the goal of an ONoC to transmit data between pairs of nodes. The data transmitted between a sending node and a receiving node is called a signal. For example, a node may send three signals to three other receiving nodes and receive five signals from five other sending nodes. In ONoCs, signals use wavelengths of light to carry data.

2.1 Physical architecture

The typical IC is fabricated layer by layer. Transistors are first etched into a silicon substrate, then multiple layers of electrical interconnections are deposited above (naturally, the exact details depend on the technology employed in the fabrication process). These layers constitute the electrical portion of the IC stack. The inclusion of an ONoC in the IC to provide optical communication capabilities between SoC components requires the addition of the following optical layers:

Routing layer. This layer contains the necessary elements for the optical transmission of data between nodes and is thus present in all ONoCs. These elements are light carrying waveguides and optical routing devices (see sections 2.2.2 and 2.2.4). In some cases, the distribution of optical power to each sending node is also done on this layer (see section 2.2.3).

Control layer. The role of this layer is to configure the routing layer, more precisely to enable and disable its communication paths as requested by the sending nodes, so that optical signals arrive at the correct receiving nodes and there is no signal interference during transmission. This layer itself requires some form of communication between nodes (although at a much lower bandwidth than the routing

[1] Segments of this chapter were previously published in [6–8].

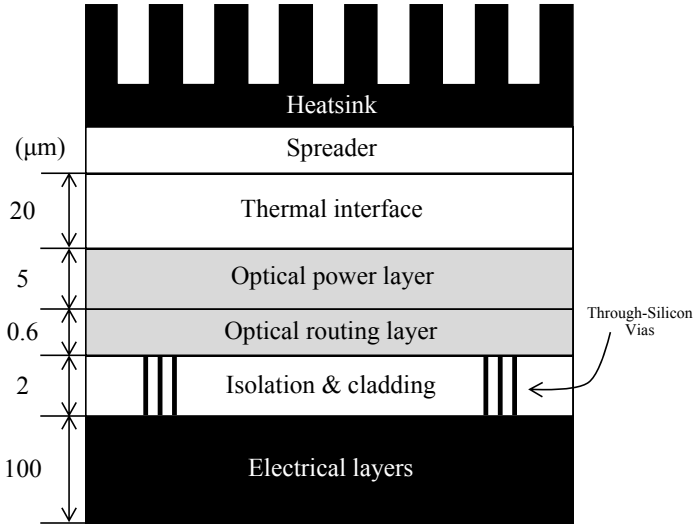

Figure 2.1: An example 3D stack of electrical and optical layers in an IC with an ONoC (adapted from [16, 17]).

layer). This communication can be achieved electrically or optically, and the layer's design and location on the IC stack are determined accordingly. ONoCs that require a control layer are called *active* (see section 2.3) and ONoCs that don't require a control layer are called *passive* (see section 2.4).

Power layer. Optical power for an ONoC is provided by one or more laser sources. These can be placed off-chip or on-chip (see section 2.2.1) [9, 10]. In the on-chip case, the laser sources are placed on the power layer, above the routing layer, and optical power is routed directly from the sources to each sending node below [11].

An example 3D layer stack is shown in fig. 2.1. An isolation and cladding layer separates the electrical and optical portions of the stack, and the connections between them are done using Through Silicon Vias (TSVs) [12–15].

2.2 Essential network elements

All ONoCs contain the following essential optical elements (in order of the propagation direction of light):

Laser sources to convert electrical power into optical power of adequate wavelengths (see section 2.2.1).

Power Distribution Network (PDN) to distribute optical power from a laser source to each sending node (necessary only for off-chip laser sources, see section 2.2.3).

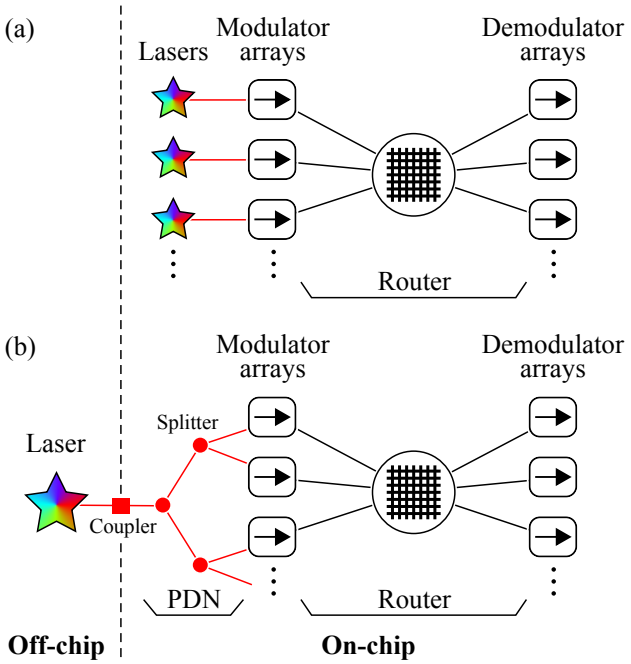

Figure 2.2: Conceptual diagram of the complete ONoC optical architecture for (a) on-chip laser sources and (b) off-chip laser sources. All elements except for the laser sources are located on the optical routing layer (adapted from [7]).

Modulators to modulate the optical power received from the laser source according to the electrical signals of each sending node (see section 2.2.5).

Optical router to propagate optical signals from sending nodes to receiving nodes with the use of waveguides and optical routing elements (see section 2.2.4).

Demodulators to demodulate the optical signals received from the router back into electrical signals at each receiving node (see section 2.2.6).

A conceptual diagram of the connections between these elements is shown in fig. 2.2. With the exception of the laser sources, all of these elements are placed on the optical routing layer. Additionally, network interfaces on the boundary between the electric and optical domains are also required. An explanation of each element is given next.

2.2.1 Laser sources

A laser source is an element that outputs a certain amount of optical power for a certain set of wavelengths, or over a certain wavelength range, for which it is configured [10]. There are two fundamental types of laser sources based on their location:

On-chip. The sources are placed on the optical power layer close to the modulators they supply. Each node has its set of sources. On-chip packaging is more complex

but the connection between each source and its corresponding modulators places no burden on the design of the optical router [10, 16, 18].

Off-chip. The sources exist outside of the IC and their optical power is transferred into the optical routing layer through an optical coupler placed on its edge [10, 18]. These create extra optical power loss. From there, power must be routed to each sending node. This requires a PDN (see section 2.2.3), which is also placed on the routing layer and thus adds additional complexity to the design of the router [9].

There are also two other categories of laser sources relevant to this work. These are orthogonal to on-chip/off-chip and are conceptually described as follows:

Type X. The source emits multiple wavelengths and the power output amount for each wavelength can be controlled independently. Therefore, the source can accommodate different power requirements for each wavelength with minimal power waste. Single-wavelength distributed-feedback laser arrays belong in this category [10].

Type Y. The source emits multiple wavelengths but the power output amounts for all wavelengths are controlled together by a single parameter. Therefore, the source must be configured according to the wavelength with the highest power requirement, which may lead to wasted power on other wavelengths. Comb lasers fall into this category [10, 19].

The distinction between these two categories is crucial for calculating and optimising optical power (see section 3.1.1).

2.2.2 Waveguides and waveguide terminators

Waveguides are the optical equivalent of electrical wires. Their core material is silicon [3, 20–22] and they propagate light in a limited frequency range. Within this range, all frequencies can be transmitted simultaneously in both directions through the same waveguide. This effect is called Wavelength-Division Multiplexing (WDM) and unlocks high bandwidth capabilities for ONoCs. To avoid interference when using WDM, all optical signals through a waveguide must use unique wavelengths. In other words, different signals can share wavelengths or waveguides, but not both. This is called the optical exclusion principle.[2]

However, unlike electrical wires, waveguides can cross on the same layer. These crossings are sources of optical loss and noise but do not alter optical paths: light traveling on a waveguide that crosses other waveguides will continue traveling in the same direction on the original waveguide.

Waveguide terminators are placed at the ends of waveguides and their purpose is to absorb as much optical power as possible to avoid back reflection.

[2] Not to be confused with the Pauli exclusion principle [23], which makes me happy because ONoCs are much easier than quantum mechanics.

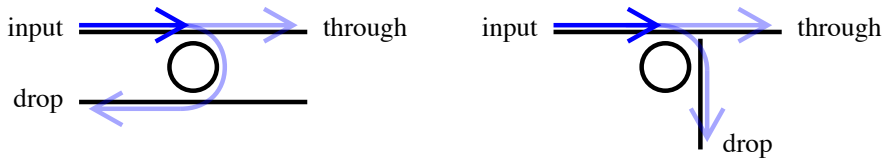

Figure 2.3: Routing mechanism of an MRR. A fraction of the energy that arrives at the input port continues on the same waveguide and is transferred to the through port; another fraction is moved to a second waveguide and transferred to the drop port. This mechanism works with both parallel and perpendicular waveguides as long as both waveguides are close enough to the MRR.

2.2.3 Power Distribution Network and optical power splitters

When using off-chip laser sources, a PDN is required to transfer optical power from the coupler on the edge of the routing layer to each sending node [1]. To achieve a given Bit Error Rate (BER) target in the ONoC, each Modulator Array (MA) must receive a minimum amount of optical power on the correct wavelengths (excess power is unnecessary).[3]

The PDN is placed on the optical routing layer and is structured as a binary tree where the root vertex is the coupler, the interior vertices are optical power splitters, the leaf vertices are MAs and the edges are waveguides [1, 24]. Optical power splitters take optical power on multiple wavelengths as input and transfer fractions of that power to each of the two outputs according to a splitting ratio chosen at design time [1]. A conceptual example of a PDN is shown in fig. 2.2(b) and a more detailed analysis is carried out in section 5.1.

2.2.4 Optical router and Micro-Ring Resonators

An optical router receives optical signals from multiple sending nodes on multiple wavelengths and propagates them to the correct receiving nodes. It is composed of waveguides, waveguide terminators and, most importantly, optical routing elements that achieve wavelength-based routing: Micro-Ring Resonators (MRRs) [25].

MRRs are silicon ring structures that can be tuned to specific wavelengths. When a signal travels on a waveguide and gets close to an MRR also adjacent to a second waveguide, some energy goes *through* the MRR and continues on the same waveguide, and some energy is *dropped* onto the second waveguide [26]. This effect, shown in fig. 2.3, works the same way with both parallel and perpendicular waveguides so long as both waveguides are close enough to the MRR. The fractions of energy transferred to the through and drop ports depend strongly on both the resonance wavelengths of the MRR and the wavelength of the signal. This is the key to wavelength-based routing using MRRs.

[3] MAs are explained in section 2.2.5 and BER is defined in section 3.1.2.

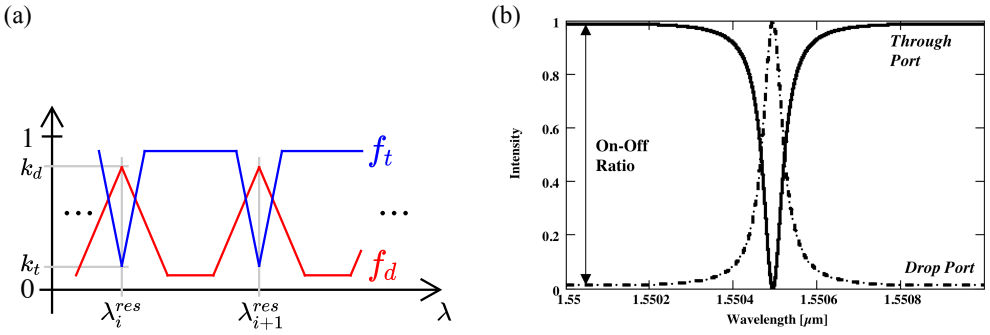

Figure 2.4: (a) Simplified shape of the fractions of the input energy transferred to the through (f_t) and drop (f_d) ports of an MRR depending on the signal wavelength λ. The resonance wavelengths of the MRR are $\lambda_i^{res} \in \Lambda^{res}$ and k_d,k_t are physical parameters. When the wavelength of light λ is far away from all λ_i^{res}, most of the input energy is transferred to the through port. When λ is close to a λ_i^{res}, most of the input energy is transferred to the drop port. When λ is equal to a λ_i^{res} (i.e. $\lambda \in \Lambda^{res}$), the fractions of input energy transferred to the drop and through ports are k_d and k_t respectively (adapted from [8]). (b) Example of the actual shape of the $f_d(\lambda)$ and $f_t(\lambda)$ functions around one λ^{res} (adapted from [30]).

An MRR with radius R has a set of resonance wavelengths Λ^{res} approximately defined by

$$\Lambda^{res} = \left\{ \mu \frac{2\pi R}{i} \;\middle|\; i \in \mathbb{Z}^+ \right\} \tag{2.1}$$

where μ is a physical parameter. Out of all $R, \lambda \in \mathbb{R}$, only radii $R \in R^{all}$ and wavelengths $\lambda \in \Lambda^{all}$ are actually available to use in an ONoC and thus considered in this work. Commonly, $R^{all} \in [5\,\mu\text{m}; 30\,\mu\text{m}]$ and $\Lambda^{all} \in [1500\,\text{nm}; 1600\,\text{nm}]$ [17, 25, 27, 28].

The fractions of the input energy transferred to the through (f_t) and drop (f_d) ports are functions dependent on the wavelength λ of the signal. An example of both is shown in fig. 2.4 and they are described approximately by

$$f_d(\lambda) = \frac{k_d \left(\frac{\lambda^{res}}{2Q}\right)^2}{(\lambda - \lambda^{res})^2 + \left(\frac{\lambda^{res}}{2Q}\right)^2} \tag{2.2}$$

$$f_t(\lambda) = \frac{(\lambda - \lambda^{res})^2 + k_t \left(\frac{\lambda^{res}}{2Q}\right)^2}{(\lambda - \lambda^{res})^2 + \left(\frac{\lambda^{res}}{2Q}\right)^2} \tag{2.3}$$

where λ^{res} is the resonance wavelength in Λ^{res} closest to λ, Q is the MRR's quality factor and k_d, k_t are other physical parameters [29]. Note that $k_d, k_t > 0$ and $k_d + k_t < 1$ because the MRR has always some intrinsic loss and so the combined output energy is less than the input energy.

Additionally, thermal and electrical effects can be leveraged to tune an MRR, i.e. shift its resonance wavelengths Λ^{res} up or down. An increase in the temperature of the MRR shifts Λ^{res} up (red shift of resonance frequencies) [31, 32]. This increase can be controlled by using a thermal heater placed next to the MRR. Similarly, injecting electric charge into a p-n junction placed at the base of the MRR shifts Λ^{res} down (blue shift of resonance frequencies) [4, 17, 20, 31, 32]. Electrical tuning is limited in the maximum wavelength shift it can produce but has low latency and uses little power. Conversely, thermal tuning benefits from a larger maximum wavelength shift but suffers from a larger thermal time constant and power consumption. Both effects are used in practice for three reasons:

- To compensate for process variations that cause permanent unintentional shifts in the resonance wavelengths of each MRR during fabrication.
- To compensate for the increase in temperature of the IC that causes temporary unintentional shifts in the resonance wavelengths of each MRR during operation.
- To modulate optical signals in modulators (see section 2.2.5) and activate/deactivate MRRs for specific wavelengths, causing optical paths through the router to change during operation (see section 2.3). For this case, electrical tuning is preferred due to its low latency.

Finally, assume an MRR has a specific radius and is tuned in such a way that results in a particular Λ^{res}, $f_d(\lambda)$ and $f_t(\lambda)$. Let its *drop-set* Λ^d and *through-set* Λ^t be the set of wavelengths that are considered to be routed to its drop and through ports respectively. As explained above, there is always some energy transferred to both ports regardless of wavelength, so the definition of these sets is, to some extent, flexible. One possibility is

$$\Lambda^d = \left\{ \lambda \in \Lambda^{all} \;\middle|\; f_t(\lambda) < p_t \right\} \tag{2.4}$$

$$\Lambda^t = \left\{ \lambda \in \Lambda^{all} \;\middle|\; f_d(\lambda) < p_d \right\} \tag{2.5}$$

for some p_t and p_d threshold values below which it is considered that the fractions of energy of wavelength λ transferred to the through and drop ports of the MRR respectively are negligible. Some wavelengths may not belong to either set.

2.2.5 Modulators and Modulator Arrays

A modulator is a device that receives optical power and outputs an optical signal modulated according to an electrical data signal. A modulator can be built with a single MRR using On-Off Keying (OOK) as the modulation scheme: the electrical data signal is connected to the MRR and shifts its resonance wavelengths, which changes how much the light is attenuated as it passes next to the MRR [20, 25, 33, 34].

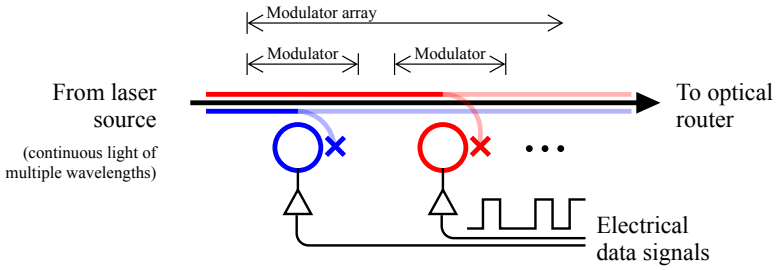

Figure 2.5: Modulator construction using one MRR and OOK. An MA is a sequence of modulators on a single waveguide. Each colour corresponds to a specific MRR radius and matching set of resonance wavelengths (adapted from [17]).

Each modulator modulates in unison all the resonance wavelengths of its MRR according to a single data signal. To modulate more wavelengths according to the same signal, or to modulate other wavelengths according to other signals, more MRRs are added to the same waveguide. A sequence of modulators on one waveguide is called an MA, shown in fig. 2.5.

For obvious reasons, each sending node requires at least one MA with one modulator. A sending node can also have multiple MAs, each with one or more modulators. This allows the node to send signals on multiple waveguides, which can help decrease wavelength usage and/or increase bit parallelism in the ONoC. MAs have to be placed on the optical routing layer above each corresponding SoC component in such a way that the TSVs can connect each modulator to its electrical controlling circuit below.

2.2.6 Demodulators and Demodulator Arrays

Demodulators perform the reverse function of modulators, i.e. convert modulated optical signals back into the electrical domain. Their construction is very similar to that of modulators: an MRR pulls light of its resonance wavelengths away from the waveguide and into a photodetector that recovers the electrical data signals. Photodetectors require a minimum amount of optical power to detect signals with a specific BER (see section 3.1.2) [35].

Multiple demodulators on a single waveguide form a Demodulator Array (DMA), shown in fig. 2.6. Each receiving node must have at least one DMA with one demodulator. Similarly to sending nodes and MAs, receiving nodes can also have multiple DMAs and their location on the optical routing layer is also defined by the location of the corresponding SoC components.

2.2.7 Network interfaces

Optical elements provide the transmission medium and routing capabilities but the optical-electrical interfaces are essential to bridge the gap between the optical and elec-

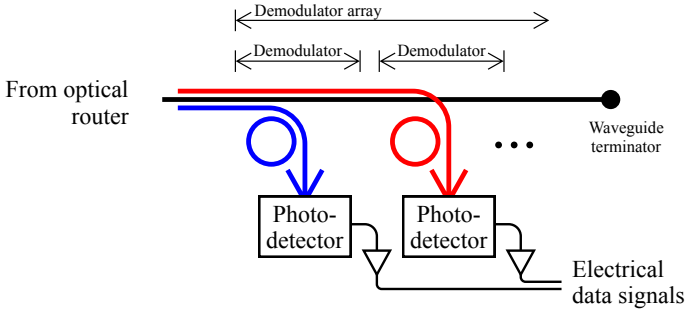

Figure 2.6: Demodulator construction using one MRR and one photodetector. A DMA is a sequence of demodulators on a single waveguide. Each colour corresponds to a specific MRR radius and matching set of resonance wavelengths (adapted from [17]).

trical domains. These are electric circuits in charge of network management tasks such as: buffering data before sending and after receiving, serialising and deserialising data according to the desired bit parallelism, reserving the required optical paths in the router before transmission (only required for active networks), synchronising the transmission and reception of optical signals between nodes (for example, by also sending an optical clock signal along with the data signals) and implementing control flow constraints when the receiving nodes are busy [33].

2.3 Active networks

In active networks, routing of optical signals is done by actively tuning MRRs so that they are on/off-resonance with the chosen signal wavelengths. Before sending a signal, an available path through the network must be chosen and configured. To be available, a path must satisfy the exclusion principle together with all other signals being routed through the network in that instant. To configure a path, its MRRs must be tuned on/off-resonance with the signal's wavelengths. Only then can the signal be sent. The optical control layer is necessary to perform these coordination and configuration duties.

2.3.1 Common architectures

Active network architectures are commonly based around multiple instances of a single router design. Each router instance is assigned to one network node and the connection between router instances is done according to a specific topology. Active router designs normally have four bi-directional connections (North, South, East, West) to four other router instances, and one bi-directional connection to their assigned node (i.e. its modulators and demodulators). Crux [36], Cygnus [37,38] and OTAR [39] are examples of such 5x5 router designs. Larger 6x6 and 7x7 router designs with additional Up and/or Down connections also exist [40, 41]. The interconnections between router instances

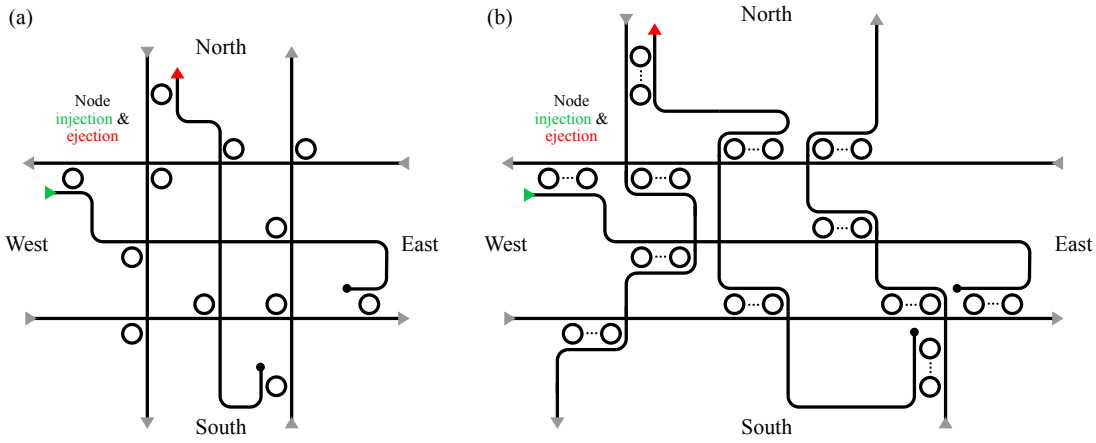

Figure 2.7: Active network router: 5x5 Crux [36]. Black dots are waveguide terminators. (a) Basic Crux router with one MRR on each routing location (adapted from [17]). (b) Adjusting the basic Crux router to support a higher bit parallelism by adding more MRRs of varying radii to each routing location.

are then provided by topologies such as mesh [42], torus [43], fat-tree [39] and others. Figure 2.7(a) shows the basic version of the Crux router and fig. 2.8 shows how multiple 5x5 router instances are connected in a torus topology. Signal routing through the network is done during operation and uses, for example, Manhattan routing for the mesh and torus topologies [44].

2.3.2 Bit parallelism capabilities

Bit parallelism can be increased by assigning more wavelengths to each signal and taking advantage of WDM. To route multiple wavelengths together in active networks, the router designs must be adapted to have multiple MRRs on each routing location, as shown in fig. 2.7(b). Maximum bit parallelism is achieved when each signal uses all possible wavelengths to send data in parallel. Because all locations route all wavelengths together, maximum bit parallelism is supported on any path, and thus between any pair of nodes. However, note that the router designs presented above only have one ejection and one injection waveguide (i.e. only one MA and one DMA) and the exclusion principle must always be satisfied. Therefore, when using these designs, nodes only support maximum bit parallelism transmission/reception with one other receiving/sending node at a time.

Assigning all wavelengths to every signal (i.e. all signals have maximum bit parallelism) is only possible because routing is active. This is a trade-off where the disadvantages are increased power consumption caused by the mandatory control layer and increased latency with fewer performance guarantees due to signal path contention.

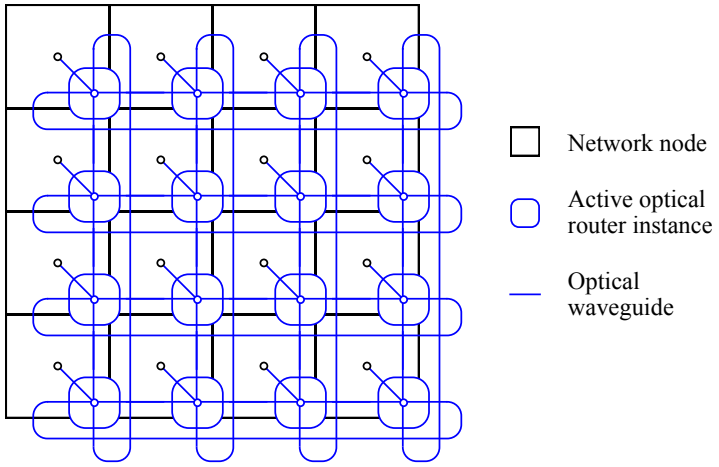

Figure 2.8: An active network of 16 nodes built from 16 instances of a 5x5 router connected in a torus topology (adapted from [17]).

2.4 Passive networks

In passive networks, routing of optical signals is done automatically based on their wavelength. For this reason, passive networks are also called Wavelength-Routed ONoCs. In a WRONoC, the path of each signal depends only on its wavelength(s) and where it starts (its MA), both of which are defined at design time. This allows the designer/program creating the WRONoC to ensure that all signals collectively obey the exclusion principle, which means that all signal paths are always available and signal contention is eliminated. As a result, WRONoCs always guarantee minimal latency. Furthermore, the lack of a control layer also reduces power consumption compared to active ONoCs. However, the disadvantage of wavelength routing is that WDM is more restricted, as explained in section 2.4.2, which limits maximum signal bit parallelism.

When designing a WRONoC, each signal is assigned a non-empty set of wavelengths. The design must guarantee that each MA sends different signals on different wavelengths, because signal origin and signal wavelength define signal path, and different signals must arrive at different DMAs. The design must also guarantee that each DMA receives different signals on different wavelengths, because DMAs may be receiving signals from multiple nodes at the same time, and the receiving node must be able to distinguish which MA sent which signal. However, because each MA and DMA is built from a single waveguide, both of these conditions are met if the exclusion principle is satisfied.

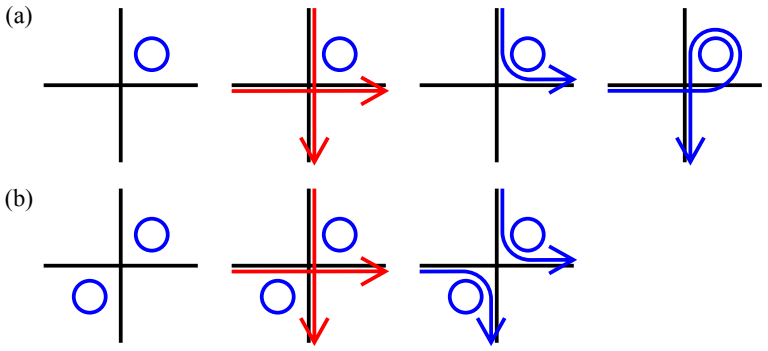

Figure 2.9: PSE structure and wavelength-based routing capabilities. Each colour corresponds to a specific MRR radius and matching set of resonance wavelengths. (a) 1x2 PSE, with one MRR. (a) 2x2 PSE, with two MRRs.

2.4.1 Common architectures

The Photonic Switching Element (PSE) is a very common WRONoC building block [3, 5, 45, 46]. It consists of a crossing between two waveguides together with one or two MRRs which are always placed diagonally opposite each other and configured with the same radius. The structure and wavelength-based routing capabilities of PSEs are shown in fig. 2.9. Multiple PSE instances can be configured and interconnected through waveguides in different ways to produce varying WRONoC logical topologies with distinct characteristics.

A logical topology is a collection of MRRs, waveguides and waveguide terminators connected and configured in a specific way to provide communication capabilities between a specific number of nodes. It includes the radius of each MRR and the set of wavelengths assigned to each signal supported by the topology. It also includes the number of MAs and DMAs per node and their internal structure. However, it contains no physical layout information (for example, the location on the optical routing layer of the MRRs and waveguides). It is only an abstract description of a WRONoC similar to the schematic of an electrical circuit.

Some well-known PSE-based logical topologies connecting N nodes, each with one MA and one DMA, are:

- **Standard crossbar**, built from a N^2 grid of 1x2 PSEs configured with N different MRR radii, shown in fig. 2.10(a) [42].
- **λ-router** and **Snake router**, built from $\frac{N(N-1)}{2}$ instances of the 2x2 PSE configured with N different MRR radii, shown in fig. 2.10(b-c) [1, 3, 5, 25, 46].
- **GWOR**, built from $\frac{N(N-2)}{2}$ instances of the 2x2 PSE configured with $N-2$ different MRR radii, shown in fig. 2.10(d) [45].

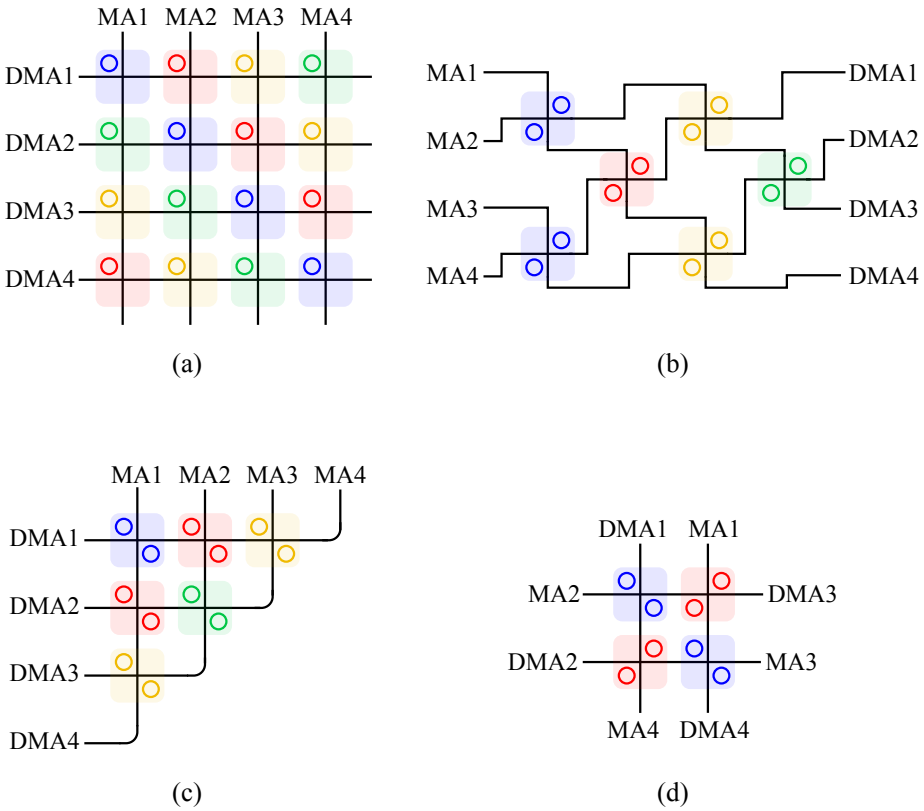

Figure 2.10: PSE-based WRONoC logical topology examples for networks with four nodes. Each MAi and DMAi pair belong to the same node. Each colour corresponds to a specific MRR radius and matching set of resonance wavelengths. (a) Standard crossbar, using 1x2 PSEs. (b) λ-router, using 2x2 PSEs. (c) Snake router, using 2x2 PSEs. (d) Generic Wavelength-routed Optical Router (GWOR), using 2x2 PSEs (adapted from [30]).

The GWOR topology requires less PSEs and MRR radii because it does not provide self-communication capabilities (i.e. routing a signal from the MA of a node to the DMA of the same node), unlike the other three topologies.

This list is definitely not exhaustive. The λ-router and Snake topologies are but two points in a very large design space of WRONoC logical topologies built from 2x2 PSEs that provide full communication between all nodes [5]. Similarly, the 1x2 PSE design space also contains more than just the standard crossbar [47, 48]. By connecting the same PSEs in different ways, many other PSE-based topologies with distinct properties (for example, number of waveguide crossings) can be created. Once the physical layout of these topologies is performed for a specific use-case, their properties diverge even further [9, 46].

Ring topologies, built from one or more loops of waveguides, are also common. The loops are routed together and pass next to each node. MRRs are then used to drop the signals into and out of the rings on the way from the MA to the DMA [1, 14, 24, 49].

Finally, many other custom logical topologies can be designed. These topologies may accommodate more than one MA or DMA per node, they may not support complete communication between all nodes and/or may not use PSEs as building blocks. In general, any arrangement of waveguides and MRRs that follows the rules explained in this chapter is valid. Clearly, a process is required that can effectively choose the best designs in such a vast design space.

2.4.2 Bit parallelism capabilities

Bit parallelism can be increased by assigning more wavelengths to each signal and taking advantage of WDM. Wavelengths must be assigned to signals and the router must be designed in such a way that all of the wavelengths of a given signal follow the same path through the router. To ensure this in WRONoCs, where wavelengths are used for routing, the wavelengths of each signal must belong to the drop-set of one specific MRR radius and also belong to the through-set of all other MRR radii they pass through. The difficulty of this combinatorial problem depends heavily on the fabrication technology: if the Q factor of the MRRs is higher, more wavelengths belong to the through set of each MRR which simplifies the problem, and *vice-versa* [25, 28]. Compared to active networks, passive networks have a lower maximum bit parallelism for each signal.

3 WRONoC design problem[1]

This work focuses exclusively on WRONoCs and the design of their optical routing layer. The previous chapter introduced the optical elements present on this layer and explained their working principles. This chapter now provides a comprehensive account of the characteristics and challenges involved in this layer's design and optimisation. It starts by specifying the three major performance characteristics that determine the quality of a WRONoC design. It then formally defines the WRONoC optimisation problem considered in this work by listing its inputs, outputs and objective function, as well as other relevant details. Next, the state of the art optimisation process for this problem is reviewed. Finally, the chapter ends with a high-level overview of a new optimisation process that seeks to solve the shortcomings of the state of the art.

3.1 Performance factors

3.1.1 Power consumption

NoC power consumption is classified into static power consumption and dynamic power consumption. The former is the amount of energy consumed per unit of time while the NoC is turned on and ready to transmit but not actively transmitting data, whereas the latter is the amount of energy consumed per unit of data (eg. bit, byte, etc) transmitted by the NoC.

In a WRONoC, laser sources, MRRs, modulators, demodulators and network interfaces consume static power, whereas only network interfaces, modulators and demodulators consume dynamic power. In order to reduce average static power consumption, the output power of laser sources can be modulated according to the network requirements in each instant. This is impractical for off-chip laser sources but increasingly possible for on-chip ones [18], with an active area of research and many developing possibilities [50–52]. This work does not assume the use of any specific modulation scheme, and instead considers the worst-case scenario: the entire network is transmitting data and the laser sources must be at peak output. Furthermore, routing in WRONoCs is performed passively, therefore dynamic power is consumed only by the components on the electrical side of the optical-electrical interfaces. Because of this, while static power consumption can be high in WRONoCs, they also consume considerably less dynamic power than equivalent ENoCs [33].

[1] Segments of this chapter were previously published in [6–8].

The design and optimisation of electrical components is outside of the scope of this work. Therefore, dynamic power consumption is not considered. For the same reason, static power consumption of the network interfaces, which are electric circuits, is equally not considered.

As explained in section 2.2.4, each MRR requires tuning circuits to compensate for fabrication and IC temperature variations. In WRONoCs these tuning circuits consist of electrical heaters. The heater element itself along with its control circuit consume an average static power of P_{mrr} in Watt per MRR. Modulators and demodulators also consume an average static power of P_{mod} and P_{demod} in Watt respectively per element.

Laser sources must output enough optical power to compensate for all optical losses in the ONoC and still guarantee that the minimum required amount of optical power arrives at the photodetector in each demodulator. The sum of all optical losses on a given optical path is called insertion loss. Insertion loss can be reported in dB (logarithmic scale) or as a multiplicative attenuation factor. The relationships in both formats between input power, output power and insertion loss are

$$\text{output}_{dB} = \text{input}_{dB} - il_{dB} \tag{3.1}$$

$$\text{output}_{f} = \text{input}_{f} \times il_{f} \tag{3.2}$$

and the corresponding conversion formula is $il_{dB} = -10\log_{10}(il_f)$. Naturally, $il_{dB} > 0$ and $il_f < 1$.

There are multiple sources of insertion loss, shown in fig. 3.1. Waveguides cause *propagation loss* (L_p) and *bend loss* (L_b). Propagation loss in dB is proportional to the length of the waveguide and bend loss in dB depends on the radius of the bend. Waveguide crossings cause a fixed amount of *crossing loss* in dB (L_c) per crossing. Optical couplers and optical power splitters, both required when using off-chip laser sources, also cause fixed amounts of *coupling loss* and *splitting loss* in dB respectively (L_{cpl} and L_s).

MRRs cause two types of insertion loss: *drop loss* (L_d) and *through loss* (L_t). These are directly related to eqs. (2.2) and (2.3):

- If a signal with wavelength λ arrives at an MRR where λ belongs to its drop-set, the signal is considered to be routed to the drop port but loses some energy in the process. Thus, the attenuation factor is $il_f = f_d(\lambda)$ and $L_d = -10\log_{10}(f_d(\lambda))$ in dB.
- If a signal with wavelength λ arrives at an MRR where λ belongs to its through-set, the signal is considered to be routed to the through port but loses some energy in the process. Thus, the attenuation factor is $il_f = f_t(\lambda)$ and $L_t = -10\log_{10}(f_t(\lambda))$ in dB.
- Otherwise, a non-negligible amount of energy is routed to both ports. In this case, there is no clear path for the signal. In ONoCs, signals always take well-defined paths, so this case constitutes a design error.

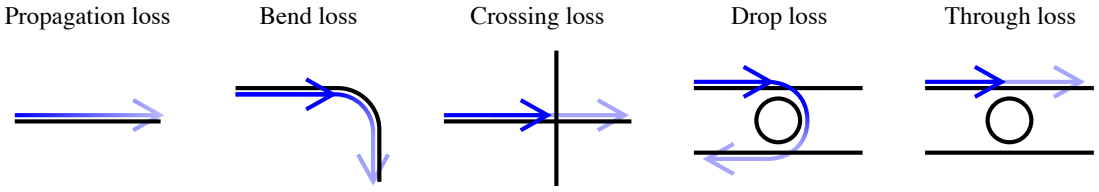

Figure 3.1: Insertion loss sources in ONoCs: propagation loss (L_p), bend loss (L_b), crossing loss (L_c), drop loss (L_d) and through loss (L_t). Not pictured: coupling loss (L_{cpl}) and splitting loss (L_s).

Some of the signal energy which is not routed to the correct port becomes noise, as explained in section 3.1.2. Furthermore, the values of L_d and L_t are very accurate but depend on both the wavelength of the signal and the radius of the MRR. In some cases, approximate (average) values, constant for all signal wavelengths and MRR radii present in the ONoC, are used instead (see section 3.3.2) [2, 42, 53, 54].

If a laser source must supply N wavelengths and each wavelength λ must carry enough power to compensate for a total insertion loss of il_λ in dB, then the total optical power in Watt required for each type (X or Y) of laser source is

$$P^X_{laser:opt} = \frac{1}{K} \sum_{\lambda=1}^{N} 10^{\frac{il_\lambda + S}{10}} \tag{3.3}$$

$$P^Y_{laser:opt} = N \frac{1}{K} 10^{\frac{\max_{\lambda=1}^{N}\{il_\lambda\} + S}{10}} \tag{3.4}$$

where K is a combination of technology and fabrication parameters and S is a parameter that depends on the demodulator sensitivity and the desired BER (see section 3.1.2) [9, 18]. Note that $P^Y_{laser:opt}$ uses the max function because the laser source must be configured according to the wavelength with the highest power requirement. In this case, some wavelengths may receive more power than the minimum dictated by their insertion loss values.

The electrical power in Watt required for a laser source is

$$P_{laser:el} = \frac{P_{laser:opt}}{\eta} \tag{3.5}$$

where η is the wall-plug efficiency of the laser source, which depends on the exact technology used [18].

In total, the electrical power consumed by the optical elements in the WRONoC is the sum of $P_{laser:el}$, P_{mod}, P_{demod} and P_{mrr} from each laser source, modulator, demodulator and MRR respectively. This sum must be minimised.

3.1.2 Crosstalk and Bit Error Rate

Crosstalk broadly refers to the influence of each signal on other signals [55]. Consider a signal traveling between two nodes. A fraction of its energy arrives at the assigned demodulator, a fraction arrives at other demodulators, causing crosstalk, and the rest is lost due to insertion loss. Any signal energy that strays from its intended path becomes noise, because when it inevitably arrives at incorrect demodulators it interferes with the detection of their assigned signals.

There are two types of crosstalk: inter-channel and intra-channel crosstalk [2]. Inter-channel crosstalk happens when the wavelength of the noise is distant enough from the wavelength of the signal. Intra-channel crosstalk happens when the wavelengths of the signal and noise are the same or close enough. The distinction is based on whether the noise can be effectively filtered out from the signal before detection. Naturally, this makes intra-channel crosstalk worse than inter-channel crosstalk.

There are five sources of noise, shown in fig. 3.2 [2]. Energy that arrives at a waveguide terminator is never fully absorbed. *Terminator back reflection* (C_{tbr}) is the fraction of noise energy reflected back onto the waveguide in the reverse direction. A signal that arrives at a crossing mostly continues in the same direction on the same waveguide, but a fraction of the energy escapes through the sides of the crossing as noise and another fraction is reflected back as noise. These are *crossing side spill* (C_{css}) and *crossing back reflection* (C_{cbr}) respectively.

Lastly, MRRs cause two types of noise, and their distinction is inherently tied to the routing goal the MRR is designed to fulfil. If a signal is meant to be routed to the drop port, energy coming out of the through port is noise, and *vice-versa*. More specifically:

- If a signal with wavelength λ arrives at an MRR where λ belongs to its drop-set, the signal is considered to be routed to the drop port but some energy is still transferred to the through port as noise. This is *MRR on-resonance through* (C_{r1t}) noise with $C_{r1t} = f_t(\lambda)$.

- If a signal with wavelength λ arrives at an MRR where λ belongs to its through-set, the signal is considered to be routed to the through port but some energy is still transferred to the drop port as noise. This is *MRR off-resonance drop* (C_{r0d}) noise with $C_{r0d} = f_d(\lambda)$.

- Otherwise, as explained in section 3.1.1, there is no clear signal path and thus also no obvious noise-producing port. This is a design error.

Just like L_d and L_t, the values of C_{r1t} and C_{r0d} also depend on both the wavelength of the signal and the radius of the MRR. Similarly, average versions of these values are also used in some cases (see section 3.3.2).

Note that these five sources of noise apply to not only signal energy but also noise energy. In other words, signals traveling through the router create first-order noise,

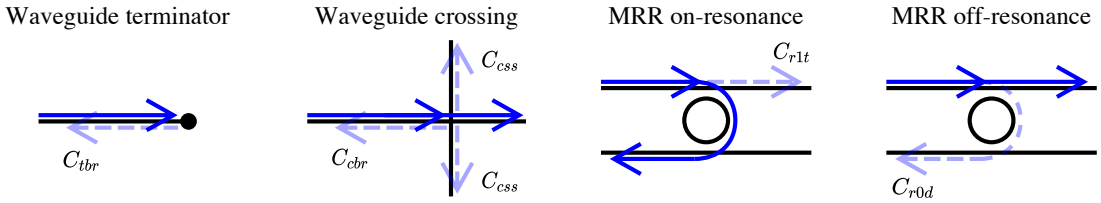

Figure 3.2: Noise sources in ONoCs: terminator back reflection (C_{tbr}), crossing side spill (C_{css}), crossing back reflection (C_{cbr}), MRR on-resonance through (C_{r1t}) and MRR off-resonance drop (C_{r0d}). Full arrows are signal paths, dashed arrows are noise paths.

which then creates second, third, and higher orders as the noise propagates. This will be explained in more detail in section 6.3.

For each signal received at a demodulator, the noise also arriving at the demodulator and interfering with the detection of the signal can be separated into three types: intra-channel noise caused by the signal itself taking a different path to arrive at its demodulator, intra-channel noise caused by other signals of the same wavelength and inter-channel noise caused by other signals of different wavelengths.

Signal-to-Noise Ratio (SNR) in dB is defined as the logarithm of the ratio between signal and noise power in Watt at the demodulator

$$SNR = 10 \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right) \tag{3.6}$$

and BER measures the proportion of incorrect bits to total bits received after demodulation. BER increases (i.e. worsens) as the optical signal power received by the demodulator decreases, and it also increases as the SNR at the demodulator decreases [9, 36]. Often, a requirement is specified for the ONoC to stay under a maximum BER (for example, of 10^{-9}) [36]. This translates into constraints of a minimum amount of signal power and a minimum SNR at the demodulators. Consequently, crosstalk-aware design of ONoCs is essential for reliability and performance, and even more so for larger network sizes where noise generation is higher [2, 56].

3.1.3 Signal bit parallelism

One of the major advantages of ONoCs is higher bandwidth than their ENoC counterparts. This is largely enabled by WDM. One wavelength, modulated at one modulator of the sending node and demodulated at one demodulator of the receiving node, is capable of transmitting one serial stream of bits (i.e. one bit transmitted at a time, in sequence). This is referred to as a bit parallelism of one and is similar in concept to a single digital data wire. To increase the bit parallelism between a pair of nodes, multiple wavelengths are used for the corresponding signal. Modulating N wavelengths in

parallel provides a parallelism of N bits, which is similar to a parallel bus of N digital data wires.

When designing WRONoCs, signal bit parallelism is an important factor that imposes constraints and defines goals for the optimisation process. The specific details depend on the intentions of the designer. At the very least, each signal must have a parallelism of one, i.e. at least one assigned wavelength. But some signals may have higher minimum parallelism targets, which further constrain the optimisation process. So far this may be enough with regards to bit parallelism, i.e. as long as these minimum targets of one or higher are met, the optimisation process is free to pursue other goals. However, the designer may also want to optimise for bit parallelism, i.e. not just achieve parallelism targets but actively pursue designs with the highest bit parallelism possible. In this case, the bit parallelism optimisation function can take different shapes. Consider K signals, each with bit parallelism p_i. Some possible bit parallelism optimisation functions are:

	Speed-based	**Time-based**
Average	$\max \sum_{i=1}^{K} \alpha_i p_i$	$\min \sum_{i=1}^{K} \frac{\alpha_i}{p_i}$
Extreme	$\max \min_{i=1}^{K} \frac{p_i}{\alpha_i}$	$\min \max_{i=1}^{K} \frac{\alpha_i}{p_i}$

Here, $\alpha_i \in \mathbb{R}^+$ are constant optimisation function parameters and a higher α_i increases the importance of the parallelism of signal i in the optimisation function. Speed-based functions maximise the speed, weighted by α_i, at which each signal sends data, and time-based functions minimise the amount of time it takes each signal to send an amount of data α_i.

3.2 Optimisation problem

The optimisation problem for the design of the optical routing layer of a WRONoC is now formally described: an outline is given of the input data, the output data and the optimisation function along with a few supplementary points. The output data (i.e. solution to the optimisation problem) fully describes a WRONoC design at the level of abstraction considered in this work.

3.2.1 Input data

Routing layer. The size and shape of the optical routing layer in which the WRONoC elements are to be placed and routed.

Network nodes. The location on the routing layer of the WRONoC nodes, i.e. where the MAs and DMAs will be placed. Note that the number of MAs and DMAs in

each node and the number of modulators and demodulators in each array is an output, not an input, of the optimisation process.

Laser sources. The type (X/Y) and location (on-chip/off-chip) of the laser sources along with the location on the routing layer of the optical coupler in the off-chip laser source cases.

Communication Matrix (CM). An adjacency matrix M with dimensions $N \times N$ where N is the number of nodes of the WRONoC, $[M]_{ij} = 1$ indicates a signal is sent from node i to node j and $[M]_{ij} = 0$ indicates the opposite. In most cases, diagonal values of M are zero because self-communication is not required in the WRONoC.

Technology parameters. Other constant values related to fabrication technology:

- Set of available wavelengths Λ^{all} and MRR radii R^{all} (section 2.2.4)
- MRRs (section 2.2.4): μ, Q, k_d, k_t, p_d, p_t
- Insertion loss (section 3.1.1): L_p, L_b, L_c, L_{cpl}, L_s (and L_d, L_t which can be obtained from the MRR parameters)
- Crosstalk (section 3.1.2): C_{tbr}, C_{cbr}, C_{css} (and C_{r1t}, C_{r0d} which can be obtained from the MRR parameters)
- Optical power consumption (section 3.1.1): K, S
- Electrical power consumption (section 3.1.1): P_{mrr}, P_{mod}, P_{demod}, η
- Feasible splitting ratio range of optical power splitters

Optimisation function. Structure and weights of the optimisation function.

General purpose WRONoCs have full CMs to cover all possible communication requirements, but application-specific WRONoCs may not require complete communication between all nodes (i.e. have sparser CMs) [46, 57, 58]. In those cases, their design can be substantially improved by customising it to meet the specific requirements and including only the necessary elements [59]. Therefore, it is essential to consider the CM during optimisation.

3.2.2 Output data

Characterisation of the following attributes:

- Set of wavelengths used for each signal
- Radius of each MRR
- Splitting ratio of each PDN splitter

Placement of the following elements:

- PDN splitters between PDN waveguides (if PDN exists)
- MRRs in MAs, DMAs and next to router waveguides
- Waveguide terminators, on the ends of waveguides

Routing of the following elements:

- PDN waveguides, between the coupler, splitters and modulators (if PDN exists)
- Router waveguides, between MAs and DMAs

All aspects of the WRONoC design are defined by this set of outputs. For example, the relative placement of waveguides and MRRs defines which MRRs can influence the signals going through each waveguide. By extension, it also defines the number and design of the MAs and DMAs of each node.

3.2.3 Optimisation function

The full optimisation function is some combination of the performance factors explained in section 3.1. The exact combination is up to the WRONoC designer to decide. As previously mentioned, BER is often specified as a maximum limit. This defines a minimum SNR constraint. Signal bit parallelism can either appear as a minimum target to achieve or as a goal to optimise (see section 3.1.3), and power consumption is always minimised. When multiple optimisation goals are considered (for example, power consumption and bit parallelism), they can be combined using various methods:

Weighted sum. Minimise an expression of the form $\alpha * power - \beta * parallelism$ with $\alpha, \beta \geqslant 0$. This produces the optimal solution only for the specific weight values chosen. Also, *power* and *parallelism* are values in different units, so the weights implicitly include a conversion factor.

Pareto front. Build a set of solutions where for each member of the set no other solution exists that is strictly better in both power consumption and bit parallelism. This set, called the optimal Pareto front, represents the best possible trade-offs between these two objectives. Power consumption and bit parallelism are in different units, but no conversion factor must be specified because this method never performs a direct comparison between both.

A Pareto front provides three distinct advantages over a weighted sum:

Flexibility. The designer is not required to commit to a specific trade-off (i.e. the weights in the sum) before the optimisation is performed. Instead, this is only done after the optimisation finishes.

Completeness. After the optimisation finishes, the designer is shown the full trade-off landscape. This facilitates a more educated decision when choosing the final solution (and multiple final solutions can be chosen, if desired).

Adaptability. The choice of the final solution can also be influenced by priorities not modelled by the optimisation function and any other quality measures that only the designer can specify, such as those based on intuition and experience.

Of course, a Pareto front is also normally more complex to implement and slower to run than a weighted sum.

3.2.4 Other aspects

3.2.4.1 Thermal hotspots

Electrical components heat up during operation and their distribution on the electrical layers may create hotspots on the IC. MRRs are thermally sensitive devices that require tuning to compensate for shifts in Λ^{res} due to changes in temperature. In order to lessen this thermal compensation effort and to ensure that MRRs do not heat up beyond the capabilities of the thermal compensation hardware, MRRs should be placed on colder areas of the IC.

To consider this during optimisation, one possibility is to add a penalty term to the optimisation function that depends on a thermal profile of the IC and the location of the MRRs [60]. This thermal profile can be obtained before optimisation by simulating the electrical layers of the IC under relevant workloads.

3.2.4.2 Problem size

The complexity of this optimisation problem is directly influenced by the total number of signals, which is itself determined by the number of network nodes and the density of the CM. It is unclear for WRONoC networks what the maximum number of signals is in practice. A high number of signals may decrease SNR below acceptable amounts, require more wavelengths than available to ensure even the absolute minimum of one wavelength per signal, or hit some other physical limitation. Naturally, this practical limit depends on the fabrication technology and the capability of WRONoC optimisation tools, so it is expected to increase over time.

Currently, it seems that most published works create, optimise or otherwise reference designs with 8 to 16 nodes [1, 5, 9, 14, 45, 46, 59, 61–64], with some going as far as 32 to 64 nodes [1, 14, 24]. With full CMs, this results in designs with a maximum of approximately 4000 signals. Therefore, a WRONoC optimisation tool should ideally be able to tackle designs of this complexity.

3.2.4.3 Solution quality, computational power and runtime constraints

In optimisation problems, the quality of the solution is often proportional to the amount of computational power and runtime consumed. In these cases, difficulties can arise

if the proportionality constant is very low or if there are constraints on any of these quantities.

For the WRONoC optimisation problem, optimal solutions are not essential. Naturally, the ultimate goal is to obtain the best solutions possible, but it is also acceptable to trade optimality guarantees for lower runtime or algorithms that are more straightforward to implement so long as the solutions are good enough. However, the definition of "good enough" can only be given on a case-by-case basis. For example, a WRONoC design can be considered good enough if it meets the requirements of the WRONoC designer or surpasses previously available designs for the same use-case.

As for computational power and runtime, the WRONoC optimisation problem does not need to be solved under limited hardware or harsh runtime constraints.[2] Using previously published works as reference, state of the art tools for WRONoC design run on commercially available desktop hardware and produce outputs within seconds to minutes. Any future tools should aim to match these results.

3.3 Optimisation process

An optimisation process is a series of steps responsible for taking a set of inputs and producing one or more valid and high quality sets of outputs. In general, the ideal optimisation process examines all facets of the optimisation problem simultaneously, allowing all trade-offs between different aspects to be evaluated, thus arriving at the absolute best solution (or set of solutions) for a given optimisation function. It also does so fast enough for the relevant problem sizes when given a viable amount of computational power, and ideally requires minimal human input.[3]

The WRONoC optimisation problem, as explained in the previous sections, is quite complex. So far no process has been published for this problem that achieves all of these lofty goals. Instead, previous works have considered only limited parts of the problem and employed simplifications to make it more manageable. These will now be discussed.

3.3.1 Sequential optimisation

One of the most common and widely used simplifications for addressing this optimisation problem is to adopt a "divide and conquer" approach: instead of considering the entire problem simultaneously, divide the problem into smaller, individual sub-problems, each with their own unique inputs, outputs, constraints, and optimisation functions. Each sub-problem is then optimised only once, in sequence.

[2] As opposed to, for example, embedded hardware in an autonomous drone running a real-time path-planning algorithm through a 3D space.

[3] The definitions of "fast enough", "relevant" and "viable" can only be precisely specified on a case-by-case basis. For WRONoCs, see sections 3.2.4.2 and 3.2.4.3 above.

Many state of the art works divide the complete WRONoC problem into three sub-problems, or steps. These are, in order:

Logical topology design. This first step takes as inputs the CM and the technology parameters and outputs a logical topology that defines the structure of the WRONoC along with the set of wavelengths used for each signal and the radius of each MRR. This resulting topology must fulfil the communication requirements of the CM.

With full CMs, this step can be as simple as choosing a standard logical topology (for example, λ-router, Snake router, etc) and adequately sizing it for the specified number of nodes. However, as mentioned in section 3.2.1, sparser CMs present opportunities to design smaller topologies. The design process for these cases commonly consists of starting with a standard logical topology fit for a full CM and optimising WRONoC performance factors by removing as many unneeded elements from the topology as possible [59, 65, 66].

All three performance factors should be considered in this step, but while the information required to optimise for bit parallelism is already fully available, only incomplete approximations of final power consumption and crosstalk are possible at this stage.

Physical router layout. This second step takes as inputs the optical routing layer dimensions, the network node locations, the technology parameters and the logical topology from the first step. It then places the MRRs and waveguide terminators on the routing layer and routes the waveguides accordingly. Multiple optimisation tools have been developed for this process [9, 60–63, 67].

Because signal bit parallelism is only dependent on the logical topology, this step only needs to consider power consumption and crosstalk. The calculations for these two performance factors at this point are exact unless a PDN must still be added.

PDN design and layout. This third step is necessary only when off-chip laser sources are used. It takes as inputs the routing layer dimensions, the technology parameters, the laser source information and the output from the previous step. It then performs its own design and layout sub-steps:

1. It selects the best structure for the PDN tree that connects to all MAs,
2. it places the optical power splitters and routes the PDN waveguides accordingly, and
3. it determines the splitting ratios necessary to supply the required power to each MA.

Both power consumption and crosstalk should be considered during these substeps. Some works include this PDN step in their WRONoC design process, but do so only for specific logical topologies, consider only a small subset of possible PDN structures, do not fully explore the PDN physical layout possibilities and/or perform PDN optimisation manually [1, 14, 24]. Also, some works perform the entire PDN step before the other two steps (which includes selecting the splitter ratios before any power consumption information is available) [24].

The defining characteristic of this simplification is that the output of each step is passed only to the subsequent step. Because there is no feedback and previous steps are not optimised again, the design decisions made at the end of each step are fixed for the rest of the optimisation process. This reduces the total search space considered during optimisation, which is why this sequential approach is very effective at making problems faster and easier to solve. However, there is one major trade-off: the consecutive search space reduction caused by each step may remove the global optimal solution of the full problem from the search space. Problems where this issue is guaranteed not to happen are said to exhibit optimal substructure.

Unfortunately, the WRONoC optimisation problem does not exhibit optimal substructure when divided in this way. This is because obtaining the optimal logical topology requires exact calculation of power consumption and crosstalk, which can only be done after the router layout and PDN layout have been performed. Therefore, it is impossible to guarantee that the global optimal solution of the full problem is still in the search space after the logical topology has been fixed. Similarly, obtaining the optimal router layout requires precise calculation of power consumption and crosstalk, which can only be done after the PDN layout has been performed. The same argument applies to obtaining the optimal PDN design if the PDN step is performed before the other two. When on-chip laser sources are used the PDN step is not required, but the logical topology design step still depends on the physical router layout step. Not only do all of these interdependencies preclude optimal solutions (unless by luck), but they also make good solutions more difficult to obtain.[4]

The interdependency between the first and second steps is very strong. This is because logical topology design has a major influence in the performance factors of the final result, while at the same time being limited by the fact that those performance

[4] An analogy may be useful. Consider a combination lock with n rotating discs, each of which has k options. The correct combination must be selected to open the lock. A thief does not have sufficient time to check all k^n combinations, so resorts to selecting a position for each disc once, in sequence. A poorly constructed lock allows the thief to independently validate the position of each disc. So, for each disc, the thief tries all k options until finding the correct one. The lock is always opened successfully because it exhibits optimal substructure (in practice, this is very common due to large manufacturing tolerances). Conversely, a well-designed lock provides no feedback until the end of the process, requiring the thief to select the position for each disc blindly. The lock does not exhibit optimal substructure and only a lucky thief will get it open.

factors can only be calculated accurately after the physical router layout is performed. More specifically, physical layout has been consistently shown to increase the length of waveguides and the number of waveguide crossings substantially and unpredictably [1, 5, 9, 62]. For example, the maximum number of crossings of a signal path in a λ-router may increase from 7 after the first step to over 90 after the second step [9], which has significant implications for the performance factors:

- More crossings increases crosstalk, which increases BER.
- More crossings or longer waveguides increases total insertion loss, which increases power consumption.
- More crossings or longer waveguides changes the insertion loss distribution over the wavelengths. This is very relevant for type Y lasers whose power consumption, as shown in eq. (3.4), is determined by a max function and is thus sensitive to changes in signals which have substantially more insertion loss than others.

Some works can avoid creating extra crossings during the second step when the logical topology is planar. This completely prevents increasing crosstalk, but substantially increases waveguide length [63]. The PDN step also causes the same three effects, although their magnitude is less studied in the literature [1,24]. Finally, larger WRONoCs with more signals have more elements in the logical topology that must be placed and routed, which increases these discrepancies and worsens their effects.

3.3.2 Symbolic wavelengths and MRRs

The complete logical topology design process for WRONoCs requires assigning a radius to each MRR and one or more wavelengths to each signal according to the rules explained in section 2.4.2. This is a difficult combinatorial problem [25, 28] that a substantial fraction of published works choose to simplify [1, 5, 14, 24, 46, 59].

This simplification starts by establishing that MRRs have exactly one resonance wavelength and that there is a one-to-one correspondence between an MRR's resonance wavelength and its radius.[5] It continues by stating that a wavelength belongs to either the drop-set or the through-set of each MRR (with no other option) and that an MRR can be defined entirely by its resonance wavelength, so its radius is no longer relevant. Since MRRs only have one resonance wavelength, signals must also be assigned only one wavelength. Therefore, when a signal gets close to an MRR, it is either using the MRR's resonance wavelength, in which case it is routed to its drop port, or it is not, in which case it is routed to its through port. The actual wavelength value also becomes irrelevant and is substituted with a symbolic entity (for example, λ_1, λ_2, λ_x, λ_n, etc) which is now only used in equality comparisons. Note that the exclusion principle can

[5] ...under constant thermal and electrical tuning conditions. Given that this is a high-level approximation of MRR behaviour, these details are also normally ignored.

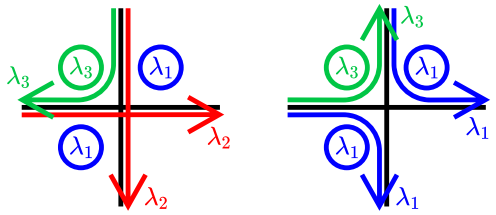

Figure 3.3: MRR and signal interactions using symbolic wavelengths. Routing: based on whether the MRR and the signal use the same symbolic wavelength. Exclusion principle: only signals with different symbolic wavelengths can use the same waveguides.

still be enforced: two signals can use the same waveguide or have the same symbolic wavelength, but not both. An example of this simplification is shown in fig. 3.3.

This approximation of MRR–signal interaction has three consequences:

- Signals only have one wavelength, so direct optimisation of signal bit parallelism is impossible. Under this simplification, the total number of symbolic wavelengths used is the only bit parallelism-related metric available for optimisation.
- Wavelength and radius values were discarded. Therefore, exact calculation of insertion loss parameters L_d, L_t and crosstalk parameters C_{r1t}, C_{r0t} as a function of wavelength and radius using eqs. (2.2) and (2.3) is no longer possible, and values constant for all wavelengths and radii must be used instead. These approximate values are equal to the average of the exact values over all wavelengths and radii [68].
- Power consumption eqs. (3.3) and (3.4) depend on the number of wavelengths used. This should be the number of actual wavelengths used, but under the symbolic wavelengths simplification, the number of symbolic wavelengths is substituted in the equations instead.

In the process of going from a WRONoC logical topology using symbolic wavelengths to a complete WRONoC design, these symbolic wavelengths must at some point be replaced with sets of wavelength values and corresponding MRR radii. An optimisation step that performs this replacement is not constrained by this simplification, by definition, and so it is able to directly optimise for bit parallelism. However, its success depends heavily on how many symbolic wavelengths were used in the original (simplified) logical topology. This is because, when the logical topology is designed to use many different symbolic wavelengths, the number of actual wavelengths available for each symbolic wavelength becomes heavily restricted by WRONoC design constraints (see section 2.4.2). This can lead to bit parallelism requirements not being satisfied (see section 3.1.3), and in the worst case some signals may not even have a single wavelength

available [25,28]. Therefore, minimising the number of symbolic wavelengths used when designing simplified logical topologies helps the later bit parallelism optimisation step complete successfully and achieve better results.

When this simplification is used in conjunction with sequential optimisation, the logical topology design step in section 3.3.1 is divided into two sub-steps: first design a symbolic logical topology while minimising the total number of symbolic wavelengths (among other performance factors), and then replace the symbolic wavelength with actual wavelengths as described above.

3.3.3 Approximate optimisation goals

Considering all three WRONoC performance factors together in full detail is complex. To simplify the optimisation process, some shortcuts are often implemented:

- A reduced number of performance factors (often, only one) is considered. For example, power consumption is typically the only factor taken into account when designing logical topologies and performing their physical layout [9, 59, 61–63]. Crosstalk is almost never considered during optimisation [69], and is instead calculated as a verification step only after the design is completed [1].

- When more than one factor is considered, the factors are combined into a single objective function using a weighted sum rather than the more complete option of constructing a Pareto front of solutions [9, 59].

- Total power consumption, which is equal to the sum of the $P_{laser:el}$, P_{mod}, P_{demod} and P_{mrr} terms (see section 3.1.1), is only approximately minimised.

  To accurately calculate total power consumption in Watt, one must first calculate the insertion loss at each MA, then, if applicable, take into account the insertion loss caused by the PDN, then convert the total insertion loss into laser power consumption using eqs. (3.3) to (3.5), and finally add the power consumption of each modulator, demodulator and MRR. This is very rarely done accurately, for multiple reasons.

  Firstly, some works do not consider the PDN during optimisation if off-chip laser sources are used [9, 61, 63]. Secondly, the conversion between insertion loss in dB and power consumption in Watt described by eqs. (3.3) and (3.4) is non-linear, and thus cannot be performed exactly by linear optimisation methods [59]. As a result, for minimisation purposes, it is very common to use the total and maximum insertion loss over all modulators as approximations for $P^X_{laser:opt}$ and $P^Y_{laser:opt}$ respectively.

  When doing this, the total power consumption can no longer be calculated: terms P_{mod}, P_{demod} and P_{mrr} are in Watt, but $P_{laser:el}$ has been substituted with a value

in dB which is in a different unit and cannot be added directly. Two further transformations are then performed:

  - If the number of modulators and demodulators is fixed before optimisation, their total power consumption is a constant value and can be ignored during optimisation.
  - Since each MRR consumes the same amount P_{mrr} of power, minimising total MRR power consumption is equivalent to minimising the number of MRRs.

  In the end, minimisation of total power consumption is simplified into the minimisation of two terms: total/maximum insertion loss over all modulators and number of MRRs. They are combined into one objective function together with any other optimisation terms (for example, those coming from the bit parallelism performance factor).

- As mentioned in section 3.3.2, the symbolic wavelengths simplification enforces approximate optimisation goals for all three performance factors.
- Crosstalk calculation considers first-order noise only and higher orders are ignored (see section 6.3).

3.3.4 Other simplifications and limitations

A few other miscellaneous simplifications and limitations are also common when designing WRONoCs:

- In section 3.2.4.2 it was argued that 64 nodes, i.e. approximately 4000 signals, is currently an upper bound for WRONoC networks in practice. Unfortunately, many published WRONoC optimisation tools cannot handle such complexity in a reasonable amount of time and are limited instead to a maximum of around 16 to 20 nodes, i.e. approximately 250 to 400 signals [9, 28, 59].
- Some works do not take into account the specific communication requirements of the network and instead always assume a full CM [1, 5]. Designs created under this assumption are likely to be sub-optimal when used for application specific cases.
- As explained in section 3.3.1, many works simplify PDN optimisation. So far, no work has fully considered all aspects of PDN design together during its optimisation process.
- Almost all works fix the number of MAs and DMAs per node (normally to one of each) before optimisation, which can limit solution quality.
- Thermal hotspots (see section 3.2.4.1) are rarely considered during optimisation [60].

3.3.5 A plan forward

As stated above, the ideal WRONoC optimisation process considers the entire design problem at once and without any simplification or limitation. Such a process does not exist yet, and the problem is complex enough that attempts in this direction are not guaranteed to be entirely successful. Nevertheless, this section introduces a plan for incrementally developing a complete, no-compromise and turn-key WRONoC optimisation process[6], and provides a rationale for its expected effectiveness.

The plan starts by acknowledging that the "divide and conquer" strategy is essential to tackle complex problems such as this one, especially in the beginning. However, the key is to divide the problem in such a way that not only are all pieces effective by themselves, but can also be merged together gradually over time with minimal effort.

At the core of the state of the art is the division between logical topology design and physical router layout. Unfortunately, this division fails on both aspects:

- The logical topology design step is fundamentally incapable of being effective by itself because it requires essential information during its optimisation that is only available later and for which there are no suitable approximations in the moment.
- There is no clear path to take current logical topology design and physical router layout tools and merge them.

Evidently, a new approach is necessary. The proposed plan divides the optimisation problem into two steps as follows:

1. Perform logical topology design and physical router layout (and PDN design and layout if necessary) together, but use symbolic wavelengths. As a result, enough information is available to optimise for power consumption directly, but the symbolic wavelength simplifications explained in section 3.3.2 must be used and so only approximate crosstalk and bit parallelism optimisation is possible.
2. Take the output of the first step and replace the symbolic wavelengths with sets of wavelength values and corresponding MRR radii. This step now also has all the information required to optimise for crosstalk and bit parallelism exactly.

This approach is expected to work better than the state of the art for multiple reasons:

- Each step has enough information to perform its main optimisation duties effectively (power consumption on the first, crosstalk and bit parallelism on the second).

[6] Complete: the entire scope of the WRONoC optimisation problem is considered together. No-compromise: the optimisation process is not limited and does not employ simplifications, such as those explained in section 3.3. Turn-key: minimal user input is required.

- The dependencies between steps can be adequately handled. The first step strongly influences crosstalk and bit parallelism but lacks all the information required to directly optimise for those performance factors. Fortunately, capable approximations (average C_{r1t}, C_{r0d} values for crosstalk and number of symbolic wavelengths for bit parallelism) are available.

- There is a clear path towards merging both steps. The first step is already required to choose symbolic wavelengths for the signals and MRRs, so the second step can be merged on to the first by selecting actual wavelengths and MRR radii instead. Naturally, this substantially increases the problem complexity targeted in one single step, which is why processes for both steps are developed separately first.

The success of this plan hinges first and foremost on how the logical topology and physical layout are combined. As such, a process that performs the first step in its entirety will undergo multiple stages of development:

1. Propose a new approach to integrate these two components during optimisation and validate it against the state of the art (see chapter 4). To allow for rapid prototyping, most process simplifications are used initially. In particular, the PDN is not considered and therefore power consumption is optimised using insertion loss values at the modulators (see section 3.3.3).

2. If this approach is successful, improve the newly developed process by including the PDN, which unlocks complete power consumption optimisation (see chapter 5).

3. Address any other unresolved simplifications or limitations (except for symbolic wavelengths).

Likewise, a process to perform the second step will also be constructed in multiple stages:

1. Develop an optimisation process that takes a WRONoC design output by the first step and assigns actual wavelengths and MRR radii while optimising only for bit parallelism. This has already been attempted, but designs with more than 40 signals cannot be handled in a reasonable amount of time (and not all bit parallelism optimisation functions in section 3.1.3 are considered) [28].

2. Establish an accurate crosstalk and SNR calculation method for any ONoC design that is fast enough to be used in the optimisation function of this second step (see chapter 6).

3. Merge the crosstalk calculation method into the process, thereby optimising for both bit parallelism and crosstalk.

Since the two steps are independent, their multi-stage development can be carried out simultaneously. This development should continue until the use of symbolic wavelengths in the first step is the only simplification or limitation present in both processes. If successful, this will result in two WRONoC design and optimisation tools that are already very capable by themselves and even more so when used together (in sequence). At this point, one final effort can be made to merge the second process into the first, achieving the original goal.

4 Combining optimisation of logical topology and physical layout[1]

The WRONoC design problem was explained in detail in the previous chapter. At the end, a plan was proposed to gradually move towards a full solution to this problem. This chapter presents the first step along this process: a way to combine logical topology and physical layout optimisation using a Physical Layout Template (PLT), and a tool[2] called PSION+ that performs this optimisation using a Linear Programming (LP) model.

This new approach is described in section 4.1 and its implementation is explained in section 4.2. Section 4.3 provides an in-depth analysis of a simple yet flexible PLT structure. Section 4.4 then proposes multiple heuristics to decrease runtime. Finally, section 4.5 compares PSION+ against the state of the art Place & Route (P&R) tools PROTON+ [9] and PlanarONoC [63] and analyses the performance of the proposed PLT designs and heuristics.

4.1 Physical Layout Template

As explained in section 3.3.5, the first phase in the proposed plan is to design an optimisation process that merges the logical topology and physical layout steps. The primary goal behind this approach is to provide the process with enough information to be able to optimise for power consumption directly while allowing any restrictions, choices or optimisation opportunities from one step influence the choices made on the other.

The logical topology step contains both synthesis and optimisation requirements, and the physical layout step contains only optimisation requirements. For example: deciding what MRRs and waveguides are used is part of logic topology synthesis, choosing the radius of each MRR is part of logic topology optimisation, and placing the MRRs and routing the waveguides is part of physical layout optimisation. Clearly, synthesis must always be performed before optimisation (for example, waveguides can only be routed after the synthesis step has decided what waveguides are used). Therefore, it is not immediately obvious how these two steps can be merged effectively, in particular

[1] Segments of this chapter were previously published in [6, 70].

[2] The tool PSION was published first [17, 70]. Later, PSION+ [6] was published as an extension to PSION. Because PSION is contained by PSION+, the latter is presented here.

so that each can influence the other (for example, waveguide crossings created by P&R influencing waveguide synthesis decisions).

Fortunately, there is some inherent structure to the WRONoC optimisation problem that can be exploited. Firstly, node locations are known and fixed before optimisation, which means many router waveguides have well-defined end points. Secondly, MRRs must be placed at a specific distance next to the waveguides they are meant to influence, and away from all other waveguides.

Motivated by this fact, the proposed solution simplifies the WRONoC optimisation problem by using a new input: a Physical Layout Template (PLT), which consists of a collection of WRONoC router elements (MAs, DMAs, waveguides and MRR placeholders) already placed and routed on the optical routing layer. The PLT simplifies the problem because the elements it contains are the only elements the final WRONoC design can use, which means all the synthesis burden is handled when creating the PLT. Consequently, an optimisation process that takes a PLT as input only has to address the optimisation requirements. Furthermore, because all the optimisation requirements are considered together, they can influence each other during optimisation and power consumption can be calculated directly. This stands in direct contrast to the state of the art, whose sequential optimisation process includes optimisation requirements on separate steps and therefore cannot achieve either of these goals.

4.1.1 Template elements

PLTs are composed of multiple instances of three basic elements, each with a fixed location on the optical routing layer:

Endpoints represent MAs and DMAs. Their location on the optical routing layer is the same as the MAs/DMAs they represent, and they connect to one waveguide section.

Generic Routing Units (GRUs) have four ports that connect to four waveguide sections (called the edges of the GRU) and contain MRR placeholders to be populated as needed. They are the only PLT element that can contain MRRs, making them the routing building blocks of the PLT, and are described further in section 4.1.2.

Waveguide sections connect two GRUs or a GRU and an endpoint. Each section has two constant associated parameters: *length* and *extraloss*. The latter is used to describe sections with other constant sources of insertion loss besides length such as bends or crossings with other sections.

One of the simplest PLTs that can be built with these elements is the Grid Template (GT), shown in fig. 4.1. It has a regular grid structure of GRUs and will be further discussed in section 4.3. Many other PLT designs are also possible and some will be presented in section 4.5.1.

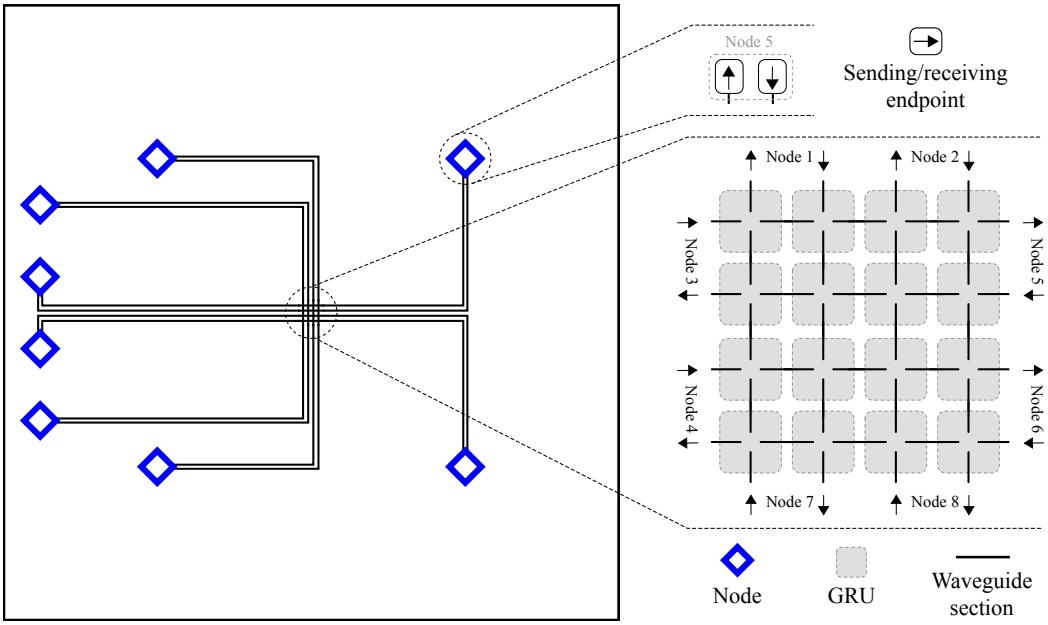

Figure 4.1: Example of a PLT for 8 nodes: a Grid Template (GT).

4.1.2 Generic Routing Unit

PSEs are commonly used in WRONoC routers. However, PSEs have some distinct shortcomings:

- They only have one or two MRRs, where in fact it is possible to place up to four MRRs on a single crossing (one on each corner).
- Both MRRs always have the same radius, where in fact all four MRRs on a crossing can have different radii.
- Their waveguide structure is fixed (PSEs always have a crossing, for example), where in fact other structures are also possible.
- Signals only travel unidirectionally (PSEs always have exactly two input ports and two output ports), where in fact signals can travel in both directions on a waveguide simultaneously.

To solve this inflexibility a new type of building block is proposed: the Generic Routing Unit (GRU). Externally, GRUs still have four ports to which waveguides are connected to, like PSEs. However, in contrast to PSEs, the internal structure of GRUs is not inherently constrained to a specific configuration of waveguides and MRRs. Internally, many different waveguide configurations are possible and MRR placeholders may be populated with MRRs of different radii. Furthermore, each GRU instance on

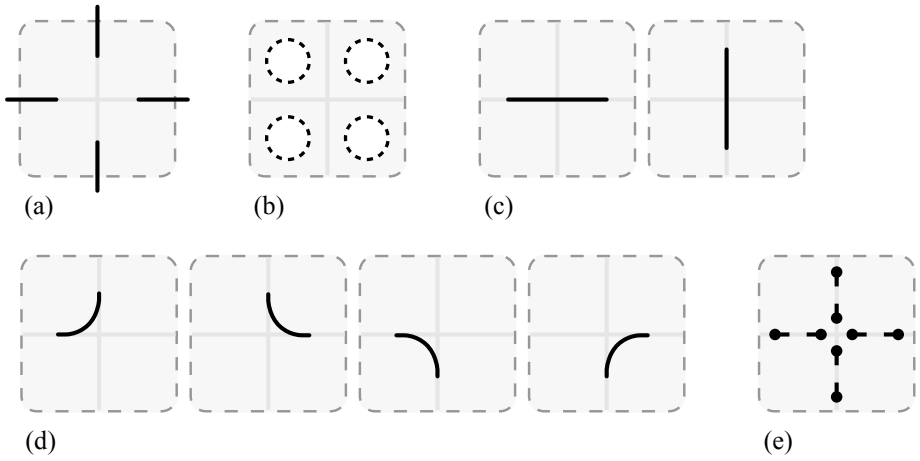

Figure 4.2: The set $\mathbb{S}$ of all 22 possible GRU internal structure elements. (a) Four edge waveguides. (b) Four MRR placeholders. (c) Two center crossing waveguides. (d) Four corner bending waveguides. (e) Eight waveguide terminators.

the PLT can be configured independently to have whatever internal structure is best suited for the given design requirements. This flexibility plays a fundamental role in unlocking new areas of the WRONoC design space.

4.1.2.1 Structure

To define the internal structure of a GRU, first a set $\mathbb{S}$ is defined which contains all possible GRU internal structure elements. This set contains MRR placeholders and various types of internal waveguide connections on certain positions of the GRU:

Edge waveguides are the ends of the four waveguide sections that connect to the GRU, shown in fig. 4.2(a).

MRR placeholders represent the predefined positions where MRRs of various radii can be placed, shown in fig. 4.2(b).

Center crossing waveguides are waveguide fragments that connect edge waveguides on opposite sides through the center of the GRU, shown in fig. 4.2(c).

Corner bending waveguides are waveguide fragments that connect edge waveguides on adjacent sides through a corner with a 90° bend, shown in fig. 4.2(d).

Waveguide terminators are required to close off the ends of other waveguide elements which are not connected to anything, shown in fig. 4.2(e).

The internal structure of any GRU instance on the PLT is built from a subset of $\mathbb{S}$, as demonstrated by fig. 4.3. An optimisation process that takes a PLT as an input

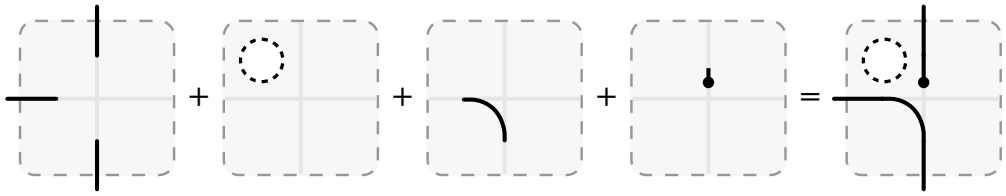

Figure 4.3: Example internal structure of a GRU instance built from a subset of $\mathbb{S}$.

must configure it for the chosen signal paths and wavelengths. This includes deciding, for each GRU instance, what subset of $\mathbb{S}$ should be used.

However, not all subsets of $\mathbb{S}$ are valid:

1. The use of center crossing waveguides and corner bending waveguides is mutually exclusive. Using any center crossing waveguide prohibits the use of any corner bending waveguide and *vice-versa*.

2. For each corner, the placement of an MRR and the use of a corner bending waveguide is mutually exclusive.

3. Any two corner bending waveguides that connect to the same edge are mutually exclusive.

4. Any unconnected ends of waveguides must be closed off with a waveguide terminator (e.g. an edge waveguide ending with a waveguide terminator like in fig. 4.3).

5. An MRR can only be placed on a corner where both of its adjacent edges have waveguides (this could be edge, center crossing or corner bending waveguides).

4.1.2.2 Routing

The different GRU structures available lead to different routing behaviours, but routing through a GRU is still based on the principles shown in figs. 2.3 and 2.9: a signal will continue on the waveguide it is traveling in (i.e. through a center crossing or corner bend) unless that waveguide passes next to an on-resonance MRR, in which case the signal will be diverted to another waveguide. Figure 4.4 shows examples of the routing possibilities in a GRU.

To ensure correct signal routing, one last rule must be followed when configuring GRUs:

6. Two MRRs on corners adjacent to the same edge must have different radii (under the symbolic wavelength simplification, this means different symbolic wavelengths).

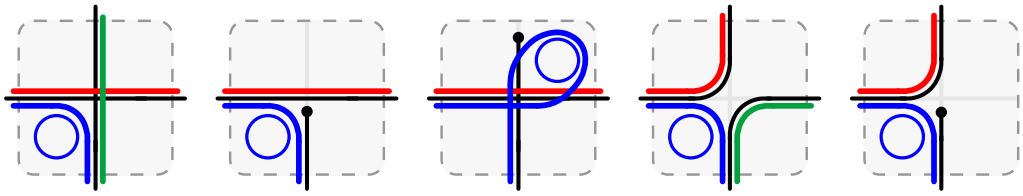

Figure 4.4: Examples of signal routing in a GRU. The signal with wavelength *blue* is routed by the MRR, whereas signals with other wavelengths (*red* and *green* in this case) are not.

Otherwise, a signal with a wavelength in the drop-set of the two MRRs will be routed by both MRRs at the same time, meaning it will not have a well-defined path.

Finally, it is important to note that none of the routing features on a GRU depend on the direction of the signal, i.e. all routing features are bidirectional. As a consequence, a signal's path between two endpoints does not depend on which endpoint is the MA and which is the DMA.

4.1.3 Endpoints

The CM can be translated into a set of signals (one for each non-zero entry) where each signal is defined by a tuple (N_m^S, N_m^R): N_m^S is the sending node and N_m^R is the receiving node of signal m.

When using a PLT, each WRONoC node n can be defined by a tuple $(\mathbb{K}_n^S, \mathbb{K}_n^R)$: $\mathbb{K}_n^S$ is the set of sending endpoints (MAs) and $\mathbb{K}_n^R$ is the set of receiving endpoints (DMAs) for node n. While simpler PLTs may only have one MA and DMA per node, multiple MAs or DMAs are perfectly feasible, and even possibly beneficial [24].

Consequently, each signal can be associated with two sets of endpoints: $\mathbb{K}_{N_m^S}^S$, the endpoints available to send the signal m, and $\mathbb{K}_{N_m^R}^R$, the endpoints available to receive the signal m (henceforth these will be referred to as $\mathbb{E}_m^S$ and $\mathbb{E}_m^R$ respectively). For each signal, when any of these two sets contain more than one endpoint, the optimisation process should figure out which endpoint in that set is the best choice.

4.1.4 Concluding comments

In summary, the PLT removes synthesis requirements from the optimisation process, thereby reducing its duties to:

- Routing each signal defined in the CM through the PLT.
- Assigning wavelengths to each signal.[3]

[3]Under the symbolic wavelength simplification, this is just one symbolic wavelength.

- Configuring the PLT for the chosen signal paths and wavelengths, i.e. activating the necessary routing features in the GRUs and removing unnecessary waveguides.

These are just optimisation requirements, as stated before, and they come from both the logical topology and physical layout steps. The PLT enables the simultaneous execution of these responsibilities, which is why logical topology and physical layout can influence each other during the optimisation process.

The PLT also has a few more advantages. Firstly, GRUs possess enough modelling capability and versatility that any WRONoC design can represented by a suitable PLT. Consequently, an optimisation process that takes a PLT as an input is automatically very general. Secondly, the PLT makes it trivial to account for thermal hotspots (see section 3.2.4.1).

Finally, the creation of the PLT itself has not been discussed yet. Generally speaking, producing a suitable PLT is relatively easy. For example:

- Any previously existing logical topology (such as λ-router, ring templates and others, as described in section 2.4.1) can be adapted to use GRUs instead. This is especially easy for topologies based on PSEs. The PLT elements are then placed and routed on the optical routing layer to form the complete PLT (some suggestions on how to do this are given in section 4.3). In this way, these pre-established designs can be optimised for application-specific WRONoCs (while taking into consideration their physical layout) because PLT elements such as MRRs and waveguide sections are only used when necessary.

- The designer can exercise his/her experience and/or intuition to create bespoke PLTs for specific situations (an example of this is given in section 4.5.1).

- New PLT topologies such as the GT can be created and explored (see section 4.3).

All of these use cases can be done manually but some can also be automated, which reduces the burden on the WRONoC designer.

4.2 Implementation using Linear Programming

Having proposed a way to combine logical topology and physical layout using a PLT, a method will now be presented that can perform this optimisation. The main goal of this method is to validate this new approach against the state of the art.

As explained in section 3.3.5, symbolic wavelengths are used and the PDN is not yet considered during optimisation at this stage.[4] As a result, signal bit parallelism is optimised by minimising the number of symbolic wavelengths and power consumption is optimised by minimising total/maximum insertion loss and MRR count. Crosstalk

[4]Nevertheless, special care is taken to avoid crossings between the router and the PDN, if one is added after optimisation (see section 4.3.1).

Table 4.1: Model constants and indices.

Constants	
N_{gru}, N_{wg}, N_m, N_{ep}, N_λ	Total number of GRUs, waveguide sections, signals, endpoints and wavelengths
L^P, L^C, L^B, L^D, L^T	Values for propagation, crossing, bending, drop and through loss
L_{wg}, L^E_{wg}	Length and extra loss of waveguide section wg
Indices	
W^T_g, W^B_g, W^L_g, W^R_g	Waveguide section connected to GRU g to the top, bottom, left and right
W^E_{ep}	Waveguide section connected to endpoint ep
$\mathbb{E}^S_m$, $\mathbb{E}^R_m$	Set of sending and receiving endpoints for signal m

is indirectly optimised by minimising MRR count and number of crossings (number of crossings is minimised because crossings contribute to insertion loss). This is consistent with the state of the art against which this new method is being compared [5, 9, 24, 25, 45, 46, 63]. Finally, thermal hotspots are not considered, but this is trivial to add at a later stage, as stated before.

The backbone of this method is LP, more specifically Mixed-Integer Linear Programming (MILP). The main reason why LP is used at this stage is that it allows for fast and effortless prototyping of this new PLT approach. This is because the effort of developing an *ad hoc* algorithm for PLT optimisation is converted into modelling PLTs using variables and linear constraints, which can then be fed to an LP solver that handles the optimisation process. An additional benefit of LP is that it provides optimal solutions, or at least an upper/lower bound to the optimal value of the optimisation function.

The model constants and indices are outlined in table 4.1. Constants L_{wg}, L^E_{wg} and indices W^*_i collectively describe the PLT and indices E^*_m define the CM. Table 4.2 lists all model variables. Section 4.2.1 explains the model constraints (note that similar constraints referring to multiple corners and the standard MILP linearisation techniques applied to some constraints are omitted for brevity and clarity) and section 4.2.2 describes the linear objective function. Finally, section 4.2.3 presents a fast proof of feasibility for the LP model.

Table 4.2: Model variables. Index $p \in \mathbb{P}$ and $\mathbb{P} = \{TL : \text{Top-Left}, TR : \text{Top-Right}, BL : \text{Bottom-Left}, BR : \text{Bottom-Right}\}$

Binary	
$mwg_{m,wg}$	Signal m uses waveguide section wg
$mwl_{m,\lambda}$	Signal m uses wavelength λ
$mwle_{m_1,m_2}$	Signals m_1 and m_2 use the same wavelength
$mlc^1_{g,m}$	Signal m has crossing loss once on GRU g
$mlc^2_{g,m}$	Signal m has crossing loss twice on GRU g
$mlb_{g,m}$	Signal m has bending loss on GRU g
$mlt_{g,p,m}$	Signal m has through loss due to MRR p in GRU g
$gr_{g,p}$	MRR on corner p of GRU g is used
$grm_{g,p,m}$	MRR on corner p of GRU g is used by signal m
$gcb_{g,p}$	Corner bending waveguide on corner p of GRU g is used
$gcch_g$	Horizontal center crossing waveguide of GRU g is used
$gccv_g$	Vertical center crossing waveguide of GRU g is used
wlu_λ	At least one signal uses wavelength λ
Integer	
nwl	Number of used wavelengths
Continuous	
mil_m	Insertion loss for signal m
$maxil$	Maximum insertion loss over all signals

4.2.1 Constraints

4.2.1.1 Signal routing

From a routing standpoint the PLT can be interpreted as a graph where endpoints and GRUs are the vertices and the waveguide sections are the edges. The routing features in GRUs are bidirectional, so the direction of a signal does not influence its path, thus making the graph undirected. To model signal paths, three sets of constraints are needed as described next.

The path must start and end at the correct endpoints, i.e. each signal must use the waveguide section (W^E_{ep}) of exactly one of the possible endpoints it can be sent from ($\mathbb{E}^S_m$) and received by ($\mathbb{E}^R_m$):

$$\sum_{ep \in \mathbb{E}^S_m} mwg_{m,W^E_{ep}} = 1 \tag{4.1}$$

$$\sum_{ep \in \mathbb{E}^R_m} mwg_{m,W^E_{ep}} = 1 \tag{4.2}$$

$$\forall m = 1...N_m$$

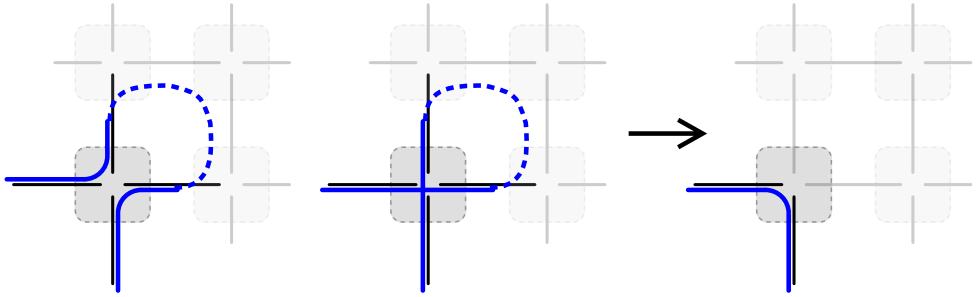

Figure 4.5: Path simplification from usage of 4 edges to 2 edges of a GRU.

Conversely, if an endpoint does not send or receive a given signal, that signal cannot use its waveguide section:

$$mwg_{m,W^E_{ep}} = 0 \qquad (4.3)$$
$$\forall ep \in \{1...N_{ep}\} \setminus (\mathbb{E}^S_m \cup \mathbb{E}^R_m), m = 1...N_m$$

Finally, the path must be continuous from the sending endpoint to the receiving endpoint. Gaps in the path appear when a signal does not exit a GRU once for every time it enters it. To avoid gaps, constraints must be added to ensure that each signal uses an even number of edges on each GRU, i.e. either 0, 2 or 4 edges. However, the choice was made to restrict those possibilities to only 0 and 2, for two reasons:

Unlikely usage in optimised solutions. A path that uses a GRU twice can almost always be simplified into a path that only uses it once, as shown in fig. 4.5. Also, a path that uses it twice has necessarily a bigger insertion loss (it has twice the loss on the GRU and the path must be longer, so it has an increased propagation loss too). Therefore, good solutions are unlikely to feature this case.

Model simplicity. The constraints for the activation of the routing features (see section 4.2.1.3) and for the calculation of the insertion loss of each signal (see section 4.2.1.4) become much simpler if a signal is restricted to use each GRU at most once (2 edges) instead of twice (4 edges).

This is expressed with the following constraint:

$$mwg_{m,W^T_g} + mwg_{m,W^R_g} + mwg_{m,W^B_g} + mwg_{m,W^L_g} \in \{0, 2\} \qquad (4.4)$$
$$\forall m = 1...N_m, g = 1...N_{gru}$$

4.2.1.2 Wavelength assignment

A wavelength must be assigned to each signal, but the actual wavelength value does not matter since the wavelengths are all symbolic (λ_1, λ_2,...). What is important,

however, is to define what signals can use the same wavelengths and what signals must use different wavelengths so that the optical exclusion principle is satisfied. This also makes wavelength selection highly dependent on the chosen signal paths.

To correctly carry out this assignment, first make sure each signal uses exactly one wavelength:

$$\sum_{\lambda=1}^{N_\lambda} mwl_{m,\lambda} = 1 \qquad \forall m = 1...N_m \tag{4.5}$$

Then keep a record of what pairs of signals use the same wavelength by setting the values of $mwle_{m_1,m_2}$ accordingly:

$$\begin{aligned} mwl_{m_1,\lambda} \wedge mwl_{m_2,\lambda} \Rightarrow mwle_{m_1,m_2} \\ \forall \lambda = 1...N_\lambda, m_1, m_2 = 1...N_m : m_2 > m_1 \end{aligned} \tag{4.6}$$

Finally, enforce the optical exclusion principle, i.e. if two signals share a wavelength, they cannot both use the same waveguide:

$$\begin{aligned} mwle_{m_1,m_2} \Rightarrow (mwg_{m_1,wg} + mwg_{m_2,wg} \leqslant 1) \\ \forall wg = 1...N_{wg}, m_1, m_2 = 1...N_m : m_2 > m_1 \end{aligned} \tag{4.7}$$

4.2.1.3 Activation of routing features

Constraints in sections 4.2.1.1 and 4.2.1.2 have forced the solver to choose a valid path through the PLT for each signal. These paths are described using the $mwg_{m,wg}$ variables. Now, constraints are required to configure each GRU accordingly.

The selected GRU configurations must fulfil the chosen signal paths (the two entrance/exit edges of each signal going through each GRU, described by the $mwg_{m,wg}$ variables) while at the same time adhering to the defined GRU structure and routing rules described in sections 4.1.2.1 and 4.1.2.2. If this is not possible, the solver will be forced to choose other signal paths.

Path fulfilment. Looking only at the two GRU edges used by a signal (variables $mwg_{m,wg}$ with $wg \in \{W_g^T, W_g^B, W_g^L, W_g^R\}$ for GRU g), there are two possible choices: either the signal uses two edges on opposite sides (i.e. top & bottom edges or left & right edges) or two edges on the same corner (i.e. top & left, top & right, bottom & left or bottom & right), as shown in fig. 4.6(a). This knowledge will be useful to force the use of certain required GRU structure elements depending on each signal's path.

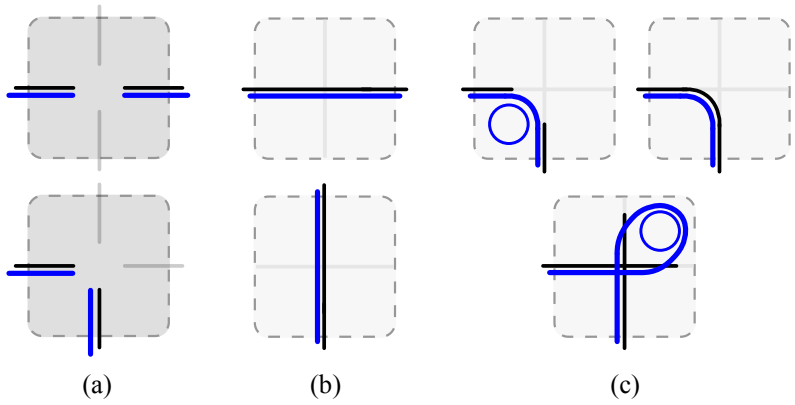

Figure 4.6: (a) Possible two-edge path choices in a GRU (only 2 of 6 choices shown). (b) Possible GRU configurations for signals that use two opposite edges. (c) Possible GRU configurations for signals that use two edges on the same corner (only one corner shown).

If a signal uses two opposite edges then the corresponding center crossing waveguide must be used, as shown in fig. 4.6(b). Thus, the following constraints are added:

$$mwg_{m,W_g^L} \wedge mwg_{m,W_g^R} \Rightarrow gcch_g \tag{4.8}$$

$$mwg_{m,W_g^T} \wedge mwg_{m,W_g^B} \Rightarrow gccv_g \tag{4.9}$$

$$\forall m = 1...N_m, g = 1...N_{gru}$$

If a signal uses two edges on the same corner, then one of three structure elements is required: the MRR on that corner, the corner bending waveguide on that corner or the MRR on the opposite corner, as shown in fig. 4.6(c). The following four constraints per GRU–signal pair are added (an example constraint for the bottom–left corner is given here):

$$\begin{aligned} &mwg_{m,W_g^B} \wedge mwg_{m,W_g^L} \\ &\quad \Rightarrow grm_{g,BL,m} \vee gcb_{g,BL} \vee grm_{g,TR,m} \\ &\forall \text{ 4 corners}, m = 1...N_m, g = 1...N_{gru} \end{aligned} \tag{4.10}$$

Note that if the signal goes through the MRR on the opposite corner then both center crossing waveguides must also be used (once again, an example constraint for the bottom–left corner is given here):

$$\begin{aligned} &mwg_{m,W_g^B} \wedge mwg_{m,W_g^L} \wedge grm_{g,TR,m} \\ &\quad \Rightarrow gcch_g \wedge gccv_g \\ &\forall \text{ 4 corners}, m = 1...N_m, g = 1...N_{gru} \end{aligned} \tag{4.11}$$

GRU structure compliance. The GRU structure rules 1–6 from sections 4.1.2.1 and 4.1.2.2 must be enforced. First, the values of $gr_{g,p}$ must be set based on the values of $grm_{g,p,m}$. The following constraints both set the value of $gr_{g,p}$ and guarantee that each MRR is only used by at most one signal:

$$gr_{g,p} = \sum_{m=1}^{N_m} grm_{g,p,m} \qquad \forall p \in \mathbb{P}, g = 1...N_{gru} \tag{4.12}$$

Rules 1–3 from section 4.1.2.1 are then enforced with the following sets of constraints:

1. Center crossing and corner bending waveguides are mutually exclusive:

$$gcch_g + gcb_{g,p} \leqslant 1 \tag{4.13}$$
$$gccv_g + gcb_{g,p} \leqslant 1 \tag{4.14}$$
$$\forall p \in \mathbb{P}, g = 1...N_{gru}$$

2. MRRs and corner bending waveguides are mutually exclusive per corner:

$$gr_{g,p} + gcb_{g,p} \leqslant 1 \qquad \forall p \in \mathbb{P}, g = 1...N_{gru} \tag{4.15}$$

3. Pairs of corner bending waveguides that connect to the same edge are mutually exclusive:

$$gcb_{g,TL} + gcb_{g,TR} \leqslant 1 \tag{4.16}$$
$$gcb_{g,TR} + gcb_{g,BR} \leqslant 1 \tag{4.17}$$
$$gcb_{g,TL} + gcb_{g,BL} \leqslant 1 \tag{4.18}$$
$$gcb_{g,BL} + gcb_{g,BR} \leqslant 1 \tag{4.19}$$
$$\forall g = 1...N_{gru}$$

Rules 5 and 6 are implicitly enforced by the path fulfilment constraints above and rule 4, which defines the placement of waveguide terminators, is irrelevant from an optimisation perspective and does not need to be considered in this model.

In later sections an analysis of the advantages and disadvantages of using corner bending waveguides will be presented. For those tests, corner bending waveguides will need to be eliminated from the model, which can be done by setting all $gcb_{g,p}$ variables to zero.

4.2.1.4 Insertion loss calculation

Crossing loss. A signal suffers crossing loss when going through a crossing with a perpendicular waveguide. Here, only crossing loss inside GRUs needs to be considered.[5]

[5] Crossing loss between two waveguide sections, i.e. outside of GRUs, is possible but is already taken into account with the *extraloss* parameter (L^E_{wg}) for waveguide sections. Note that this parameter depends only on the PLT and is thus constant during optimisation.

If a signal goes through two opposite edges of a GRU and the perpendicular center crossing waveguide is also used, then that signal suffers one instance of crossing loss on that GRU:

$$mwg_{m,W_g^T} \wedge mwg_{m,W_g^B} \wedge gcch_g \Rightarrow mlc_{g,m}^1 \tag{4.20}$$

$$mwg_{m,W_g^L} \wedge mwg_{m,W_g^R} \wedge gccv_g \Rightarrow mlc_{g,m}^1 \tag{4.21}$$

$$\forall m = 1...N_m, g = 1...N_{gru}$$

Additionally, if a signal uses two edges on the same corner but routes through the MRR on the opposite corner, then it suffers two instances of crossing loss (example constraint for the bottom–left corner given here):

$$mwg_{m,W_g^B} \wedge mwg_{m,W_g^L} \wedge grm_{g,TR,m} \Rightarrow mlc_{g,m}^2 \tag{4.22}$$

$$\forall \text{ 4 corners}, m = 1...N_m, g = 1...N_{gru}$$

Through loss. A signal has through loss if it goes through a center crossing waveguide on a GRU while the GRU has instantiated MRRs, with the signal being off-resonance with the MRRs:

$$mwg_{m,W_g^L} \wedge mwg_{m,W_g^R} \wedge gr_{g,p} \Rightarrow mlt_{g,p,m} \tag{4.23}$$

$$mwg_{m,W_g^T} \wedge mwg_{m,W_g^B} \wedge gr_{g,p} \Rightarrow mlt_{g,p,m} \tag{4.24}$$

$$\forall m = 1...N_m, p \in \mathbb{P}, g = 1...N_{gru}$$

Bending loss. A signal has bending loss[6] on a GRU if it routes through a corner that uses its corner bending waveguide (an example constraint for the bottom–left corner is given here):

$$mwg_{m,W_g^B} \wedge mwg_{m,W_g^L} \wedge gcb_{g,BL} \Rightarrow mlb_{g,m} \tag{4.25}$$

$$\forall \text{ 4 corners}, m = 1...N_m, g = 1...N_{gru}$$

Drop loss and propagation loss. Drop loss of a signal is proportional to the number of MRRs used by that signal (variables $grm_{g,p,m}$) and propagation loss of a signal is proportional to the length of the waveguides the signal goes through.

Signal insertion loss. The total insertion loss of a signal over all waveguides and GRUs is given by the following weighted sum of propagation, crossing, bending, through

[6] Similar to crossing loss, bending outside of GRUs on waveguide sections is also possible and already taken into account with the *extraloss* parameter (L_{wg}^E) for waveguide sections.

and drop loss:

$$\begin{aligned} mil_m = &\sum_{i=1}^{N_{wg}} (L^P L_i + L_i^E) * mwg_{m,i} \\ &+ \sum_{g=1}^{N_{gru}} (L^C mlc_{g,m}^1 + 2L^C mlc_{g,m}^2 + L^B mlb_{g,m}) \\ &+ \sum_{g=1}^{N_{gru}} \sum_{p \in \mathbb{P}} (L^T mlt_{g,p,m} + L^D grm_{g,p,m}) \\ &\forall m = 1...N_m \end{aligned} \tag{4.26}$$

4.2.2 Objective function

As explained above, three optimisation objectives are considered: number of symbolic wavelengths, signal insertion loss and number of MRRs. For signal insertion loss, both the total and maximum over all signals is considered.

Calculating the number of wavelengths is done with the following set of constraints:

$$wlu_\lambda \geqslant mwl_{m,\lambda} \qquad \forall m = 1...N_m, \lambda = 1...N_\lambda \tag{4.27}$$

$$nwl = \sum_{\lambda=1}^{N_\lambda} wlu_\lambda \tag{4.28}$$

Determining the maximum insertion loss over all signals is done with the following set of constraints:

$$maxil \geqslant mil_m \qquad \forall m = 1...N_m \tag{4.29}$$

Finally, the MILP optimisation problem is formulated as follows:

Minimize

$$\alpha_1 * nwl + \alpha_2 * maxil + \alpha_3 * \sum_{m=1}^{N_m} mil_m + \alpha_4 * \sum_{g=1}^{N_{gru}} \sum_{p \in \mathbb{P}} gr_{g,p} \tag{4.30}$$

Subject to (4.1)–(4.29)

where α_i are optimisation weights chosen by the designer.

Since the value for the insertion loss of each signal is available through the mil_m variables, functions other than the total/maximum of the insertion loss can also be added to the model and used for optimisation assuming they are linear or linearisable (see section 5.3).

4.2.3 Proof of feasibility

It is possible that the chosen PLT cannot satisfy the entire CM (for example, if the PLT is too small and the CM is dense). In these cases, the PLT is said to be "saturated" and the model above will be unfeasible. Verifying the existence of a solution can be done much faster using a simplified version of the model. In this case, consider $N_\lambda = N_m$ and uniquely assign a wavelength to each signal by adding these constraints:

$$mwl_{m,\lambda} = 1 \qquad \forall m = 1...N_m, \lambda = m \tag{4.31}$$

$$mwl_{m,\lambda} = 0 \qquad \forall m = 1...N_m, \lambda \neq m \tag{4.32}$$

The resulting model can be solved much faster and if the solver is unable to find a feasible solution for this simplified model, the complete model is also unfeasible.

Proof. Assume a feasible solution exists. It will have $nwl \leqslant N_m$. From that solution build another where each signal uses its own wavelength (thus either maintaining or increasing nwl). Any signal that changes its wavelength must also change the wavelength of the MRRs it uses. This is always possible because each MRR routes only one signal. Furthermore, the optical exclusion principle is always satisfied. Hence, the feasibility of the complete model implies the existence of a solution for the simplified version. In conclusion, if the simplified model is unfeasible, the complete model is also unfeasible. □

4.3 Grid Template design

This section introduces and analyses a simple PLT structure that can be used in virtually any WRONoC problem while requiring practically no synthesis effort. This simple PLT is called the GT and is shown in fig. 4.1. It has GRUs arranged in a grid pattern connected by waveguide sections. Waveguide sections on the sides of the grid are arranged in pairs, called terminals, with each terminal connected to one sending and one receiving endpoint. Note that sending endpoints are always placed opposite receiving endpoints and *vice-versa*. Each WRONoC node is assigned one terminal and the endpoints connected to that terminal are placed on the node's location on the optical routing layer.

In general, the grid of GRUs can be distributed throughout the optical routing layer, Distributed Grid Template (DGT), which raises the question of what location is optimal for each GRU. A simpler option which mostly evades this issue is packing the GRUs as closely as possible and then placing the entire grid on one location, leading to the Centralised Grid Template (CGT). This location can still be optimised, but simple yet sensible locations for centralised grids are the center of mass of the nodes or just the center of the routing layer.

Another design choice with GTs is terminal assignment. This can have a measurable impact on the solution because it influences signal paths through the grid, which in turn influences wavelength usage and insertion loss. Without deeper analysis, however, it is difficult to predict the best assignment. Nonetheless, having decided on a position for the CGT on the routing layer, there will ordinarily exist one simple assignment of terminals to nodes where no crossings external to the grid are created and which also has the advantages explained in section 4.3.1. Since crossing loss strongly influences overall power usage, the procedure which is in equal parts simple and effective is just to use that assignment.

Finally, the waveguides connecting the endpoints to the GRUs can be manually routed to minimise their length and number of bends (external waveguide routing).

4.3.1 PDN awareness

When using off-chip laser sources, a PDN is required. Importantly, crossings between the PDN waveguides and the router waveguides will significantly increase the minimum optical power required [24]. An easy way to avoid these crossings is to split the routing layer into two areas, the "router area" and the "PDN area", such that a path free from crossings from each MA to the optical coupler is always available, as shown in fig. 4.7. While most P&R tools presented thus far do not guarantee zero crossings with the PDN [9, 62, 63], CGTs can always easily be designed to be confined to the "router area", thus having no crossings with the PDN.

A proof of the previous statement follows. Consider a graph where each node, containing one sending endpoint (i.e. MA) and one receiving endpoint (i.e. DMA), is a vertex. Add also one vertex for the optical power source and one vertex for the CGT. Naturally, the CGT vertex must be connected to each sending and receiving endpoint and the optical power source vertex must be connected to each sending endpoint. Each edge of the graph is thus either a waveguide belonging to the PDN or a waveguide section external to the CGT belonging to the router. The requirement that waveguides do not cross outside the CGT is equivalent to stating that the constructed graph is planar. As seen on fig. 4.7(b), the constructed graph is clearly planar.[7]

4.3.2 Types of signal paths on GTs

The regular structure of GTs allows for an *a priori* analysis of the optimal paths signals will likely take through the grid. If corner bending waveguides are turned off (these are not always necessary and this significantly reduces runtime, as explained in section 4.5.2), signals are prone to following one of three path types:

[7] If any node contains more than one sending or receiving endpoint then more vertices can be added but planarity is still guaranteed.

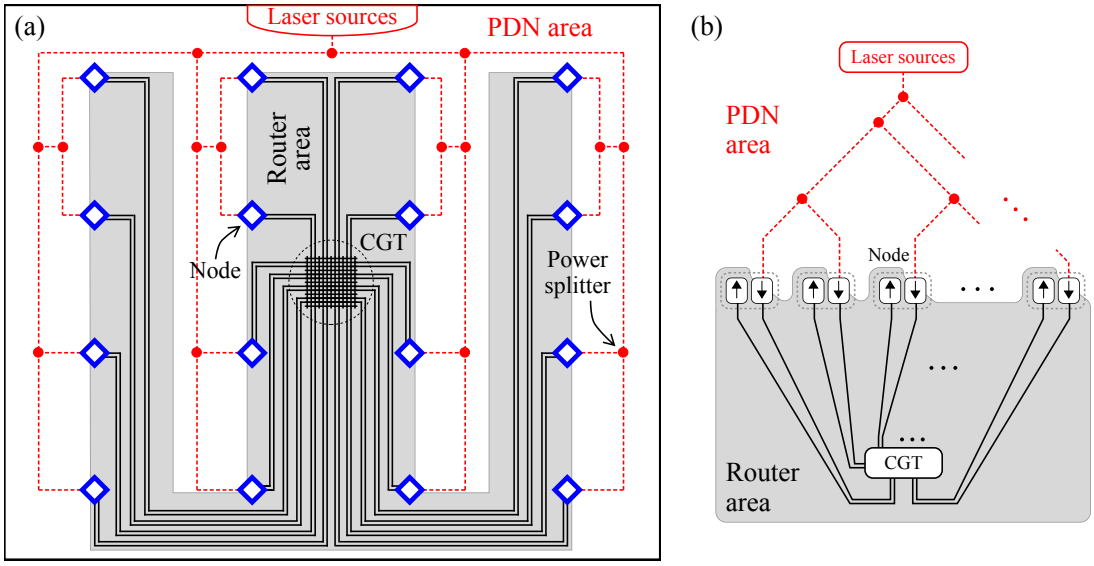

Figure 4.7: Breaking the inter-dependency between the PDN and the router by dividing the routing layer into a "PDN area" where only the PDN is placed, and a "router area" where only the CGT and external waveguide sections are placed. (a) Example with a grid of 16 nodes from [1]. (b) General topology of a PDN and a router using a CGT with no waveguide crossings outside the CGT.

0-MRR paths. Signals whose entrance and exit terminals are directly aligned are very likely to take the direct path and use 0 MRRs (blue path in fig. 4.8).

1-MRR paths. Signals whose entrance and exit terminals are on perpendicular sides of the grid are very likely to use 1 MRR (red path in fig. 4.8).

2-MRR paths. Other signals have multiple 2-MRR paths available (green paths in fig. 4.8).

Other paths are possible but they must use more MRRs. In fact, this increase in MRRs must always be in multiples of two (for example, if a signal that can take a 1-MRR path instead follows a path with more than 1 MRR, then that path has 3, 5, 7... MRRs). This drastically increases insertion loss. Thus, paths with more than 2 MRRs are unlikely to feature in optimised solutions. This fact will be exploited by heuristics presented in section 4.4.

If corner bending waveguides are allowed then these 0/1/2-MRR paths, while still valid, are less likely to feature in an optimised solution for sparse CMs (see section 4.5.2).

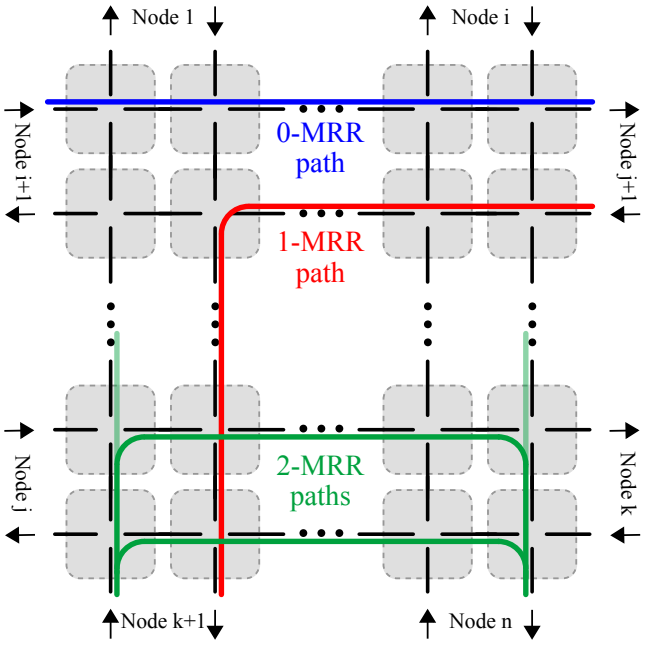

Figure 4.8: Types of paths through a GT.

4.3.3 Open problems in GT design

As stated in section 4.2.3, PLTs may become saturated with dense CMs, and GTs are no exception.

Consider a GT with width w and height h in terminals (pairs of endpoints). The maximum number of nodes that can connect to the grid is $N = 2w + 2h$ (one per terminal) and the number of GRUs in the grid is $G = 2w * 2h = 4wh$. The total number of available MRRs is $R_a = G * 4 = 16wh$.

As explained in section 4.3.2, signals will tend to use the 0/1/2-MRR paths. More importantly, any other path would use more MRRs. Thus, the minimum number of MRRs required to support these paths with a full CM without self-communication will now be calculated. Such a matrix contains $M = N * (N - 1)$ signals, where:

- $2w + 2h$ signals use 0 MRRs
- $8wh$ signals use 1 MRR
- $4w(w - 1) + 4h(h - 1)$ signals use 2 MRRs

Multiplying each expression by the corresponding number of used MRRs yields the minimum number of MRRs required to route M signals, which is $R_r = 8(w^2 + h^2 + wh - w - h)$. The condition $R_a \geqslant R_r$ must be true for a GT to support M signals, but unfortunately it is not true for all values of w and h. For example, a 2×2 grid (8

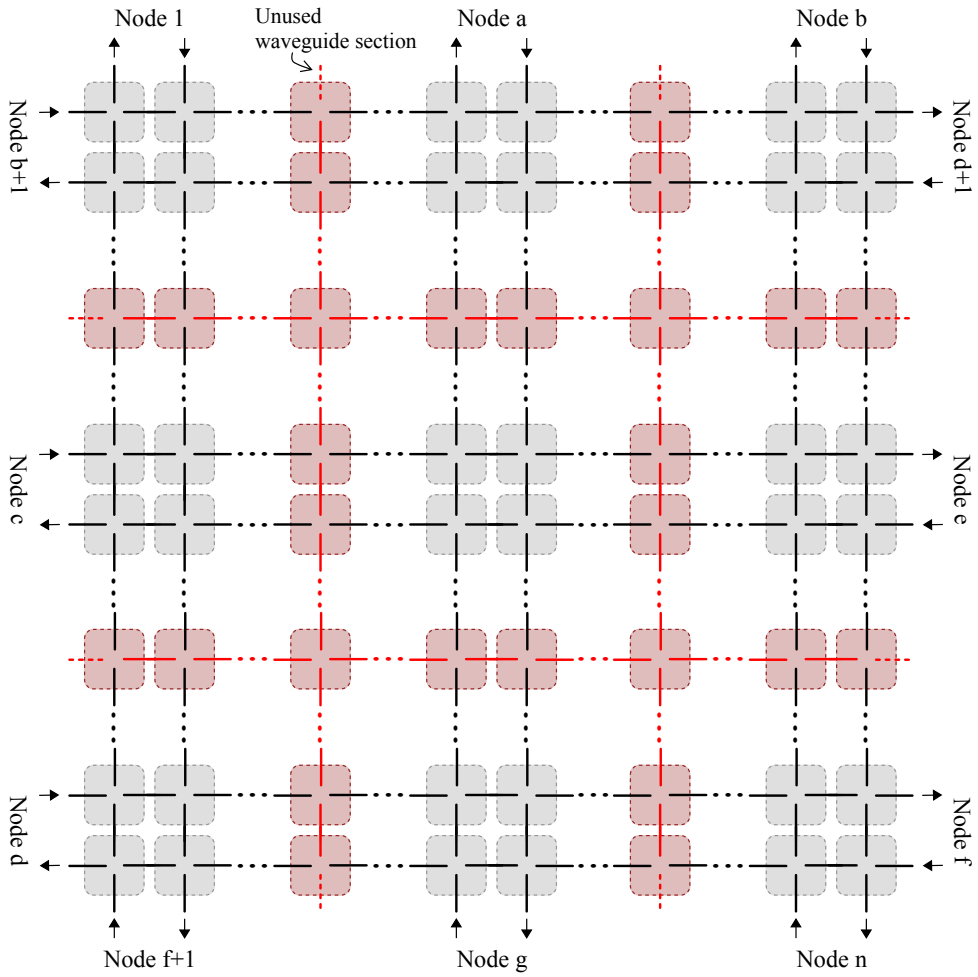

Figure 4.9: Expanding a GT by adding extra sets of waveguide sections and GRUs (pictured in red).

nodes) supports all $8 * 7 = 56$ signals, but a 3×3 grid (12 nodes) will not support all $12 * 11 = 132$ signals. Corner bending does not change this result because it is never used in PLTs close to saturation.

To add more MRRs, more GRUs must be added, which is possible by adding extra sets of horizontal and/or vertical waveguide sections to the grid as shown in fig. 4.9. The open question then becomes: how many extra sets to add, and where on the grid to add them.

An expanded grid will surely satisfy $R_a \geqslant R_r$ but each extra set may significantly increase runtime, so they should be added cautiously. Additionally, extra sets can be beneficial even when $R_a \geqslant R_r$ is already true, since they can decrease the minimum

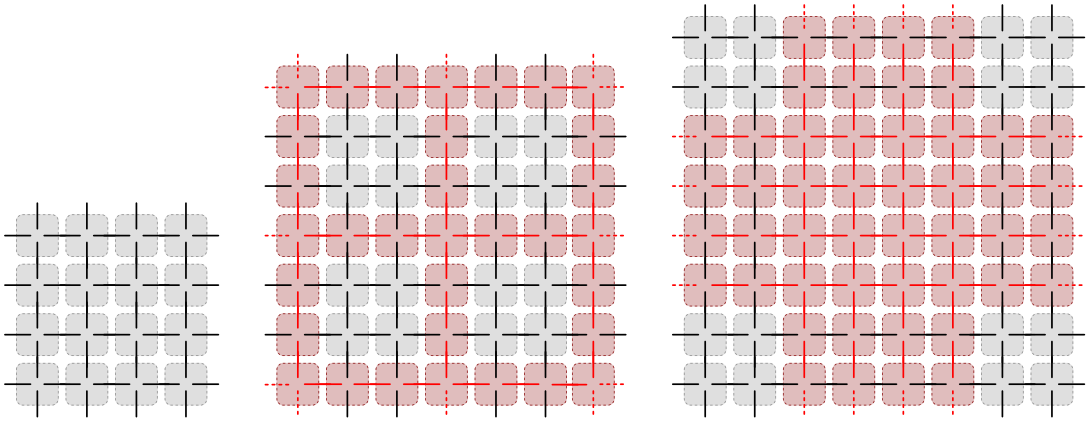

Figure 4.10: Comparison between possible expanded grid designs for an 8 node GT. #WL is the minimum amount of wavelengths required to support all 56 signals and *i. loss* is maximum insertion loss. In this case, (b) is clearly a better option than (c) even though it uses fewer extra sets.

required number of wavelengths (see fig. 4.10). In both cases, their position also contributes to the quality of the final result.

Nevertheless, in many cases it is acceptable to ignore these concerns. PSION+ performs better in application specific design (see section 4.5.2) where the number of signals rarely reaches M. Here, GTs are much less likely to be saturated thus making corner bending useful and lowering the benefits of adding extra waveguide sets.

4.4 Heuristics and model reduction techniques

The MILP method outlined in this chapter produces optimal solutions, which makes the solving process inherently slow. Thus, multiple heuristics were developed that speed up this process and allow targeting of larger WRONoCs with full CMs. These will now be presented.

4.4.1 A: Restrictions on usage of wavelengths

The following constraints can be added to the model:

$$mwl_{m,\lambda} = 0 \qquad \forall \lambda = (m+1)...N_\lambda, m = 1...N_m \tag{4.33}$$

Their effect is to restrict the possible wavelengths for each signal: signal 1 uses wavelength 1, signal 2 uses wavelengths 1 or 2, etc. This way, some meaningless variations

around the same practical solution are removed. The optimal solution, however, is never removed from the search space.

4.4.2 B: Restrictions on usage of MRRs (R^{max})

When minimising insertion loss, each signal in an optimal solution will tend to use a low number of MRRs. Yet, the solver will still look at complex paths with many MRRs each during its optimisation process. To avoid this wasted effort, constraints can be added to the model that force a maximum number of MRRs per signal (R^{max}):

$$\sum_{g=1}^{N_{gru}} \sum_{p \in \mathbb{P}} grm_{g,p,m} \leqslant R^{max} \qquad \forall m = 1...N_m \tag{4.34}$$

This forces the solver to only consider simpler paths right from the onset and so leads to noticeably lower runtime. Corner bending also contributes to path complexity but, unlike MRRs, corner bending should not be restricted because bends are much cheaper in terms of insertion loss. Hence, it can be observed that most optimised solutions will take advantage of corner bending by making signals snake through the PLT in unexpected ways.

The choice of R^{max} is paramount in defining the usefulness of this heuristic. Too big and this heuristic has little impact, but too small and the optimal solution might be removed or even make the model unfeasible. This choice needs to be done through specific PLT analysis. For example, given the analysis in section 4.3.2, $R^{max} = 2$ turns out to be an attractive option for GTs.

4.4.3 C: Path assignment

As explained in section 4.1.4, one of the optimisation process duties is signal routing. Thus, knowing *a priori* the paths that certain signals will take can be highly beneficial in reducing runtime. In the case of GTs, the paths of all 0-MRR and 1-MRR signals can be predicted with reasonable confidence[8] when corner bending waveguides are turned off. This knowledge can be utilised effectively by adding extra constraints that fix the values of variables $mwg_{m,wg}$ for 0/1-MRR signals according to their path.

The effectiveness of this heuristic depends on how correct the path assumptions are and how high the proportion of signals with well-defined paths to total signals is. It is shown below that GTs succeed in both criteria, leading to substantial runtime reductions without practically any penalty in solution quality.

[8] 2-MRR signals have multiple path choices, so no clear path information is available *a priori*.

Algorithm 1 Wavelength assignment heuristic

Input: predicted signal paths p, complete waveguide set w
Output: assigned signal wavelengths

```
λ ← 1
loop
    find one exact cover of w using paths p
    if no cover found then
        end algorithm
    end if
    for signal in cover do
        signal wavelength ← λ
        remove signal from p
    end for
    λ ← λ + 1
end loop
```

4.4.4 D: Wavelength assignment

Similar to signal routing, wavelength assignment is another major source of problem complexity. The challenge with this assignment is in finding the smallest[9] packing of signals into wavelengths such that the paths of signals with the same wavelength do not overlap, i.e. use the same waveguide sections.

Let $\mathbb{M}$ be the set of all signals and $\mathbb{W}$ the set of all waveguide sections in the PLT. Then let $\mathbb{M}^*$ be a subset of $\mathbb{M}$ and $p(\mathbb{M}^*)$ be the collection of subsets of $\mathbb{W}$ which contain the waveguide sections used by signals in $\mathbb{M}^*$. This heuristic is based on the following observation: if $p(\mathbb{M}^*)$ is an exact cover of $\mathbb{W}$, i.e. if all elements in $\mathbb{W}$ are present exactly once in $p(\mathbb{M}^*)$, then signals in $\mathbb{M}^*$ can all be assigned one wavelength λ_x without increasing the minimum possible number of used wavelengths. In doing this the initial assignment problem $(\mathbb{M}, \mathbb{W}, p)$ is reduced into a new smaller problem $(\mathbb{M} \setminus \mathbb{M}^*, \mathbb{W}, p)$. Thus, the heuristic works as explained in algorithm 1 (to find exact covers Algorithm DLX is used [71])[10] and is run before solving the MILP model.

Note that while this heuristic guarantees wavelength optimality, it does not necessarily keep optimality in other objectives such as insertion loss. Also, the effectiveness of this heuristic depends once again on the same two conditions as the path assignment heuristic in section 4.4.3.

[9] This is assuming the number of wavelengths is part of the optimisation function, which is almost certain. Otherwise, wavelength assignment is not an optimisation problem but a feasibility one, which can be solved extremely fast using the feasibility proof from section 4.2.3.

[10] A small exception must be considered here: extra waveguide sets added to GTs should not be considered in the exact cover since they are never used by 0/1-MRR signals.

4.4.5 E: 3-step optimisation

Solving the presented MILP model once for the required optimisation function is enough to get the optimal solution. However, due to the nature of the problem, it is possible to slightly alter the optimisation process yielding more control and faster results. This leads to the 3-step optimisation process proposed below. In this process each step optimises a slightly different version of the model and produces a solution used at the start of the next step.

In the first step set $N_\lambda = N_m$ and apply the feasibility proof from section 4.2.3. In this way the first feasible solution can be generated much faster if one exists and then used to warm start the optimisation. This has the added bonus of stopping the process as quickly as possible if unfeasible.

In the second step only the number of wavelengths (nwl) is minimised, for two reasons. Firstly, the designer will almost certainly want to use fewer wavelengths than signals. Hence, even if nwl is not the main minimisation goal, the number of used wavelengths should still be partly reduced. Secondly, because, after completing this step, a feasible solution for a smaller number of wavelengths is available, so the model can again be simplified by eliminating from it the $N_m - nwl$ unused wavelengths. In this step finding a feasible solution with a reduced number of wavelengths is enough (as opposed to finding the minimum nwl). Thus, this optimisation procedure can be stopped once a solution with a reasonably small nwl has been found.

In the third step first the model is simplified by removing unused wavelengths with the following constraints:

$$mwl_{m,\lambda} = 0 \quad \forall m = 1...N_m, \text{ unused wavelengths } \lambda \tag{4.35}$$

Then the model is solved to optimality for the optimisation function chosen by the designer, reaching the final solution.

Using this process it is possible to noticeably simplify the search space during the optimisation. Moreover, results show that the partial optimisation of nwl in the second step can be subtle enough to not remove the optimal solution from the optimisation in the third step while still being aggressive enough to reduce total runtime (see section 4.5.2).

4.5 Results

The MILP model and accompanying heuristics were programmed in C++ and make use of Gurobi, an MILP solver [72].

4.5.1 Comparison to state of the art P&R tools

The presented model and optimisation procedure was tested against the state of the art PROTON+ [9] and PlanarONoC [63] P&R tools. Most of their result analysis

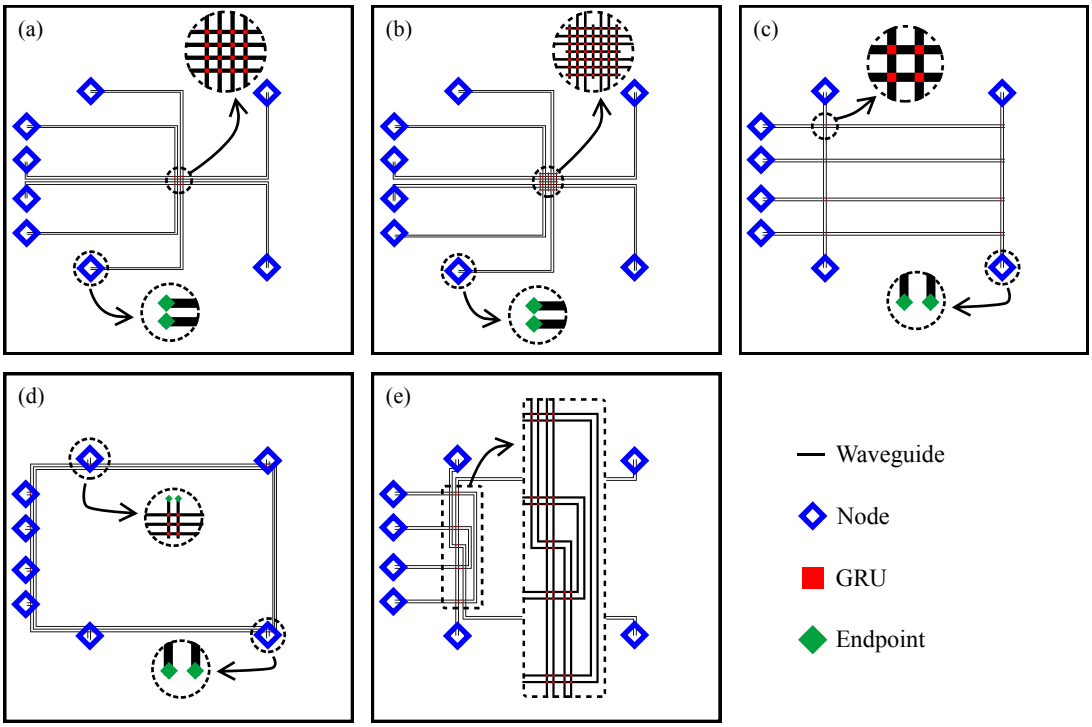

Figure 4.11: (a) A non-expanded Centralised Grid Template (CGT-e0) connecting the nodes on the positions that produce the best result in PROTON+. (b) An expanded CGT with 6 extra waveguide section sets (CGT-e6). (c) A Distributed Grid Template (DGT). (d) A ring template with 3 rings. (e) A custom template.

is dedicated to an 8 node test case where four nodes are clusters of processors and four nodes are memory controllers for off-chip memory. The clusters of processors communicate with each other bidirectionally ($4 \times 3 = 12$ signals) and each cluster communicates with all memory controllers bidirectionally ($4 \times 4 \times 2 = 32$ signals). Memory controllers do not communicate between themselves, thus lowering the number of signals from a full CM (56 signals) to only $12 + 32 = 44$ signals. This test case was solved with this CM and the same optical routing layer size, crossing size, loss parameters and optimisation function (maximum insertion loss).

The node positions that produced the best result over all presented in PROTON+ were used. PROTON+ and PlanarONoC both employ three standard logical topologies (λ-Router, GWOR and Standard $\times$bar), whereas five simple PLTs were manually designed for PSION+, presented in fig. 4.11(a-e), that connect to those node positions. No effort was spent in trying to optimise the design of the GTs: the grids of the CGTs were placed in the middle of the routing layer, the positions of the GRUs in the DGT follow directly from the node positions and the most straightforward terminal assignments were used. The ring template has enough rings (three) to achieve

Table 4.3: Results for 8 nodes, 44 signals. #WLs, #MRRs: number of wavelengths and MRRs. Max IL: maximum insertion loss. Runtime in seconds, insertion loss in dB. PSION+ results are optimal for the given PLTs.

		#WLs	**Max IL**	#MRRs	Runtime
PROTON+	λ-Router	**8**	**6.6 - 9.0**	56	134
	GWOR	**7**	**8.1 - 11.3**	48	79
	Std. $\times$bar	**8**	**10.5 - 13.0**	64	602
PlanarONoC	λ-Router	**8**	**5.2**	56	<1
	GWOR	**7**	**6.4**	48	<1
	Std. $\times$bar	**8**	**7.4**	64	<1
PSION+	CGT-e0	**8**	**3.1**	52	13
	CGT-e6	**7**	**3.7**	52	75
	DGT	**8**	**3.6**	48	3
	Ring	**7**	**3.1**	88	31347
	Custom	**7**	**4.1**	40	<1

the minimum number of wavelengths required by this test case (seven). The custom template was built specifically for this test case so that no signal needs to use more than one MRR. Therefore, R^{max} was set to 1 for this template while the GTs were solved with $R^{max} = 2$. All templates use one MA/DMA pair per node for consistency with the state of the art designs. All heuristics from section 4.4 were used.

Table 4.3 presents the various comparisons. On average, results from PSION+ are either equal or better except in comparison to PlanarONoC's runtime. Using PSION+ and without any PLT synthesis effort (with CGT-e0) it is possible to reduce insertion loss by 52% and 41% compared to Proton+ and PlanarONoC respectively in the same time frame. Also, this difference will be increased when adding a PDN since Proton+ and PlanarONoC do not guarantee zero crossings between router and PDN but all five PLTs used here do. By spending slightly more effort in designing good PLTs it is possible to improve upon CGT-e0 in some areas: CGT-e6 and Custom use fewer wavelengths and DGT and Custom both use fewer MRRs and are faster to solve. The CGT-e6 already achieves the minimum possible number of wavelengths, i.e. 7, so CGTs was not tested with more extra sets of waveguide sections as that would increase insertion loss but bring no further advantage.

The ring template is an exception in some areas. It achieves positive results in number of wavelengths and insertion loss, but it takes much longer to solve. This is because it is more complex (more GRUs and waveguide sections), it does not take advantage of the heuristics from sections 4.4.3 and 4.4.4 and its combinatorial complexity is further increased because the total number of possible signal paths is multiple orders of

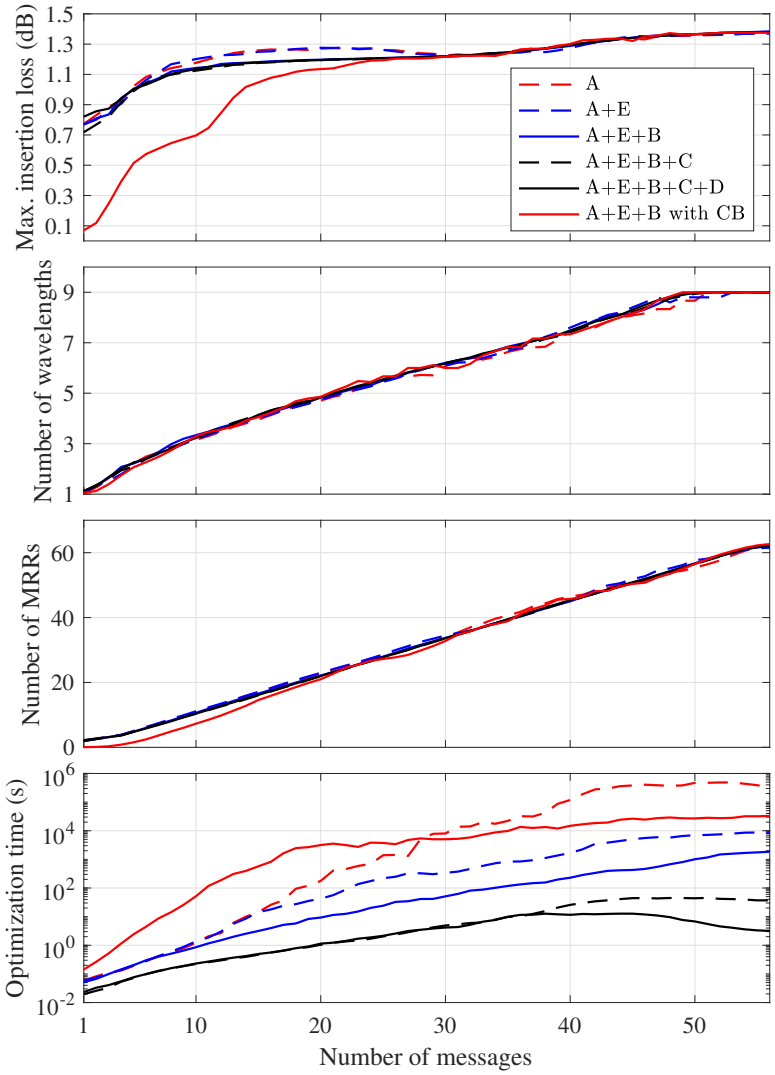

Figure 4.12: Results for maximum insertion loss, #WLs, #MRRs and runtime for 1–56 signals on an 8-node GT with multiple solver configurations. Curves *A+E+...* use the corresponding heuristics from section 4.4.1, section 4.4.5, etc, with corner bending waveguides turned off. Curve *A+E+B with CB* uses the corresponding three heuristics with corner bending waveguides turned on.

magnitude higher than with GTs. It also uses more MRRs because all signals require 2 MRRs: one to drop into the ring and another to drop out of the ring.

Note that $1 * maxil$ was chosen as the optimisation function for all PLTs to ensure fairness in the comparison, but a reduction in the usage of MRRs was still obtained when possible. Each MRR has a potential contribution of $1\times$ drop or through loss to the maximum insertion loss, so minimising *maxil* indirectly optimises the number of MRRs. Thus, there is little need for minimising the number of MRRs directly and so

the recommendation is to use $1 * maxil$ as the optimisation function[11] for the third step for any PLT.

Finally, a simple analysis of crosstalk is presented. The major sources of crosstalk are waveguide crossings and MRRs. PROTON+ designs have between 27 and 105 crossings and PlanarONoC designs have between 24 and 64 crossings (these values do not include extra crossings with the PDN). Conversely, CGT-e0 and CGT-e6 templates have 16 and 23 crossings respectively (including the PDN, which contributes zero crossings). Additionally, CGT designs have equivalent numbers of MRRs. Thus, they should not present higher crosstalk than the state of the art.

4.5.2 Grid Template analysis and heuristic techniques

The purpose of this section is twofold:

- To assess the quality of the results possible with GTs, more specifically, how they are affected by the density of the CM and the use of corner bending waveguides.
- To analyse how each of the proposed heuristics impacts solution quality and runtime on GTs.

Hence, a non-expanded GT of size 2×2 (8 nodes) was solved[12] for random CMs with 1 to 56 signals in total and the relevant results were recorded: maximum insertion loss, number of wavelengths (#WLs), number of MRRs (#MRRs) and runtime. Since all four results depend on the exact set of signals chosen, multiple tests with random CMs were done for each number of signals and their results averaged. All tests use the technology parameters from [2] and were solved to optimality.

To test the impact of each heuristic, the described sequence of random tests from 1 to 56 signals was completed five times. The first time, only the model reduction from section 4.4.1 was used, which gives the baseline, i.e. the provably optimal solution curves. Then, each of the four subsequent sequence runs increased the number of used heuristics by one.

Runs using the heuristic from section 4.4.5 minimized nwl and $100 * nwl + 1 * maxil$ on the second and third steps respectively whereas runs without it solved the model only once with $100 * nwl + 1 * maxil$ as the minimisation function. Corner bending was turned off for these tests since heuristics in sections 4.4.3 and 4.4.4 are incompatible with it. If all heuristics are successful, the quality of the results should remain constant while runtime decreases.

[11] Or $100 * nwl + 1 * maxil$ in the case the minimum number of wavelengths was not achieved during the second step.

[12] Here the endpoints are placed next to the grid since the purpose of these tests is to analyse the performance of the routing through the grid, not how node positioning, GRU positioning or external routing affects insertion loss.

To test the usefulness of corner bending waveguides, the sequence of random tests was run a sixth time with all heuristics except those from sections 4.4.3 and 4.4.4. The corresponding curves should be compared against the sequence run without corner bending that also used the same heuristics. An example of a GT optimised for a sparse CM using corner bending is shown in fig. A.2.

Figure 4.12 presents the results of all 6 sequence runs. The following conclusions can be drawn:

- There is a linear relationship between the number of signals and both #MRRs and #WLs. Thus, conventional logical topologies are clearly sub-optimal when used with non-complete CMs, since reducing #MRRs and #WLs helps to improve all three performance factors.

- Runs without the heuristic from section 4.4.2 show a slight increase in maximum insertion loss on average. It can observed that removing the R^{max} restriction enables the substitution of some 1-MRR paths by 3-MRR paths on some cases. This is done by the solver to reduce #WLs.[13] However, no appreciable difference can be seen in the #WLs or #MRRs since this happens infrequently. Thus, the use of this heuristic is still recommended given its remarkable impact on runtime.

- In general, the proposed heuristics lower runtime without appreciable reductions in result quality. In particular, the path and wavelength assignment heuristics can be used safely when corner bending is turned off. On average, 3-step optimisation reduces runtime by 96%, $R^{max} = 2$ further reduces runtime by 82%, path assignment further reduces runtime by 91% and wavelength assignment further reduces runtime by 60%. In total, using all heuristics is 18463× faster. This amount of reduction is essential for tackling larger WRONoC designs. Wavelength assignment only produces measurable reductions in runtime for denser CMs because the probability of finding exact covers is low when there are fewer signals.

- Using corner bending waveguides in a GT can substantially decrease maximum insertion loss and marginally decrease #MRRs for sparse CMs (for 8 nodes, up to 25 signals or 45% of a full CM) at the cost of being noticeably slower. However, results show corner bending is not used with dense CMs. This is because dense CMs have many signals to be routed thus requiring more MRRs per GRU than what corner bending allows. Therefore, a good guideline is to use corner bending only for sparse CMs (below 45%), turning it off and using all heuristics otherwise.

Note that the runtime results shown here are for obtaining optimal solutions. This explains their exponential growth with the increase in problem size (number of signals).

[13] This is because the optimisation function $100 * nwl + 1 * maxil$ used here prioritises the minimisation of #WLs over insertion loss.

If a designer is willing to forgo proof of optimality, good enough solutions can be obtained faster. For example, on average it is possible to obtain a solution with quality within 90% of the optimal solution in only 10% of the total optimisation runtime (the remaining 90% runtime is used to achieve and prove optimality).

4.5.3 Full power consumption comparison for 16 node WRONoC

With the help of the heuristic methods shown here, PSION+ was compared against the previously reported manual design of a 16-node WRONoC laid out on a 16 mm×16 mm optical routing layer [1].

This comparison was performed with the total static power consumption of the WRONoC, which is the sum of the laser power, the static MRR thermal tuning power and the static modulator and demodulator power. The two designs with the lowest total static power consumption in [1] are a centralised λ-Router topology and a centralised Snake topology (*λ-Router SB* and *Snake SB* where *SB* stands for *Single Box*).

The CGT design philosophy explained in section 4.3 was applied to this test case. First the optical routing layer was divided into a PDN area and a router area such that the PDN area contained the same PDN used in the SB designs. Then a CGT-e10 was placed on the router area on two locations: the center (Center CGT-e10) and the mid-bottom (Bottom CGT-e10) of the routing layer. Next, an assignment of CGT terminals to nodes was chosen that does not produce any external router waveguide crossings. Finally, the external router waveguides were manually routed. The resulting Center CGT-e10 design can be seen in fig. 4.7(a). The Bottom CGT-e10 has the CGT itself moved 4 mm down.

The Center CGT-e10 and both SB designs are placed on the same location on the routing layer, making them directly comparable. The Bottom CGT-e10 was also tested because it better balances the length of the PDN waveguides with the length of the external router waveguides compared to the Center CGT-e10, which could potentially lead to lower laser power consumption. In both cases the CGT-e10 contains 5 extra horizontal and 5 extra vertical waveguide section sets to ensure design feasibility.

For this comparison the same routing layer size, node positions, PDN design, technology parameters, static thermal tuning power per MRR, static power per modulator and demodulator, crossing size and CM were used. Here the CM is full ($16 \times 15 = 240$ signals), so corner bending was turned off and all heuristics were used (according to the conclusions drawn in section 4.5.2). On the third optimisation step the model was optimised for *maxil*. The results are given in table 4.4. A higher number of wavelengths and a higher number of MRRs were required, which lead to higher total P_{md} and P_{mrr} values. However:

- Total static power consumption (P_{total}) was reduced by 4.2% to 6.9%.

Table 4.4: Power consumption comparison to [1]. #WLs, #MRRs: number of wavelengths and MRRs. $P_{laser:el}$ = total laser power (router + PDN). P_{mrr} = total static MRR thermal tuning power. $P_{md} = P_{mod} + P_{demod}$ = total static modulator + demodulator power. $P_{total} = \sum P_i$. Power values in mW.

	#WLs	#MRRs	$P_{laser:el}$	P_{md}	P_{mrr}	$\boldsymbol{P_{total}}$
[1]						
λ-Router SB	15	240	76	19.4	14.4	**109.8**
Snake SB	15	240	78	19.4	14.4	**111.8**
PSION+						
Center CGT-e10	17	320	65.6	20.4	19.2	**105.2**
Bottom CGT-e10	17	320	64.4	20.4	19.2	**104.0**

- The design of both CGT-e10 routers required little manual effort, whereas the λ-Router and Snake designs are complex and were manually conceived.
- This test case deals with a full CM. While the results from [1] do not change with sparser CMs, it is clear from section 4.5.2 that PSION+ can achieve substantially better results with sparser CMs. If this test case was for an application specific design with fewer signals, PSION+ would achieve designs with even smaller total static power consumption.

This automated optimisation is a step forward with respect to current error-prone manual design frameworks [1] and took less than a week to run.

5 Optimising physical layout for laser power[1]

The previous chapter introduced the PLT approach and the PSION/PSION+ tools which make use of an LP model to perform PLT optimisation. This chapter will now build on this approach by extending the LP model with the components required to perform, among other things, full power consumption optimisation. The resulting improved tool is called PSION 2.

While optical power splitters can be configured with any splitting ratio in theory, in practice, unbalanced ratios pose fabrication problems and are therefore uncommon [1,9]. The 50% ratio is the most reliable from a fabrication standpoint and is consequently the most widely used. It is also the most challenging to design for, from a power consumption optimisation perspective. In order to ensure this work is independent of fabrication capabilities and is relevant in all scenarios, it is restricted to using only 50% splitting ratios.

Here it is also assumed for simplicity that each node contains only one MA and, if on-chip laser sources are used, that only one laser source of multiple wavelengths exists per node. A more general case is straightforward to develop based on this work.

Section 5.1 describes in detail how the PDN works and section 5.2 explains the motivation behind PSION 2. Its methodology is detailed further in sections 5.3 and 5.4. Finally, section 5.5 compares PSION 2 against PSION+.

5.1 Power Distribution Network

A PDN is required when using off-chip laser sources. A conceptual PDN example powering four WRONoC nodes is shown in fig. 5.1 and the following notation is used (from the bottom to the top):

- $il^N_{n,\lambda}$ — Insertion loss at MA n for wavelength λ. This includes the insertion loss caused by the modulator itself, plus the insertion loss caused by the router (from the modulator to the demodulator, excluding both), plus the insertion loss caused by the demodulator. Router insertion loss depends on the exact design of the router whereas (de)modulator losses are only technology dependent constants.

[1] Segments of this chapter were previously published in [7].

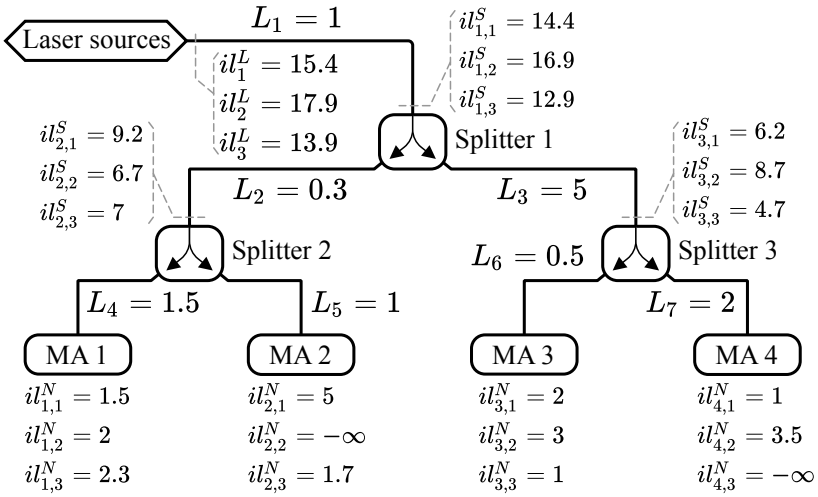

Figure 5.1: Example of a PDN distributing three wavelengths to four MAs of a WRONoC. Each MA must receive enough power to overcome the insertion loss values caused by the WRONoC router, $il^N_{n,\lambda}$. Splitters have a 50% splitting ratio. The extra insertion loss caused by each edge (PDN waveguide) of the tree, L_i, is also shown.

- L_e — Insertion loss caused by PDN waveguide (edge of the PDN tree) e of the PDN. This loss is created by a combination of propagation, bending and crossing losses in the waveguide and depends on the exact routing of the waveguide through the optical routing layer.
- $il^S_{s,\lambda}$ — Insertion loss up to the input of splitter s for wavelength λ.
- il^L_λ — Insertion loss up to the laser source for wavelength λ.

To show how il^L_λ is calculated, an example is explained from the bottom-up. At the left output of splitter 2, power must be available to overcome $il^N_{1,\lambda} + L_4$ of insertion loss on each wavelength λ. Similarly, at the right output of splitter 2, power must be available to overcome $il^N_{2,\lambda} + L_5$ $\forall\lambda$. As stated above, splitters are configured with a 50% splitting ratio, which results in the following value for $il^S_{2,\lambda}$ $\forall\lambda$:

$$il^S_{2,\lambda} = \max\{il^N_{1,\lambda} + L_4, il^N_{2,\lambda} + L_5\} + 10\log_{10}(2) + L^S \qquad \forall\lambda \tag{5.1}$$

Here, L^S is the extra insertion loss caused by the splitter itself ($L^S = 0.2\,\text{dB}$ in this example) and the max operation guarantees each output of the splitter will have enough power for one of the child branches. Naturally, the splitter must receive enough power for both child branches together, which is double the power given by the max operation. The input power requirement from one child branch to two is doubled by adding the constant term $10\log_{10}(2) \approx 3\,\text{dB}$ [1]. At this point, these values in dB are no longer strictly interpreted just as insertion loss, but also as a measure of the optical power

required at the location they are calculated so that all downstream demodulators receive signals correctly, as described by eqs. (3.3) and (3.4).

To calculate $il^S_{1,\lambda}\ \forall\lambda$ (one level above in the tree), the process is identical:

$$il^S_{1,\lambda} = \max\{il^S_{2,\lambda} + L_2, il^S_{3,\lambda} + L_3\} + 10\log_{10}(2) + L^S \qquad \forall\lambda \tag{5.2}$$

Finally, calculating $il^L_\lambda\ \forall\lambda$ is straightforward:

$$il^L_\lambda = il^S_{1,\lambda} + L_1 \qquad \forall\lambda \tag{5.3}$$

Note that if MA n does not send a signal on wavelength λ then it does not require power on that wavelength and thus $il^N_{n,\lambda} = -\infty$ dB. Equations (1)–(3) are still valid in this case. For example, in fig. 5.1, the power required at the input of splitter 3 for wavelength 3 is:

$$\begin{aligned} il^S_{3,3} &= \max\{il^N_{3,3} + L_6, il^N_{4,3} + L_7\} + 10\log_{10}(2) + L^S \\ &= \max\{1 + 0.5, -\infty + 2\} + 10\log_{10}(2) + 0.2 \\ &= 1.5 + 10\log_{10}(2) + 0.2 \approx 4.7\,\text{dB} \end{aligned} \tag{5.4}$$

Likewise, if $il^S_{s,\lambda} = -\infty$ dB for some λ then the splitter does not need to output power on wavelength λ (and this can only happen if no downstream MA uses wavelength λ).

When on-chip laser sources are used there is no need for a PDN because a laser source exists for each node, each source is next to the node it powers and is connected directly to the node's MA. Thus, the power requirements of the laser sources depend not only on the wavelength λ, but also on the node n they belong to, which means il^L_λ must be replaced by $il^L_{n,\lambda}$. Finally, $il^L_{n,\lambda} = il^N_{n,\lambda}$ because there is no insertion loss between the source and the MA.

5.2 Motivation

PSION 2 adopts the MILP model from PSION+ (see chapter 4) as a foundation and addresses two of its shortcomings. Firstly, PSION+ simplifies power consumption optimisation (see sections 3.3.3 and 4.2) and minimises the total/maximum insertion loss at the MAs. PSION 2 improves upon PSION+ by considering the laser source type and the PDN, if one exists, during optimisation.

Secondly, PSION+ considers that PLTs are static during optimisation, i.e. GRU locations and router waveguide paths are fixed. However, this need not be the case. PSION 2 optimally places GRUs (defined in the PLT given as input) for power optimisation using a GRU placement and waveguide routing strategy. This has two advantages. To begin with, this reduces total propagation loss in the router, which leads to lower power consumption. Additionally, this helps rebalance the router when a PDN is required. To understand why this also reduces power consumption, consider the example

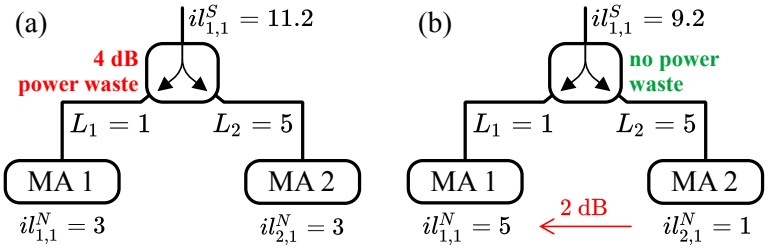

Figure 5.2: Example of how changing the design of the router may lower power requirements. (a) 50% splitting ratio in the splitter causes Node 1 to receive 4 dB extra due to asymmetries in the PDN. (b) Changes in the design of the router can shift insertion loss amounts between nodes to compensate for the PDN and thus result in lower total insertion loss. This compensation is only possible if the PDN is considered in the optimisation.

in fig. 5.2. All max operations, both in the PDN and in $P^Y_{laser:opt}$, have the potential to lead to power losses if the operands of max have very different values: in fig. 5.2(a), the max operation in the splitter causes 4 dB of extra power to be sent to MA 1. By allowing GRUs to be moved, insertion loss loads can be relocated between nodes: in fig. 5.2(b), both branches of the splitter now require the same amount of power, which leads to no power waste and lower power requirements. This is expected to lead to substantial reductions in laser power.

5.3 Laser power optimisation

This section explains the proposed new optimisation functions required to accurately model on-chip and off-chip lasers of both types in the MILP model.

5.3.1 Calculating il^L_λ (off-chip) and $il^L_{n,\lambda}$ (on-chip)

Variables defined in the MILP model from PSION+ which represent the insertion loss through the router of each signal sent by each MA are used to set the value of the $il^N_{n,\lambda}$ variables. Specifically, constraints set $il^N_{n,\lambda}$ equal to the insertion loss of the signal sent by MA n on wavelength λ (if no signal is sent by node n on wavelength λ then $il^N_{n,\lambda} = -\infty$ dB).

Then, variables $il^L_{n,\lambda}$ and il^L_λ are created for on-chip and off-chip designs respectively, and their values are set with constraints that mirror the PDN calculations explained in section 5.1.

5.3.2 Minimising $P_{laser:opt}$

With the values of il^L_λ and $il^L_{n,\lambda}$ established, optimisation functions to minimise $P_{laser:opt}$ according to eqs. (3.3) and (3.4) for on-chip and off-chip lasers are now defined. These are more accurate than $\min \sum il^N_{n,\lambda}$ and $\min \max\{il^N_{n,\lambda}\}$.

The challenge with minimising $P_{laser:opt}$ is to synchronise the measures of power consumption in Watt and insertion loss in dB. The relationship between both is non-linear and thus strategies to convert $P_{laser:opt}$ into a linear function of variables il^L_λ and $il^L_{n,\lambda}$ must be adopted. The essence of the employed strategy is to do the approximation

$$\min \sum x_i \;\widehat{\approx}\; \min \prod x_i \;\Leftrightarrow\; \min \log_{10}(\prod x_i) \;=\; \min \sum y_i \tag{5.5}$$

where x_i are measures of power consumption in Watt which cannot be represented precisely with the available MILP model variables and $y_i = \log_{10}(x_i)$ are measures of insertion loss in dB which can be represented by a linear combination of the available MILP model variables. In eqs. (5.5), (5.6) and (5.9) to (5.11), = indicates that the two surrounding expressions are mathematically equivalent, $\Leftrightarrow$ indicates that the minimisation of the two surrounding expressions is exactly equivalent, i.e. leads to the same optimal assignment of values to model variables even though the expressions themselves are not equivalent, and $\widehat{\approx}$ indicates that the minimisation of the two surrounding expressions is not exactly equivalent but is still likely to lead to the same or a similar optimal assignment of values to model variables.

5.3.2.1 Off-chip lasers

For type Y off-chip lasers, the following simplification can be done:

$$\begin{aligned}
&\min\ P^Y_{laser:opt} = \ \min\ wl * \frac{1}{K} 10^{\frac{\max_{\lambda=1}^{wl}\{il^L_\lambda\}+S}{10}} \\
\Leftrightarrow\ &\min\ wl * 10^{\frac{\max_{\lambda=1}^{wl}\{il^L_\lambda\}}{10}} \\
\Leftrightarrow\ &\min\ 10\log_{10}(wl * 10^{\frac{\max_{\lambda=1}^{wl}\{il^L_\lambda\}}{10}}) \\
\Leftrightarrow\ &\min\ 10\log_{10}(wl) + \max_{\lambda=1}^{wl}\{il^L_\lambda\}
\end{aligned} \tag{5.6}$$

Here, wl is the number of wavelengths of the laser source. Because it is an integer variable, the expression $10\log_{10}(wl)$ can be precisely linearised into the variable $logwl$ with the use of M auxiliary binary variables α_i, where M is the maximum number of wavelengths available (that is, M is a constant and $wl \leqslant M$ is always true), and the following $M + 1$ constraints:

$$logwl = \sum_{i=1}^{M} \alpha_i * 10\log_{10}(i) \tag{5.7}$$

$$(wl = i) \Rightarrow (\alpha_i = 1) \qquad \forall i = 1...M \tag{5.8}$$

Thus eq. (5.6) is precisely equivalent to minimising $P^{Y}_{laser:opt}$. In this case, no approximation is made.

For type X off-chip lasers, the same simplification will now be only an approximation since the summation must be changed to a product before applying the log function:

$$\begin{aligned} \min\ P^{X}_{laser:opt} &= \min \sum_{\lambda=1}^{wl} \frac{1}{K} 10^{\frac{il^{L}_{\lambda}+S}{10}} \\ \Leftrightarrow \min \sum_{\lambda=1}^{wl} 10^{\frac{il^{L}_{\lambda}}{10}} &\approxeq \min \prod_{\lambda=1}^{wl} 10^{\frac{il^{L}_{\lambda}}{10}} \\ \Leftrightarrow \min\ 10\log_{10}\left(\prod_{\lambda=1}^{wl} 10^{\frac{il^{L}_{\lambda}}{10}}\right) &\Leftrightarrow \min \sum_{\lambda=1}^{wl} il^{L}_{\lambda} \end{aligned} \tag{5.9}$$

5.3.2.2 On-chip lasers

For on-chip lasers the simplifications for both types are only approximations since both contain a summation (of the optical power over all laser sources) which must be converted to a product before applying the log function.

For type Y on-chip lasers, the following is done:

$$\begin{aligned} \min \sum_{n=1}^{N} P^{Y}_{laser:opt_n} &= \min \sum_{n=1}^{N} wl_n * \frac{1}{K} 10^{\frac{\max_{\lambda=1}^{wl_n}\{il^{L}_{n,\lambda}\}+S}{10}} \\ \approxeq \min \prod_{n=1}^{N} wl_n * 10^{\frac{\max_{\lambda=1}^{wl_n}\{il^{L}_{n,\lambda}\}}{10}} & \\ \Leftrightarrow \min \sum_{n=1}^{N} \left(10\log_{10}(wl_n) + \max_{\lambda=1}^{wl_n}\{il^{L}_{n,\lambda}\}\right) & \end{aligned} \tag{5.10}$$

Here, $P^{Y}_{laser:opt_n}$ and wl_n are respectively the power consumption and the number of wavelengths of the laser source n and N is the total number of laser sources.

Finally, for type X on-chip lasers, the following is done:

$$\begin{aligned} \min \sum_{n=1}^{N} P^{X}_{laser:opt_n} &= \min \sum_{n=1}^{N} \sum_{\lambda=1}^{wl_n} \frac{1}{K} 10^{\frac{il^{L}_{n,\lambda}+S}{10}} \\ \approxeq \min \prod_{n=1}^{N} \prod_{\lambda=1}^{wl_n} 10^{\frac{il^{L}_{n,\lambda}}{10}} &\Leftrightarrow \min \sum_{n=1}^{N} \sum_{\lambda=1}^{wl_n} il^{L}_{n,\lambda} \end{aligned} \tag{5.11}$$

5.4 GRU positioning optimisation

As explained in section 5.2, good router design is essential for balancing power usage in the PDN and reducing propagation loss in the router. GRU positioning is proposed

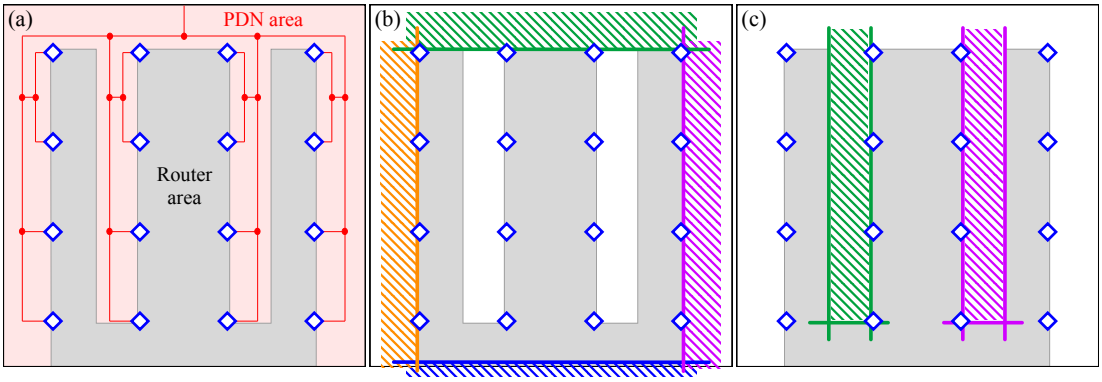

Figure 5.3: (a) Router area and PDN area for a 16 node WRONoC. Router area must be concave in this case. (b) The four convex edges for this router area. (c) The two concave edge sets for this router area. Striped areas are infeasible areas for the router.

as the key to provide sufficient flexibility during router optimisation. This section describes the modelling of GRU positioning. First, the feasible area in which GRUs and waveguides can exist is described. Then, the constraints for GRU positioning are explained. Finally, the modelling of waveguide section routing is clarified, including how the resulting changes in the propagation and bending loss of the router waveguides are taken into consideration during optimisation.

5.4.1 Feasible router area

Both the router and the PDN, when one exists, are placed on the optical routing layer. This can lead to crossings between PDN waveguides and router waveguides which substantially increases total laser power [1, 24]. PSION+ avoids this problem by dividing the layer into two areas: one for the PDN and one for the router (see section 4.3.1). As long as the two areas are mutually exclusive, no crossings between router and PDN are possible. An example of these two areas for a 16 node WRONoC with a PDN is shown in fig. 5.3(a). Both areas are continuous and only polygons with horizontal and vertical edges are considered because this reduces modelling complexity significantly and it is unlikely that more complex cases will be necessary in practice.

For node placements such as in fig. 4.1, a PDN can be designed such that the router area is convex. In other cases, for instance the one in fig. 5.3(a), the router area is always concave. For modelling purposes, all router areas are decomposed into convex edges and concave edge sets. A convex edge is an edge where one of the sides is an infeasible area for the router. The four convex edges for the example in fig. 5.3(a) are shown in fig. 5.3(b). A concave edge set is a set of three edges (two parallel and one perpendicular) where the area within the edges is an infeasible area for the router, as shown in fig. 5.3(c). The union of the infeasible areas defined by each convex edge and

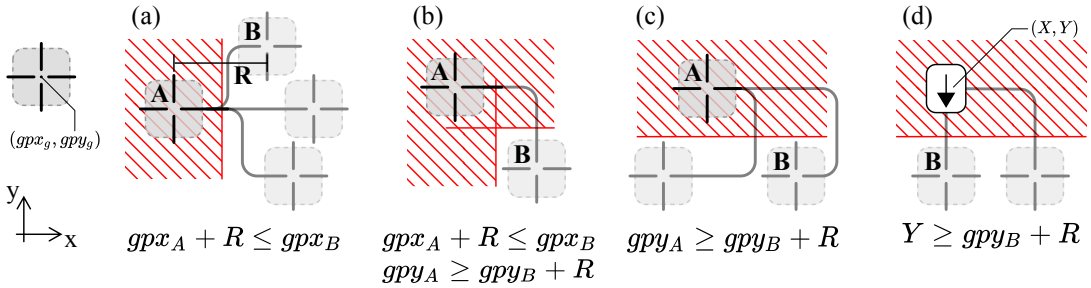

Figure 5.4: Positioning constraints for GRU–GRU and GRU–endpoint connections. Striped areas are infeasible locations for GRU B. (a) Type I: ports on opposite sides. (b) Type L: ports on orthogonal sides. (c) Type D: ports on the same side. (d) Type A: ports on all four sides.

each concave edge set results in a complete definition of the total infeasible area of the router, whose complement is the feasible router area. If no PDN exists, the feasible router area is the entire optical routing layer, which can be modelled with just four convex edges on its boundary.

To avoid the insertion loss penalty caused by crossings between router and PDN, the GRU positioning and waveguide routing method employed by PSION 2 must comply with the defined router area, that is, GRUs must be placed and waveguides must be routed within the feasible area defined by the convex edges and the concave edge sets.

5.4.2 GRU positioning

This GRU placement method is designed to be simple to model in order to keep runtime overhead as low as possible. The original topology of the PLT is also kept intact, i.e. when optimising GRU positions no crossings are created between waveguides that did not exist in the original PLT already. Both of these goals are possible with a straightforward set of constraints that keeps relative positions between interconnected pairs of GRUs and between interconnected GRUs and endpoints.

For GRU–GRU connections, all possible ways GRUs can be connected together (left port of GRU A with left port of GRU B, left port of GRU A with top port of GRU B, etc) are considered. Those 16 ways are categorised into three types, as shown in fig. 5.4: type I (4 ways), type L (8 ways) and type D (4 ways). Variables gpx_g, gpy_g are added to the MILP model for each GRU g, where (gpx_g, gpy_g) is the (x, y) coordinate of the center of GRU g. These variables are constrained appropriately for each pair of interconnected GRUs based on the connection type from the PLT while always keeping a minimum distance R between each GRU pair.

For GRU–endpoint connections, there are only four possible ways (left port of GRU connected to endpoint, top port of GRU connected to endpoint, etc) and all four ways are categorised into the same type, type A, as shown in fig. 5.4. Constraints for gpx_g

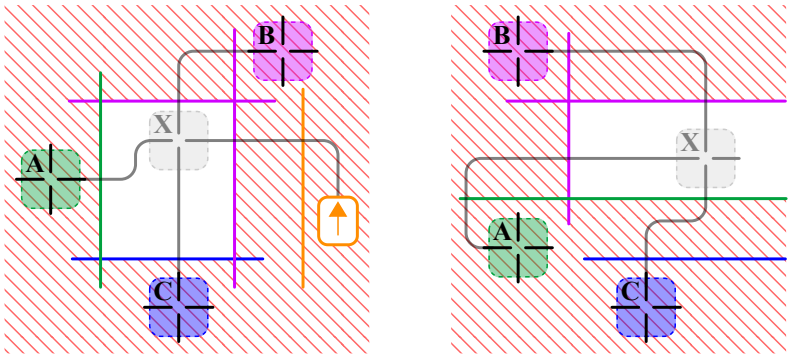

Figure 5.5: Two examples of how the location of GRU X is constrained by the locations of the elements connected to it, in this case GRUs A, B, C and an endpoint. Striped areas are infeasible areas for GRU X.

and gpy_g are added based on the port of the GRU that connects to the endpoint while keeping a minimum distance R between both.

The aforementioned set of constraints results in an infeasible area for each GRU which is defined by its connected elements. Two examples are shown in fig. 5.5. Note that all constraints in an MILP model are considered together during optimisation. Thus, if a location of a GRU changes during optimisation, so will the infeasible area (and consequently possibly also the location) of its connected GRUs.

Finally, to ensure that all GRUs are placed within the router area, a set of constraints for each GRU g is created for each convex edge and each concave edge set such that GRU g cannot be in any of their infeasible areas. Thus, the complete infeasible area for a GRU is the union of the infeasible areas defined by the convex and concave edges with the infeasible areas created by that GRU's connected elements.

5.4.3 Waveguide section routing

Since the topology of the PLT is kept intact when moving GRUs, waveguide routing is straightforward: waveguides are routed vertically and horizontally while minimising their length and number of bends.

If no concave edge sets exist (only convex edges are present), the length of the waveguides can be approximated by the Manhattan distance between their starting and ending locations, which are given by the gpx_g and gpy_g variables and the fixed positions of the endpoints. For example, the length of a waveguide section connecting GRUs g_1 and g_2 is $|gpx_{g_1} - gpx_{g_2}| + |gpy_{g_1} - gpy_{g_2}|$ for any type of GRU–GRU connection.[2] This is because no GRUs or endpoints are ever in the infeasible area of convex edges.

[2] This is only an approximation as some waveguide sections may be slightly longer. For example, with Type D GRU–GRU connections, or to add sufficient spacing between parallel waveguides. But the error is small enough to be ignored for modelling purposes.

Linearisation techniques are applied here to model the absolute value expressions. The number of bends is also easily modelled based on the type (I, L, D or A) and the relative positions of the starting and ending locations.

If concave edge sets exist, more complex modelling is required. If the starting and ending locations of a waveguide are such that the Manhattan routing of the waveguide goes through the infeasible area of a concave edge set, then the waveguide must make a detour around the edges of the concave edge set. The total length of the waveguide is now the Manhattan distance as written above plus the extra length caused by the detour. This detour also adds extra bends. These cases are modelled accordingly such that the correct length and number of bends is always considered by the MILP solver during optimisation.

The changes in length and number of bends for each waveguide section induce changes in the amount of propagation and bending loss for signals going through that waveguide section. The constraints that calculate the insertion loss for each signal (see section 4.2.1.4) take into consideration the effects of GRU positioning and waveguide routing. These alterations to the amount of propagation and bending loss of signals not only reduce total insertion loss in the router but also rebalance the router design and thus help avoid power losses caused by max operations.

5.5 Results

This new methodology, which includes PDN modelling, GRU positioning and accurate optimisation functions, was tested against the same WRONoC test cases from multiple other works [1,6,9,61–63,70] to ensure a fair comparison. These are an 8 node WRONoC and a 16 node WRONoC. PSION+ was shown to outperform the other optimisation tools and manual designs in both test cases (see section 4.5) so it will be used as the baseline for this comparison.

5.5.1 8 node WRONoC

This test case contains 8 nodes on a 9 mm square optical routing layer: four computing hubs and four memory controllers for off-chip memory. The hubs are equally spaced on a 2×2 grid and the memory controllers are next to the edges of the routing layer. PSION+ tested five PLTs, shown in fig. 5.6(a)–(e): CGT-e0, CGT-e6, DGT, a ring template with 3 rings and a custom template. To test off-chip lasers, two PDN designs that connect to those 8 nodes were created and are shown in fig. 5.6(f)(g): one PDN is symmetric in its construction while the other is asymmetric.

This test case yielded 30 design configurations where each is a combination of PLT design, laser location, laser type and PDN design (when off-chip lasers are used). To compare PSION 2 against PSION+, each configuration was optimised twice: once using PSION+ (i.e. no GRU positioning and $\max\{il^N_{n,\lambda}\}$ in the optimisation function)

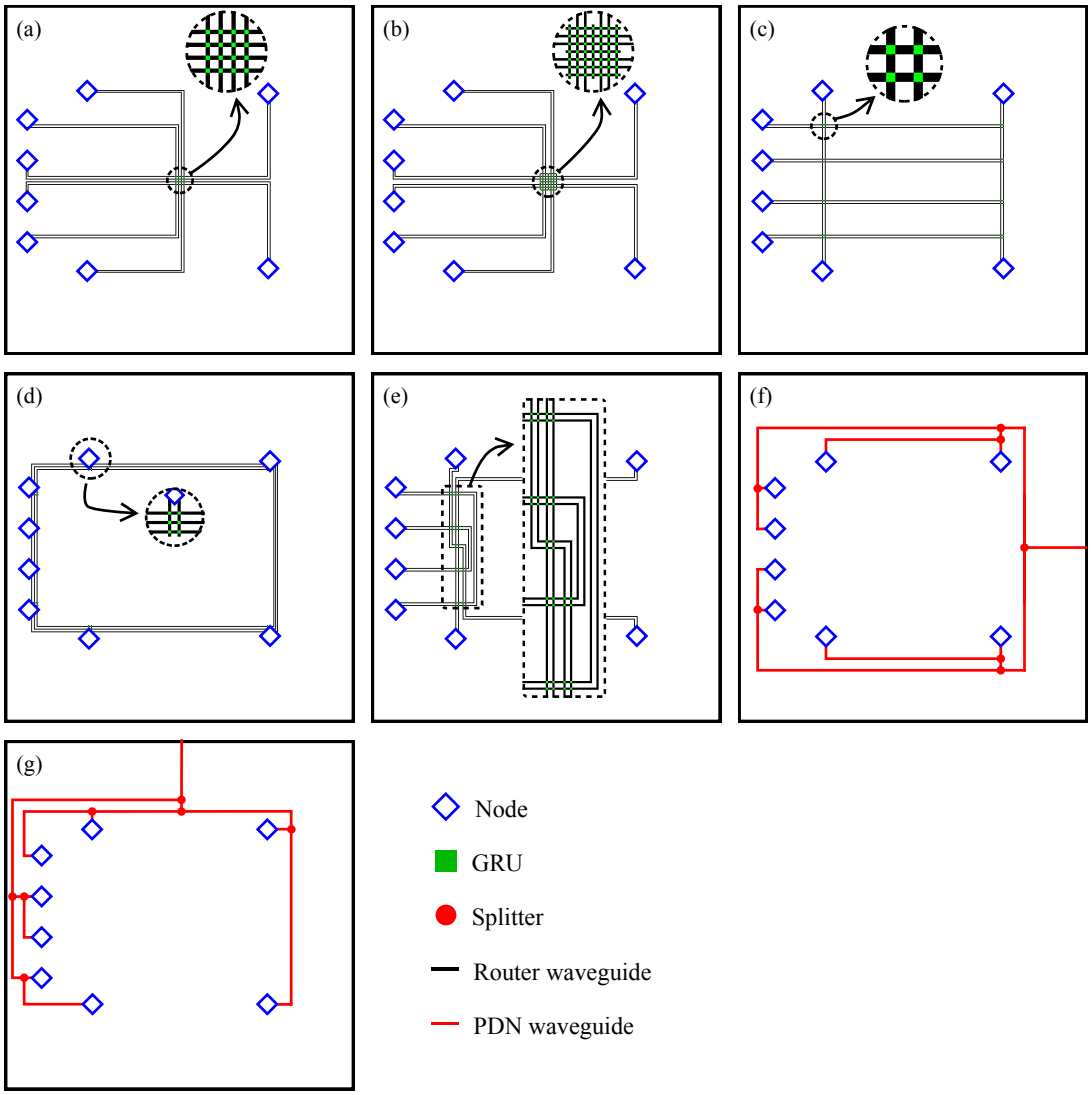

Figure 5.6: PLTs and PDN designs for an 8 node WRONoC. (a) Non-expanded CGT-e0. (b) Expanded CGT with 6 extra waveguide sets (CGT-e6). (c) DGT. (d) Ring template with 3 rings. (e) Custom template. (f) Symmetric PDN. (g) Asymmetric PDN.

to serve as a baseline, and once using the new methodology containing GRU positioning and the accurate optimisation function for each laser type and location as explained in section 5.3.2.

The results are shown in table 5.1. On average, laser power is reduced by 10.2%. The following conclusions can be drawn:

- For 4 out of 5 PLTs, off-chip lasers (i.e. with a PDN) produce a higher average improvement compared to on-chip lasers (no PDN). This is because considering

Table 5.1: Laser power reduction using GRU positioning and accurate optimisation functions for 8 node WRONoC.

Laser & PDN		**Laser power reduction**				
Location	Type	CGT-e0	CGT-e6	DGT	Ring	Custom
Off-chip &	X	13.6%	17.6%	22.5%	11.8%	3.5%
Sym. PDN	Y	7.9%	4.5%	11.5%	10.0%	1.8%
Off-chip &	X	19.6%	20.4%	20.5%	13.1%	4.9%
Asym. PDN	Y	19.3%	15.1%	0.0%	15.0%	11.5%
On-chip &	X	10.7%	11.5%	3.6%	2.1%	6.7%
no PDN	Y	9.1%	6.6%	4.5%	1.8%	7.1%

the PDN in the optimisation allows the solver to compensate for its effects, which is something that PSION+ cannot do.

- The asymmetric PDN yields higher reductions compared to the symmetric PDN on 80% of the cases. This is expected since the asymmetric PDN has a higher imbalance in the distribution of insertion loss between its branches, which leads to more potential for rebalancing like in fig. 5.2.

- Type X lasers yield higher reductions compared to type Y lasers on 73% of the cases. This is because both the optimisation function used by PSION+ in this test case and the optimisation function for type Y lasers contain a max operation, whereas the optimisation function for type X lasers does not. This makes the optimisation function used by PSION+ a worse approximation to the function for type X lasers than for type Y. Hence, using the accurate optimisation function for type X lasers results in a higher improvement.

- Even when on-chip laser sources are used and no PDN is required, the amount of laser power consumed is still reduced significantly. Here, the reduction is achieved not only because of the accurate optimisation functions, but also by optimally positioning the GRUs. Running the test case for on-chip lasers with the highest improvement (CGT-e6 type X, improvement of 11.5%) without GRU positioning reduces the improvement to only 7.6%.

Different PLTs were also found to achieve their power reduction results in different ways. For example:

- The CGTs are originally concentrated on the center of the routing layer, so GRU positioning is paramount to achieving high amounts of power reduction by scattering the GRUs to their optimal locations. Running the test case for the CGT PLT with the highest improvement (CGT-e6 off-chip type X asymmetric PDN,

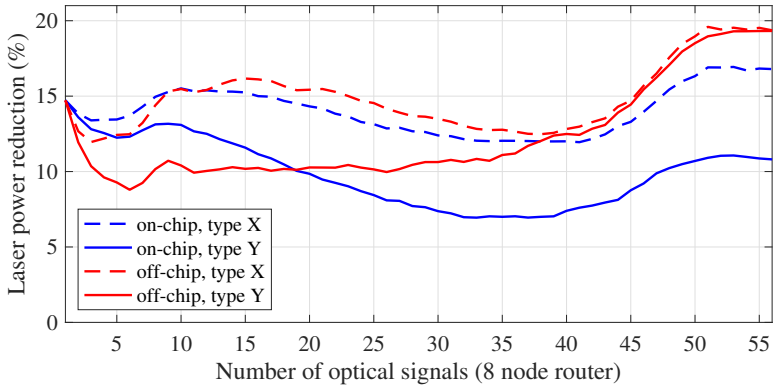

Figure 5.7: Laser power reduction for different laser configurations and CM densities for an 8 node CGT-e0 with an Asymmetric PDN.

improvement of 20.4%) without GRU positioning reduces the improvement to only 12.5%.

- The DGT already has the GRUs distributed throughout the routing layer, so the gains due to GRU positioning are much smaller. For these cases, the power reduction is mostly due to the inclusion of the PDN in the optimisation process and the use of better optimisation functions. In fact, these two aspects are so crucial that the DGT off-chip type X symmetric PDN case is able to achieve 22% reduction based solely on them, i.e. without moving any GRUs.

The results above use the same technology parameters as [9]. Other sets of technology parameters were also tested and similar results were obtained. More specifically, it was found that technology parameter sets with high propagation loss coefficients lead to higher improvements. From this, it can be deduced that the GRU positioning method is effective at reducing propagation loss in the router. In conclusion, all aspects of this work (accurate optimisation functions, consideration of the PDN and GRU positioning) play a decisive role in reducing laser power for WRONoCs.

These test cases took on average about 5× longer to run than PSION+. This seems like a worthwhile quality/runtime trade-off as all of PSION+'s cases, except for the ring template, complete in under two minutes.

Finally, to understand how the reduction in laser power depends on the density of the CM, the CGT-e0 asymmetric PDN test case was chosen as an example. Random testing for CMs from one to 56 signals was performed on this test case to obtain average laser power reduction values for all four combinations of laser locations and types.[3] The results are shown in fig. 5.7. Even with very sparse CMs, PSION 2 achieves substantial reductions in laser power, and over the full range of number of signals the improvement is between 7% and 20%. The improvement for off-chip lasers is in general higher than

[3]For on-chip laser tests the PDN was removed.

for on-chip lasers. Once again, this is due to the presence of the PDN, which provides more opportunities for insertion loss rebalancing.

5.5.2 16 node WRONoC

This test case contains 16 equally spaced nodes in a 4×4 grid configuration on a 16 mm square optical routing layer (see section 4.5.3). Here the laser source is off-chip, so a PDN is required. Since the nodes are equally spaced in a grid pattern, the PDN is symmetric. The node placement along with the PDN design for this test case is equal to those in fig. 5.3(a).

PSION+ uses an expanded CGT-e10 for this test case. It outperforms the manual designs from [1] and so it is the baseline for this comparison. By using PSION 2 with GRU positioning, PDN aware optimisation and accurate optimisation functions to optimise the exact same PLT, laser power consumption is reduced by 20.2% and 8.0% for type X and type Y laser sources respectively.

6 Calculating accurate infinite-order crosstalk[1]

BER is one of the WRONoC performance factors, as explained in section 3.1.2. To create WRONoC designs optimised for this performance factor, a method to calculate crosstalk is required. Multiple works have tackled the calculation of ONoC crosstalk [2, 42,53,54,69], but they all suffer from one or more limitations. Firstly, they only consider first-order noise, that is, noise caused directly by signals. However, this first-order noise goes on to create second, third and higher orders of noise as it travels through the ONoC, which worsens the SNR. Calculating higher orders of noise is considered to be computationally difficult [54]. Secondly, they only operate on specific ONoC designs. Finally, they also commonly use only approximate (average) versions of the insertion loss and crosstalk coefficients (see section 3.3.2).

The method presented in this chapter is the first capable of lifting all of these restrictions. It uses a linear algebra method based on flow graphs to calculate first-order and infinite-order incoherent noise levels and SNR values for any active or passive ONoC design using either the approximate (average) or the accurate (wavelength-dependent) versions of the insertion loss and crosstalk coefficients. This method can calculate the total amount of noise generated or discriminate the results on a signal by signal basis. Furthermore, its mathematical backbone can be applied to any system fulfilling a simple set of rules, not just ONoCs.

This chapter is organised as follows. Section 6.1 establishes the mathematical foundation of this method and section 6.2 follows with a matrix reduction technique that is useful to increase its performance. These two sections stand on their own and their utility extends beyond crosstalk calculation for ONoCs, so they are presented first. Section 6.3 then explains how the principles from section 6.1 are applied to calculate SNR in ONoCs. Section 6.4 touches on a few practical considerations of this method, namely how the reduction technique in section 6.2 increases performance. Finally, section 6.5 presents results obtained by this method and section 6.6 summarizes the chapter.

6.1 Steady-state analysis of a flow graph

To start, a general version of this method is explained. This general version can be applied to any system composed of multiple elements that obeys the following rules:

[1] Segments of this chapter were previously published in [8].

1. Each element holds some amount of a quantifiable property of the system. Here this property is called energy.
2. Some elements may be sources or sinks of energy.
3. Fixed ratios of energy are transferred between elements over time.
4. If there are no active source elements inserting energy into the system, the total amount of energy in the system decreases to zero over time.

Under the condition that all sources of energy input a constant amount of energy per unit of time into the system forever, the goal is to calculate the distribution of energy over the system's elements (the state of the system) after an infinite amount of time has passed, i.e. its steady-state.

For a system with n elements, consider a weighted directed graph $G = (V, E)$, abbreviated as flow graph in this chapter, with n vertices and edge weights $w(e)\ \forall e \in E$. The graph structure can be described by an adjacency matrix $A \in \mathbb{R}^{n\times n}$ where:

$$[A]_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

A transfer matrix $T \in \mathbb{R}^{n\times n}$ can be created based on the transpose of the adjacency matrix A and the edge weights $w(e)$

$$[T]_{ij} = w(j,i) \tag{6.2}$$

where the definition of $w(e)\ \forall e \in E$ is extended with $w(e) = 0\ \forall e \notin E$ for simplicity. This matrix fully defines the flow graph.

Each vertex $v \in V$ is associated with a state variable x_v. Both the state variables and the edge weights are real and non-negative. The weights are constant but the state variables are time-varying so they are indexed on discrete time steps: $x_v[k]$. Each state variable $x_v[k]$ indicates how much energy is flowing through vertex v at time instant k. The propagation of energy through each vertex v in the graph can be described by the following relationship:

$$x_v[k+1] = \sum_{i\in V} w(i,v)\, x_i[k] \tag{6.3}$$

In other words, some fraction of the energy currently flowing at time k through vertices whose outgoing edges end at v will be propagated at time $k+1$ to vertex v, and some fraction of the energy flowing at time k through vertex v will be propagated at time $k+1$ to the vertices where the outgoing edges of v end at. An example is shown in fig. 6.1.

Note that the energy in a vertex v is not necessarily propagated in full to the rest of the system. Define $f_v = \sum_{i\in V} w(v,i)$. Vertex v is a lossless vertex if $f_v = 1$, a partial sink vertex if $f_v \in]0,1[$ and a full sink vertex if $f_v = 0$.

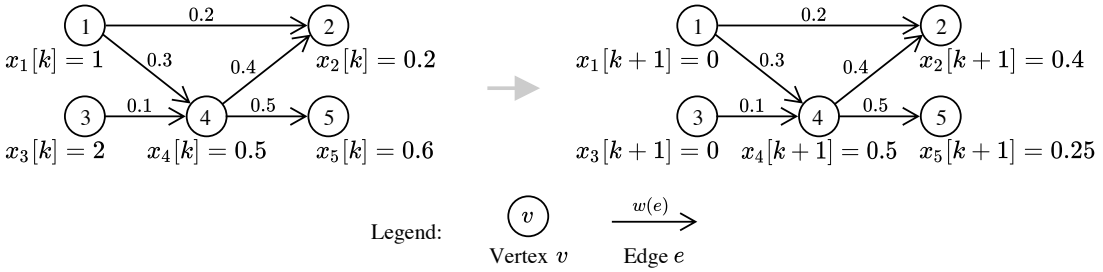

Figure 6.1: Example of how energy is propagated through the flow graph over one time step, as described by eq. (6.3).

If all state variables $x_v[k]$ are combined into a state vector $x[k] \in \mathbb{R}^n$, the propagation relationship in eq. (6.3) can be written simply as:

$$x[k+1] = Tx[k] \tag{6.4}$$

That is, the transfer matrix T propagates energy one step forward in the flow graph.

Some vertices may be energy sources which insert energy into the system. Define δ_k as a one-hot column vector:

$$[\delta_k]_i = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{otherwise} \end{cases} \tag{6.5}$$

If one source vertex v produces a pulse (i.e. lasting only one time step) of energy with amount g, the addition of energy into the system can be expressed in a vector $b \in \mathbb{R}^n$ with $b = g\delta_v$. If the system starts with no energy, i.e. $x[k < 0] = 0$, and the pulse is produced at $k = 0$, then $x[0] = b$. To calculate the propagation of energy through the graph just multiply by T, resulting in the relationship $x[k] = Tx[k-1] = T^k x[0]$.

Since multiple outgoing edges may exist for each vertex, energy that starts as a pulse in one vertex v spreads out through the graph over time. Multiplying by T^k does not calculate one single propagation path of energy k steps through the graph, but instead a wavefront of energy propagating in parallel through all possible k-step paths starting at v in the graph.

In general, multiple source vertices may produce a pulse on the same time step. Thus, the complete definition of vector b is

$$b = \sum_{v \in V} g(v)\delta_v \tag{6.6}$$

where $g(v)$ is the amount of energy per pulse released by vertex v. Source vertices have $g(v) > 0$ and other vertices have $g(v) = 0$.

This method's goal is now formalised according to the explanation in the beginning of this section:

- Assume $x[k < 0] = 0$.

- One or more sources are turned on at $k = 0$, each producing a constant amount of energy continuously (i.e. on each time step) forever. The release of energy into the system on each time step is defined in eq. (6.6).
- Calculate $x[\infty]$, that is, the steady-state value of the state vector x.

The value of $x[\infty]$ can be calculated step-by-step. Start with $x[0] = b$. At $k = 1$, the energy from $k = 0$ has been propagated forward once and a new amount b of energy has been produced by the sources, resulting in:

$$x[1] = Tx[0] + b = Tb + b = (T + I)b \tag{6.7}$$

At $k = 2$, the process is repeated:

$$x[2] = Tx[1] + b = (T^2 + T + I)b \tag{6.8}$$

Clearly, the conclusion is:

$$x[\infty] = \left(\sum_{i=0}^{\infty} T^i\right) b \tag{6.9}$$

This result is sensible: the steady-state amount of energy in each vertex is the sum of the contributions of all possible paths of energy of all lengths (between 0 and ∞ steps) starting at each active source (the non-zero entries in b).

Calculating $x[\infty]$ using an infinite sum of powers is not reasonable in practice. Fortunately, the well-known closed-form formula for the geometric series with a real number r

$$\sum_{i=0}^{\infty} r^i = \frac{1}{1-r} \quad \forall r \in \mathbb{R} : |r| < 1 \tag{6.10}$$

has a matrix analogue

$$\sum_{i=0}^{\infty} M^i = (I - M)^{-1} \quad \forall M \in \mathbb{R}^{n \times n} : \rho(M) < 1 \tag{6.11}$$

which is true for all square matrices M whose spectral radius $\rho(M) < 1$.

Define $\hat{M}$ as shorthand for $I - M$ for all matrices M. With the use of theorem 1, the calculation of $x[\infty]$ is reduced to solving a single linear system of equations:

$$x[\infty] = \left(\sum_{i=0}^{\infty} T^i\right) b \;\Leftrightarrow\; x[\infty] = (I - T)^{-1} b \;\Leftrightarrow\; \hat{T}x[\infty] = b \tag{6.12}$$

Note that the convergence of $x[\infty]$ does not depend on how much energy is input into the system, i.e. the amplitude of b. Convergence is guaranteed just based on b being finite and constant and $\rho(T) < 1$.

Section 6.3 will explain how eq. (6.12) can be used to calculate insertion loss and crosstalk on ONoCs.

Theorem 1. *The transfer matrix T satisfies the condition $\rho(T) < 1$.*

Proof. Let $E(x[k])$ be the total energy in the system given a state-vector $x[k]$ at time instant k. Then

$$E(x[k]) = \sum_{i=1}^{n} [x[k]]_i = \sum_{i=1}^{n} |[x[k]]_i| = \|x[k]\|_1 \tag{6.13}$$

since all the entries in x are non-negative.

If the system has the state $x[k]$ at time instant k, the total energy in the system after r time steps is:

$$E(x[k+r]) = E(T^r x[k]) = \|T^r x[k]\|_1 \tag{6.14}$$

Since the total energy in a system with no active source vertices must eventually decrease to zero (rule 4 of the system), there must be a value $r \in \mathbb{Z}^+$ of propagation steps where the following is true for all[2] state vectors $x[k]$:

$$E(x[k+r]) < E(x[k]) \;\Leftrightarrow\; \|T^r x[k]\|_1 < \|x[k]\|_1 \tag{6.15}$$

By the definition of induced matrix norm, eq. (6.15) implies $\|T^r\|_1 < 1$. The following is true for all induced matrix norms

$$\rho(T) \leqslant \|T^r\|^{1/r} \quad \forall r \in \mathbb{Z}^+ \tag{6.16}$$

which proves $\rho(T) < 1$ is true. □

Corollary 1. *The flow graph cannot have lossless cycles.*

Proof. A more intuitive and verifiable interpretation of rule 4 of the system is that the flow graph cannot have lossless cycles.

A lossless vertex is a vertex whose weights of its outgoing edges sum to 1. All energy held by a lossless vertex is kept in the system during propagation. A lossless tree is a connected series of lossless vertices where all energy held by the vertices in the tree is kept in the system during propagation. A lossless cycle is a lossless tree that closes back on itself with no energy losses. Once energy enters any vertex in the cycle, it will never escape and will stay in the system forever. Most, if not all, real physical systems have no perfect stores of energy, thus have no lossless cycles.

Clearly, for rule 4 to be satisfied, lossless cycles cannot exist. The reverse implication is also true, making rule 4 and the absence of lossless cycles equivalent statements.

Furthermore, if there are no lossless cycles, then by definition there must be a value $r \in \mathbb{Z}^+$ of propagation steps (larger than the length of the longest lossless tree) after which the total energy in the system has decreased, which also proves $\rho(T) < 1$. The reverse implication is also true.

In conclusion: rule 4, $\rho(T) < 1$ and the absence of lossless cycles are all equivalent statements. □

[2]Except the zero state vector where, by definition, the total energy is already zero.

6.2 Equivalent reduction of a flow graph

In this section a process is presented that reduces the number of vertices in the flow graph, and thus reduces the size of the associated transfer matrix T, while keeping the relevant steady-state analysis results unchanged.

On any given flow graph, it is possible that some vertices are not source vertices. Likewise, when calculating the steady-state vertex values, it is possible that some values are not relevant as an analysis result. If a vertex is not a source vertex (so does not contribute to a non-zero value in the b vector) and is also not a vertex relevant for the result (its corresponding steady-state energy value in the x vector is not used after), then it can be eliminated before the $\hat{T}x = b$ system is solved.

A process will now be described that allows for a set of vertices to be removed from the flow graph without altering any of the relevant results, and in section 6.4.2 this process will be used specifically in the context of ONoCs to speed up computation time.

Consider the graph $G = (V, E)$ with n vertices. Define R as the set of vertices to remove, Q as the set of intermediate vertices and $S = V \backslash (R \bigcup Q)$ as the rest of the vertices in V. Intermediate vertices are all vertices not in R that are connected directly to one or more vertices in R. Sets R, Q and S are disjoint.

To eliminate the vertices in R from the graph, all edges connected to vertices in R must be removed. These are edges internal to R and edges between R and Q. The problem can be represented in matrix form using the transfer matrix of the graph. Start with the original matrix T, then partition it into 9 sub-matrices:

$$T = \begin{bmatrix} A & B & 0 \\ C & D & F \\ 0 & H & J \end{bmatrix} \tag{6.17}$$

It can be assumed, without loss of generality[3], that:

- the first set of rows/columns (sub-matrices A, B and A, C) are the in/out-edges respectively for the vertices in R,
- the second set of rows/columns (C, D, F and B, D, H) are the in/out-edges respectively for the vertices in Q, and
- the last set of rows/columns (H, J and F, J) are the in/out-edges respectively for the vertices in S.

This is shown graphically in fig. 6.2(a). Note that the dimensions of these sub-matrices are defined by the number of vertices in each set R, Q, S, and vertices in R are not connected to vertices in S so the two corresponding sub-matrices (on the corners) are filled with zeros.

[3]The order of the vertices in T can be changed at will by re-ordering rows and columns with $T' = PTP^T$, where P is a permutation matrix, and does not influence any result.

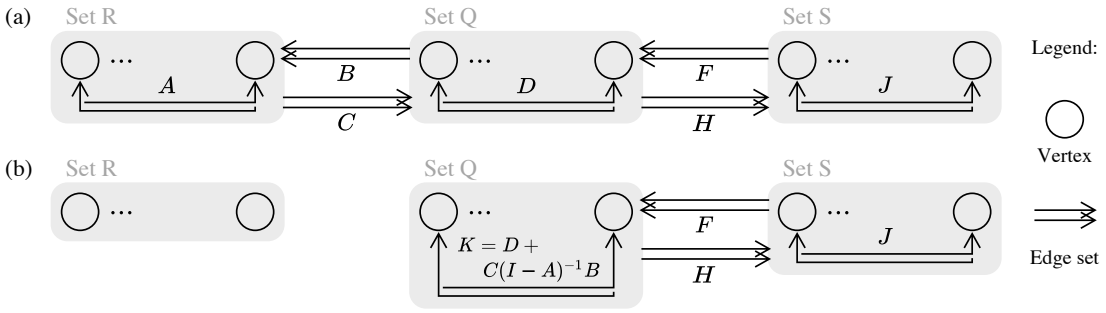

Figure 6.2: Simplifying a flow graph by removing all edges connected to vertices in the set R. Each set of edges shown is associated with the corresponding sub-matrix of the transfer matrix T of the flow graph. (a) Before simplification. (b) After simplification.

By eliminating all edges connected to vertices in R, a new transfer matrix U is created:

$$U = \begin{bmatrix} 0 & 0 & 0 \\ 0 & K & F \\ 0 & H & J \end{bmatrix} \tag{6.18}$$

A graphical representation of U is shown in fig. 6.2(b). Sub-matrix K must be determined so that transfer matrices T and U are equivalent for vertices in $Q \bigcup S$.

The equivalence of two matrices for a set of vertices $V^* \subseteq V$ is defined based on eq. (6.12). Consider $x_i = \hat{T_i}^{-1} b$. Two transfer matrices T_1 and T_2 are equivalent for vertices V^* if any b vector which only contains non-zero values for vertices in V^* results in vectors x_1 and x_2 with the same values for all rows corresponding to vertices in V^*:

$$\begin{aligned}
& T_1 \overset{V*}{\longleftrightarrow} T_2 \\
\Leftrightarrow\ & \delta_i^T \hat{T_1}^{-1} \left(\sum_{j \in V^*} g(j)\delta_j \right) = \delta_i^T \hat{T_2}^{-1} \left(\sum_{j \in V^*} g(j)\delta_j \right) && \forall g(j) \in \mathbb{R}, i \in V^* \\
\Leftrightarrow\ & \delta_i^T \hat{T_1}^{-1} g(j)\delta_j = \delta_i^T \hat{T_2}^{-1} g(j)\delta_j && g(j) \in \mathbb{R}, i, j \in V^* \\
\Leftrightarrow\ & \delta_i^T \hat{T_1}^{-1} \delta_j = \delta_i^T \hat{T_2}^{-1} \delta_j && \forall i, j \in V^* \\
\Leftrightarrow\ & \left[\hat{T_1}^{-1} \right]_{ij} = \left[\hat{T_2}^{-1} \right]_{ij} && \forall i, j \in V^*
\end{aligned} \tag{6.19}$$

To determine K such that matrix U is equivalent to T for vertices in $Q \bigcup S$, first $\hat{T}^{-1}$ and $\hat{U}^{-1}$ must be calculated. The inverse of a 2×2 block matrix is

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix} \tag{6.20}$$

assuming A and D are invertible [73]. Matrices T and U are partitioned into 4 sub-matrices each along the following boundaries

$$T = \left[\begin{array}{c|cc} A & B & 0 \\ \hline C & D & F \\ 0 & H & J \end{array}\right] \qquad U = \left[\begin{array}{c|cc} 0 & 0 & 0 \\ \hline 0 & K & F \\ 0 & H & J \end{array}\right] \tag{6.21}$$

and apply[4] eq. (6.20) to $\hat{T}$ and $\hat{U}$, resulting in

$$\hat{T}^{-1} = \begin{bmatrix} [\hat{T}^{-1}]_{11} & [\hat{T}^{-1}]_{12} \\ [\hat{T}^{-1}]_{21} & [\hat{T}^{-1}]_{22} \end{bmatrix} \tag{6.22}$$

$$\hat{U}^{-1} = \begin{bmatrix} I & 0 \\ 0 & [\hat{U}^{-1}]_{22} \end{bmatrix} \tag{6.23}$$

where:

$$\begin{aligned} \left[\hat{T}^{-1}\right]_{22} &= \left(I - \begin{bmatrix} D & F \\ H & J \end{bmatrix} - \begin{bmatrix} C \\ 0 \end{bmatrix} (I-A)^{-1} \begin{bmatrix} B & 0 \end{bmatrix}\right)^{-1} \\ &= \left(I - \begin{bmatrix} D + C(I-A)^{-1}B & F \\ H & J \end{bmatrix}\right)^{-1} \end{aligned} \tag{6.24}$$

$$\left[\hat{U}^{-1}\right]_{22} = \left(I - \begin{bmatrix} K & F \\ H & J \end{bmatrix}\right)^{-1} \tag{6.25}$$

According to eq. (6.19), if matrices T and U are equivalent for vertices in $Q \bigcup S$, then $[\hat{U}^{-1}]_{22} = [\hat{T}^{-1}]_{22}$ must be true. By comparing eq. (6.24) with eq. (6.25) the conclusion is clear:

$$K = D + C(I-A)^{-1}B \tag{6.26}$$

This result is sensible: the flow of energy from Q to R and back to Q is added to the already existing flow between vertices in Q.

In the case where R contains only one vertex, matrix B becomes vector b, matrix C becomes vector c, matrix A becomes scalar a and:

$$[K]_{ij} = [D]_{ij} + \frac{[c]_i [b]_j}{1-a} \tag{6.27}$$

This simplifies further to $[K]_{ij} = [D]_{ij} + [c]_i[b]_j$ when the vertex in R does not have a self-looping edge ($a = 0$).

In summary, to eliminate a set R of vertices from the flow graph, start with the associated transfer matrix T, calculate $C(I-A)^{-1}B$ for the corresponding sub-matrices A, B, C in T, add the result to the corresponding sub-matrix D and zero the sub-matrices A, B and C. At this point, the size of the T matrix can be reduced by eliminating from it all the zero rows and columns.

[4]The sub-matrices on the diagonal of the block decompositions of $\hat{T}$ and $\hat{U}$ are invertible.

6.3 Steady-state crosstalk calculation

This section now explains how to calculate SNR for any ONoC design using the method in section 6.1. This is possible because ONoCs satisfy all four rules: waveguides hold specific amounts of energy in transit, modulators/demodulators are sources/sinks of energy, insertion loss and crosstalk factors transfer fixed fractions of energy between waveguide locations, and all ONoC elements cause insertion loss so ONoCs cannot have lossless cycles.

Assume a modulator sends a signal into the router. As it travels through the router, two types of interactions are possible: the signal may lose a fraction of its energy due to insertion loss or the signal may bifurcate due to noise-inducing elements. Therefore, there are two types of light streams traveling through the router and two types of transformations that can be applied to them. A light stream can either be signal or noise, and both stream types can suffer insertion loss and bifurcation. Signal/noise streams contribute to signal/noise energy levels respectively at the demodulators they arrive at.

When a signal goes through a noise-inducing element, it bifurcates and a fraction of its energy becomes first-order noise. This noise stream travels through the router suffering insertion loss like any other light stream. If it encounters a noise-inducing element, it bifurcates again and a fraction of the first-order noise energy becomes second-order noise (no transformation can be applied to a noise stream that generates a signal stream back). This process may happen up to an infinite number of times depending on the design of the router. The energy of all $\{1, 2, ...\infty\}$-order noise streams is summed together to contribute to the noise energy levels for the demodulators the streams arrive at.

Three steps are required to model the propagation of light through an ONoC and calculate all SNR results. First, translate the ONoC into a corresponding set of flow graphs, then convert the graphs into transfer matrices and then apply the mathematical method in section 6.1. Since streams go through different paths and suffer different transformations depending on their wavelength, a flow graph G_λ must be created for each wavelength λ used in the ONoC. This results in a different transfer matrix T_λ per wavelength.

6.3.1 Building a flow graph

A three-stage process is used to convert an ONoC design into a flow graph G_λ for a specific wavelength λ: first the locations of compound vertices on the router are identified, then each compound vertex is converted into multiple single vertices and then directed edges between single vertices are added with the appropriate edge weights.

Compound vertices are placed on waveguides such that, for all locations where a light stream can suffer insertion loss and/or bifurcation, there are two adjacent compound

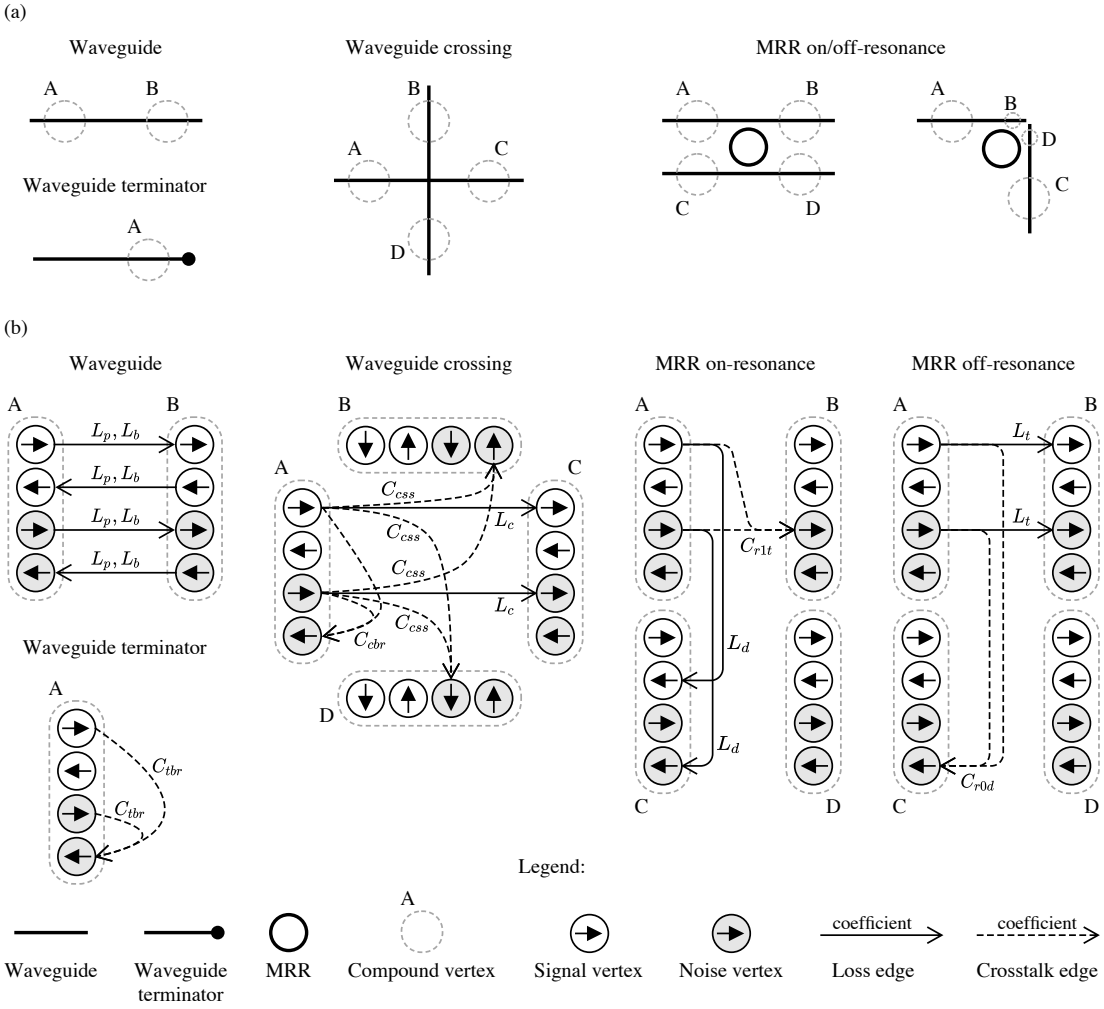

Figure 6.3: (a) Location of compound vertices for ONoC elements. For the MRR on/off-resonance case, the two possible waveguide placement configurations around an MRR are shown. Both configurations have 4 compound vertices A,B,C,D. Although they are placed in different locations relative to the MRR, they work exactly in the same way when converting to a flow graph. (b) The decomposition of the compound vertices into single (signal and noise) vertices and the edges connecting them. The loss/crosstalk weight used in each edge is also shown. In the waveguide case, the weights are propagation loss (L_p) and/or bending loss (L_b) depending on the path of the waveguide. For simplicity and clarity, in the waveguide crossing and MRR cases only the edges leaving vertices in the compound vertex A are shown. The edges leaving compound vertices B, C and D are analogous and thus not drawn.

vertices, one just before the transformation and the other just after. Figure 6.3(a) shows the locations of compound vertices for ONoC elements. A compound vertex is placed on each end of a waveguide, just before a waveguide terminator, on the four sides of a waveguide crossing and on the four locations around an MRR. A compound vertex is also placed just after each modulator and just before each demodulator.

Each compound vertex tracks the amount of signal and noise energy going through the waveguide on its location. Since there are two types of streams and streams can travel in two directions on a waveguide, each compound vertex is decomposed into four single vertices: signal direction top/left, signal direction bottom/right, noise direction top/left, noise direction bottom/right. Figure 6.3(b) shows examples of this decomposition. Flow graphs are composed of these single vertices.

To build a flow graph of an ONoC comprised of multiple individual ONoC elements, the compound vertices (and, by extension, the single vertices) of each element are merged with the ones from its adjacent elements. An example is shown in fig. 6.4.

Having listed all the vertices of a flow graph, the final step is to add edges with the correct weights. Edges describe what fractions of energy are transferred between vertices, and thus how light is propagated through the router. Since a light stream can suffer insertion loss and/or bifurcation between vertices, the edge weights are of type loss and crosstalk respectively. Figure 6.3(b) shows the edges required for each ONoC element. Edge direction follows intuitively from the light direction associated with each vertex.

For each MRR, the choice to use the on- or off-resonance case is based on the path the signal should take according to the design of the network. The L_d, L_t, C_{r0d} and C_{r1t} weights can also be the average or the accurate versions. The average weights are the same for all G_λ graphs but the accurate weights also depend on the wavelength λ of the flow graph G_λ and the resonance wavelengths of the MRR.

6.3.2 Transfer matrix types

Edges are categorised into four types depending on the types of the vertices they are connected between and the type of their weight:

Type SL from a signal (S) vertex to a signal vertex with a loss (L) weight.

Type SX from a signal (S) vertex to a noise vertex with a crosstalk (X) weight.

Type NL from a noise (N) vertex to a noise vertex with a loss (L) weight.

Type NX from a noise (N) vertex to a noise vertex with a crosstalk (X) weight.

The distinction between these four edge types is crucial to the method and can be explained in three stages:

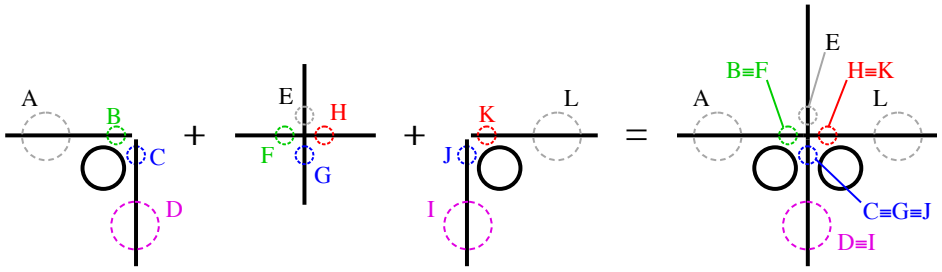

Figure 6.4: Example of how the compound vertices of isolated elements shown in fig. 6.3(a) are merged to form a flow graph of an ONoC. In this case, the four compound vertices of two MRRs each and the four compound vertices of a waveguide crossing are combined into a total of seven compound vertices.

1. If only SL edges are considered, then only signals are propagated through the graph. Signal streams suffer insertion loss before arriving at their respective demodulators, but no noise streams are generated and thus demodulators will receive no noise energy.

2. If SL, SX and NL edges are considered then only signal streams and first-order noise streams are propagated through the graph. SL edges are responsible for the propagation of signal streams, SX edges are responsible for the generation of first-order noise streams from signal streams and NL edges are responsible for the propagation of first-order noise streams. Demodulators receive both signal and first-order noise energy. Therefore, first-order SNR can be calculated.

3. If NX edges are also considered along with SL, SX and NL edges, then n-order noise streams generate $n+1$-order noise streams. Noise received by the demodulators will be the sum of all $\{1, 2, ...\infty\}$-order noise and the SNR values calculated will be all-order SNR.

These three stages result in three possible T_λ transfer matrix types for each G_λ flow graph: T_λ^S containing only SL edges, $T_\lambda^{SN_1}$ containing SL, SX and NL edges, and $T_\lambda^{SN_\infty}$ containing all edges. All these matrices have the same size because the number of vertices in the graph does not change[5] but the fewer edge types considered, the sparser the matrix is.

Depending on the choice of T_λ matrix type and on how the b vector is constructed, different kinds of information can be obtained from solving $\hat{T}_\lambda x = b$. Matrix T_λ^S can only be used to calculate insertion loss for signals whereas matrices $T_\lambda^{SN_*}$ (where $*$ is 1 or ∞) can also be used to calculate noise energy levels and SNR values at the demodulators.

[5]There is an exception to this statement that will be explained in section 6.4.2.

6.3.3 Calculating signal insertion loss

Assume a signal is sent on wavelength λ. Let i be the vertex where the signal's modulator sends signal energy into the router and j be the vertex where the signal's demodulator receives signal energy from the router.

To calculate this signal's total insertion loss, first solve $\hat{T}_\lambda^* x = b$ for vector x where $*$ is any of the three matrix types and $b = 1\delta_i$. Then, the insertion loss on this signal's path through the router from vertex i to vertex j ($il_{i\to j}$) is the logarithm of the ratio of signal energy sent into the router ($[x]_i$) to signal energy received from the router ($[x]_j$):

$$\begin{aligned} il_{i\to j} &= 10\log_{10}\left(\frac{[x]_i}{[x]_j}\right) = 10\log_{10}\left(\frac{1}{[x]_j}\right) \\ &= -10\log_{10}\left([x]_j\right) \end{aligned} \tag{6.28}$$

This reasoning can be extended to multiple signals at once as long as their wavelengths are the same and their paths do not interfere, i.e. do not contain the same waveguides.[6] Just use eq. (6.6) with $g(v) = 1\ \forall v \in V^*$ where V^* is the set of vertices of the corresponding modulators sending signal energy into the router. Then solve $\hat{T}_\lambda^* x = b$ once and apply eq. (6.28) for each signal. This reduces the computational burden of calculating the insertion loss of all signals.

6.3.4 Calculating noise at demodulators

Assume a signal is sent on wavelength λ. Let i be the vertex where the signal's modulator sends signal energy into the router, j be the vertex where the signal's demodulator receives signal energy from the router and v_k be all the vertices where the demodulators receive noise energy from the router.

To calculate the noise caused by this signal on all demodulators, solve $\hat{T}_\lambda^{SN_*} x = b$ where $*$ is the chosen noise type and $b = g\delta_i$ where g is the signal's energy at the modulator. The signal's energy at the demodulator is given by $[x]_j$ and the noise energies at the demodulators are given by $[x]_{v_k}$.

By solving one system per signal, the noise caused by each signal on each demodulator can be calculated separately. Intra-channel noise caused by the signal itself, intra-channel noise caused by other signals and inter-channel noise can all be calculated separately in this way. However, like in section 6.3.3, the computational burden can be reduced by combining signals with the same wavelength and non-interfering paths into the same b vector. In this case, values $[x]_{v_k}$ contain the total amount of noise generated by all the selected signals and received by each demodulator.

[6] The optical exclusion principle guarantees that this is always true for WRONoCs.

6.3.5 Calculating SNR at demodulators

To obtain all SNR values, first calculate the insertion loss of all signals using section 6.3.3 and from the insertion loss determine the required energy of each signal at the modulator.[7] Then, calculate the signal and noise energies at the demodulators using section 6.3.4 and from there derive the SNR values. To reduce the computational burden, multiple signals with the same wavelength and non-interfering paths can be combined into the same b vector in which case only the sum of the generated noise is obtained instead of individual noise energy levels.

6.4 Practical considerations

There are multiple practical aspects to take into consideration when solving $\hat{T}x = b$ that can make this method faster and more memory efficient. These will be discussed now.

6.4.1 Sparse matrix methods

The average transfer matrix density (ratio of non-zero entries to total entries in the matrix) is very low and approaches zero as the number of vertices in the flow graph increases. The use of sparse matrices and sparse matrix methods is thus essential to storing T efficiently and solving $\hat{T}x = b$ quickly.

Methods for solving linear systems can be divided into two types: direct and iterative. Direct methods guarantee a full precision solution in a finite number of steps. Conversely, iterative methods start from an initial guess and converge (some methods, asymptotically) to the solution with each iteration. Because of their iterative nature, iterative methods can trade accuracy for speed by stopping earlier when a good enough solution is achieved.

Direct methods for sparse linear systems include sparse versions of Cholesky, LU and QR factorisations. To avoid fill-in during factorisation, multiple strategies exist. One such strategy is ordering vertices in the matrix to reduce its bandwidth, which can be applied here very easily. Through testing, LU factorisation with Cuthill-McKee vertex ordering has shown itself to work well.

Iterative methods for sparse linear systems include Conjugated Gradient (CG) methods, the Generalised Minimal Residual (GMRES) method and the Biconjugate Gradient Stabilised (BiCGSTAB) method. Iterative methods may also benefit greatly from using a preconditioning matrix. Testing has shown that BiCGSTAB with an incomplete LU preconditioner works very well.

[7] In case the energy levels of the signals are already available, this step can be skipped.

6.4.2 Transfer matrix reduction techniques

Any process by which a transfer matrix's size is reduced or its sparsity is increased without affecting the result can have the potential to save memory and speed up the matrix method.

A vertex whose corresponding row and column of the transfer matrix are empty, i.e. filled with zeros, is a vertex without any ingoing or outgoing edges. These vertices can be eliminated from the flow graph and transfer matrix without consequence. Because of this, since noise vertices in T^S are not used, T^S can actually be half the width and height of the T^{SN_*} matrices. Likewise, when the process from section 6.2 is used, the resulting empty rows/columns in sub-matrices A, B, C can be removed. However, good sparse matrix method implementations should be able to handle empty rows/columns with little to no overhead, dismissing the requirement to remove them from the matrix once they are emptied.

Many active and passive ONoC routers are built from multiple copies of one basic, configurable, routing structure. These structures are self-contained collections of ONoC elements that perform simple routing tasks between a fixed number of ports (commonly four or five). They are copied multiple times and interconnected through a specific topology to form an ONoC router for as many network nodes as required. Each structure is configured internally so that all structures work together to accomplish the routing requirements of the full ONoC router. For example, as explained in section 2.3.1, active ONoCs can be built from multiple structures (Crux [36], Cygnus [37], OTAR [39]) interconnected through different topologies. WRONoCs can be built using PSEs [9] or GRUs (see section 4.1.2). Crucially, for each of these routing structures, there is a relatively small finite amount of possible internal configurations. Because of this, it is possible to list these configurations and use the process from section 6.2 to reduce the transfer matrices of each configuration substantially. This reduction process only has to be done once because all the simplified transfer matrices can be saved in a library. Then, when calculating results for an ONoC design (which uses a specific combination of routing structure configurations) its transfer matrix can be built directly by combining the simplified transfer matrices from the library. When solving multiple $\hat{T}x = b$ systems for different ONoC designs based on the same structure library, the repeated effort caused by indirectly solving multiple copies of the same structure configurations is eliminated.

An example of a WRONoC built using GRUs is given. GRUs have a total of 73 possible internal configurations[8] of crossing, MRR and corner bend placements. Some examples are shown in fig. 6.5(a). According to section 6.3.1, a GRU requires eight compound vertices to fully describe its routing behaviour: four internal vertices and four port vertices. Thus, a naïve flow graph for a grid of GRUs requires 8 compound vertices per GRU. This is shown in fig. 6.5(b).

[8]Including rotations and reflections.

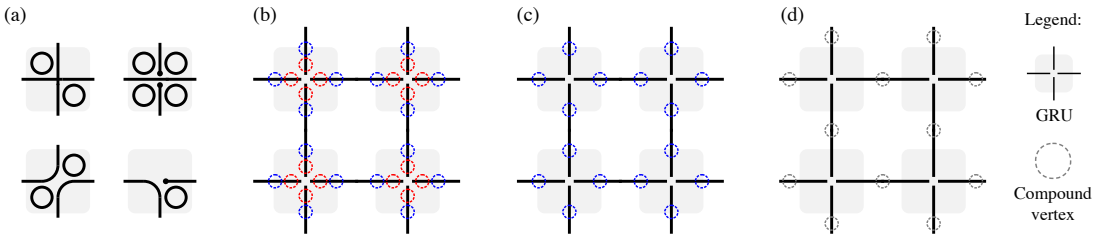

Figure 6.5: (a) Example GRU configurations (see section 4.1.2 for more details). (b) Location of the eight compound vertices on a GRU: four internal vertices (in red) and four port vertices (in blue). (c) Removing internal vertices. (d) Merging port vertices. Final result has up to 4× lower vertex count for GRU-based designs.

Since the routing structures are self-contained, reusable and finite in number, a library of their simplified versions can be built *a priori*. The process from section 6.2 is used to remove all internal vertices of each structure. For a GRU-based design, this results in a flow graph with only 4 compound vertices per GRU, i.e. a 2× reduction in vertex count. This is shown in fig. 6.5(c).

A further optimisation is possible when connecting the structures together: instead of having a compound vertex per port for each structure, have a compound vertex per pair of connected ports. The insertion loss of the waveguide section connecting the two ports is shared between both structures. In the 2×2 GRU grid example of fig. 6.5, merging port vertices brings the total number of compound vertices down to 12, as shown in fig. 6.5(d). This is a reduction of $\frac{16}{12} = 1.33\times$ compared to fig. 6.5(c), for a total reduction of $2 * 1.33 = 2.66\times$ compared to fig. 6.5(b). Note that compound vertices on the boundaries of a GRU grid cannot be merged. As the GRU grid size n increases, the number of vertices that cannot be merged increases with $O(n)$, but the number of vertices that can be merged increases with $O(n^2)$. Thus, this provides an asymptotical vertex count reduction of 2×, for a total reduction that approaches $2 * 2 = 4\times$.

More complex structures lead to larger matrix reductions. For example, a Crux router [2] used in an active ONoC has only 780 possible configurations regardless of wavelength count. By analysis of the Crux router's design together with the vertex placement rules described in section 6.3.1, it can be determined that each structure requires $24n + 34$ internal compound vertices and 10 port compound vertices, where n is the number of wavelengths. Since all internal vertices can be removed and most port vertices can be merged, this process leads to a reduction in vertex count of up to $\frac{24n+34+10}{10} \times 2$, which is at least 13.6× for $n = 1$ and already 56.8× for $n = 10$.

Table 6.1: Insertion loss and crosstalk technology parameter values from [2].

Insertion loss	
Propagation loss	0.274 dB/cm
Bend loss	0.005 dB/90°
Crossing loss	0.05 dB
Drop loss	1 dB
Through loss	0.005 dB

Crosstalk	
Terminator back reflection	50 dB
Crossing side spill	40 dB
Crossing back reflection	—
MRR on-resonance through	25 dB
MRR off-resonance drop	20 dB

6.5 Experimental results

This method was implemented in C++ and made use of Eigen [74], a C++ linear algebra library with sparse matrix solvers. To exemplify the use of this method, SNR results were calculated for the 16 node WRONoC design created in section 5.5.2. This design uses 17 wavelengths to provide connectivity between 16 network nodes with a total of $16 \times 15 = 240$ signals.[9] The design of this ONoC allows all signals to travel without contention, meaning that all 240 signals can be active at the same time. Naturally, this causes the highest amount of crosstalk in the router and is the case considered in the following results analysis. For this analysis the technology parameters from [2] shown in table 6.1 were used.

The process in section 6.3.5 was followed. First, 17 linear systems (one per wavelength) were solved to calculate the insertion loss of all 240 signals. Each signal is sent into the network with the minimum amount of energy required to overcome its insertion loss considering a demodulator sensitivity of $S = -20$ dBm (see section 3.1.1) [24]. Then, 17 linear systems were solved again to calculate the signal and noise energy levels in the network. To compare between first-order and infinite-order SNR, the second batch of 17 linear systems was solved twice: once with $T_\lambda^{SN_1}$ matrices and once with $T_\lambda^{SN_\infty}$ matrices. BiCGSTAB with an incomplete LU preconditioner was used to solve all linear systems.

The noise energy level that each signal's demodulator receives on each wavelength can be obtained from the resulting steady-state vectors. The noise received on the signal's wavelength is intra-channel noise and the sum of the noise received on other

[9] The network does not have communication paths from a node to itself.

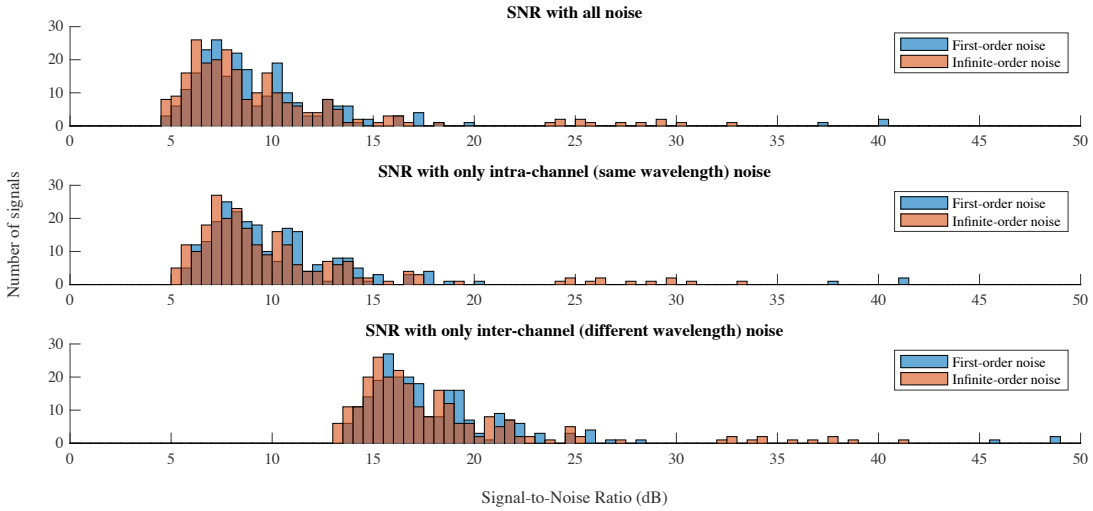

Figure 6.6: Histogram of SNR results for all 240 signals of a 17 wavelength WRONoC router.

wavelengths is inter-channel noise. A histogram of the SNR results for all signals is shown in fig. 6.6.

It is clear that in this design SNR is limited mostly by the intra-channel noise. While inter-channel noise energy levels are higher (there are more signals on more wavelengths compared to intra-channel noise), this noise is filtered just before arriving at the demodulator. The average (over all signals) infinite-order SNR with just inter-channel noise is 18.08 dB, with just intra-channel noise is 10.12 dB and with all noise is 9.47 dB. The effect of inter-channel noise becomes more pronounced as the number of used wavelengths increases.

On average, infinite-order SNR with all noise is only 0.65 dB lower than first-order SNR with all noise. However, nine signals do not receive any first-order noise, making their first-order SNR infinite, while in fact the average infinite-order SNR with all noise for these signals is 27.06 dB. Furthermore, even for signals whose first-order SNR is not infinite, the maximum difference between first-order SNR and infinite-order SNR (with all noise) is as high as $40.38 - 24.23 = 16.15$ dB. The conclusion is that, while for most signals the infinite order and first order results are close, some outliers can only be correctly characterised using infinite-order noise calculations. Infinite-order calculations are thus necessary for a full and accurate analysis of an ONoC.

Noise-enabled ($T_\lambda^{SN_*}$) transfer matrices obtained directly from the process in section 6.3.5 have a size of 5408×5408. Each first-order noise and infinite-order noise linear system is solved on average in 658 µs and 2232 µs respectively.[10] Using the matrix reduction techniques from section 6.4.2, the matrix size becomes 1376×1376 and each system is solved on average in 123 µs and 332 µs respectively. This is a 3.93×

[10] Using a 2.3 GHz 8-Core Intel Core i9 on a single thread.

reduction in matrix size (approximately 4× as expected from GRU based designs) and a 5.35× and 6.72× increase in speed respectively.

6.5.1 Steady-state flow map

The steady-state result vector x can be used for more than just calculate SNR results. In general, it contains the steady-state energy levels for all vertices in the flow graph. All this information can be superimposed on the flow graph itself to create useful visualisations. For example, in the case of ONoCs, it can convey graphically the steady-state optical energy distribution on a per signal basis, per wavelength basis, or in total. An example is shown in fig. 6.7. Furthermore, by using the flow propagation relationships in eqs. (6.7) and (6.8), flow animations can also be produced frame by frame. This provides ONoC designers with powerful analysis capabilities in a very hands-on approach, such as discerning whether the ONoC contains any optical energy hotspots, understanding which ONoC elements are most responsible for noise generation, exploring how noise flows through the ONoC and from there deriving the best locations for design modifications (for example, adding extra MRRs and waveguide terminators to filter noise on certain wavelengths).

6.6 Summary

This chapter described a new general method to analyse the steady-state of a system. This method can be applied to any system obeying a few simple rules, namely not having any perfect stores of energy. The steady-state is obtained by solving $\hat{T}x = b$ with a transfer matrix obtained from the system's flow graph. This chapter also derives a process through which the flow graph and transfer matrix can be reduced in complexity without affecting the steady-state results.

This method was applied to the calculation of first-order and infinite-order noise and SNR of any ONoC design. It is quick and efficient through the use of sparse matrices and methods for sparse linear systems and accurate by supporting the wavelength-dependent versions of the insertion loss and crosstalk factors. It also allows for powerful analysis and visualisation capabilities.

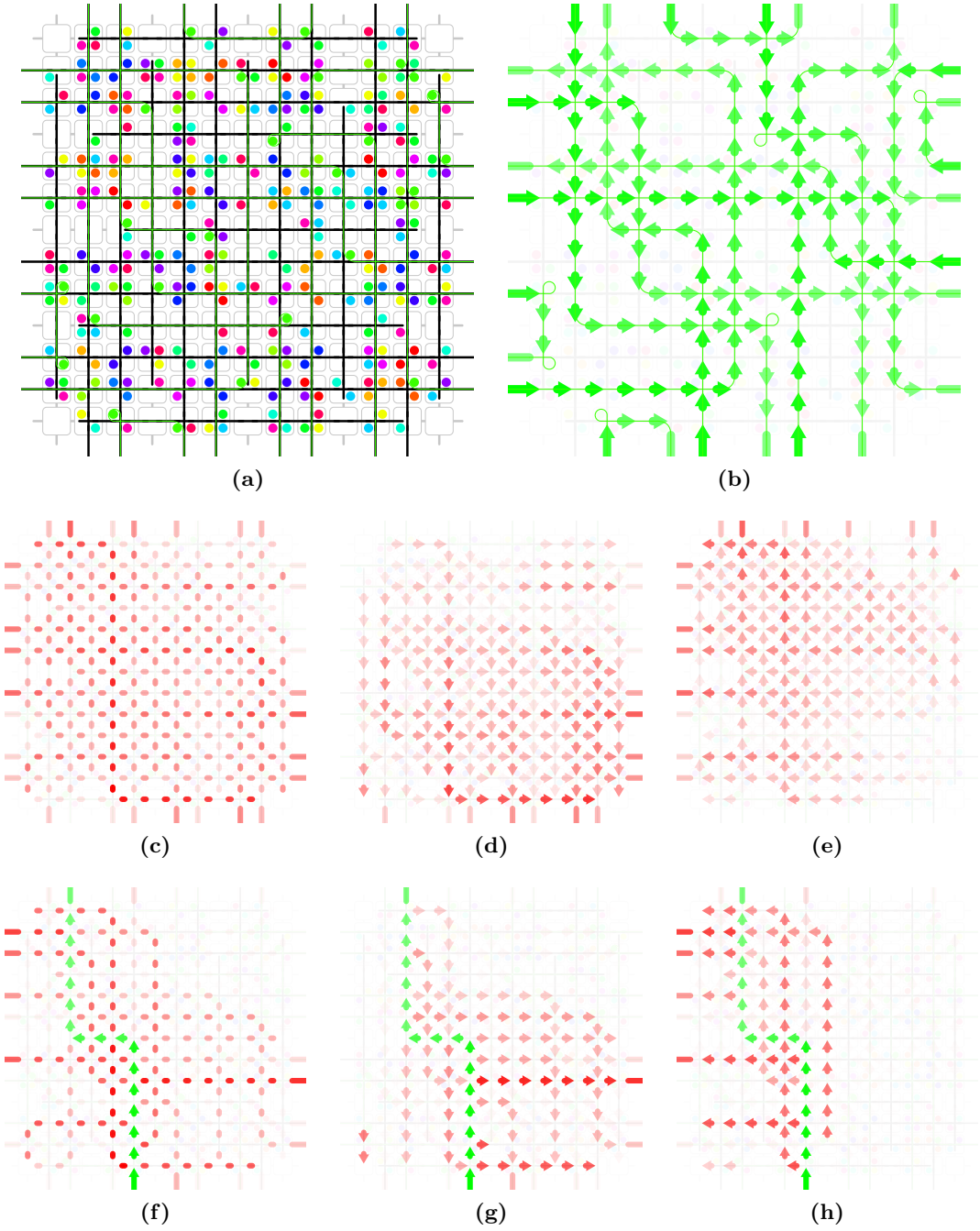

Figure 6.7: Breakdown of noise generation and propagation in a router using a steady-state flow map. (a) Complete router design. The modulators and demodulators of the 16 nodes are connected to the sides of the router. Circles are MRRs and color indicates their resonance wavelength. (b) Signal paths of one specific wavelength. (c) Distribution of infinite-order noise on the router waveguides caused by the signals in (b). Stronger red indicates a higher noise level. (d-e) Noise levels in (c) discriminated by direction. (f-h) Same as (c-e) but for one signal only.

7 Conclusion

The objective of this work was to advance the state of the art in WRONoC design automation. Therefore, it began with a review of the WRONoC design problem. A plan was then laid out for incrementally developing a complete, no-compromise and turn-key WRONoC optimisation process. Finally, some of the first steps in this plan were executed.

The first major idea presented in this work is the PLT (which includes the new GRU building block). Its development was driven by the need to enable simultaneous optimisation of logical topology and physical layout and allow direct and complete optimisation of power consumption. While chapters 4 and 5 successfully showed this idea achieves those requirements and thus raises the quality of state of the art WRONoC designs, it is also crucial to consider the trade-offs the PLT creates, especially with regards to the no-compromise and turn-key goals.

Much like the simplifications explained in section 3.3, the PLT constrains the optimisation process in the following ways:

- It limits the maximum number of MRRs, MAs, DMAs and waveguides of the final design: it is always possible to use fewer elements than contained in the PLT, but never more.

- It restricts the location of elements (a limitation that is partially lifted in chapter 5).

- It fixes the structure of the PDN: the optimisation process does not have the flexibility to change the PDN present in the PLT during optimisation.

Additionally, the PLT, as presented in chapters 4 and 5, is not entirely turn-key either. Even the structures whose design can be more easily automated (the GTs, for example) may require expansion to support dense CMs of larger WRONoCs. In the end, some (small) effort on the part of the designer is always required. In fact, this is the comment most frequently made on the PSION family of tools by later published work.

Fortunately, there is one PLT structure that strongly mitigates these issues: cover the optical routing layer in an homogeneous grid of GRUs. An example of this structure applied to the test case from section 4.5.1 is shown in fig. A.1. This structure has multiple advantages:

- Its design can be easily automated, requiring minimal manual effort: the WRONoC designer only needs to choose the (maximum) number of MAs and DMAs per node, the PDN structure, and the spacing of the GRU grid.
- There is no longer a need to divide the optical routing layer into a "router area" and a "PDN area" (see section 4.3.1). Instead, both are allowed to coexist and any crossings created between them during optimisation are automatically considered in the power consumption calculations. This creates potential trade-offs that the optimisation process can explore, such as choosing between shorter signal paths that cross the PDN and longer detours that avoid the crossings but increase propagation loss.[1]
- It allows for the constraints created by the PLT to be incrementally lifted solely through improving the PLT optimisation process.

  P&R of WRONoC elements is inherently a continuous problem because the coordinates of each element are continuous values. By using a homogeneous GRU grid, the PLT is effectively approximating a continuous problem with a discrete version where the approximation error is directly proportional to the grid spacing.

  This fact "future-proofs" the PLT idea. Sparser grids contain fewer GRUs and are therefore easier and faster to optimise. However, as PLT optimisation processes evolve, denser grids (which are expected to produce better results) also become feasible. Researchers can focus exclusively on improving these processes and, as grid density is increased over time, the difference between the discrete and continuous versions is likely to become negligible for all practical purposes.[2]
- Related to the previous point, this structure encompasses all other PLT structures with decreasing error as grid density increases, which means that optimisation of a large GRU grid supersedes optimisation of any other template.[3] In other words, the WRONoC designer no longer needs to carry the PLT synthesis burden, knowing that the optimisation process will carve out the best design from the GRU grid for each case.

The PLT idea and this specific structure has been explored by works published after PSION 2, to very good results (although many of the simplifications explained in section 3.3 are still used) [75, 76].

The main reason currently holding back the PSION family of tools from exploring large GRU grids is that the LP method used in chapters 4 and 5 cannot handle that amount of complexity. LP was essential to allow for rapid prototyping at a stage when the effectiveness of the PLT had not yet been proven, but its limits eventually revealed themselves:

[1] Dividing the optical routing layer into two effectively forces this second option, as is evident in fig. 4.7.
[2] The modelling complexity of optimising GRU positioning (see section 5.4) is also no longer needed.
[3] One could say, one structure to rule them all.

- It cannot handle larger amounts of complexity, such as a higher number of GRUs in the PLT or signals in the CM (as stated in section 3.2.4.2, ideally up to approximately 4000).
- It is slower with corner bending turned on.
- It cannot optimise for non-linear functions, which is required for exact power consumption optimisation, SNR calculation, and for some of the bit parallelism optimisation functions in section 3.1.3.
- It most likely cannot model crosstalk to the same level of accuracy as the method in chapter 6, and even if it can it will almost surely result in a prohibitively large LP model.

Clearly, a new PLT optimisation process is required and work has already started in this direction. The design in fig. A.1 was produced by a prototype optimisation process based on local search that is focused on scalability and optimisation of non-linear functions. Its goal is to completely fulfil the three items that belong to the first step mentioned in section 3.3.5.

Simultaneously, progress is being made on the third item of the second step: using the method from chapter 6 as part of the optimisation function of a process that optimises bit parallelism and crosstalk. The previously available bit parallelism optimisation tools also rely on LP and suffer from the same issues [28], so research in this area is also deeply focused on scalability.

As explained in section 3.3.5, success in both of these efforts results in two optimisation tools that can be used in sequence to create WRONoC designs. At this point, it may be possible to combine these tools and produce a complete, no-compromise and turn-key WRONoC optimisation process.

Bibliography

[1] M. Ortín-Obón et al. Contrasting laser power requirements of wavelength-routed optical noc topologies subject to the floorplanning, placement, and routing constraints of a 3-d-stacked system. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pages 2081–2094, July 2017. doi:10.1109/TVLSI.2017.2677779.

[2] M. Nikdast et al. Crosstalk noise in wdm-based optical networks-on-chip: A formal study and comparison. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pages 2552–2565, Nov 2015. doi:10.1109/TVLSI.2014.2370892.

[3] I. O'Connor et al. Towards reconfigurable optical networks on chip. In *Proceedings of the 1st International Workshop on Reconfigurable Communication-centric Systems-on-Chip*, pages 121–128, 2005.

[4] A. Shacham et al. Photonic networks-on-chip for future generations of chip multiprocessors. *IEEE Transactions on Computers*, 57(9):1246–1260, 2008. doi:10.1109/tc.2008.78.

[5] M. Tala et al. Populating and exploring the design space of wavelength-routed optical network-on-chip topologies by leveraging the add-drop filtering primitive. In *2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip*, pages 1–8, Aug 2016. doi:10.1109/NOCS.2016.7579331.

[6] A. Truppel et al. Psion+: Combining logical topology and physical layout optimization for wavelength-routed onocs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 5197–5210, 2020. doi:10.1109/TCAD.2020.2971536.

[7] A. Truppel et al. Psion 2: Optimizing physical layout of wavelength-routed onocs for laser power reduction. In *Proceedings of the 39th International Conference on Computer-Aided Design*, 2020. doi:10.1145/3400302.3415655.

[8] A. Truppel et al. Accurate infinite-order crosstalk calculation for optical networks-on-chip. *Journal of Lightwave Technology*, pages 1–13, 2022. doi:10.1109/JLT.2022.3210159.

[9] A. von Beuningen et al. Proton+: A placement and routing tool for 3d optical networks-on-chip with a single optical layer. *J. Emerg. Technol. Comput. Syst.*, pages 44:1–44:28, Dec 2015. doi:10.1145/2830716.

[10] M. J. R. Heck et al. Energy efficient and energy proportional optical interconnects for multi-core processors: Driving the need for on-chip sources. *IEEE Journal of Selected Topics in Quantum Electronics*, 20(4):332–343, July 2014. doi:10.1109/JSTQE.2013.2293271.

[11] C. Chen et al. Sharing and placement of on-chip laser sources in silicon-photonic nocs. 09 2014. doi:10.1109/NOCS.2014.7008766.

[12] F. Liu et al. Dynamic ring-based multicast with wavelength reuse for optical network on chips. pages 153–160, 2016. doi:10.1109/MCSoC.2016.9.

[13] S. Bahirat et al. Exploring hybrid photonic networks-on-chip foremerging chip multiprocessors. pages 129–136. ACM.

[14] S. L. Beux et al. Optical ring network-on-chip (ornoc): Architecture and design methodology. In *2011 Design, Automation & Test in Europe*, pages 1–6. doi:10.1109/DATE.2011.5763134.

[15] L. B. Sébastien et al. Optical crossbars on chip, a comparative study based on worst-case losses. *Concurrency and Computation: Practice and Experience*, 26(15):2492–2503, 2014. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.3336, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.3336, doi:10.1002/cpe.3336.

[16] C. Chen et al. Sharing and placement of on-chip laser sources in silicon-photonic nocs. In *2014 Eighth IEEE/ACM International Symposium on Networks-on-Chip (NoCS)*, pages 88–95, Sept 2014. doi:10.1109/NOCS.2014.7008766.

[17] A. Truppel. Layout aware router design and optimization for wavelength-routed optical nocs. Master's thesis, 2018.

[18] S. Werner et al. A survey on optical network-on-chip architectures. *ACM Comput. Surv.*, 50(6), dec 2017. URL: https://doi.org/10.1145/3131346, doi:10.1145/3131346.

[19] Q. Cheng et al. Recent advances in optical technologies for data centers: A review. *Optica*, 5:1354–1370, 11 2018. doi:10.1364/OPTICA.5.001354.

[20] M. Georgas et al. A monolithically-integrated optical receiver in standard 45-nm soi. In *2011 Proceedings of the ESSCIRC (ESSCIRC)*, pages 407–410, Sept 2011. doi:10.1109/ESSCIRC.2011.6044993.

[21] C. Batten et al. Building many-core processor-to-dram networks with monolithic cmos silicon photonics. *IEEE Micro*, 29(4):8–21, 2009. doi:10.1109/MM.2009.60.

[22] C. Gunn. Cmos photonics for high-speed interconnects. *IEEE Micro*, 26(2):58–66, 2006. doi:10.1109/MM.2006.32.

[23] W. Pauli. Über den zusammenhang des abschlusses der elektronengruppen im atom mit der komplexstruktur der spektren. *Zeitschrift für Physik*, 31(1):765–783, Feb 1925. doi:10.1007/BF02980631.

[24] M. Ortín-Obón et al. A tool for synthesizing power-efficient and custom-tailored wavelength-routed optical rings. In *Asia and South Pacific Design Automation Conference*, pages 300–305, Jan 2017. doi:10.1109/ASPDAC.2017.7858339.

[25] A. Peano et al. Design technology for fault-free and maximally-parallel wavelength-routed optical networks-on-chip. In *2016 IEEE/ACM International Conference on Computer-Aided Design*, pages 1–8, Nov 2016. doi:10.1145/2966986.2967023.

[26] B. Little et al. Microring resonator channel dropping filters. *Journal of Lightwave Technology*, pages 998–1005, 1997. doi:10.1109/50.588673.

[27] M. Tala et al. Understanding the design space of wavelength-routed optical noc topologies for power-performance optimization. In *2018 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, pages 255–260, 2018. doi:10.1109/VLSI-SoC.2018.8644998.

[28] M. Li et al. Maximizing the communication parallelism for wavelength-routed optical networks-on-chips. In *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 109–114, 2020. doi:10.1109/ASP-DAC47756.2020.9045163.

[29] S. Xiao et al. Modeling and measurement of losses in silicon-on-insulator resonators and bends. *Opt. Express*, pages 10553–10561, Aug 2007. doi:10.1364/OE.15.010553.

[30] D. Rabus. *Realization of optical filters using ring resonators with integrated semiconductor optical amplifiers in GaInAsP/InP*. PhD thesis, 01 2002. doi:10.14279/depositonce-565.

[31] A. Mirza et al. Opportunities for cross-layer design in high-performance computing systems with integrated silicon photonic networks. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1622–1627, 2020. doi:10.23919/DATE48585.2020.9116234.

[32] J. Ahn et al. Devices and architectures for photonic chip-scale integration. *Applied Physics A*, 95(4):989–997, Jun 2009. doi:10.1007/s00339-009-5109-2.

[33] M. Ortín-Obón et al. A complete electronic network interface architecture for global contention-free communication over emerging optical networks-on-chip. In *Proceedings of the 24th Edition of the Great Lakes Symposium on VLSI*, pages 267–272, May 2014. doi:10.1145/2591513.2591536.

[34] Q. Xu et al. Micrometre-scale silicon electro-optic modulator. *Nature*, 435(7040):325–327, May 2005. URL: https://doi.org/10.1038/nature03569, doi:10.1038/nature03569.

[35] G. Masini et al. A 1550nm, 10gbps monolithic optical receiver in 130nm cmos with integrated ge waveguide photodetector. In *2007 4th IEEE International Conference on Group IV Photonics*, pages 1–3, 2007. doi:10.1109/GROUP4.2007.4347656.

[36] Y. Xie et al. Crosstalk noise and bit error rate analysis for optical network-on-chip. In *Proceedings of the 47th Design Automation Conference*, pages 657–660, Jan 2010. doi:10.1145/1837274.1837441.

[37] H. Gu et al. A low-power low-cost optical router for optical networks-on-chip in multiprocessor systems-on-chip. In *2009 IEEE Computer Society Annual Symposium on VLSI*, pages 19–24, May 2009. doi:10.1109/ISVLSI.2009.19.

[38] R. Ji et al. Five-port optical router for photonic networks-on-chip. *Opt. Express*, 19(21):20258–20268, Oct 2011. URL: https://opg.optica.org/oe/abstract.cfm?URI=oe-19-21-20258, doi:10.1364/OE.19.020258.

[39] H. Gu et al. A low-power fat tree-based optical network-on-chip for multiprocessor system-on-chip. In *2009 Design, Automation Test in Europe Conference Exhibition*, pages 3–8, 2009. doi:10.1109/DATE.2009.5090624.

[40] Y. Ye et al. 3-d mesh-based optical network-on-chip for multiprocessor system-on-chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(4):584–596, 2013. doi:10.1109/TCAD.2012.2228739.

[41] K. Zhu et al. Votex: A non-blocking optical router design for 3d optical network on chip. In *2015 14th International Conference on Optical Communications and Networks (ICOCN)*, pages 1–3, 2015. doi:10.1109/ICOCN.2015.7203771.

[42] Y. Xie et al. Formal worst-case analysis of crosstalk noise in mesh-based optical networks-on-chip. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pages 1823–1836, 2013. doi:10.1109/TVLSI.2012.2220573.

[43] T. Song et al. Crosstalk analysis and performance evaluation for torus-based optical networks-on-chip using wdm. *Micromachines*, 11(11), 2020. URL: https://www.mdpi.com/2072-666X/11/11/985, doi:10.3390/mi11110985.

[44] Y. Wang et al. An optimized optical router for mesh based optical networks-on-chip. In *2015 14th International Conference on Optical Communications and Networks (ICOCN)*, pages 1–3, 2015. doi:10.1109/ICOCN.2015.7203618.

[45] X. Tan et al. On a scalable, non-blocking optical router for photonic networks-on-chip designs. In *2011 Symposium on Photonics and Optoelectronics*, pages 1–4, May 2011. doi:10.1109/SOPO.2011.5780550.

[46] L. Ramini et al. Contrasting wavelength-routed optical noc topologies for power-efficient 3d-stacked multicore processors using physical-layer analysis. In *2013 Design, Automation & Test in Europe Conference & Exhibition*, pages 1589–1594, March 2013. doi:10.7873/DATE.2013.323.

[47] L. Huang et al. Lace: A non-blocking on-chip optical router by utilizing the wavelength routing technology. In *2017 16th International Conference on Optical Communications and Networks (ICOCN)*, pages 1–3, 2017. doi:10.1109/ICOCN.2017.8121405.

[48] Y. Yang et al. Taonoc: A regular passive optical network-on-chip architecture based on comb switches. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 27(4):954–963, 2019. doi:10.1109/TVLSI.2018.2885141.

[49] M. Tala et al. Exploring communication protocols for optical networks-on-chip based on ring topologies. In *Asia Communications and Photonics Conference 2014*, page ATh3A.165. Optica Publishing Group, 2014. URL: https://opg.optica.org/abstract.cfm?URI=ACPC-2014-ATh3A.165, doi:10.1364/ACPC.2014.ATh3A.165.

[50] M. Ortin et al. Partitioning strategies of wavelength-routed optical networks-on-chip for laser power minimization. pages 17–24, 2015. doi:10.1109/SiPhotonics.2015.13.

[51] X. Wang et al. A group-based laser power supply scheme for photonic network on chip. *IEEE Photonics Journal*, 11(3):1–14, 2019. doi:10.1109/JPHOT.2019.2913676.

[52] J. Luo et al. Offline optimization of wavelength allocation and laser power in nanophotonic interconnects. *J. Emerg. Technol. Comput. Syst.*, 14(2), jul 2018. URL: https://doi.org/10.1145/3178453, doi:10.1145/3178453.

[53] M. Nikdast et al. Fat-tree-based optical interconnection networks under crosstalk noise constraint. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pages 156–169, 2015. doi:10.1109/tvlsi.2014.2300534.

[54] Y. W. Kim et al. Extended worst-case osnr searching algorithm for optical network-on-chip using a semi-greedy heuristic with adaptive scan range. *IEEE Access*, pages 125863–125873, 2020. doi:10.1109/ACCESS.2020.3007148.

[55] R. Ramaswami et al. *Optical Networks A Practical Perspective 3ed.* Elsevier, 2010. doi:10.1016/C2009-0-17339-7.

[56] R. Cao et al. A crosstalk-aware wavelength assignment method for optical network-on-chip. *IEICE Electronics Express*, 13(18):20160821–20160821, 2016. doi:10.1587/elex.13.20160821.

[57] A. Jalabert et al. ×pipesCompiler: a tool for instantiating application specific networks on chip. In *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, volume 2, pages 884–889 Vol.2, Feb 2004. doi:10.1109/DATE.2004.1268999.

[58] Z. Wang et al. Moca: An inter/intra-chip optical network for memory. In *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, 2017. doi:10.1145/3061639.3062286.

[59] M. Li et al. Customtopo: A topology generation method for application-specific wavelength-routed optical nocs. In *Proceedings of the 37th International Conference on Computer-Aided Design*, pages 1–8, 11 2018. doi:10.1145/3240765.3240789.

[60] F. Jiao et al. Thermal-aware placement and routing for 3d optical networks-on-chips. In *2018 IEEE International Symposium on Circuits and Systems*, pages 1–4, May 2018. doi:10.1109/ISCAS.2018.8351101.

[61] A. Boos et al. Proton: An automatic place-and-route tool for optical networks-on-chip. In *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 138–145, 2013.

[62] A. von Beuningen et al. Platon: A force-directed placement algorithm for 3d optical networks-on-chip. In *Proceedings of the 2016 on International Symposium on Physical Design*, pages 27–34, April 2016. doi:10.1145/2872334.2872356.

[63] Y. Chuang et al. Planaronoc: Concurrent placement and routing considering crossing minimization for optical networks-on-chip. In *Proceedings of the 55th Annual Design Automation Conference*, pages 151:1–151:6, June 2018.

[64] C. Qiu, Y. Liu, et al. Wavelength-routed optical network with interleave microring switches for optical network-on-chips. In *2014 Optical Interconnects Conference*, pages 67–68, 2014. doi:10.1109/OIC.2014.6886074.

[65] S. L. Beux et al. Reduction methods for adapting optical network on chip topologies to 3d architectures. *Microprocessors and Microsystems*, pages 87 – 98, Feb. 2013. doi:10.1016/j.micpro.2012.11.001.

[66] I. O'Connor et al. Reduction methods for adapting optical network on chip topologies to specific routing applications. 2007.

[67] D. Ding et al. O-router: An optical routing framework for low power on-chip silicon nano-photonic integration. pages 264–269, 01 2009. doi:10.1145/1629911.1629983.

[68] B. G. Lee et al. All-optical comb switch for multiwavelength message routing in silicon photonic networks. *IEEE Photonics Technology Letters*, 20(10):767–769, 2008. doi:10.1109/LPT.2008.921100.

[69] M. Xiao et al. Crosstalk-aware automatic topology customization and optimization for wavelength-routed optical nocs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022. doi:10.1109/TCAD.2022.3151247.

[70] A. Truppel et al. Psion: Combining logical topology and physical layout optimization for wavelength-routed onocs. In *Proceedings of the 2019 International Symposium on Physical Design*, pages 49–56, April 2019. doi:10.1145/3299902.3309747.

[71] D. E. Knuth. Dancing links. *arXiv e-prints*, page cs/0011047, November 2000.

[72] Gurobi Optimization, Inc. *Gurobi Optimizer Reference Manual*. http://www.gurobi.com, 2018.

[73] T.-T. Lu et al. Inverses of 2×2 block matrices. *Computers & Mathematics with Applications*, pages 119–129, 2002. doi:10.1016/S0898-1221(01)00278-4.

[74] Eigen Development Team. Eigen, 2022. Version 3.4.0. URL: https://eigen.tuxfamily.org.

[75] Y.-S. Lu et al. Topological structure and physical layout codesign for wavelength-routed optical networks-on-chip. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6, 2020. doi:10.1109/DAC18072.2020.9218625.

[76] Y.-S. Lu et al. Topological structure and physical layout co-design for wavelength-routed optical networks-on-chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(7):2237–2249, 2022. doi:10.1109/TCAD.2021.3101145.

A Optimised PLT examples

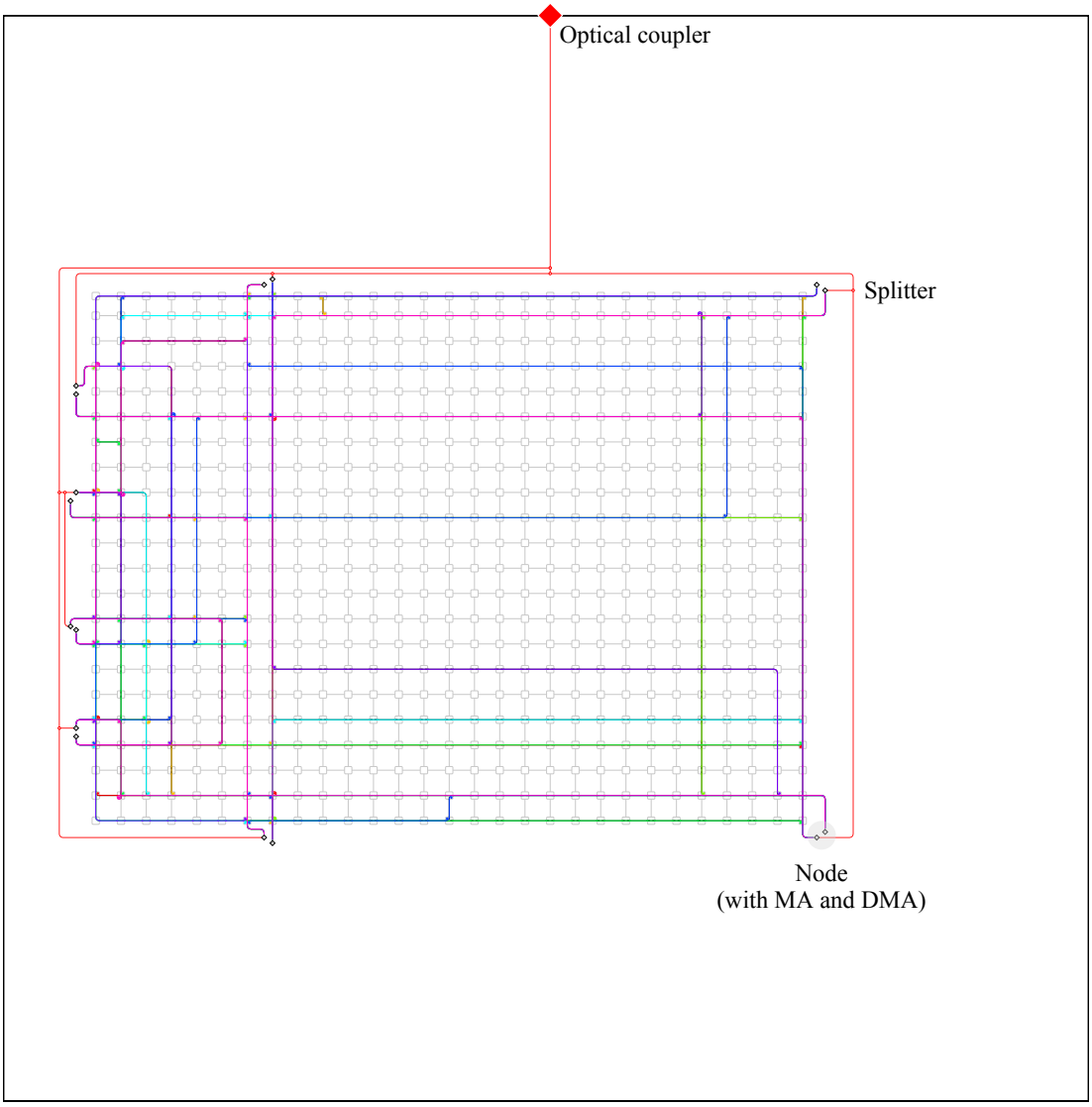

Figure A.1: Example of a WRONoC design using a PLT consisting of a large and homogeneous grid of GRUs. In this particular example the grid covers only a portion of the optical routing layer, but grids covering larger areas can also be used.

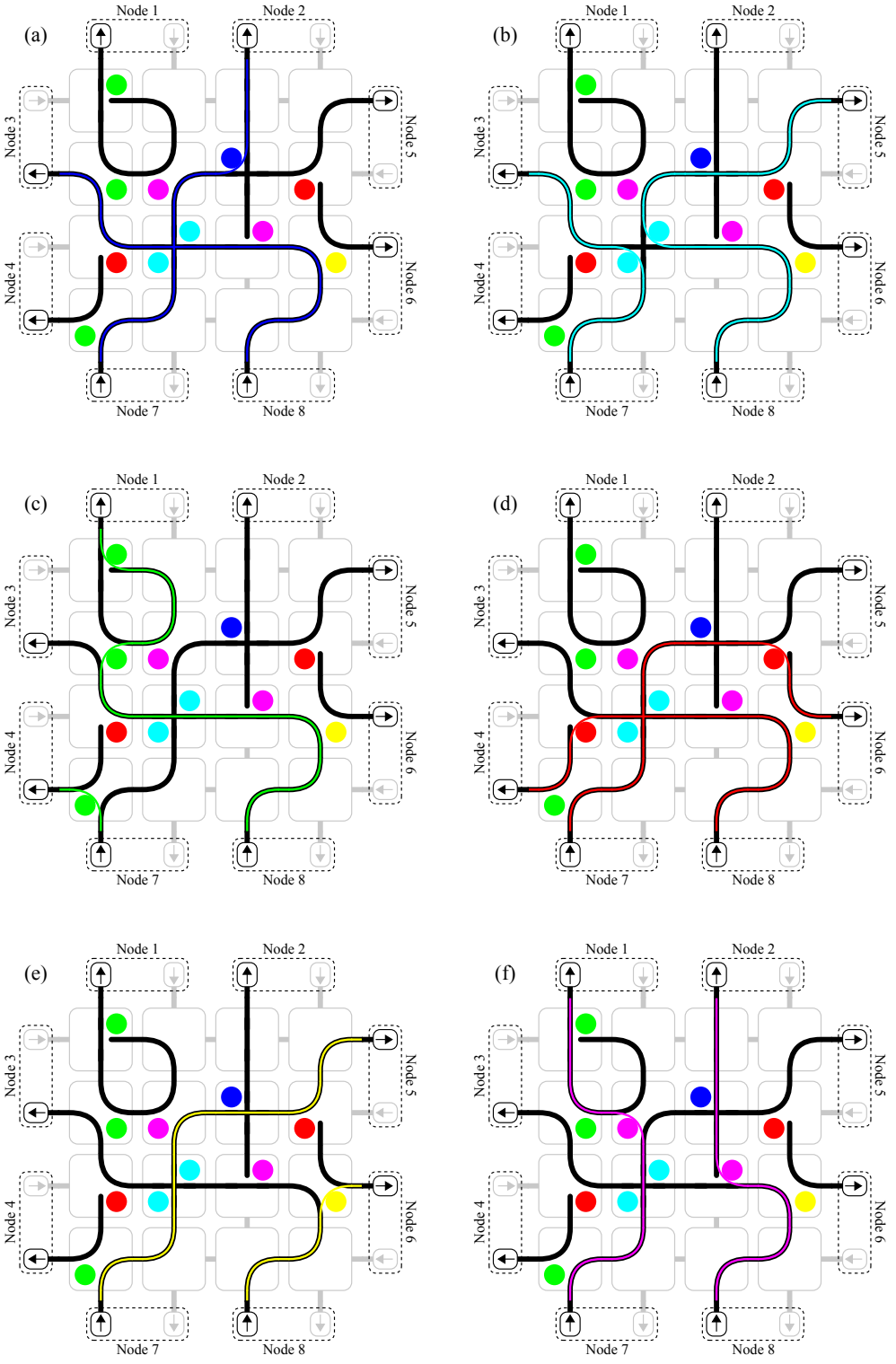

Figure A.2: Example of an optimal WRONoC design for an 8 node network with a sparse CM based on a GT and using corner bending. Each sub-figure shows the signal paths of each one of the six wavelengths.