This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2020.2971536, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

PSION+: Combining logical topology and physical layout optimization for Wavelength-Routed ONoCs

Alexandre Truppel, Tsun-Ming Tseng, *Member, IEEE*, Davide Bertozzi, *Member, IEEE*, José Carlos Alves, and Ulf Schlichtmann, *Senior Member, IEEE*

***Abstract*—Optical Networks-on-Chip (ONoCs) are a promising solution for high-performance multi-core integration with better latency and bandwidth than traditional Electrical NoCs. Wavelength-routed ONoCs (WRONoCs) offer yet additional performance guarantees. However, WRONoC design presents new EDA challenges which have not yet been fully addressed. So far, most topology analysis is abstract, i.e. overlooks layout concerns, while for layout the tools available perform Place and Route (P&R) but no topology optimization. Thus, a need arises for a novel optimization method combining both aspects of WRONoC design. In this paper such a method, PSION+, is laid out. This new procedure uses a linear programming model to optimize a WRONoC physical layout template to optimality. This template-based optimization scheme is a new idea in this area that seeks to minimize problem complexity while keeping design flexibility. A simple layout template format is introduced and explored. Finally, multiple model reduction techniques to reduce solver run-time are also presented and tested. When compared to the state-of-the-art design procedure, results show a decrease in maximum optical insertion loss of 41%.**

***Index Terms*—optical networks-on-chip, mixed integer linear programming.**

I. Introduction

Optical Networks-on-Chip (ONoCs) have been proposed as a solution for the ever-increasing integration requirements of large System-on-Chip designs. Compared to traditional Electrical Networks-on-Chip, ONoCs present not only lower dynamic power consumption but also extremely low signal delay and higher bandwidth [1].

The use of light as opposed to electrical signals to send information between network nodes requires the following four main components on the optical layer: **1)** *modulators* to convert electrical signals into optical signals at every node (electrical-optical interface) of the optical network, **2)** *demodulators* to do the opposite, **3)** *waveguides* acting as optical wires and **4)** *optical routing elements* to transfer optical signals between waveguides [2].

ONoCs can be organized into two main categories: **1)** *active networks* [3]–[5] and **2)** *passive networks*, also termed Wavelength-Routed ONoCs (WRONoCs). Active networks require a control layer for routing. Passive networks use routing elements which resonate with different frequencies such that a message is passively routed according to the wavelength of the carrier light. Hence, a message's path is completely defined, at design time, by its origin and wavelength alone. This eliminates network delay resulting from path setup and dynamic power consumption required for the extra control layer, and enables a simple all-optical implementation of the interconnection fabric.

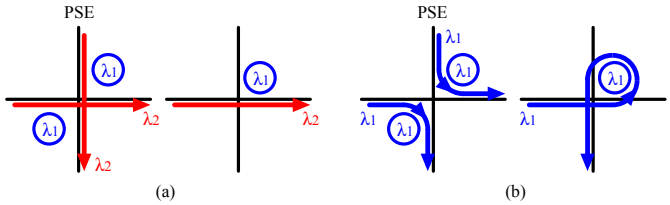


Figure 1. Wavelength routing using an MRR. The configuration with two MRRs is called a Photonic Switching Element (PSE). (a) The light signal continues its path on the same waveguide because it has a different wavelength than the resonant one of the MRR. (b) The light signal is routed through the MRR to another intersecting waveguide.

WRONoCs trade performance predictability with scalability limitations. In fact, the more I/O connections are to be established, the larger the number of laser sources required to power them. It has been demonstrated that passive networks turn out to be more power efficient with respect to active optical networks only for systems up to 16 nodes [6]. Furthermore, current technology also struggles to provide high bandwidth for all possible connections between high amounts of nodes simultaneously [7]. However, these limitations are actually not severe, since the energy cost for E/O and O/E conversions is such that providing optical network access with core granularity is currently not realistic. Typically, processing cores are aggregated in clusters of 8, 16 or more, and given a shared access to a network hub. As a result, a 16-node WRONoC may reasonably interconnect a manycore system with up to 256 cores.

In wavelength-routed topologies, multiple light sources of different wavelengths can be used to transmit separate data streams on the same waveguide without interference (wavelength-division multiplexing). This enables conflict-free communications with increased bandwidth. The only requirement is to make sure at design time that no two messages with the same wavelength are allowed to share the same waveguides.

The optical switching element in ONoCs is the Micro-Ring Resonator (MRR). It has a circular silicon structure whose radius defines the periodic transmittance characteristic (i.e. the resonance frequencies). A light signal with a certain wavelength propagating on a waveguide close to an MRR with a matching resonance frequency will be coupled to the MRR

The preliminary version of this paper was published in the Proceedings of the 2019 International Symposium on Physical Design.

Alexandre Truppel, Tsun-Ming Tseng, and Ulf Schlichtmann are with the Institute for Electronic Design Automation, Technical University of Munich, Arcisstr. 21, Munich 80333, Germany (e-mail: alex.truppel@tum.de; tsun-ming.tseng@tum.de; ulf.schlichtmann@tum.de).

Davide Bertozzi is with University of Ferrara, Via Ludovico Ariosto 35, 44121 Ferrara, Italy (e-mail: davide.bertozzi@unife.it).

Joseé Carlos Alves is with Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal (e-mail: jca@fe.up.pt).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2020.2971536, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

and moved onto another waveguide also close to that MRR [7]. Figure 1 shows an example of this behavior.

The design of a WRONoC router is an optimization process with *two aspects* to consider: the logical topology and the physical layout of the router. The former assigns a wavelength to each message and each MRR and also connects the nodes through waveguides and MRRs such that the communication matrix, which specifies the communication requirements between nodes, is fulfilled. The latter optimally places and routes those elements on the optical layer while considering the physical positions of the nodes and constraints related to the physical placement of the waveguides.

So far both aspects have only been considered separately or with restrictions. However, neither aspect should be considered in isolation, as each influences the other [8]–[10]. During generation of the logical topology we are unable to accurately predict important physical characteristics, e.g. the number of waveguide crossings, of the final design after P&R. Furthermore, during P&R, if the logical topology has already been chosen and fixed, any subsequent optimization is being done only around a local minimum of the solution space.

Ideally, a design tool would take as inputs the communication matrix and the physical positions of the nodes and, by working on both aspects simultaneously, produce a fully-optimized fully-custom logical topology and a matching physical layout. In reality, the problem space of such an optimization is discouragingly vast for any but the simplest cases. Thus, in this paper we propose and solve a constrained version of the complete problem. In this version, PSION+, a physical layout template is also given as an input to the optimization. The template mainly consists of MRR and waveguide placeholders already placed and routed on the optical layer, and connects all nodes. By optimizing within the programmable features of the template, we improve power-efficiency of state-of-the-art WRONoC topologies with an affordable optimization time even for the largest realistic network sizes.

The major contributions of PSION+ are the following:

- We propose a new way of designing WRONoCs using a physical layout template which enables simultaneous optimization of synthesis, placement and routing.
- We introduce a new basic element used in physical layout templates, the General Routing Unit (GRU), which is much more customizable than the Photonic Switching Element (PSE) used thus far in the literature.
- We establish a Mixed Integer Programming (MIP) model to solve any combination of physical layout template and communication matrix for its corresponding *optimal solution*.
- We analyze a straightforward physical layout template structure, the grid template, that leads to simple yet effective WRONoC designs. We highlight its strengths and weaknesses and we mention some solutions to the latter.
- We develop heuristic methods that can reduce runtime by multiple orders of magnitude with no meaningful impact on solution quality, thus bringing even the largest realistic WRONoC instances within reach of our methodology.

We define the optimization problem in Section III. Physical layout templates are described in Section IV and the Mixed Integer Programming (MIP) model used to optimize them is presented in Section V. Section VI provides an in-depth analysis of a simple yet flexible template format: the grid template. Section VII then proposes multiple heuristics to speed up solver run-time. Section VIII compares PSION+ against the state-of-the-art P&R tools PROTON+ [10] and PlanarONoC [11] and analyzes the performance of the proposed grid template design and solver heuristics on realistic network sizes. Finally, conclusions are drawn in Section IX along with a brief enumeration of future research directions.

II. Related work

As stated above, the two design aspects have not yet been considered together. Regarding the logical topology, various works have presented specific topologies with few concerns about their layout [2], [9], [12]. Ramini et al. [8] presented a topology designed in tandem with placement constraints, yet it results from a manual optimization effort for one specific set of node positions. Ortín-Obón et al. [1] took into consideration physical constraints, but analyzed only the ring topology. The same authors later extended their work to more topologies, but still based their designs on manual layout for one specific set of node positions [13]. Hui Li et al. [14] also studied the physical layout of WRONoC topologies but did not consider non-complete communication matrices. On the other hand, attempts to optimize for non-complete communication matrices do not include layout constraints in their optimization [15], [16]. Regarding the physical layout, tools to optimize the second aspect have been developed. Some perform placement and routing [10], [11], [17] and some perform only routing [18], but all take a topology as the input, forcing the designer to choose the topology beforehand.

III. WRONoC design problem

We formally define the optimization problem for the design of WRONoC routers as follows.

A. Input data

- Communication matrix (CM): a square binary matrix $M_{i,j} \in \mathbb{R}^{N \times N}$ with N being the number of nodes and where $M_{i,j} = 1$ if node i sends a message to node j.
- Physical positions of the modulators and demodulators of each node on the optical layer.
- Technology parameters: insertion loss (i.e. optical power loss values).

B. Output data

- Wavelength (symbolic) of each message and MRR.
- Placement of each MRR.
- Routing of each waveguide.

C. Minimization objectives

The choice of minimization objectives depends on the technology and the needs of the design. We consider the same as in previous publications [1], [7]–[10], [12]:

- Number of wavelengths.
- Message insertion loss.
- Number of MRRs.

Message insertion loss is the sum of seven types of losses: **1)** crossing loss, **2)** drop loss, **3)** through loss, **4)** bending loss, **5)** propagation loss, **6)** modulator loss and **7)** demodulator loss [10], [19]. We consider all except the last two, which are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2020.2971536, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

constant and equal for all messages and thus can be ignored from an optimization perspective.

IV. Physical layout template

As stated before, the global optimum to the WRONoC design problem is achieved by optimizing both the logical topology and the physical layout together. For complex designs this results in an extremely large solution space. Thus, our method seeks to solve a constrained version of the complete problem. However, whereas the state of the art constrains the problem by considering both aspects separately, our goal is to do it such that the developed solver is still given enough flexibility to design the logical topology and the physical layout together, letting any restrictions, choices or optimization opportunities from one aspect influence the choices made on the other.

For this reason, we propose a new input to the optimization process: a **physical layout template**. This input consists of a collection of WRONoC router elements (modulators, demodulators, waveguides and MRR placeholders) already placed and routed on the optical layer, to which the solution must conform.

This new input constrains the problem because the synthesis from scratch of a physical layout is turned into an optimization of the given template. In other words, to design the physical layout of the solution, the solver is now required to decide only on which elements (waveguides and MRR placeholders) to keep and which to remove from the template. Thus, most importantly, it will never be asked to place any new elements in new locations. This reduces the role of the solver with this new input to:

- Routing each message defined in the CM through the template.
- Assigning a wavelength to each message.
- Configuring the template for the chosen message paths and wavelengths (i.e. activating the necessary routing features and removing extraneous waveguides).

This way we significantly reduce the complexity of the complete problem while still considering both design aspects and thus improving upon the state-of-the-art solutions.

A. Template elements

Physical layout templates are composed of multiple instances of three basic elements, each with a fixed location on the optical layer:

- **Endpoints** represent modulator and demodulator arrays. They are placed wherever the (de)modulator arrays for each node are and connect to one waveguide section.
- **General Routing Units (GRUs)** are elements that connect through *ports* to multiple waveguide sections, called the *edges* of the GRU, and contain MRR placeholders to be populated by the solver as needed. They are the only template element that can contain MRRs, making them the routing building blocks of the template, and are described further in the next section.
- **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 sections with $90°$ bends or crossings with other sections.

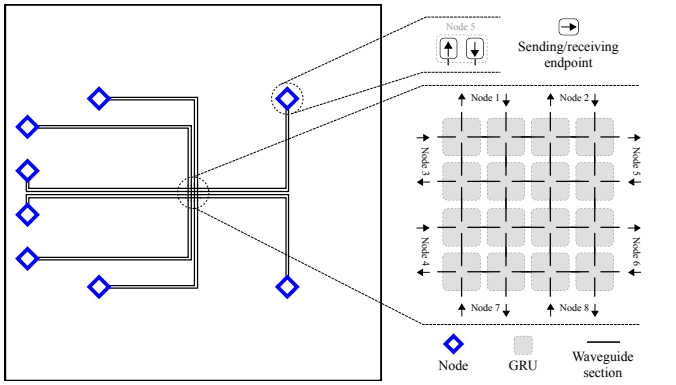


Figure 2. Example of a physical layout template for 8 nodes: a grid template.

One of the simplest templates that can be built with these elements is the grid template, shown in Figure 2. It has a regular grid structure of GRUs and endpoints and will be the main focus of this paper (see Section VI).

However, templates with different structures are also possible and two such examples are presented in Section VIII-A. In fact, we propose that any WRONoC design based on waveguides and MRRs can be contained in a suitably designed template. Yet, PSION+ can optimize any template, which makes PSION+ a powerful tool in WRONoC design.

B. General Routing Unit

Photonic Switching Elements (PSEs) are commonly applied in WRONoC routers [2], [8], [9], [12]. Yet, 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 resonance frequency, where in fact all four MRRs on a crossing can have different resonance frequencies.
- Their waveguide structure is fixed (PSEs always have a crossing), where in fact other routing designs are also possible.
- Messages only travel unidirectionally (PSEs always have exactly two input ports and two output ports), where in fact messages can travel in both directions on a waveguide simultaneously.

To solve this inflexibility a new type of optical switch is proposed in this paper: the General 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 resonance frequencies. Furthermore, each GRU instance on the template can be configured independently to have whatever internal structure is best suited for the current design problem. This flexibility enables exploration of areas of the WRONoC design space not yet analyzed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2020.2971536, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

1) Structure

To define the internal structure of a GRU, we first define a set $\mathbb{S}$ 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 — Figure 3(a).
- **MRR placeholders** represent the predefined positions where MRRs of various resonance frequencies can be placed — Figure 3(b).
- **Center crossing waveguides** are waveguide fragments that connect edge waveguides on opposite sides through the center of the GRU — Figure 3(c).
- **Corner bending waveguides** are waveguide fragments that connect edge waveguides on adjacent sides through a corner with a 90° bend — Figure 3(d).
- **Waveguide terminators** are required to close off the ends of other waveguide elements which aren't connected to anything — Figure 3(e).

The internal structure of any GRU instance on the template is built from a subset of $\mathbb{S}$ — an example is shown in Figure 4. As explained above, one of the roles of the solver is to configure the template for the chosen message paths and wavelengths. This includes deciding, for each GRU instance, what subset of $\mathbb{S}$ is optimal. However, the solver must also take into consideration that 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 Figure 4).
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).

2) Routing

The different GRU structures available lead to different routing behaviors, but routing through a GRU is still based on the principles shown in Figure 1: a message will follow the waveguide it's in (i.e. through a center crossing or corner bend) unless that waveguide passes next to an on-resonance MRR, in which case the message will be diverted to another waveguide. Figure 5 shows examples of the routing possibilities in a GRU.

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

6) Two MRRs on corners adjacent to the same edge cannot have the same resonance frequency.

Otherwise, a message of the same wavelength as the two MRRs would be affected by both MRRs at the same time.

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

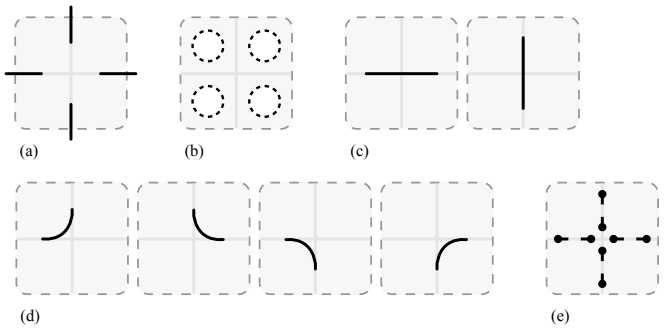


Figure 3. The set $\mathbb{S}$ of all 22 possible GRU internal structure elements. (a) Four edge waveguides. (b) Four MRR placeholders. (c) Two center crossing waveguides. (d) Four corner bending waveguides. (e) Eight waveguide terminators.

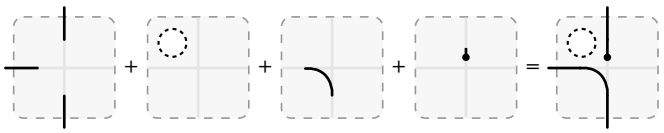


Figure 4. Example of an internal structure of a GRU instance built from a subset of $\mathbb{S}$.

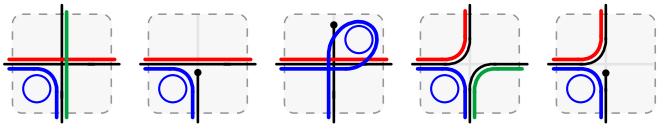


Figure 5. Examples of message routing in a GRU. The message with wavelength *blue* is affected by the MRR, whereas messages with other wavelengths (*red* and *green* in this case) aren't.

C. Communication Matrix

The other major input to the WRONoC optimization, the communication matrix (CM), can be translated into a set of messages (one for each nonzero entry) where each message 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 message m.

When using a layout template, 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 (modulator arrays) and $\mathbb{K}_n^R$ is the set of receiving endpoints (demodulator arrays) for node n. While simpler templates may only have one modulator and demodulator array per node, multiple modulator or demodulator arrays are perfectly feasible, and even possibly beneficial [1].

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

V. Mathematical Model

We solve the constrained version (with a layout template) of the complete problem using a Mixed Integer Programming model. Advantages of MIP models include:

- An MIP model can give optimal solutions, or at least an upper/lower bound to the optimal value of the optimization function.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2020.2971536, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

TABLE I
MODEL CONSTANTS & INDICES

Constants	
N_{gru}, N_{wg}, N_m, N_{ep}, N_λ	Total number of GRUs, waveguide sections, messages, 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 message m

TABLE II
MODEL VARIABLES

Binary	
$mwg_{m,wg}$	Message m uses waveguide section wg
$mwl_{m,\lambda}$	Message m uses wavelength λ
$mwle_{m_1,m_2}$	Messages m_1 and m_2 use the same wavelength
$mlc^1_{g,m}$	Message m has crossing loss once on GRU g
$mlc^2_{g,m}$	Message m has crossing loss twice on GRU g
$mlb_{g,m}$	Message m has bending loss on GRU g
$mlt_{g,p,m}$	Message 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 message 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 message uses wavelength λ
Integer	
nwl	Number of used wavelengths
Continuous	
mil_m	Insertion loss for message m
$maxil$	Maximum insertion loss over all messages

Index $p \in \mathbb{P}$, $\mathbb{P} = \{TL$: Top-Left, TR : Top-Right, BL : Bottom-Left, BR : Bottom-Right$\}$.

- The same MIP model can be used to optimize different objectives, therefore giving the designer more flexibility.
- MIP models are adaptable, so in the future more features can be added and optimized (see Section IX).

The model constants and indices are outlined in Table I. Constants L_{wg}, L^E_{wg} and indices W^*_i collectively describe the physical layout template and indices E^*_m define the CM. Table II lists all model variables.

We now specify the constraints and the optimization function (note that similar constraints referring to multiple corners and the standard MIP linearization techniques applied to some constraints are omitted for brevity and clarity). Finally, we present a fast proof of feasibility for the model.

A. Constraints

1) Message routing

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

The path must start and end at the correct endpoints, i.e. each message 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 ($\mathbb{E}^R_m$):

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

$$\sum_{ep \in \mathbb{E}^R_m} mwg_{m,W^E_{ep}} = 1 \tag{2}$$

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

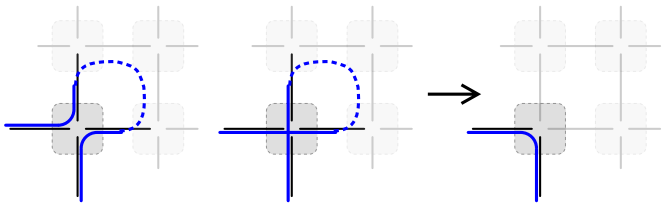


Figure 6. Path simplification from usage of 4 edges to 2 edges of a GRU.

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

$$mwg_{m,W^E_{ep}} = 0 \tag{3}$$
$$\forall ep \in \{1...N_{ep}\} \setminus (\mathbb{E}^S_m \cup \mathbb{E}^R_m), m = 1...N_m$$

Finally, the path must be continuous from the sending endpoint to the receiving endpoint. Gaps in the path appear when a message does not exit a GRU once for every time it enters it. To avoid gaps, constraints must be added to ensure that each message 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 optimized solutions:** A path that uses a GRU twice can almost always be simplified into a path that only uses it once, as shown in Figure 6. 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 V-A3) and for the calculation of the insertion loss of each message (see Section V-A4) become much simpler if a message is restricted to use each GRU at most once (2 edges) instead of twice (4 edges).

This is expressed with the following constraint:

$$mwg_{m,W^T_g} + mwg_{m,W^R_g} + mwg_{m,W^B_g} + mwg_{m,W^L_g} \in \{0, 2\} \tag{4}$$
$$\forall m = 1...N_m, g = 1...N_{gru}$$

2) Wavelength assignment

A wavelength must be assigned to each message, 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 messages can use the same wavelengths and what messages must use different wavelengths, which follows from the ONoC restriction that each waveguide section can have at most one message going through it for each wavelength. This also makes wavelength selection highly dependent on the chosen message paths.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2020.2971536, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

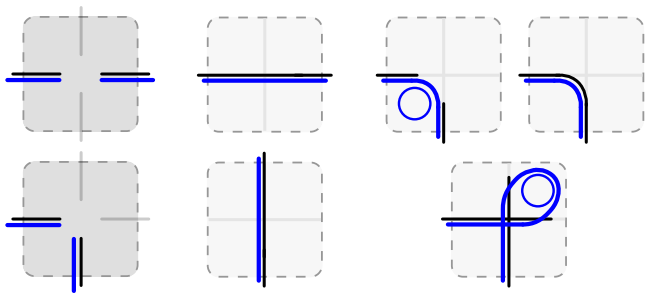


Figure 7. (a) Possible two-edge path choices in a GRU (only 2 of 6 choices shown). (b) Possible GRU configurations for messages that use two opposite edges. (c) Possible GRU configurations for messages that use two edges on the same corner (only one corner shown).

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

$$\sum_{\lambda=1}^{N_\lambda} mwl_{m,\lambda} = 1 \qquad \forall m = 1...N_m \tag{5}$$

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

$$mwl_{m_1,\lambda} \wedge mwl_{m_2,\lambda} \Rightarrow mwle_{m_1,m_2} \tag{6}$$
$$\forall \lambda = 1...N_\lambda, m_1, m_2 = 1...N_m : m_2 > m_1$$

Finally, enforce exclusivity of wavelengths on all waveguides, i.e., if two messages share a wavelength, they cannot both use the same waveguide:

$$mwle_{m_1,m_2} \Rightarrow (mwg_{m_1,wg} + mwg_{m_2,wg} \leqslant 1) \tag{7}$$
$$\forall wg = 1...N_{wg}, m_1, m_2 = 1...N_m : m_2 > m_1$$

3) Activation of routing features

Constraints in Section V-A1 and Section V-A2 have forced the solver to choose a valid path through the template for each message. These paths are described using the $mwg_{m,wg}$ variables. Now, constraints are required to configure each GRU accordingly. The chosen GRU configurations must satisfy two requirements, which will be explained in more detail next:

- **Path fulfillment:** The two entrance/exit edges of each message going through each GRU ($mwg_{m,wg}$ variables).
- **GRU structure compliance:** The GRU structure and routing rules described in Sections IV-B1 and IV-B2.

In other words, the chosen GRU configurations must support the chosen message paths while at the same time adhering to the defined GRU structure rules. If this is not possible, the solver will be forced to choose other message paths.

a) Path fulfillment

Looking only at the two GRU edges used by a message (variables $mwg_{m,wg}$ with $wg \in \{W_g^T, W_g^B, W_g^L, W_g^R\}$ for GRU g), there are only two possible choices: either the message 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 Figure 7(a). We will use this knowledge to force certain required GRU structure elements to be used depending on each message's path.

If a message uses two opposite edges then the corresponding center crossing waveguide must be used — Figure 7(b). Thus, the following constraints are added:

$$mwg_{m,W_g^L} \wedge mwg_{m,W_g^R} \Rightarrow gcch_g \tag{8}$$
$$mwg_{m,W_g^T} \wedge mwg_{m,W_g^B} \Rightarrow gccv_g \tag{9}$$
$$\forall m = 1...N_m, g = 1...N_{gru}$$

If a message 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 — Figure 7(c). The following four constraints per GRU–message pair are added (example constraint for the bottom–left corner given here):

$$mwg_{m,W_g^B} \wedge mwg_{m,W_g^L} \Rightarrow grm_{g,BL,m} \vee gcb_{g,BL} \vee grm_{g,TR,m} \tag{10}$$
$$\forall \text{ 4 corners}, m = 1...N_m, g = 1...N_{gru}$$

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

$$mwg_{m,W_g^B} \wedge mwg_{m,W_g^L} \wedge grm_{g,TR,m} \Rightarrow gcch_g \wedge gccv_g \tag{11}$$
$$\forall \text{ 4 corners}, m = 1...N_m, g = 1...N_{gru}$$

b) GRU structure compliance

The GRU structure rules 1–6 from Section IV-B1 and Section IV-B2 must be enforced. First, we must set the values of $gr_{g,p}$ 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 message:

$$gr_{g,p} = \sum_{m=1}^{N_m} grm_{g,p,m} \qquad \forall p \in \mathbb{P}, g = 1...N_{gru} \tag{12}$$

Rules 1–3 from Section IV-B1 are then enforced with the following sets of constraints:

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

$$gcch_g + gcb_{g,p} \leqslant 1 \tag{13}$$
$$gccv_g + gcb_{g,p} \leqslant 1 \tag{14}$$
$$\forall p \in \mathbb{P}, g = 1...N_{gru}$$

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

$$gr_{g,p} + gcb_{g,p} \leqslant 1 \qquad \forall p \in \mathbb{P}, g = 1...N_{gru} \tag{15}$$

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

$$gcb_{g,TL} + gcb_{g,TR} \leqslant 1 \tag{16}$$
$$gcb_{g,TR} + gcb_{g,BR} \leqslant 1 \tag{17}$$
$$gcb_{g,TL} + gcb_{g,BL} \leqslant 1 \tag{18}$$
$$gcb_{g,BL} + gcb_{g,BR} \leqslant 1 \tag{19}$$
$$\forall g = 1...N_{gru}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2020.2971536, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

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

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

4) Insertion loss calculation

a) Crossing loss

A message suffers crossing loss when going through a crossing with a perpendicular waveguide. Here we only need to consider crossing loss inside GRUs[1].

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

$$mwg_{m,W_g^T} \wedge mwg_{m,W_g^B} \wedge gcch_g \Rightarrow mlc^1_{g,m} \quad (20)$$

$$mwg_{m,W_g^L} \wedge mwg_{m,W_g^R} \wedge gccv_g \Rightarrow mlc^1_{g,m} \quad (21)$$

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

Additionally, if a message 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):

$$mwg_{m,W_g^B} \wedge mwg_{m,W_g^L} \wedge grm_{g,TR,m} \Rightarrow mlc^2_{g,m} \quad (22)$$

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

b) Through loss

A message 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:

$$mwg_{m,W_g^L} \wedge mwg_{m,W_g^R} \wedge gr_{g,p} \Rightarrow mlt_{g,p,m} \quad (23)$$

$$mwg_{m,W_g^T} \wedge mwg_{m,W_g^B} \wedge gr_{g,p} \Rightarrow mlt_{g,p,m} \quad (24)$$

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

c) Bending loss

A message has bending loss on a GRU[2] if it routes through a corner that uses its corner bending waveguide (example constraint for the bottom–left corner given here):

$$mwg_{m,W_g^B} \wedge mwg_{m,W_g^L} \wedge gcb_{g,BL} \Rightarrow mlb_{g,m} \quad (25)$$

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

d) Drop loss and propagation loss

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

[1]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 layout template and is thus constant during optimization.

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

e) Message insertion loss

The total insertion loss of a message over all waveguides and GRUs is given by the following weighted sum of propagation, crossing, bending, through and drop loss:

$$\begin{aligned} mil_m = &\sum_{i=1}^{N_{wg}} (L^P L_i + L^E_i) * mwg_{m,i} \\ &+ \sum_{g=1}^{N_{gru}} (L^C mlc^1_{g,m} + 2L^C mlc^2_{g,m} + L^B mlb_{g,m}) \\ &+ \sum_{g=1}^{N_{gru}} \sum_{p \in \mathbb{P}} (L^T mlt_{g,p,m} + L^D grm_{g,p,m}) \quad (26) \\ &\forall m = 1...N_m \end{aligned}$$

B. Objective function

As explained in Section III-C, we will consider three optimization objectives: number of wavelengths, message insertion loss and number of MRRs. For message insertion loss, we will consider both the maximum and the sum over all messages.

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

$$wlu_\lambda \geqslant mwl_{m,\lambda} \qquad \forall m = 1...N_m, \lambda = 1...N_\lambda \quad (27)$$

$$nwl = \sum_{\lambda=1}^{N_\lambda} wlu_\lambda \quad (28)$$

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

$$maxil \geqslant mil_m \qquad \forall m = 1...N_m \quad (29)$$

Finally, the MIP optimization problem is formulated as follows:

Minimize:

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

Subject to: (1)–(29)

where α_i are optimization weights chosen by the designer.

Since the value for the insertion loss of each message is available through the mil_m variables, functions other than the maximum of the insertion loss can also be added to the model and used for optimization assuming they are linear or linearizable.

C. Proof of feasibility

It is possible that the chosen layout template cannot satisfy the entire CM (for example, if the template is too small and the CM is dense). In these cases, we call the template "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. For that we consider $N_\lambda = N_m$ and uniquely assign a wavelength to each message by adding these constraints:

$$mwl_{m,\lambda} = 1 \qquad \forall m = 1...N_m, \lambda = m \quad (30)$$

$$mwl_{m,\lambda} = 0 \qquad \forall m = 1...N_m, \lambda \neq m \quad (31)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2020.2971536, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

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 message uses its own wavelength (thus either maintaining or increasing nwl). Any message that changes its wavelength must also change the wavelength of the MRRs it uses. This is always possible because each MRR routes only one message. Furthermore, the wavelength exclusion rule is always satisfied. Hence, the feasibility of the complete model implies the existence of a solution for the simplified version. ■

VI. Grid template design

In Section IV we explained that considering a physical layout template as an extra input to the WRONoC design problem turned a synthesis problem into an easier optimization problem. While this statement is true, it disregards the synthesis effort required to create a physical layout template in the first place. This might seem like an oversight: after all, creating a suitable layout template for each WRONoC design problem might be as difficult as it is crucial. However, in this section we will prove this need not be a concern by introducing and analyzing a simple layout template format that can be used in virtually any WRONoC problem while requiring no synthesis effort. We will also show some of its other favorable characteristics besides simplicity. Finally we will end with a discussion on more advanced synthesis concerns.

This simple template is called the grid template (GT) [20] and is shown in Figure 2. 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 position on the die.

In general, the grid of GRUs can be distributed throughout the die (distributed grid template — DGT [20]) 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 (centralized grid template — CGT [20]). This location can still be optimized, but simple yet sensible locations for centralized grids are the center of mass of the nodes or just the center of the die.

Another design choice with grid templates is terminal assignment. This can have a measurable impact on the solution because it influences message 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 die, 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 VI-A. 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, without loss of generality, the waveguides connecting the endpoints to the GRUs can be manually routed to minimize their length and number of bends (external waveguide routing).

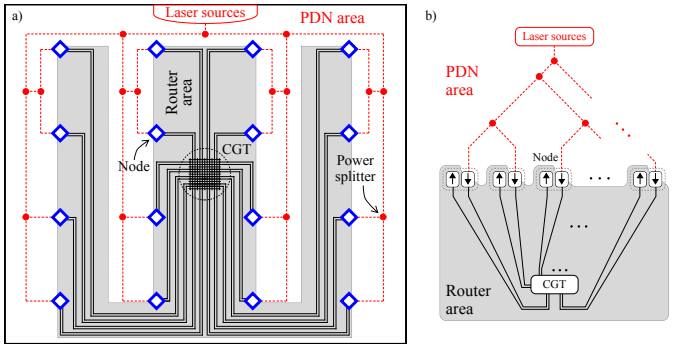


Figure 8. Breaking the inter-dependency between the PDN and the router by dividing the die into a "PDN area" where only the PDN is placed, and a "router area" where only the CGT and external waveguide sections are placed. (a) Example with a grid of 16 nodes from [13]. (b) General topology of a PDN and a router using a CGT with no waveguide crossings outside the CGT.

A. Power Distribution Network (PDN) awareness

Modulators require a source of optical power obtained from on-chip or off-chip lasers [13]. For off-chip lasers, laser power is transferred into the chip through optical couplers placed on the edge of the die. This power must then reach each modulator array, which calls for a PDN composed of waveguides and optical splitters. Most crucially, crossings between the PDN waveguides and the router waveguides will significantly increase the minimum laser power required [1]. An easy way to avoid these crossings is to split the die into two areas, the "PDN area" and the "router area", such that a path free from crossings from each modulator array to the laser source is always available, as shown in Figure 8. While most P&R tools presented thus far do not guarantee zero crossings with the PDN [10], [11], [17], CGTs can always easily be designed to be confined to the "router area", thus having no crossings with the PDN.

A proof of this last statement follows. Consider a graph where each node, containing one sending endpoint (modulator array) and one receiving endpoint, 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 don't cross outside the CGT is equivalent to stating that the constructed graph is planar. As seen on Figure 8(b), the constructed graph is clearly planar[3].

B. Types of message paths on grid templates

The regular structure of GTs allows for an *a priori* analysis of the optimal paths messages will likely take through the grid. If corner bending waveguides are turned off (these aren't always necessary and this significantly reduces solver run-time — see Section VIII-B), messages are prone to following one of three path types:

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2020.2971536, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

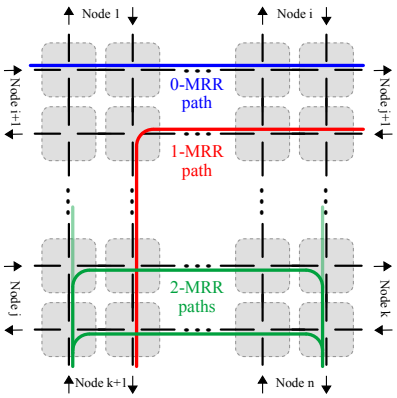


Figure 9. Types of paths through a GT.

- **0-MRR paths:** Messages whose entrance and exit terminals are directly aligned are very likely to take the direct path and use 0 MRRs — blue path in Figure 9.
- **1-MRR paths:** Messages whose entrance and exit terminals are on perpendicular sides of the grid are very likely to use 1 MRR — red path in Figure 9.
- **2-MRR paths:** Other messages have multiple 2-MRR paths available — green paths in Figure 9.

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 message 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 optimized solutions. This fact will be exploited by heuristics presented in Section VII.

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

C. Open problems in grid template design

As stated in Section V-C, layout templates 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 VI-B, messages will tend to use the 0/1/2-MRR paths. More importantly, any other path would use more MRRs. Thus, we will now calculate the minimum number of MRRs required to support these paths with a full CM without loopback. Such a matrix contains $M = N*(N-1)$ messages, where:

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

Multiplying each expression by the corresponding number of used MRRs we get the minimum number of MRRs required to route M messages, 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 messages, but unfortunately it is not true for all values of w and h. For example, a 2×2 grid (8 nodes) supports all $8 * 7 = 56$ messages, but a 3×3 grid (12 nodes) will not support all $12 * 11 = 132$ messages. Corner bending does not change this result because it is never used in templates close to saturation.

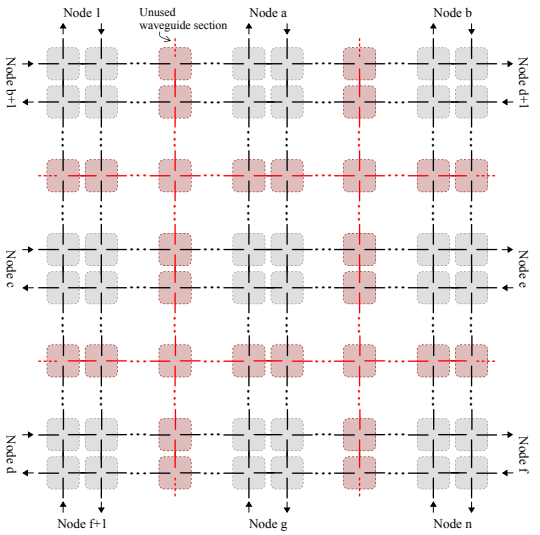


Figure 10. Expanding a GT by adding extra sets of waveguide sections and GRUs (pictured in red).

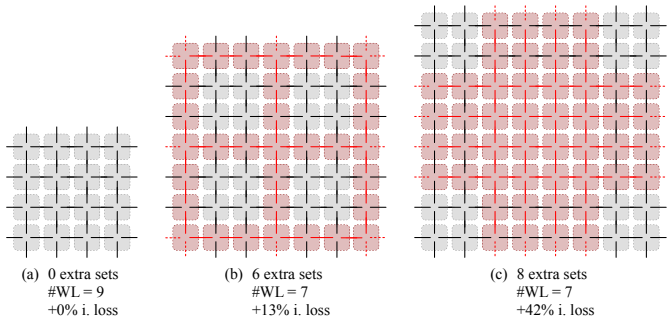


Figure 11. Comparison between possible expanded grid designs for an 8 node GT. #WL is the minimum amount of wavelengths required to support all 56 messages and *i. loss* is maximum insertion loss. In this case, (b) is clearly a better option than (c) even though it uses fewer extra sets.

To add more MRRs we must add more GRUs, which is possible by adding extra sets of horizontal and/or vertical waveguide sections to the grid as shown in Figure 10. 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 solver run-time, 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 required number of wavelengths — Figure 11. In both cases, their position also contributes to the quality of the final result.

Nevertheless, in many cases it's acceptable to ignore these concerns. PSION+ performs better in application specific design (see Section VIII-B) where the number of messages 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.

VII. Heuristics and Model Reduction Techniques

The MIP approach outlined in this work produces optimal solutions, which makes the solving process inherently slow. Thus, multiple heuristics were developed that speed up this process and allow targeting of bigger WRONoC sizes with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2020.2971536, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

full CMs. They will now be presented.

A. Restrictions on usage of wavelengths

The following constraints can be added to the model:

$$mwl_{m,\lambda} = 0 \qquad \forall \lambda = (m+1)...N_\lambda, m = 1...N_m \qquad (32)$$

Their effect is to restrict the possible wavelengths for each message: message 1 uses wavelength 1, message 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 solution space.

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

When minimizing insertion loss, each message 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 optimization process. To avoid this wasted effort, constraints can be added to the model that force a maximum number of MRRs per message (R^{max}):

$$\sum_{g=1}^{N_{gru}} \sum_{p \in \mathbb{P}} grm_{g,p,m} \leqslant R^{max} \qquad \forall m = 1...N_m \qquad (33)$$

This forces the solver to only consider simpler paths right from the onset and so leads to noticeably reduced solver run-time. 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 optimized solutions will take advantage of corner bending by making messages snake through the template 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 layout template analysis. For example, given the analysis in Section VI-B, $R^{max} = 2$ turns out to be an attractive option for GTs.

C. Path assignment

As explained in Section IV, one of the major sources of complexity in the WRONoC design problem is message routing. Thus, if for some reason we know *a priori* what paths some messages are going to take, this information can be very useful in reducing solver run-time.

In the case of GTs we can predict with reasonable confidence the paths of all 0-MRR and 1-MRR messages[4] when corner bending waveguides are turned off. This knowledge can be put to good use by adding extra constraints that fix the values of variables $mwg_{m,wg}$ for 0/1-MRR messages according to their path.

The effectiveness of this heuristic depends on **1)** how correct our path assumptions are and on **2)** how high the proportion of messages with well-defined paths to total messages is. We will show that GTs succeed in both criteria, leading to substantial solver run-time reductions without practically any penalty in solution quality.

[4]2-MRR messages have multiple path choices, so no clear path information is available *a priori*.

Algorithm 1 Wavelength assignment heuristic

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

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

D. Wavelength assignment

Similar to message routing, wavelength assignment is another major source of problem complexity. The challenge with this assignment is in finding the smallest[5] packing of messages into wavelengths such that the paths of messages with the same wavelength do not overlap, i.e. use the same waveguide sections.

Let $\mathbb{M}$ be the set of all messages and $\mathbb{W}$ the set of all wavelength sections in the template. 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 messages 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 messages in $\mathbb{M}^*$ can all be assigned one wavelength λ_x without increasing the minimum possible number of used wavelengths. In doing this we reduce the initial assignment problem $(\mathbb{M}, \mathbb{W}, p)$ 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 we used Algorithm DLX [21])[6] and is run before solving the MIP 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 VII-C.

E. 3-step optimization

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

In the **first step** we consider $N_\lambda = N_m$ and apply the feasibility proof from Section V-C. In this way we can generate

[5]This is assuming the number of wavelengths is part of the optimization 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 V-C.

[6]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 messages.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2020.2971536, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

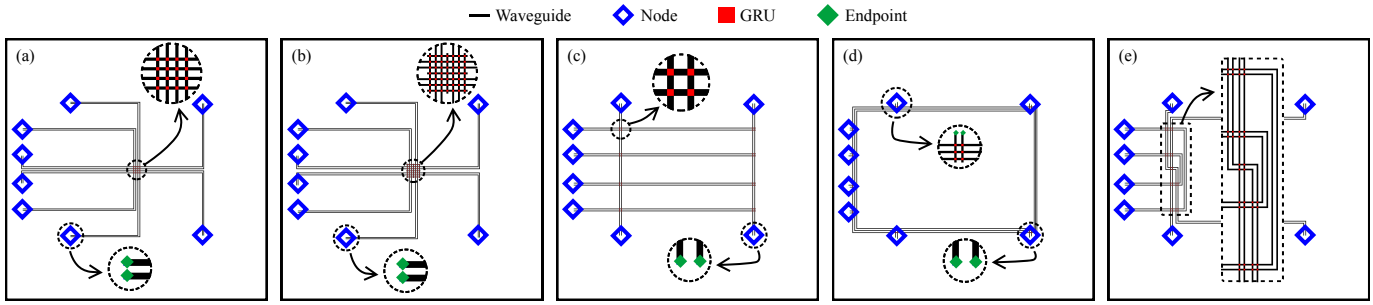


Figure 12. (a) A non-expanded centralized grid template (CGT-e0) connecting the nodes on the positions that produce the best result in PROTON+. (b) An expanded CGT with 6 extra waveguide section sets (CGT-e6). (c) A distributed grid template (DGT). (d) A ring template with 3 rings. (e) A custom template.

the first feasible solution much faster if one exists and then use it to warm start the optimization. This has the added bonus of stopping the process as quickly as possible if unfeasible.

In the **second step** we only minimize the number of wavelengths (nwl), for two reasons. Firstly, the designer will almost certainly want to use fewer wavelengths than messages. Hence, even if nwl is not the main minimization 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 we are only interested in finding a feasible solution with a reduced number of wavelengths, not in finding the minimum of nwl. Thus, this optimization procedure can be stopped once a solution with a reasonably small nwl has been found.

In the **third step** we first simplify the model by removing unused wavelengths with the following constraints:

$$mwl_{m,\lambda} = 0 \quad \forall m = 1...N_m, \text{ unused wavelengths } \lambda \quad (34)$$

We then solve the model to optimality for the optimization function chosen by the designer, reaching the final solution.

Using this process we can notably simplify the problem space during the optimization. Moreover, results show that the partial optimization of nwl in the second step can be subtle enough to not remove the optimal solution from the optimization in the third step while still being agressive enough to reduce total solver run-time (see Section VIII-B).

VIII. Results

The MIP model and accompanying heuristics are programmed in C++ and make use of Gurobi [22], an MIP solver, on a 2.6 GHz CPU.

A. Comparison to state-of-the-art P&R tools

We tested our model and optimization procedure against the state-of-the-art PROTON+ and PlanarONoC P&R tools. Most of their result analysis 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 ($4 \times 3 = 12$ messages) and each cluster communicates with all memory controllers bidirectionally ($4 \times 4 \times 2 = 32$ messages). Memory controllers do not communicate between themselves, thus lowering the number of messages from a full CM (56 messages) to only $12 + 32 = 44$ messages. We solved this test case with this CM and the same die size, crossing size, loss parameters and optimization function (max. insertion loss).

TABLE III
Results for 8 nodes, 44 messages

		#WLs	Max IL	#MRRs	Time
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

#WLs, #MRRs: number of wavelengths and MRRs.
Max IL: maximum insertion loss. Time in seconds, insertion loss in dB.
PSION+ results are optimal for the given templates.

We used the node positions that produced the best result over all presented in PROTON+. PROTON+ and PlanarONoC both employ three standard logical topologies (λ-Router, GWOR and Standard $\times$bar) whereas we manually designed five simple layout templates, presented in Figure 12(a-e), that connect to those node positions. No effort was spent in trying to optimize the design of the GTs: the grids of the CGTs were placed in the middle of the die, 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 (3) to achieve the minimum number of wavelengths required by this test case (7). The custom template was built specifically for this test case so that no message 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 heuristics from Section VII were used.

Table III presents the various comparisons. On average results from PSION+ are either equal or better except in comparison to PlanarONoC's execution time. Using PSION+ and without any template 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 templates used here do. By spending slightly more effort in designing good templates 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2020.2971536, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

CGT-e6 already achieves the minimum possible number of wavelengths, i.e. 7, so we did not test CGTs 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: **1)** it is more complex (more GRUs and waveguide sections), **2)** it does not take advantage of the heuristics from Section VII-C and Section VII-D and **3)** the combinatorial complexity is further increased because the total number of possible message paths is multiple orders of magnitude higher than with GTs. It also uses more MRRs because all messages require 2 MRRs: one to drop into the ring and another to drop out of the ring.

We note that we chose $1*maxil$ as the optimization function for all templates to ensure fairness in the comparison, but we still obtained a reduction in the usage of MRRs when possible. Each MRR has a potential contribution of $1\times$ drop or through loss to the max. insertion loss, so minimizing $maxil$ indirectly optimizes the number of MRRs. Thus, there is little need for minimizing the number of MRRs directly and so we recommend using $1 * maxil$ as the optimization function[7] for the third step for any template.

B. Analysis of grid templates 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 analyze how each of the proposed heuristics impacts solution quality and solver run-time on GTs.

Hence, we solved a non-expanded GT[8] of size 2×2 (8 nodes) for random CMs with 1 to 56 messages in total and recorded the relevant results: maximum insertion loss, number of wavelengths (#WLs), number of MRRs (#MRRs) and optimization time. Since all four results depend on the exact set of messages chosen, multiple tests with random CMs were done for each number of messages and their results averaged. All tests use the technology parameters from [19] and were solved to optimality.

To test the impact of each heuristic, the described sequence of random tests from 1 to 56 messages was completed five times. The first time, only the model reduction from Section VII-A was used, which gives us the "ground truth", 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 VII-E 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 minimization function. Corner bending was turned off for these tests since heuristics in Section VII-C and Section VII-D are incompatible with it. If all heuristics are successful, the quality of the results should remain constant while solver run-time decreases.

To test the usefulness of corner bending waveguides we ran the sequence of random tests a sixth time with all heuristics except those from Section VII-C and Section VII-D. The corresponding curves should be compared against the sequence run without corner bending that also used the same heuristics.

Figure 13 presents the results of all 6 sequence runs. The following conclusions can be drawn:

- There is a linear relationship between the number of messages and both #MRRs and #WLs. Thus, conventional logical topologies waste a considerable amount of unnecessary resources when used with non-complete CMs. Reducing #MRRs directly reduces static MRR thermal tuning power and reducing #WLs reduces total static (de)modulator power and can reduce laser power.
- Runs without the heuristic from Section VII-B show a slight increase in maximum insertion loss on average. We can observe 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[9]. However, no appreciable difference can be seen in the #WLs or #MRRs since this happens infrequently. Thus, we still recommend the use of this heuristic given its remarkable impact on solver run-time.
- In general, the proposed heuristics reduce solver run-time 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 optimization reduces run-time by 96%, $R^{max} = 2$ further reduces run-time by 82%, path assignment further reduces run-time by 91% and wavelength assignment further reduces run-time by 60%. In total, using all heuristics is $18463\times$ faster. This amount of reduction is essential for tackling larger WRONoC designs while targeting optimality. Wavelength assignment only produces measurable reductions in run-time for denser CMs because the probability of finding exact covers is low when there are fewer messages.
- 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 messages or 45% of a full CM) at the cost of being noticeably slower. However, results show corner bending isn't used with dense CMs. This is because dense CMs have many messages 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 matrices (below 45%), turning it off and using all heuristics otherwise.

Note that the solver run-time results shown here are for obtaining optimal solutions. This explains their exponential growth with the increase in problem size (number of messages). If a designer is willing to forgo proof of optimality, good enough solutions can be obtained faster. For example, we observed that on average we achieve a solution up to only 10% worse than an optimal solution in 10% of the total optimization time (the remaining 90% is used to achieve optimality).

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

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

[9]This is because the optimization function $100 * nwl + 1 * maxil$ used here prioritizes the minimization of #WLs over insertion loss.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2020.2971536, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

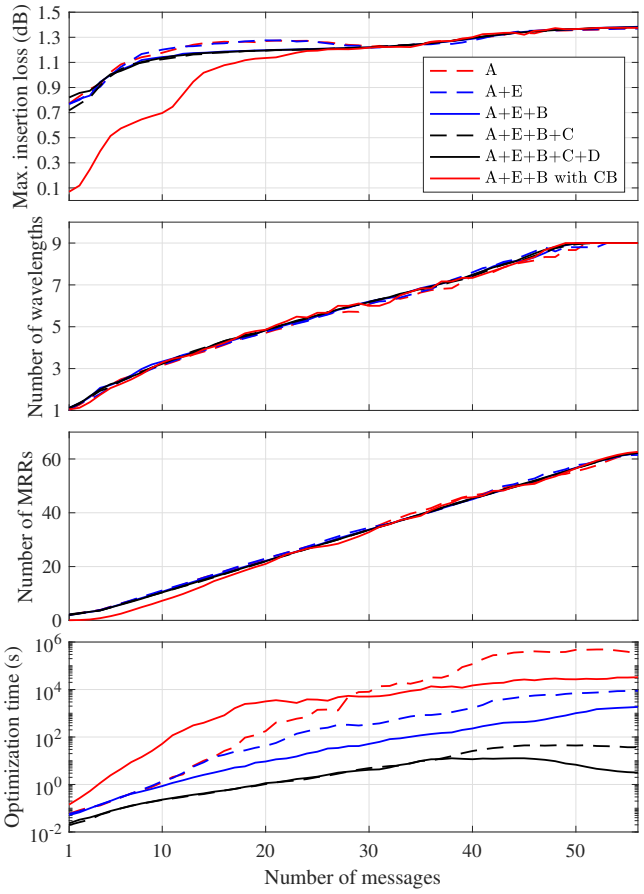


Figure 13. Results for max. insertion loss, #WLs, #MRRs and solver runtime for 1–56 messages on an 8-node GT with multiple solver configurations. Curves *A+E+...* use the corresponding heuristics from sections VII-A, VII-E, etc, with corner bending waveguides turned off. Curve *A+E+B with CB* uses the corresponding three heuristics with corner bending waveguides turned on.

C. Full power consumption comparison for 16 node WRONoC

The heuristic methods that we have developed enable us to extend our approach to state-of-the-art realistic problem sizes for wavelength routing applications. In particular, we compared PSION+ against the previously-reported manual design of a 16-node WRONoC, which is laid out on a $16\,\mathrm{mm}\times16\,\mathrm{mm}$ optical layer vertically stacked on top of a 3D manycore system with $16 \times 16 = 256$ cores [13].

For this comparison we use 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 (de)modulator power. The two designs with the lowest total static power consumption in [13] are a centralized λ-Router topology and a centralized Snake topology (*λ-Router SB* and *Snake SB* where *SB* stands for *Single Box*).

We applied our CGT design philosophy to this test case. We first divided the die into a PDN area and a router area such that the PDN area contained the same PDN used in the SB designs. We then placed a CGT-e10 on the router area on two locations: the center (Center CGT-e10) and the mid-bottom (Bottom CGT-e10) of the die. Next, we chose the assignment of CGT terminals to nodes that does not produce any external router waveguide crossings. Finally, we manually routed the external router waveguides. The resulting Center CGT-e10 design can be seen in Figure 8(a) (the Bottom CGT-e10 has the CGT itself moved $4\,\mathrm{mm}$ down).

TABLE IV
COMPARISON TO [13]

	#WLs	#MRRs	P_L	P_{MD}	P_T	$\boldsymbol{P_{total}}$
[13]						
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**

#WLs, #MRRs: number of wavelengths and MRRs. P_L = total laser power (router + PDN). P_T = static MRR thermal tuning power. P_{MD} = static modulator + demodulator power. $P_{total} = \sum P_i$. Power values in mW.

The Center CGT-e10 and both SB designs are placed on the same location on the die, 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 we used the same die size, node positions, PDN design, technology parameters, static thermal tuning power per MRR, static power per modulator and demodulator, crossing size and communication matrix. Here the communication matrix is full ($16 \times 15 = 240$ messages), so we turned off corner bending and used all heuristics (see Section VIII-B). On the third optimization step we optimized for $maxil$. The results are given in Table IV. We use a higher number of wavelengths and a higher number of MRRs which leads to higher static (de)modulator and thermal tuning power. However:

- We reduce total static power consumption by 4.2% to 6.9%.
- 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 [13] do not change with sparser CMs, it is clear from Section VIII-B that PSION+ can achieve substantially better results with sparser CMs. If this test case was for an application specific design with fewer messages, PSION+ would achieve designs with even smaller total static power consumption.

This fully automated optimization is a step forward with respect to current error-prone manual design frameworks [13] and took less than one week to run, which is still a reasonable one-shot design time overhead for a complete system-level interconnection network.

IX. CONCLUSION

In this work we defined the WRONoC design problem and presented PSION+, a novel method for solving it. This method uses a physical layout template to combine logical topology and physical layout optimization. We also presented a new, flexible, routing element, the GRU, which improves upon the PSE used thus far. We used an MIP model along with multiple heuristics to quickly solve physical layout templates for their optimal solution. We also introduced a very simple yet general layout template, the grid template, and described its strengths and weaknesses. Finally, we analyzed the perfor-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2020.2971536, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

mance of PSION+ in general and in comparison to previous WRONoC design tools. We concluded that these combined efforts produce results superior to the state-of-the-art.

In the future, the presented MIP model may be expanded to include other areas of WRONoC design, such as the PDN. Also, as feasible WRONoC sizes increase, so will PSION+ be improved to handle them within acceptable optimization timeframes. More specifically, PSION+ may yet be enhanced with further reduction techniques, or maybe by off-loading part of the combinatorial search burden (fundamentally created by path assignment and wavelength assignment) to other combinatorial optimization methods such as local search or genetic algorithms. These trade off solution quality for runtime much more effectively than a MIP solver, which should improve scalability at the cost of optimality.

Template synthesis is another area where many improvements are most likely possible. Template designs other than the GT (ring templates [1], for example) should be explored and characterized further. Grid templates specifically can still be improved, as there are yet many opportunities for optimization: grid expansion, terminal assignment, GRU positioning and external waveguide routing are just some examples.

REFERENCES

[1] 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*, Jan 2017.

[2] 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*, 2005.

[3] 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*, May 2009.

[4] Y. Xie *et al.*, "Crosstalk noise and bit error rate analysis for optical network-on-chip," in *Proceedings of the 47th Design Automation Conference*, Jan 2010, pp. 657–660.

[5] M. A. Seyedi *et al.*, "Crosstalk analysis of ring resonator switches for all-optical routing," *Opt. Express*, pp. 11 668–11 676, May 2016.

[6] M. Tala *et al.*, "Exploring communication protocols for optical networks-on-chip based on ring topologies," 2014, p. ATh3A.165.

[7] 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*, Nov 2016, pp. 1–8.

[8] 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*, March 2013, pp. 1589–1594.

[9] 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*, Aug 2016, pp. 1–8.

[10] 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.*, pp. 44:1–44:28, Dec. 2015.

[11] Y.-K. 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*, June 2018, pp. 151:1–151:6.

[12] X. Tan *et al.*, "On a scalable, non-blocking optical router for photonic networks-on-chip designs," in *2011 Symposium on Photonics and Optoelectronics*, May 2011, pp. 1–4.

[13] M. Ortn-Obn *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*, July 2017.

[14] H. Li *et al.*, "Optical crossbars on chip: a comparative study based on worst-case losses," *SiPhotonics*, Oct. 2014.

[15] S. L. Beux *et al.*, "Reduction methods for adapting optical network on chip topologies to 3d architectures," *Microprocessors and Microsystems*, pp. 87 – 98, Feb. 2013.

[16] 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*, 11 2018, pp. 1–8.

[17] 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*, April 2016, pp. 27–34.

[18] D. Ding *et al.*, "O-router: An optical routing framework for low power on-chip silicon nano-photonic integration," 01 2009.

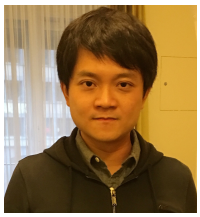
[19] M. Nikdast *et al.*, "Crosstalk noise in wdm-based optical networks-on-chip: A formal study and comparison," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 2552–2565, Nov 2015.

[20] 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*, April 2019.

[21] D. E. Knuth, "Dancing links," *arXiv e-prints*, Nov. 2000.

[22] Gurobi Optimization, Inc., *Gurobi Optimizer Reference Manual*. http://www.gurobi.com, 2018.

Alexandre Truppel received the bachelor and master degrees in electrical and computer engineering from University of Porto (UP), Porto, Portugal, in 2016 and 2018 respectively. He is currently pursuing a PhD degree as a doctoral researcher at the Technical University of Munich (TUM), Munich, Germany. His research focuses on the development and use of optimization techniques in emerging technologies such as optical networks-on-chip.



Tsun-Ming Tseng (S'10 – M'15) received the bachelor degree in electronics engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 2010, and the master and the Dr.-Ing. Degrees from Technical University of Munich (TUM), Munich, Germany, in 2013 and 2017, respectively. He currently works on his habilitation with the Chair of Electronic Design Automation at TUM. His research interests focus on mathematical modeling methods for computer-aided design for emerging technologies, such as microfluidic biochips and optical networks-on-chips.

Davide Bertozzi (M'13) is an Associate Professor at University of Ferrara (Italy). His research focuses on design methods and tools to exploit the enabling features of communication architectures and emerging technologies. He has worked on several EU-funded projects (NaNoC, vIrtical, Galaxy) and led a pioneering national project on the applications of silicon photonics to on-chip communication (Photonica). In 2018 he received the Wolfgang Mehr Award from IHP Microelectronics for his research in the field of electro-optical interconnect technologies, capable of bridging the research areas of system design and semiconductor technology.

José Carlos Alves received the PhD degree in electrical and computer engineering from University of Porto (UP), Porto, Portugal, in 1998. He is currently an Associate Professor with the Faculty of Engineering, University of Porto, and Senior Researcher at INESC TEC. His current research interests are focused on high-performance recongurable custom computing, FPGA applications and digital design methodologies and tools.

Ulf Schlichtmann (S'88 - M'90 - SM'18) received the Dipl.-Ing. and Dr.-Ing. degrees in electrical engineering and information technology from Technical University of Munich (TUM), Munich, Germany, in 1990 and 1995, respectively. He is Professor and the Head of the Chair of Electronic Design Automation at TUM. He joined TUM in 2003, following 10 years in industry. His current research interests include computer-aided design of electronic circuits and systems, with an emphasis on designing reliable and robust systems. Increasingly, he focuses on emerging technologies such as lab-on-chip and photonics.