Technical University of Munich
School of Computation, Information and Technology
Chair of Electronic Design Automation

Physical-Design-Oriented Topology Generator for Application-Specific WRONoC

Research Internship Report

You-Jen Chang

Technical University of Munich
School of Computation, Information and Technology
Chair of Electronic Design Automation

Physical-Design-Oriented Topology Generator for Application-Specific WRONoC

Research Internship Report

You-Jen Chang

Advisor :	M.Sc. Liaoyuan Cheng
Advising Professor :	Prof. Dr.-Ing. Ulf Schlichtmann
Topic issued :	24.05.2024
Date of submission :	24.11.2024
Working period :	24.5.2024 - 24.11.2024

You-Jen Chang
Arcisstraße 21
80333 Munich Germany

Abstract

Optical Networks-on-Chip (ONoC) is widely considered a critical method to overcome electronic interconnects' bandwidth and energy limitations. Specifically, the Wavelength-Routed Optical Networks-on-Chip (WRONoC) has high bandwidth and low signal delay features. A typical design flow of WRONoC has two steps: the design of topological and physical design. Some topology generators seem very advantageous from the perspective of worst insertion loss; after the routing of the physical design, the insertion loss of the routing can make them less appealing than others. This massive gap between design steps can occur if these two steps are not considered together. Even though some works have considered these two steps together in the design process, they still cannot handle the customized cases without manual efforts due to the limits of their algorithms. This research internship aims to consider both factors from topological and physical designs for application-specific cases so the synthesized layout can still guarantee its worst insertion loss at a decent level and be generated automatically. As shown in this report, the flexibility to accommodate different applications for topological design can also benefit physical design, so my method in topological design is based on the customized topology generator. Because of this, the work is called **CustomTopo+**. In addition, by solving the assignment problem, the physical design can predict pairs of a port, on a topology, and a master or slave, on an IP core, to realize the integration of considerations from both designs. Then, I will build up two mixed-integer-linear programming (MILP) models to finish the physical and topological design. Further, the assignments of the ports on topology are determined by the physical design to realize the physical-design-oriented topology generator in the end. Compared to the state-of-the-art routers, CustomTopo+ can reach an average 5% improvement of the worst insertion loss in the network, and its performance for different inputs has a 30 % lower variance. At the end of this report, the design flow of CustomTopo+ will be shown, and the future of improving the worst insertion loss further by modifying the routing of customized topology generators will be discussed.

Contents

List of Figures

List of Tables

1. Introduction

The need for communication between different IP cores inside the circuit is increasing while the complexity and popularity of System-on-Chips are rising (Vantrease et al. 2008). To relieve the stress of these needs, Optical Networks-on-Chips (ONoCs) have been widely considered as a future solution. So far, there have been two categories of ONOCs: Control-Networks-Based and Wavelength-Routed ONoCs (WRONoCs) (Werner et al. 2017). Because both categories apply wavelength-division-multiplexing (WDM), a single waveguide can be reused for multiple signals simultaneously with different wavelengths (Cheng et al. 2024). On Control-Networks-Based ONoCs, active networks are adopted, so the signal paths between masters (senders) and slaves (receivers) are set up dynamically whenever needed. On the other hand, WRONoCs provide a passive network. The paths and associated wavelength are all fixed at design time (Zheng et al. 2024). Due to its use of the fixed path, it does not have the energy and latency overhead of arbitration, and this methodology is used in this research (Truppel et al. 2019, Manolatou et al. 2002, Li et al. 2018).

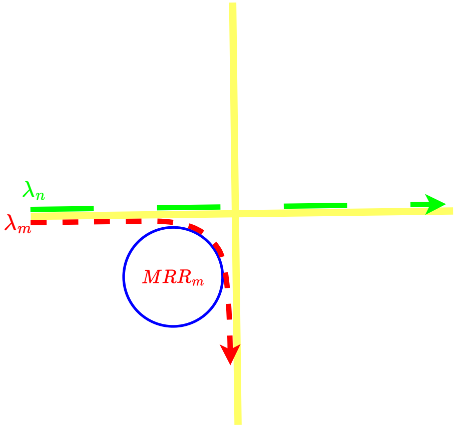


Figure 1.1.: MRR mechanism on a CSE.

Optical switching elements (OSEs) consisting of microring resonators (MRRs) should guide the specific signal to its destination from multiple signals in the same waveguide. An MRR comprises a looped optical waveguide and a coupling mechanism to resonate with particular wavelengths (Li et al. 2020, Bogaerts et al. 2012). Therefore, whenever a signal approaches an MRR along its nearby waveguide, and the wavelength of the signal is on-resonance with the MRR, the signal will experience coupling to the looped waveguide by some power consumption, called **drop loss**, and leave the MRR via another nearby waveguide, which can make the change of the direction of a signal (Zheng et al. 2025). The change of directions for a signal due to an MRR will be called **drop** in the following context. In the case of off-resonance, the signal will go through an MRR by a different form of power consumption, called **through loss**. State-of-the-art WRONoC topologies, such as GWOR (Tan et al. 2011) and CustomTopo (Li et al. 2018), are usually built upon crossing switching elements (CSEs), where MRRs can be placed near the crossing of waveguides to distribute the signals for four directions.

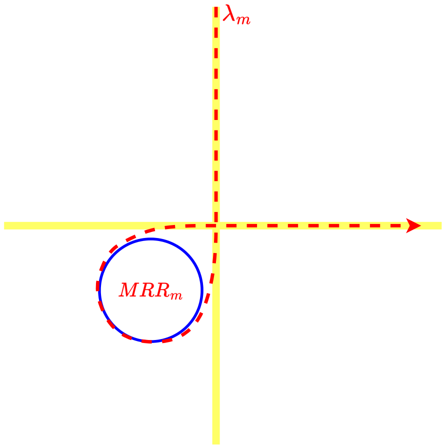


Figure 1.2.: 270° turn in CSEs.

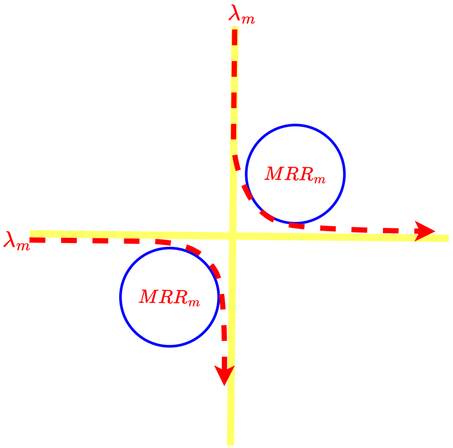


Figure 1.3.: An add-drop filter (ADF) to realize only 90° turns in CSEs.

As shown in Figure 1.1, there are two signals with different wavelengths: one uses wavelength λ_m, called $Signal_A$, and the other one uses wavelength λ_n, called $Signal_B$. Because the MRR is set to wavelength λ_m, it will couple $Signal_A$ and let $Signal_B$ pass by. Further, if the direction of a signal enters a CSE change, it has been shown that a signal can also reach 270° turn in CSEs in Figure 1.2. In this case, a signal will suffer more power loss. To maintain signal-to-noise ratio (SNR) for signals that need 270° turns, add-drop filters (ADFs) are usually used to allow only 90° turns in CSEs for some specific wavelengths. In Figure 1.3, an ADF, which couples wavelength λ_m, is shown. This research will also consider the loss of bending and propagation, depending on the lengths of passed waveguides, during the physical design and the loss of crossing made by two waveguides. Summing up all the drop loss, through loss, crossing loss, propagation loss, and bending loss of a signal is called the **insertion loss**.

1. Introduction

The flow of designing WRONoCs is usually separated into two consecutive steps: topological design and physical design. The former focuses on the configurations of MRRs and the wavelength assigned for each signal. The goal of this stage is usually achieving as much re-usage of wavelength as possible without conflicts and as little insertion loss as possible for each signal. The letter focuses on the routing of waveguides from IP cores to the positions of corresponding ports of topology. Therefore, the goal of this stage is to minimize the amount of extra crossing loss, bending loss, and propagation loss. However, it has already been reported that the unavoidable crossing loss during this stage can degrade the quality of the signals from the perspective of both insertion loss and SNR if the topological design does not consider the constraints of the physical design (Zheng et al. 2021), such as λ-router (Briere et al. 2007).

One of the mismatches between these two stages is that the location of both the master and slave from the same IP core do not have close ports in topology. This kind of mismatch may incur extra loss of power. Because of the mismatches, some works have already considered both stages, like PSION (Truppel et al. 2019). Nevertheless, their work usually cannot cater to every customized need in physical and topological designs. For example, in PSION, the routing was done using the manually designed physical templates.

This work proposed an automatic generator method that considered topological and physical design features for WRONoC to minimize insertion loss. First, several rectangular areas will be found by solving the assignment problem to predict the possible routing resources used for each master and slave on an IP core. Second, considering the predicted used resources from each IP core to its assigned ports based on the abovementioned areas, a mixed-integer-linear programming (MILP) model can solve the grid routing problem for physical design. Third, with the insertion loss in physical design for each signal, a topology generator based on Customtop can optimize the usage of MRRs, wavelength resources, and the worst insertion loss among all signals. Finally, some post-routing automatic techniques can be applied to further minimize the worst insertion loss in the network.

Even though the mismatch from the location of both the master and slave from the same IP core was mentioned above, it is not the only cause of extra power loss in physical designs. For example, the density of the spaces around each port deu to surrounding IP cores or routing resources is also a critical factor that should be considered in both designs. For more flexibility in the design to overcome mismatches between the two stages, I will not restrict the location of both the master and slave from the same IP core to be close enough. Instead, I will prove that the difference of the worst insertion loss among the network due to the changes in the ports for masters and slaves in a specific topology generator is within an acceptable range, so the physical design should be done first.

The following context of this report is organized as follows: Section 2 shows the modification of CustomTopo, the reasons for considering finishing the physical design first, and the design of CutomTopo+. Section 3 shows how to build up a MILP model for physical design and integrate the prediction of ports on topology within it. Section 4 proposes several techniques in this work to reduce the problem size of the MILP model and methods that can further minimize the worst insertion loss after the physical design. Section 5 shows the results of the experiments of CustomTopo+ and compares the data to other state-of-the-art routers while analyzing it reasonably. Section 6 concludes the contribution of CustomTopo+ and discusses the improvement of its routing of topological design in the future.

2. Background and Previous Work

In this Section, I will present the topology generator used in the program based on CustomTopo and give some additional formulations for the MILP model in the topological design. After this, I will use a network given a fixed size to show the range of changes in the worst insertion loss within the network while the assigned ports of masters and slaves change. Finally, I will present a rough idea and motivation for my physical design.

2.1. The choice and modification of topology generator

To determine the ports' assignments after physical designs, this research needs one customized topology generator where masters and slaves can be assigned to any port on the topology so that it can be physical-design-oriented. Because most topology generators either restrict the locations of ports for masters and slaves or are not designed for customized cases, based on the following reasons, I will modify CustomTopo to serve this purpose (Li et al. 2018). There are a few advantages to using it as the based topology generator. First, the crossbar structure makes it very easy to generalize ports for different combinations of masters and slaves, no matter where the port is located. In the original CustomTopo, the communication paths from one side to the other have been discussed, so the communication paths from the same sides should be added to this work. Second, this work focuses on the worst insertion loss, so one MRR in a CSE is preferred if only one signal passing by needs to drop instead of ADFs. Third, the change of the positions of masters and slaves will not affect the resulting worst insertion loss a lot. This will be explained further in Section 2.2. The advantages above are why I chose CustomTopo as the topology generator.

From Figure 2.1, the possible paths from the original CustomTopo have been shown. There are three potential paths for every signal in total. The first is the default path: from $Master_1$ to $Slave_2$ in the picture. The second one is the direct demultiplexing: in the picture, from $Master_2$ to $Slave_2$. The third one is demultiplexing with a detour for the ADF sharing: in the picture, from $Master_1$ to $Slave_1$. This work will consider the assumption of choosing a default path from the different sides as CustomTopo does so that the communication from the same side adopts similar ideas from CustomTopo.

Figure 2.2 shows the examples for communication on the same side. Because the default path is set to the different side, the possible paths for communication on the same side are only two: One is the path using direct demultiplexing shown on the left-hand side, and the other one is demultiplexing with a detour for the ADF sharing shown on the right-hand side. In CustomTopo, these two ways of paths are different because direct demultiplexing paths will

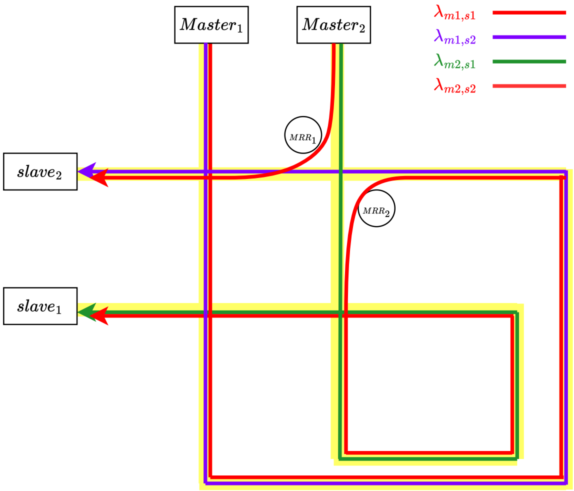
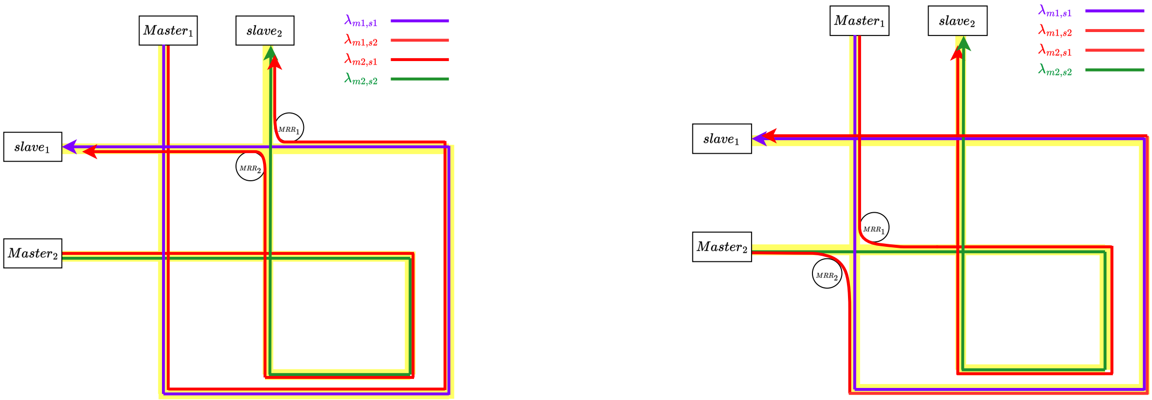


Figure 2.1.: A simple 2x2 node network for the original CuctomTopo with the permission of the self-connection. Besides, different colors of signals mean different wavelengths, and the MRRs are set to couple the same wavelength because of the ADF sharing.



(a) This is an example of direct demultiplexing.

(b) This is an example of demultiplexing with a detour.

Figure 2.2.: The examples routes for communication on the same side.

encounter the coupling ADF with signals before the signals proceed in a different direction. In other words, the signal passes at most only half of the CSEs of a default path before the signal drops. In contrast, as noticed in the picture, these two kinds of paths all need to pass at least half of the CSEs of a default path before the signal drops. Because these two paths are similar when communication happens on the same side, I will generalize them in the following context.

The structure inside the topology will be transformed into a Cartesian coordinate system. This simple transformation will assign a location with (x, y) to each CSE, providing a straightforward

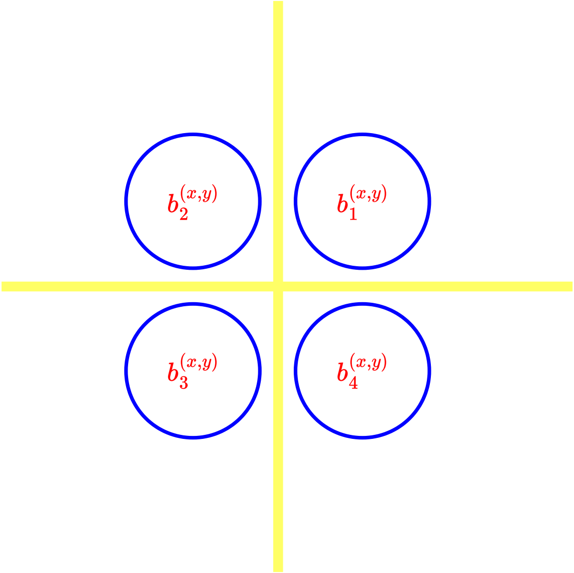


Figure 2.3.: The positions of MRR for binary variables $b_k^{(x,y)} = 1$.

way to understand the topology. Before I formulate the MILP model for these two paths, I will introduce the following four binary variables: $b_k^{(x,y)}$, where k is an integer and $1 \leq k \leq 4$. As shown in Figure 2.3, $b_k^{(x,y)} = 1$ if the corresponding MRR is used in the CSE located in (x, y). Then, to achieve the preference of using one MRR for signals instead of ADF, the following constraint will apply:

$$\sum_{1 \leq k \leq 4} b_k^{(x,y)} \leq 2 \tag{2.1}$$

This equation means that a CSE located in (x, y) can have at most 2 MRRs inside its structure, and this will be applied to each CSE inside the topology.

For every two masters m_{i1},m_{i2} from different sides and every two slaves s_{j1},s_{j2} from different sides, if they can form two kinds of paths as in Figure 2.2, then they have to satisfy the following constraint:

$$b_{i1,j1,i2,j2} \geq 1 - M(\sum_{1 \leq k \leq \omega} b_k^{\lambda_{i1,j2}}) - M(\sum_{1 \leq k \leq \omega} b_k^{\lambda_{i2,j1}}) - (1 - \sum_{1 \leq k \leq \omega} b_k^{\lambda_{i1,j1}}) \tag{2.2}$$

where M is an extremely large auxiliary constant, $b_{i1,j1,i2,j2} = 1$ if they satisfy the constraint to form the abovementioned paths, and $b_k^{\lambda_{i,j}}$ follows the exact representation in the paper of CustomTopo (Li et al. 2018). When $b_{i1,j1,i2,j2} = 1$, it implies that $\lambda_{i1,j2}$ and $\lambda_{i2,j1}$ will be assigned 0 as default paths, and m_{i1} has a connection with s_{i1} so that they can form the paths as in Figure 2.2. Otherwise, the above constraint trivially holds.

The common feature of paths in Figure 2.2 is that they all only use the same relative locations of MRRs corresponding to variables $b_1^{(x,y)}$ and $b_3^{(x,y)}$. Furthermore, I define the location of the

CSE whose MRRs are used for the direct demultiplexing path as $(j1, j2)$ and the location of the CSE whose MRRs are used for the demultiplexing with a detour path as $(i1, i2)$. It is noticeable that $b_1^{(j1,j2)}$ and $b_1^{(i1,i2)}$ are all used for the communication between $Master_1$ and $Slave_2$. A similar notice can be applied to $b_3^{(j1,j2)}$ and $b_3^{(i1,i2)}$ with the communication between $Master_2$ and $Slave_1$. This is related to which side these masters are located on and can be proven trivially because the sides masters can be located on are only the upper and left sides of the topology. Thus, I conclude that for two pairs of a master and slave with communication on the same side, $b_k^{(j1,j2)}$ and $b_k^{(i1,i2)}$, where $k \in \{1, 3\}$, are used for the same and only one communication in the network. To set the corresponding MRRs, the following two constraints are needed:

$$b_1^{(i1,i2)} + b_1^{(j1,j2)} \geq b_{i1,j1,i2,j2} \tag{2.3}$$

This constraint shows the above notice, which will only apply to communications where masters are on the upper side of the topology.

$$b_3^{(i1,i2)} + b_3^{(j1,j2)} \geq b_{i1,j1,i2,j2} \tag{2.4}$$

Similarly, this constraint will only apply to communications where masters are on the left side of the topology.

Following the ideas in CustomTopo, the formulation of the insertion loss in these two new paths can also be built up trivially. With the above consideration of paths for communication on the same side and the formulation in CustomTopo, it is now possible to put masters and slaves in any port.

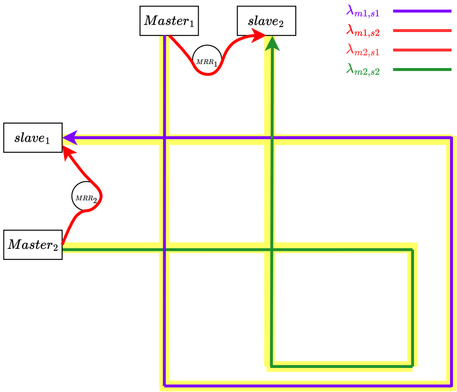


Figure 2.4.: Using one parallel switching element to achieve communication when the master and slave are put next to each other and need communication.

As shown in Figure 2.4, because paths of demultiplexing with a detour can usually become the bottleneck of the worst insertion loss, using one MRR for 180° turns, called parallel switching elements (PSE), to ease the communications of masters and slaves that are placed close is also used in this work. I will put this MRR in the middle of the space between a master and a slave if it is applicable and can cut down the worst insertion loss within the network.

In my methods, the physical design should have a higher priority in finishing than the topological one as long as the change of worst insertion loss due to the change of assigned ports for masters and slaves has a predictable bound.

2.2. Predictable bounds of CustomTopo

In this subsection, I will calculate the difference in the worst insertion loss within an 8x8 topology due to changing the assigned ports for masters and slaves. Given that for an 8x8 topology, there are 64 crossing points and 8 default paths, and there can be 256 MRRs in this topology, two extreme cases will be considered to find the maximum difference in the worst insertion loss. The effect of extra PSEs is not considered in the following context because it will add the fixed maximum amount of 4 through loss for the difference of the worst insertion loss, which is negligible to the following calculation.

The first is the fully connected communication matrix. Each master should communicate with seven different slaves, so 56 MRRs are needed at most. Because demultiplexing with a detour will go through the most CSEs for all paths, I assume the worst insertion loss happens in the path of demultiplexing with a detour for the worst scenario. Notably, even if the worst insertion loss happens in other kinds of paths, their bounds are still smaller than my following calculation. For the path of demultiplexing with a detour, a signal should go through 30 crossing points with one drop in the longest path; there are 15 crossing points from one default path and the rest from another. Because of using one MRR for each signal path other than default paths, there are, in total, at most, 13 MRRs passed by for each default path: a master to 6 different slaves and other masters to the slave of the corresponding default path. In the end, for the worst case, a signal path should have an insertion loss with at most 26 through loss, 30 crossing loss, and 1 drop loss. By changing the assigned ports of the masters and slaves, if MRRs can be distributed evenly to each default path to minimize the worst insertion loss of a single signal path, each default path should have 7 MRRs. In the end, 12 through losses can be improved, so the improvement from the change of assigned ports is minimal.

The second case is the non-fully connected matrix with a very extreme case: only one master needs to communicate with all the other slaves, and all the other masters have no communication. The worst case of the worst insertion loss will then become an insertion loss with at most 5 through loss, 30 crossing loss, and 1 drop loss. However, the best scenario for this case is to rearrange the assigned ports, where the signal should use a direct demultiplexing path. Thus, a signal path should have an insertion loss with at most 5 through loss, 5 crossing loss, and 1 drop loss. Improvements of 25 crossing losses can be achieved. In other words, the most significant change in resulting insertion loss by changing the assigned ports for 8x8 topology should be around 25 crossing loss, and this only happens when all kinds of paths drop of demultiplexing with a detour are not used in the whole topology while considering minimizing the usages of overall wavelengths. This should be the case when some masters have no communication, which is rarely the case nowadays. Besides, for most communication matrices, the use of the paths of demultiplexing with a detour usually has less crossing from around 14 to 20. Therefore, the best achievable improvements in most cases should be under 7 crossing loss and several through loss.

In conclusion, the worst bounds of the change in insertion loss when reassigning ports of masters and slaves are shown. However, for physical design, this kind of bound is more complicated to calculate, and in most cases, to cater to the physical design constraints, more than the costs mentioned above will be needed if the topological design is done first.

2.3. Considerations of physical routing

WRONoCs are usually applied on the three-dimensional integrated circuits (3D ICs) platform with through silicon vias (TSVs). Due to the structure of staking the electronic and optical layers, the locations of the hubs are usually given as the input to the physical design. Adopting the method from ToPro, the topology generated from CustomTopo will be placed directly on the layout in the middle to save the time spent on OSEs. In ToPro, the dynamic pushing algorithm minimizes the crossing outside the topology because the available space for routing in the whole layout is usually huge. Thus, finding a detour to decrease the crossing is generally worthwhile. However, with its method, the overall usage of routing cannot be considered. Therefore, in this research internship, the first step is to find the minimized routing usage without the fixed ports for masters and slaves to do the minimization more flexibly. Next, the result of minimized routing usage with the determined assignments of ports for masters and slaves can be obtained. Then, further post-processing should be done to reduce the number of crossings and bends outside the topology while fixing the amount of used routing resources. Because of keeping the same routing usage after the first step, this additional processing will not worsen the results from the perspective of extra propagation loss, unlike ToPro.

As mentioned previously, the physical design should have higher priority because the flexibility of routing resources can add unpredictable insertion loss if the topological design cannot consider some factors in the physical design. To further give the physical design orientation in topological design, the assignments of the ports for masters and slaves are decided by the physical design. This can further minimize the physical design and realize the physical-design-oriented topology generator.

3. Methodology

In my method, I build up a MILP model to formulate the routing problem and objective functions to minimize the overall usage of routing outside the topology. This problem should be suitable for solving in MILP because the problem of single-layer routing is an NP-complete problem (Richards 1984). Further, it can be regarded as a multiple-source, multiple-destination, multipath routing problem without specifying the destinated destination for each source point with constraints and an objective goal that considers all the paths. Each source will match a destination point that can minimize the global objective function when considering all the other source and destination points. Due to the complexity of the problem and dynamically assigning a destination point for each source point, a MILP model can provide a more straightforward formulation to tackle this problem. In addition, only part of the whole layout will be considered input to the model to reduce the complexity of the MILP problems.

3.1. Topological Feature in Physical Design by Separating Areas

In the following context, the layout will be transformed into a Cartesian coordinate system, so grids will be transformed into points. Besides, **source points** correspond to the locations of masters and slaves of all IP cores, **destination points** correspond to the locations of ports on the topology, and all the other points, not source and destination points, will be regarded as **other points**. The ports on topology will be assigned by physical design, so the routes of signals cannot be kwon during the physical design. However, the worst insertion loss within the network should have the information on the assigned ports. It is then important that the physical design can predict the area each signal will use as precisely as possible so that the worst insertion loss within the network can still be considered. This prediction in physical design can add a topological feature to it. To formulate the worst insertion loss for each route, I will cut several rectangular areas out of the layout, called **boxes** in the following context. These boxes have two goals. One is to include a pair of the source and destination points inside a box to predict the final assignments of ports on the topology. The other is to reduce the overall complexity by restricting the considered area from the whole layout into the areas of boxes so that the number of points in the MILP model can be less. Because each box has a pair of source and destination points, it is easy to draw the area by setting the two points as the diagonal points of a box. The method of determining which source point should match which destination point is the critical problem here. This can be modeled as:

Input: source points and destination points.

Output: pairs of a source point and a destination point with no point selected twice or zero time, while the sum of the area should be minimal, and the predictability should be as precise as possible.

Because the area of a rectangle increases while the distance of its diagonal points increases, and the best solution in the physical design is to connect the two points that are the closest if it is just the shortest path problem without the interference between paths, it can be simplified to the objective goal of minimizing the sum of the distances for each pair. After calculating the distances to all destination points for each source point, this problem can be easily transformed into the assignment problem. I use the Hungarian algorithm to solve this problem (Kuhn 1955). It is worth knowing that the ports assigned to masters and slaves obtained after the assignment problem cannot guarantee they will still be after the physical design. This comes from the fact that the physical design problem is not the shortest path problem, and their common features are few. After getting all the boxes, I will enlarge each box for 3 units according to experiments to ensure the feasibility of the following model, and the points considered in the MILP model will only be those inside one of the boxes. Also, all the areas occupied by IP cores will be called **obstacles**, and the points belonging to obstacles will be removed. This can lead to the following set:

$$P = \{(i,j)|(i,j) \in B^{area} \cap (i,j) \notin O^{area}\} \tag{3.1}$$

Where (i,j) means a point on the Cartesian coordinate system, B^{area} is the area occupied by all boxes, and O^{area} is the area occupied by obstacles. Thus, in the following model, all considered points belong to the set P.

3.2. Set-Up for Physical Design Models

In Section 4.1, I will discuss resizing the grid to speed up the solver. To simplify the model, no matter how big the grid is used in the model, all the points can accommodate only one vertical and horizontal waveguide. The compensation for this simplification to improve the results of the MILP solver will be discussed in Section 4.2. For each point(i,j), I introduce two binary variables $V^{(i,j)}$ and $H^{(i,j)}$, where $V^{(i,j)} = 1$ means there is a vertical waveguide in the point (i,j) and $H^{(i,j)} = 1$ means there is a horizental waveguide in the point (i,j). Although I do not need any label to mark which routes a point belongs to, I still need to differentiate the proceeding directions for each route in points to ensure each source point can route to only one destination point. Therefore, for each vertical waveguide, I also have the two binary variables $V_{up}^{(i,j)}$ and $V_{down}^{(i,j)}$ to represent the direction of a route from a source point to a destination point upward and downward respectively. That is, $V_{up}^{(i,j)} = 1$ when the route starting from a source point to a destination point passes the vertical waveguide used in the point(i,j) upwardly. Similarily, $V_{down}^{(i,j)} = 1$ when the it passes the vertical waveguide used in the point(i,j) downwardly. The same concept can also apply to the horizontal one, so $H_{left}^{(i,j)} = 1$ when it passes the horizontal waveguide used in the point(i,j) leftward, and $H_{right}^{(i,j)} = 1$ when it passes the horizontal waveguide used in the point(i,j) rightward. I can then formulate the constraints for each grid:

$$\forall(i,j) \in P, V^{(i,j)} = V_{up}^{(i,j)} + V_{down}^{(i,j)} \tag{3.2}$$

and

$$\forall(i,j) \in P, H^{(i,j)} = H_{left}^{(i,j)} + H_{right}^{(i,j)} \tag{3.3}$$

These two constraints will apply to each point.

To smoothly explain the constraints in the following context, I also introduce four abbreviations: $D^{(i,j)} = V_{up}^{(i,j-1)}$ only if $(i, j-1) \in P$, $U^{(i,j)} = V_{down}^{(i,j+1)}$ only if $(i, j+1) \in P$, $R^{(i,j)} = H_{left}^{(i+1,j)}$ only if $(i+1, j) \in P$, and $L^{(i,j)} = H_{right}^{(i-1,j)}$ only if $(i-1, j) \in P$. These abbreviations are used to indicate in two ways. First, they represent all the possible directions and locations for a route into point (i, j). Second, if the point from the specific direction is not considered in P, then the corresponding abbreviation will not exist.

Constraints for different kinds of points will be presented in the following context. According to the conditions for the abbreviations above, some point (i, j) on the edge of the layout or next to the obstacles will only define less than four abbreviations mentioned above. However, all these points should still have at least one defined abbreviation to apply the following constraints after removing a few items. Otherwise, the point can be removed from P during the program. Thus, for conciseness, I will only consider the case when four abbreviations are defined for the point (i, j). In other cases, removing the items that do not exist can be done trivially.

3.2.1. Source Points

In the following context, I will use S to represent a set of all source points. Source points should choose one of the directions to start its route, so I first have the constraint:

$$\forall(i,j) \in S, V_{up}^{(i,j)} + V_{down}^{(i,j)} + H_{left}^{(i,j)} + H_{right}^{(i,j)} = 1 \tag{3.4}$$

This equation means that only one of the directions inside the waveguides of the source points must be chosen. If the next point following the direction does not exist, the variable item from that corresponding direction can be removed.

Besides, all the neighbor points should not go into a source point:

$$\forall(i,j) \in S, U^{(i,j)} + D^{(i,j)} + L^{(i,j)} + R^{(i,j)} = 0 \tag{3.5}$$

This equation means that all the possible directions and locations for a route into source points are forbidden. The item can be removed if the variable item is not defined for the reasons above.

3.2.2. Destination points

In the following context, I will use D to represent a set of all destination points. Destination points should follow the directions from the previous one in the route so that I can get the following constraint:

$$\forall(i,j) \in D, V_{down}^{(i,j)} + V_{up}^{(i,j)} + H_{right}^{(i,j)} + H_{left}^{(i,j)} = 1 \tag{3.6}$$

This equation means that only one of the directions inside the waveguides of the destination points must be chosen. If the previous point following the direction does not exist, the variable item from that corresponding direction can be removed.

Besides, one of the neighbor points should go into a destination point:

$$\forall(i,j) \in D, U^{(i,j)} + D^{(i,j)} + L^{(i,j)} + R^{(i,j)} = 1 \tag{3.7}$$

This equation means that only one of all the possible directions and locations for a route into destination points can be chosen. The item can be removed if the variable item is not defined for the mentioned reasons.

3.2.3. Other Points

In the following context, I will use OP to represent a set of all other points. Other points should have the same number of waveguides going inward and going outward so I can obtain the following constraints:

$$\forall(i,j) \in OP, (U^{(i,j)} - V_{up}^{(i,j)}) + (D^{(i,j)} - V_{down}^{(i,j)}) + (L^{(i,j)} - H_{left}^{(i,j)}) + (R^{(i,j)} - H_{right}^{(i,j)}) = 0 \tag{3.8}$$

This equation means that the number of the used waveguides inside the point equals the number of the waveguides going into it. Suppose the previous point from the direction does not exist. In that case, the variable item from that corresponding direction, as well as the variable item of the waveguide going in that direction inside the point, can be removed.

3.2.4. Insertion Loss and Objective Fuctions

The objective function of this MILP model is the worst insertion loss within all the boxes. To know the insertion loss inside a box, there is a need to formulate the propagation, crossing, and bending loss situations. For a crossing to occur, it means that there are both existing vertical and horizontal waveguides within a grid. I introduce a binary variable $C^{(i,j)}$ and its constraint, where $C^{(i,j)} = 1$ means there is a crossing in the point (i,j).

$$\forall(i,j) \in P, V^{(i,j)} + H^{(i,j)} <= 1 + C^{(i,j)} \tag{3.9}$$

For a bending to occur, it means that there is a change of the direction in the point(i, j) for a route, and in other words, it means that the previous direction of a route cannot proceed in the point (i, j). I introduce a binary variable $B^{(i,j)}$ and its constraint, where $B^{(i,j)} = 1$ means there is a bending in the point (i, j).

$$\forall (i, j) \in P, if(i, j + 1) \in P, -M * V^{(i,j+1)} + V_{up}^{(i,j)} - B^{(i,j)} <= 0 \tag{3.10}$$

$$\forall (i, j) \in P, if(i, j - 1) \in P, -M * V^{(i,j-1)} + V_{down}^{(i,j)} - B^{(i,j)} <= 0 \tag{3.11}$$

$$\forall (i, j) \in P, if(i - 1, j) \in P, -M * H^{(i-1,j)} + H_{left}^{(i,j)} - B^{(i,j)} <= 0 \tag{3.12}$$

$$\forall (i, j) \in P, if(i + 1, j) \in P, -M * H^{(i+1,j)} + H_{right}^{(i,j)} - B^{(i,j)} <= 0 \tag{3.13}$$

Where M is a huge auxiliary constant. For the propagation loss, it is easy to calculate from the knowledge of $V^{(i,j)}$ and $H^{(i,j)}$. Thus, I can get the insertion loss inside a box now. For each box, I introduce a continuous variable P_i for box_i and assign the value as follows:

$$\begin{aligned} &\forall box_i \in B, \\ P_i = &\sum_{point(i,j) \in box_i} (V^{(i,j)} + H^{(i,j)}) * L * loss_{propogation} + B^{(i,j)} * loss_{bending} + C^{(i,j)} * loss_{crossing} \end{aligned} \tag{3.14}$$

Where B is a set of all boxes, L is the length of a grid, $loss_{propogation}$ is the constant value of propagation loss, $loss_{bending}$ is the constant value for a bending loss, and $loss_{crossing}$ is the constant value for a crossing loss. At the end of this model, I introduce a continuous variable P_{worst} to represent the worst insertion loss within all boxes and set the objective function to minimize this variable:

$$\forall box_i \in B, P_i <= P_{worst} \tag{3.15}$$

$$Minimize: \ P_{worst} \tag{3.16}$$

4. Improvements for Model and Flow Process

The complexity of the problem in physical design can stress out the slowness of MILP solvers. The following techniques of reducing the input problem size are used to speed up the solvers. Further, methods to reduce crossing and bending loss will be presented to improve the solutions from a MILP solver when running overtime.

4.1. Pre-Model Processing for Improvements

In this section, the focus will be on how to further reduce the complexity of the problem. Because most of the constraints and variables in our MILP model bind to a single point, the direct factor of the speed in the solver is the number of points or grids. Assume the grids used are all rectangles; the length of the minimum grid, L_{min}, is according to the minimum distance between waveguides and must be satisfied. The maximal length, L_{max}, has the following relation:

$$L_{max} = \text{Max}\{L_{min}, \text{Min}\{D^{port}_{min}, D^{master,slave}_{min}\}\} \tag{4.1}$$

Where D^{port}_{min} is the minimal distance between any two ports on a topology, and $D^{master,slave}_{min}$ is the minimal distance between a slave and a master from any IP core. Thus, to reduce the number of grids on the layout, we can increase the size by increasing L_{max}. It is noteworthy that after resizing the grids, the routing done with the MILP solver is grained and should be transformed into the layout of the grids of minimal size, which is called refined routing in the following context. Further, if masters and slaves of IP cores are located very sparsely, the considered area is also bigger than the dense case. Although we cannot move the locations of IP cores, we can rotate them to become closer to the topology for some improvements.

4.1.1. Auto-Adjusting: The Spaces Between Ports

In most cases, only the distance between ports can be adjusted for users. D^{port}_{min} also has the minimal value to satisfy according to the design of a topology. Because CustomTopo is used as the topology generator, D^{port}_{min} must be larger than twice the diameter of an MRR to support the full function of the CustomTopo. In the program, I adopt the diameter of an MRR to be 35 µm, as Proton+ did, so D^{port}_{min} is at least 70 µm. The maximal value of D^{port}_{min} is bounded by the solvability of the given input, and in my experiments, I set this value to be 75 µm, so the enlargement of the size of the topology will not over 1.14 times. Besides, I will only adjust the length of grids one by one until the total number of points is less than 8000.

4.1.2. Auto-Adjusting: The Rotation of IP Cores

The rotation of an IP core can also change the locations of the master and slave on it without any move for an IP core. I will utilize the easiest way to adjust each IP core automatically. Because the rotation of one IP core cannot affect the final rotation of other IP cores, I can try rotating each IP core for 0°, 90°, 180° and 270°, and choosing the one rotation, where the master and slave of an IP core can be closest the center point of the topology.

4.2. Post-Model Processing for Improvements

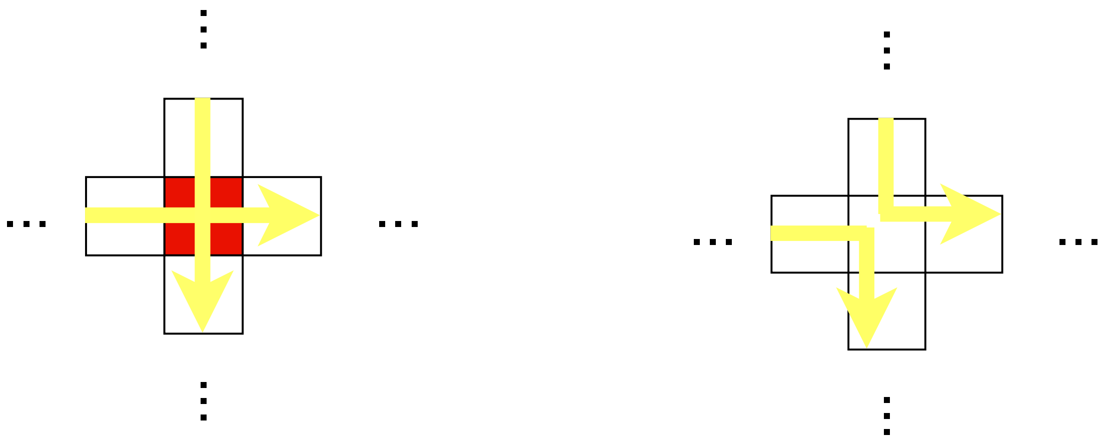
Because of the complexity of the MILP model, the MILP solver cannot explore all feasible solutions after the time limit, which is 20 minutes in the program. The final solution can not be guaranteed to be globally optimized. This means improving the solution obtained from this MILP solver is necessary. Although I can directly copy the same routes back to the layout with the minimal size of grids without any modification, to improve the final result of the physical design, I will focus on the trade-offs between crossing loss and bending loss and between bending loss and propagation loss while transforming from grained routing to refined routing.

4.2.1. Crossing

Crossings in the physical design have been reported as the main reason for the deterioration of the quality of final signals in ToPro. Thus, removing crossings is essential, especially considering their effect on the signal-to-noise ratio. Crossings on the grained routing can be easily removed while transforming into the refined routing. The reason for a crossing to occur at a point is that one route continues its direction and intersects the proceeding direction of another route. To avoid this, swapping the directions of both routes to each other can be used so that there will be no crossing at this point. Figure 4.1 shows that crossings in the grained routing can be removed when mapped onto the refined routing by swapping their proceeding directions. This also ensures the total number of used points remains the same while replacing a crossing loss with a bending loss for a signal. However, due to the particular positions required to connect the ports on topology and a master or slave on an IP core, there are still a few extra crossings left after the transformation.

4.2.2. Bending

After transformation, I can reduce the number of bendings for each route under a certain threshold. The potential unnecessary bendings could happen if there are at least four bendings or, in other words, five pieces in a route. The horizontal and vertical directions change should be known by finding this section's starting and end points. In most cases, the first direction of



(a) This is an example of the crossing in the grained routing. The red-filled rectangle is where the crossing happens.

(b) This is an example of removing the crossing from the grained routing to the refined routing. There is no crossing in the figure.

Figure 4.1.: The example of removing crossings.

the section should be kept because it is usually necessary, or the change of it cannot make the process easier and effective in the final result. Therefore, I will start changing the route from the second piece and third piece of the section. Alternative routes that begin changing from the later pieces cannot make much difference compared to the original section. Thus, for this section of a route, there are three possibilities for most cases.

Firstly, the original section is the optimized one because there are many obstacles. Secondly, the route can keep the first piece of the section and extend the first direction of the route to satisfy the amount of change in either vertical or horizontal direction. Then, there is one bending for the amount of the other direction. Thirdly, the route can keep the first and second pieces of the section and extend the second direction of the route to satisfy the remaining change in either the vertical or horizontal direction. Then, there is one bending for the remaining amount in the other direction. As shown in Figure 4.2, the alternative paths can be found by extending the original route's first and second pieces. This way, the area around the original path can be explored without extra propagation loss. I will compare the section with other alternative paths from the perspective of insertion loss and exclude the paths that cannot be valid, for example, using the same waveguide at a point as another route. The experiments will show that this method can guarantee that most final signals have a bending loss of less than or equal to 6 if the obstacles are not many and are spared.

4.3. Overall Flow

In this section, I will present the process of CustomTopo+:

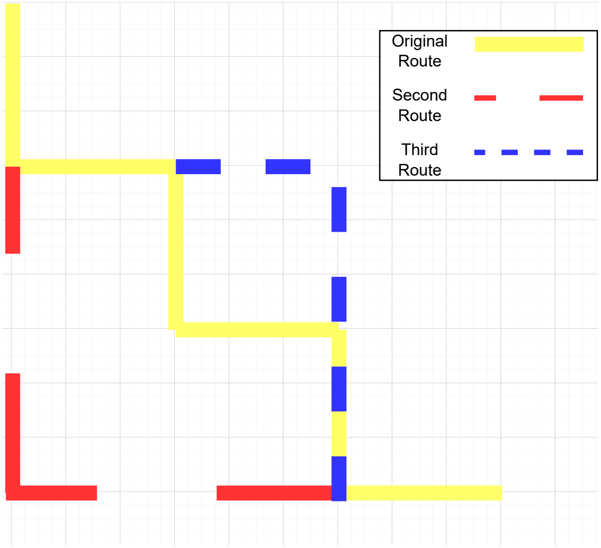


Figure 4.2.: This example shows how the two alternative paths can be found from the original path.

Input: the communication matrix, the positions of all IP cores with length and width, and the layout information.

Output: a physical design with the embedded topology in the middle and all the routing and placement of the IP cores.

Step 1 — Preprocessing: Getting the exact locations of each master and slave of IP cores and the center point of the placed topology. Then, as mentioned in the section 4.1, the adjustments to the rotations of each IP core and the distance between the ports will be applied automatically.

Step 2 — Assignment Problem Formulating: Getting the exact locations of each port of topology and inputting all the locations mentioned above into the assignment problem model as Section 3.1 does. After solving it with the Hungarian algorithm for a minimal sum of costs, each source point, as a master or a slave from an IP core, will be temporarily assigned a destination point, a port on the topology.

Step 3 — MILP formulating: Using the maximum size of a grid to map the original layout to a smaller one and develop the MILP model for each box derived from a pair of source and destination points, as described in Section 3.2, while considering the points within boxes with removal of all the points that belong to obstacles.

Step 4 — Physical Design: Solving the MILP model for minimal worst insertion loss within all boxes and transforming the layout to the original layout with further optimization mentioned in the section 4.2. The MILP solver here can run for a maximum of 20 minutes. At this step's end, the ports are assigned to specific masters and slaves while the insertion loss in physical design can be known.

Step 5 — Topological Design: With the insertion loss already calculated and the ports determined by the physical design, the topology design is physical-design-oriented. The modified version of CustomTopo, following the instructions from Section 2.1, can configure the used MRRs and the wavelengths for each signal for minimal worst insertion loss within all signals, minimal usage of MRRs, and minimal wavelength usage.

5. Analysis and Comparisons

Table 5.1.: Results from CustomTopo+ compared with other existing routers for 8-node networks with different positions of memory controllers.

Test	Router	Topology	il_w	L	C	T
position(a)	Proton+	GWOR	8.4	13.0	38	88.5
	PlanarONoC	GWOR	6.4	28.6	10	0.1
	ToPro	GWOR	3.8	14.2	8	0.19
		Light	5.5	21	12	0.19
	CustomTopo+	CustomTopo	4.2	11.7	16	1279.2
position(b)	Proton+	GWOR	9.1	14.7	41	81.5
	PlanarONoC	GWOR	n/a	n/a	n/a	n/a
	ToPro	GWOR	5.0	22.2	8	0.15
		Light	6.4	33.3	6	0.2
	CustomTopo+	CustomTopo	4.5	12.9	18	1274.6
position(c)	Proton+	GWOR	8.1	11.0	38	88.5
	PlanarONoC	GWOR	n/a	n/a	n/a	n/a
	ToPro	GWOR	4.5	18.4	8	0.14
		Light	5.2	19	12	0.15
	CustomTopo+	CustomTopo	3.7	10.0	16	1280.9
position(d)	Proton+	GWOR	8.1	13.8	35	79
	PlanarONoC	GWOR	n/a	n/a	n/a	n/a
	ToPro	GWOR	4.0	13.5	10	0.17
		Light	4.3	13.5	12	0.07
	CustomTopo+	CustomTopo	3.8	10.3	16	1277.2

il_w: the maximal insertion loss value denoted by dB. L: the path length of the signal with maximal insertion loss denoted in mm. C: the number of crossings (including the crossings in the OSEs) passed by the signal with maximal insertion loss. T : the program runtime denoted in seconds.

CustomTopo+ is implemented in C++, and all experiments discussed here were carried out on a 1.7GHz CPU. In Section 5.1, I will compare the same experiments conducted from ToPro with CustomTopo+ and give the reasons for the data. In Section 5.2, I will analyze CustomTopo+ from the perspective of its runtime, bending, and crossing from both topological and physical design. I will also show the usefulness of techniques from Section 4.

5.1. Comparisons with State-of-the-art Routers

I synthesized one layout with CustomTopo+ for an 8-node processor-memory network with four hubs and four memory controllers for 44 signals. The node locations, die dimension, size of OSEs, and loss parameters are the same ones applied in ToPro. The node locations have four different settings: memory controllers located (a) pairwise at the periphery, (b) at the corner, (c) on the four sides, and (d) at the leftmost side of the photonic layer. In PlanarONoC, there is only an experiment considering the position(a) setting, so the synthesis results for other position settings are unavailable. Table 5.1 shows the results of the comparison. In general, CutomTopo+ outperforms all the other routers from the perspective of the worst insertion loss. Benefiting from integrating features in both topological and physical designs during the process, on average, CustomTopo+ reaches 5% improvements in the worst insertion loss compared to ToPro. For all cases, because the crossings of CSEs are tricky to remove for a bit dense network, CustomTopo+ always has a higher number of crossings than ToPro. Thus, on position(a), CutomTopo+ has a worse value of inertion loss than ToPro with GWOR implementation because position(a) is very friendly for getting a good physical design. In contrast, the weakness of the topological design of Customtopo+ will be stressed out. Due to the use of a MILP solver, CutomTopo+ can have the least signal length with the worst insertion loss. On the other hand, it also becomes the bottleneck for CutomTopo+ to have fast processing as ToPro does. Besides, CustomTopo+ also demonstrates more stable performance for different node locations. The variance of worst insertion loss of CustomTopo+ can reach 30% improvements compared to ToPro.

5.2. Analysis of CustomTopo+

Table 5.2.: Detailed statistics from CustomTopo+ for 8-node networks with different positions of memory controllers.

Test	$\|P\|$	D_{min}^{port}	$C_{topological}$	$C_{physical}$	B	$T_{topological}$	$T_{physical}$
position(a)	7919	70	16	2	6	71.2	1200
position(b)	13559	75	16	2	6	63.9	1200
position(c)	4834	75	16	0	6	75.0	1200
position(d)	5652	70	16	0	6	73.5	1200

$|P|$: the number of considered points in the physical design. D_{min}^{port}: the minimal distance between two ports on a topology denoted in mm. $C_{topological}$: the most crossings in the OSEs among all signals. $C_{physical}$: the most crossings, excluding the ones in OSEs among all signals. B: the most bendings among all signals. $T_{topological}$: the program runtime of topological design denoted in seconds. $T_{physical}$: the program runtime of physical design denoted in seconds.

From Table 5.2, it is evident that the crossing of topological design is the bottleneck for better worst insertion loss of CustomTopo+, and the time spent on the physical design also contributes to more than 90% of programming time. However, the resulting physical designs show their strength with little crossing and bending less than 7. Therefore, the post-routing techniques are proven valuable and critical in this work. The values of $|P|$ also explain why CustomTopo+ has the worst value in position(b). The following Figures show the synthesized results of CustomTopo+ ignoring the routing part of the default paths.

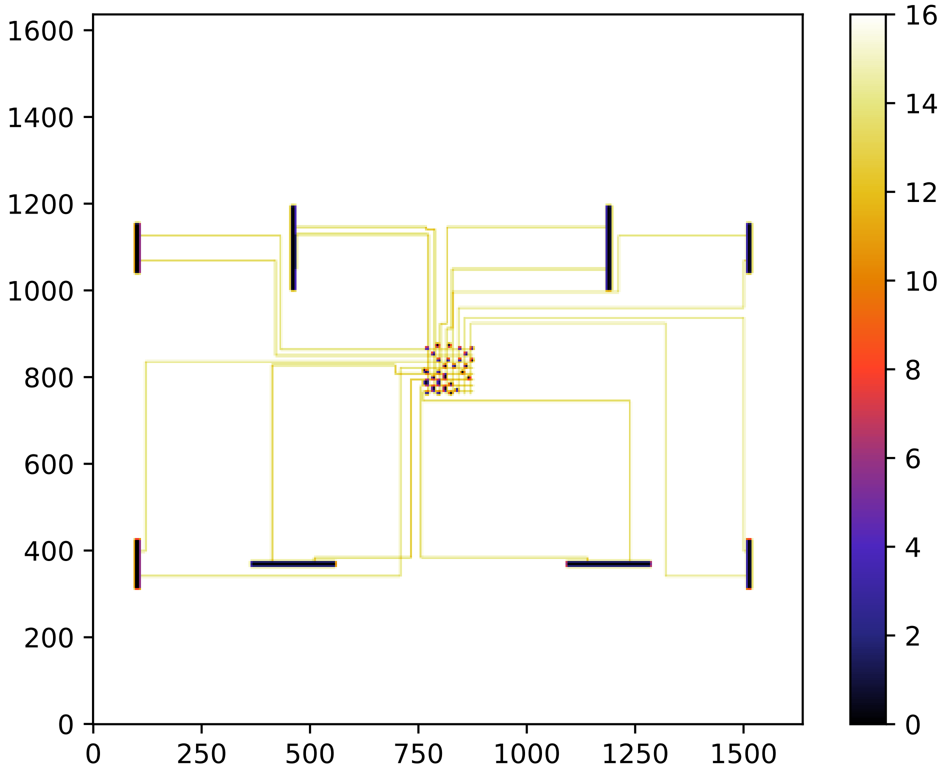


Figure 5.1.: The synthesized layout of position (a) by CustomTopo+.

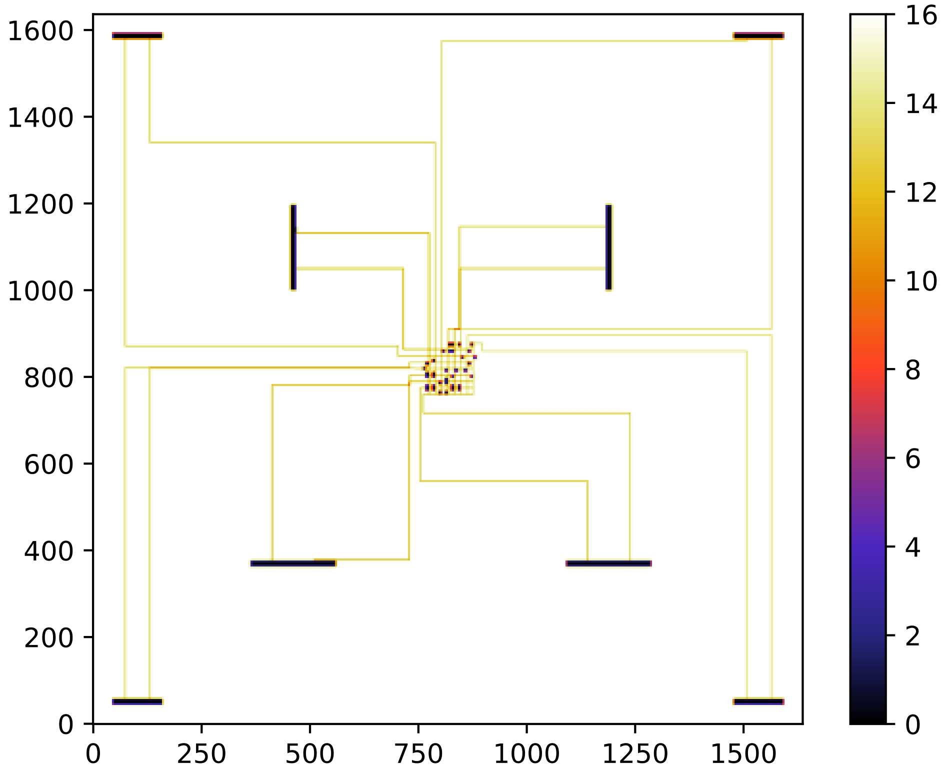


Figure 5.2.: The synthesized layout of position (b) by CustomTopo+.

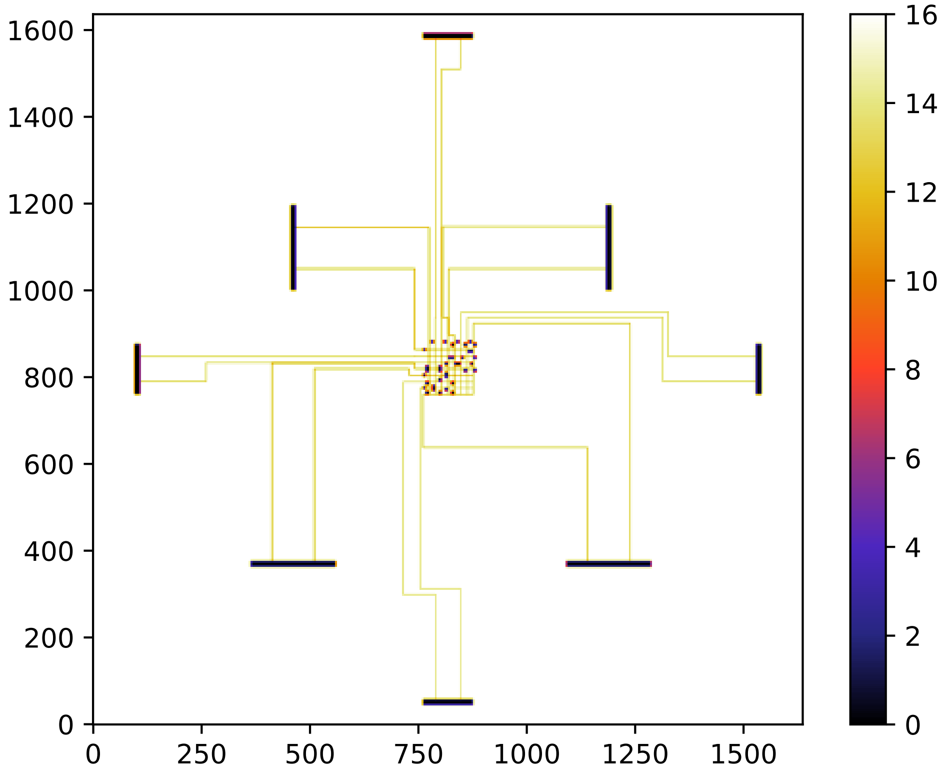


Figure 5.3.: The synthesized layout of position (c) by CustomTopo+.

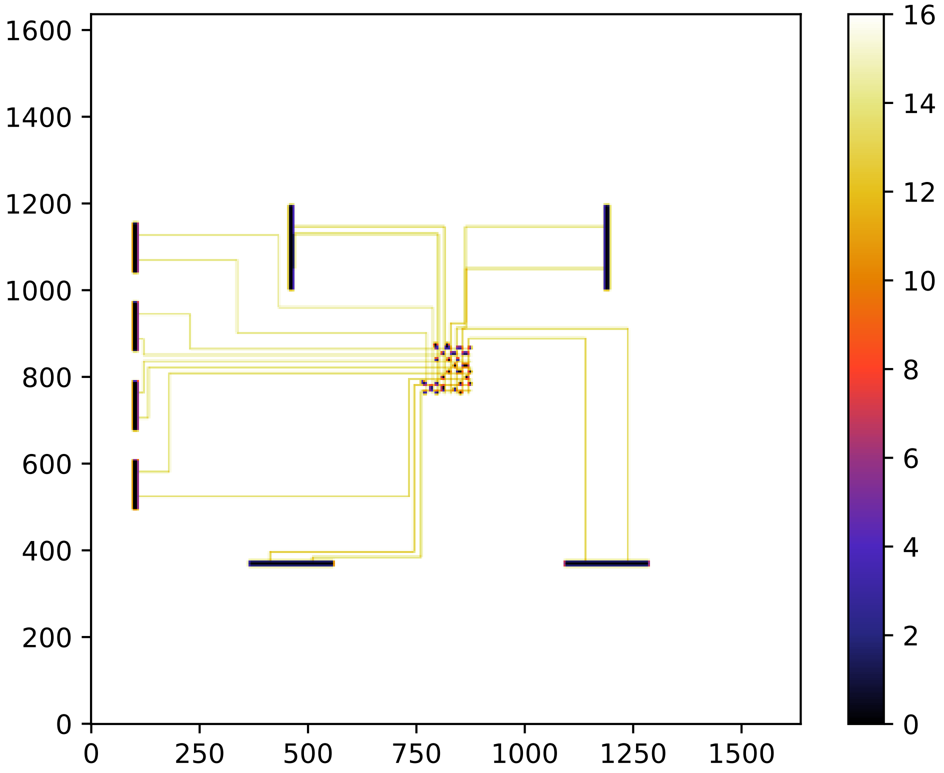


Figure 5.4.: The synthesized layout of position (d) by CustomTopo+.

6. Conclusion and Discussion

In this work, I propose a tool to finish topological and physical designs based on the customized topology generator: CumtomTopo+. Due to the flexibility of its topology generator, the physical design can be completed first while integrating the features of topological design. It also generates several alternative paths from the perspective of crossing and bending and chooses the one with less insertion loss to improve the design. By finishing the physical design first with much flexibility of unassigned ports for masters and slaves, the topological design can be physical-design-oriented while giving a predictable bound of its degradation in the worst insertion loss, which is almost impossible if the topological design is done first. Compared with the state-of-the-art physical design tools, CustomTopo+ shows its stable performance and outperforms them in the worst insertion loss. To further enhance the SNR of signals, considering the topological design and physical design simultaneously and the same platform could be possible for future implementation because of the flexible crossbar structure of the topology used in CustomTopo. In summary, CustomTopo+ can automatically generate synthesized layouts without manual effort while guaranteeing its performance for any input due to its physical design information during topological design.

Bibliography

Bogaerts, Wim, De Heyn, Peter, Van Vaerenbergh, Thomas, De Vos, Katrien, Kumar Selvaraja, Shankar, Claes, Tom, Dumon, Pieter, Bienstman, Peter, Van Thourhout, Dries & Baets, Roel (2012): Silicon microring resonators, Laser & Photonics Reviews **6**(1): 47–73.

Briere, Matthieu, Girodias, Bruno, Bouchebaba, Youcef, Nicolescu, Gabriela, Mieyeville, Fabien, Gaffiot, Frédéric & O'Connor, Ian (2007): System level assessment of an optical noc in an mpsoc platform, 2007 Design, Automation & Test in Europe Conference & Exhibition, IEEE, S. 1–6.

Cheng, Liaoyuan, Li, Mengchu, Tseng, Tsun-Ming & Schlichtmann, Ulf (2024): Minimizing worst-case data transmission cycles in wavelength-routed optical noc through bandwidth allocation, IEEE/ACM International Conference on Computer-Aided Design (ICCAD).

Kuhn, Harold W (1955): The hungarian method for the assignment problem, Naval research logistics quarterly **2**(1-2): 83–97.

Li, Mengchu, Tseng, Tsun-Ming, Bertozzi, Davide, Tala, Mahdi & Schlichtmann, Ulf (2018): Customtopo: A topology generation method for application-specific wavelength-routed optical nocs, 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), IEEE, S. 1–8.

Li, Mengchu, Tseng, Tsun-Ming, Tala, Mahdi & Schlichtmann, Ulf (2020): Maximizing the communication parallelism for wavelength-routed optical networks-on-chips, 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), IEEE, S. 109–114.

Manolatou, Christina, Haus, Hermann A, Manolatou, Christina & Haus, Hermann A (2002): High density integrated optics, Passive Components for Dense Optical Integration S. 97–125.

Richards (1984): Complexity of single-layer routing, IEEE Transactions on Computers **100**(3): 286–288.

Tan, Xianfang, Yang, Mei, Zhang, Lei, Jiang, Yingtao & Yang, Jianyi (2011): On a scalable, non-blocking optical router for photonic networks-on-chip designs, 2011 Symposium on Photonics and Optoelectronics (SOPO), IEEE, S. 1–4.

Truppel, Alexandre, Tseng, Tsun-Ming, Bertozzi, Davide, Alves, José Carlos & Schlichtmann, Ulf (2019): Psion: Combining logical topology and physical layout optimization for wavelength-routed onocs, Proceedings of the 2019 International Symposium on Physical Design, S. 49–56.

Vantrease, Dana, Schreiber, Robert, Monchiero, Matteo, McLaren, Moray, Jouppi, Norman P, Fiorentino, Marco, Davis, Al, Binkert, Nathan, Beausoleil, Raymond G & Ahn, Jung Ho (2008): Corona: System implications of emerging nanophotonic technology, ACM SIGARCH Computer Architecture News **36**(3): 153–164.

Werner, Sebastian, Navaridas, Javier & Luján, Mikel (2017): A survey on optical network-on-chip architectures, ACM Computing Surveys (CSUR) **50**(6): 1–37.

Zheng, Zhidan, Chang, You-Jen, Cheng, Liaoyuan, Tseng, Tsun-Ming & Schlichtmann, Ulf (2025): A backup resource customization and allocation method for wavelength-routed optical networks-on-chip topologies, IEEE/ACM Asia and South Pacific Design Automation Conference (ASP-DAC).

Zheng, Zhidan, Cheng, Liaoyuan, Arisawa, Kanta, Li, Qingyu, Truppel, Alexandre, Yamashita, Shigeru, Tseng, Tsun-Ming & Schlichtmann, Ulf (2024): Multi-resonance mesh-based wavelength-routed optical networks-on-chip, ACM/IEEE Design Automation Conference (DAC).

Zheng, Zhidan, Li, Mengchu, Tseng, Tsun-Ming & Schlichtmann, Ulf (2021): Topro: A topology projector and waveguide router for wavelength-routed optical networks-on-chip, 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD), IEEE, S. 1–9.