Automatic design of NxN integrated Benes optical switch

(Article begins on next page)

19 April 2024

Automatic design of NxN integrated Beneš optical switch

Lorenzo Tunesi[a], Giuseppe Giannuzzi[b], Ihtesham Khan[a], Jigesh Patel[c], Enrico Ghillino[c], Vittorio Curri[a], Andrea Carena[a], and Paolo Bardella[a]

[a]Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino, Italy
[b]Dipartimento Interateneo di Fisica, Politecnico di Bari, via Amendola 173, 70126 Bari, Italy
[c]Synopsys, Inc., 400 Executive Blvd Ste 101, Ossining, NY 10562, United States

ABSTRACT

We propose a general procedure for the automatic design of $N \times N$ integrated optical switches using the Beneš configuration based on ring filters, suitable for multi-hop routing in meshed optical networks. The basic switch components (waveguides, couplers, optical crossings) are selected from the process development kit (PDK) of the target platform, and can be integrated with user-defined custom components.

The described procedure is able to automatically select the optimal number of rings and their proper routing, minimizing the number of optical crossings and reducing the system transmission losses. The generated topology can be directly simulated in the Synopsys OptSim© environment at a system level. There, the impact of the design options on transmission performances is evaluated for the chosen signal modulation scheme and numerical simulations are automatically executed to determine, among the 2^M possible optical paths across the electrically tuned M rings, the only N! configurations required to generate all the permutations of the inputs signals to the output ports minimizing the transmission BER.

Finally, the generated Beneš switch can be converted to a GDSII mask targeting the foundry of choice and manual adjustment to the layout can be made. The automatic procedure takes advantage of the vertical integration of the Synopsys environment, which allows to perform a comprehensive investigation of the whole photonic device, from the component design to the system level analysis, thanks to the integration of Optsim-optical communication system, Optsim schematic-driven photonic circuit, Rsoft component design tools, and OptoDesigner© mask layout.

Keywords: Optical switches, Beneš networks, Automatic design

1. INTRODUCTION

The ever increasing global Internet traffic, due to 5G technology deployment, increase in connected users, as well as bandwidth intensive applications as FHD/4K streaming, will require an increase in the optical communications and networking perfomance in the following years. This trend of rapid increase in demands over the years will require an update of optical networking that supports all metro and core data transport, as well as intra- and inter- data center connections. It will require a full implementation of the openness and disaggregated paradigms, to enable full virtualization of network functions down to the physical layer. Thus, lightpaths carrying high-rate Wavelength-division multiplexing (WDM) data transport can be fully managed and controlled by a virtualized environment within the network controller.[1]

The core networking operation at the physical layer is the transparent routing of lightpaths performed by optical switches,[2] enabling a dynamic deployment of transparent optical circuits from source to destination within a meshed optical infrastructure.

The traditional implementation of optical switches through electro-optic conversion cannot handle the high speed, low latency requirements of the present and future streaming and communications demands, thereby requiring a fully transparent implementations through photonic integrated circuits (PICs).[3]

Further author information: paolo.bardella@polito.it

The Beneš switch is a prime example of a completely transparent optical switch implementable through a PIC, with a small foot-print and rearrangeable non-blocking properties, making it a viable switch for general-purpose open networks.

This switching topology is able to route any permutation of the N inputs, through a network of 2×2 Optical Switching Elements (OSE).
The design of this network and its elements depends on the transmission parameters of the lines, as well as the wanted technological and integrated implementation of the physical OSEs.

In this work, we show for the first time a procedure for the automatic design of a Beneš switch, given a reference PDK and the properties of the communication system. Starting from the characterization of the physical layer, we move then to the circuit construction and the system level simulations and finally we generate the layout mask for its fabrication. These operations can be chained in a continuous workflow thanks to the strong integration between the various components of the Synopsys multi-layer design environment,[4,5]

The RSoft Photonic Component Design Suite operates at the component level, where the Maxwell equations are integrated by using algorithms of computational electromagnetics such as beam propagation method (BPM), finite difference time domain (FDTD), eigenmode expansion (EME). The versatility of these methods and their versatile implementation in RSoft allow the user to efficiently simulate a vast number of complex device geometries taking into account the peculiar properties of the used materials.[6,7] OptSim Circuit operates at link and system level. Advanced modulation formats and digital signal processing (DSP) can be used in the simulation of optical communication systems, whose performances can be evaluated in terms of bit error ratio (BER), optical signal to noise ratio (OSNR), eye diagram, etc. The generated circuits can be directly converted to a layout ready for fabrication in a foundry thanks to OptoDesigner which is able to use the circuital description used in Optsim Circuit to automatically create the mask, allowing the user to fine tune the result using its scripting capabilities and perform the final design rule check (DRC).

Through the use of the Synopsys photonic design environment[4] and an external tool for the interaction with the user and the numerical elaboration of some results (in our case, implemented in MATLAB®), we propose a generalized approach to the design of optical Beneš networks, from the microring resonator (MRR)[8] design, to the internal routings control, up to the production masks. The developed tools allow the user to tailor the specifications of the device to the wanted requirements, with the ability to test and change the individual components, while the physical parameters are automatically evaluated from the transmission specifications. Feedback on the performance of the OSE, as well as for the complete Beneš network are evaluated through the use of Synopsys Optsim and shown to the user to allows step by step validation of the design.
In the tools, we implemented a full routing algorithm, obtaining the driving voltages controlling the Beneš switch, as a function of the required output. This allows the user to test specific routings in a system level abstraction, with evaluation of the BER and GSNR for the given modulation format and characteristic.
As in the Beneš switch multiple equivalent routings for each output permutation are present, the routing algorithm allows the test and selection of the best driving voltages, eliminating the sub-optimal results, based on an empiric metric defined by the user (number of ON states, Power loss, crosstalk, BER). These tools allow the user to design a customizable structure through a guided bottom-up approach, layering each design choice from the PIC elements, up to desired communication abstraction, to create and analyze an in-depth, generalized Beneš transparent optical switch.

At the current stage, the tool only focuses on the optical routing and does not include an automatic routing of the electrical part yet.

In the following, first a brief description of the main characteristics and properties of Beneš networks is presented. Then, each step of our design algorithm is discussed. Results are reported for a reference 6x6 Beneš switch, with channels spaced by 100 GHz in the C band; the device is based on Silicon waveguides, 500 nm × 220 nm (width x high) grown on Silicon Dioxide substrate; components are extracted from the Synopsys demo PDK (DEMOFAB, version 1.0).

(a) Logical 2×2 crossbar switch

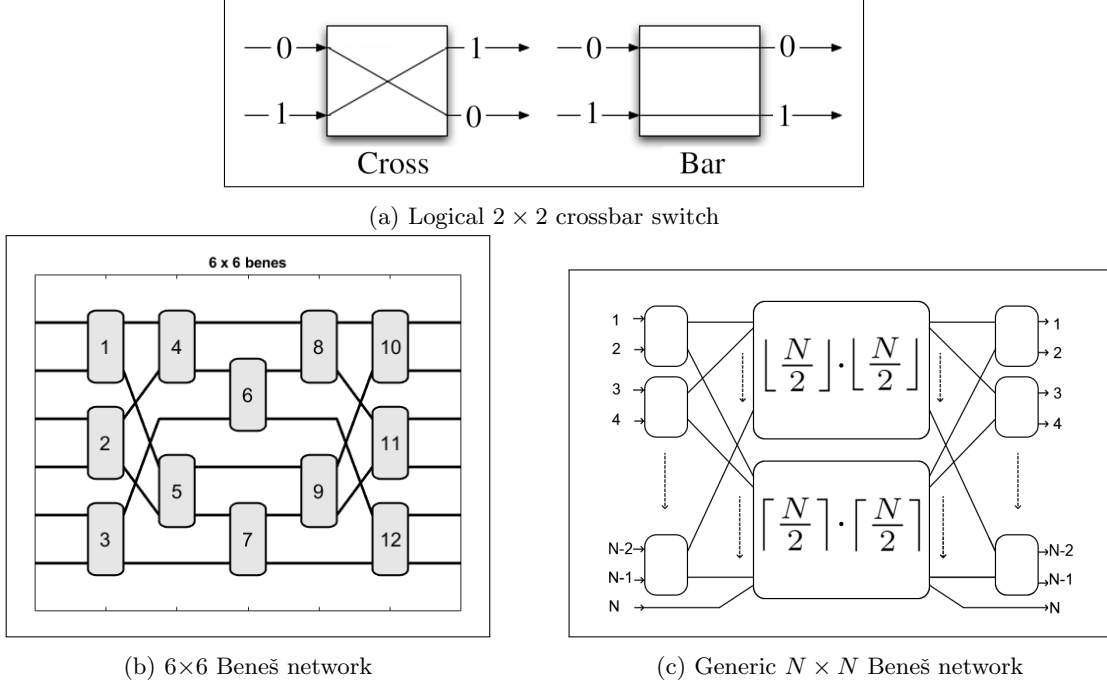(b) 6×6 Beneš network

(c) Generic $N \times N$ Beneš network

Figure 1: test

2. CHARACTERISTICS OF BENEŠ NETWORKS

The Beneš network is a sub-class of the Clos topologies: it consists of a multistage crossover $N \times N$ network, with 2×2 crossbar switches as fundamental elements.

The Beneš switch is a rearreangable non-blocking network, meaning that for a given number of input signals N, the network is able to route all possible signals permutations to the output ports. The rearreangable property of this topology is due to its inability to guarantee routing in cases of already established input-output calls: while in a strict-sense non-blocking network the new connections are always possible if the output port is unoccupied, in this class of topologies the pre-existing calls may need to be routed through different paths.

As a result all $N!$ output combinations are possible, but the control algorithm needs to account for the already existing traffic in the network.

The main advantage of this structure is the reduced number of basic switching elements needed, with respect to the strict-sense counterpart, leading to a smaller footprint and lesser power consumption.

The basic logical switch of the network, as depicted in Fig. 1a, is the 2×2 crossbar switch. This element is driven by a binary control signal and can be in one of two states: the BAR state, with input and output maintaining the same order $(\left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right) \to \left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right))$ and the CROSS state, with input and output order inverted $(\left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right) \to \left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right))$.

In our application of an Optical Beneš switch the crossbar in implemented through a first-order MRR add-drop filter, as depicted in Fig. 5a.

The interconnections between the basic elements follows a recursive definition,[9] depicted in Fig. 1c: while the Beneš is theoretically defined for a number of inputs $N = 2^x$, $x \in \mathbb{N}$, a generalization to any arbitrary size is possible, maintaining the same properties and overall behavior.

The basic rule for the construction a Beneš network is the top-bottom interconnection of each switch with the following sub-networks: at each stage the nth switch is connected with the respective n^{th} port of both the following top and bottom recursive networks.

These successive networks are generated as a stage of $\left\lfloor \frac{N}{2} \right\rfloor$ and $\left\lceil \frac{N}{2} \right\rceil$ switches respectively, applying the recursion up to the central stage of the network.

Beneš networks can be analyzed from a combinatorics point of view, yielding two important parameters:

- The number of inputs N, leading to the total unique output combinations $N!$.
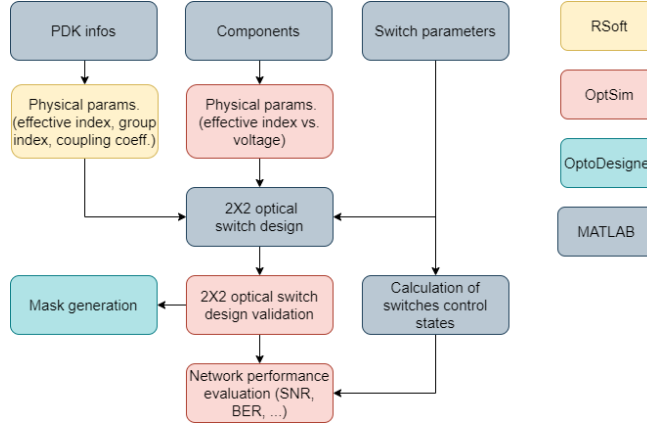
Figure 2: Flow chart of the proposed design algorithm. Blocks are colored according to the program mainly used for that step.

- The number of control signals (switches) M, leading to the total number of states of the network 2^M

Considering that $M \geq N \cdot \log_2 N - \frac{N}{2} \gg N$ and $2^M > N!$, it's clear that Beneš networks allow multiple logically-equivalent routings for the same output goal, enabling the optimization and study of the best configuration for the required parameters.

3. AUTOMATIC DESIGN ALGORITHM

The automatic design algorithm is briefly summarized in Fig. 2. The version 2019.09 of the Synopsys suite was used for all the simulations.

3.1 User specified parameters

Our automatic design of the Beneš switch starts from the collection of details about the technological platform and the requirements of the specific application. A MATLAB® Graphic User Interface (GUI) has been designed to guide the user, simplifying the selection of the proper parameters and reducing the probability of human errors. The GUI, shown in Fig. 3, is divided in three main sections. The first one, *PDK infos*, is used to gather the info on the PDK which are required for a proper configuration of Rsoft and Optsim.

In particular, Rsoft requires a PDK info file, which contains the geometrical details of the waveguides, and a PDK materials file which lists the materials used for substrate, cladding and the waveguides core: this parameters are stored in text files (with extension .txt and .ind, respectively) that are simple to elaborate. On the other hand, Optsim PDKs have a more articulated structure, and are organized in a number of subfolders containing for each component of the PDK its mathematical model, the list of parameters the user can modify in the program, the corresponding GDS layout, etc. Moreover, more versions of the same PDK can be installed at the same time. The list of installed Optsim PDKs is automatically loaded in the GUI to simplify the selection of the PDK name and version. The last parameter of this section is the minimum waveguide curvature radius. This parameter is used in the design of the add-drop rings, which are fundamental components of the Beneš switch; it is also employed to limit the radius of automatically generated optical routing during the creation of the GDS layout.

The second section of the GUI asks the user to identify, among the various components of the previously selected PDK, the ones which describe the fundamental blocks used for the Benes constructions, namely the straight waveguide, the arc/bend waveguide, the directional coupler, the temperature controlled phase shifter, the waveguide crossing, and the edge coupler or vertical grating coupler used to optically connect the PIC to the external fibers. The list of components available in a PDK can be easily retrieved scanning the file PDKModels.xml, available in each Optsim PDK. However, the names of the single components are arbitrary, and
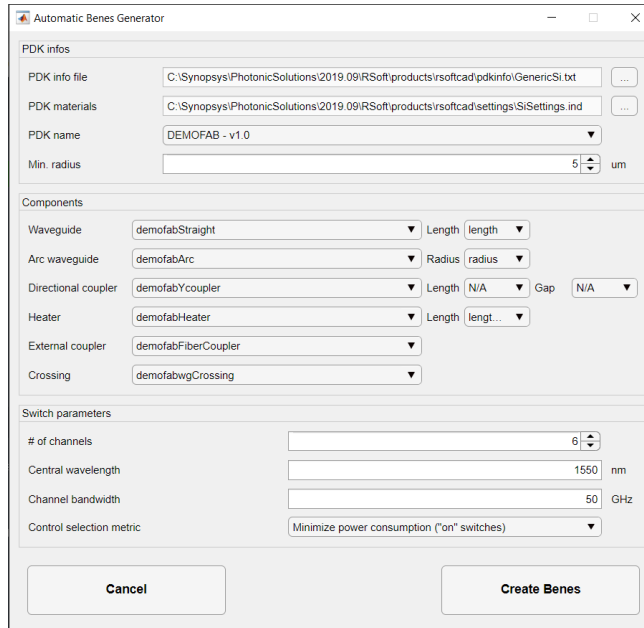
4

Figure 3: Input GUI of the Beneš design procedure

it's not possible to reliably identify the functionality of each component in an automatic way. While the GUI tries to infer the goal of a component searching for certain strings in its name (such as 'straight', 'arc', 'heater',...) the user has verify or perform the matching manually. In a similar way, in Optsim, each component is characterized by a variable number of parameters (e.g., the angle and the radius of the arc waveguide) whose name can be arbitrary chosen by the authors of the PDK. The GUI asks the user to select, for a few fundamental parameters, the right names among the list of available ones, extracted from the file OCHybridPDK.xml. Unfortunately, not all the required PDK components are always developed in a general and flexible way. For instance, in many PDKs the thermally controlled phase shifter is shipped with a fixed length, which cannot be changed, and the directional coupler does not allow to modify the length of the waveguides of their gap, simply asking for the coupling coefficient parameter. In these cases, the user can specify Not Available as parameter name; as described in the next sections the program will try to integrate as much as possible the missing elements.

It is worth observing that the data extracted by our automatic procedure are accessible even when encrypted PDKs are used to protect the intellectual property of the foundry. In fact, in those cases, the models (in the .moml files) are protected, but the name of the components and their parameters, which are of course visible in the Optsim interface, are not encoded.

The last section is used to gather design parameters of the switch, such as the number of channels N, the central wavelength, the bandwidth of each channel.

3.2 Material parameters extraction from simulations

The first step in the generation of the Beneš is the calculation of a set of material parameters which are essential for the correct design of the add-drop rings, but are not directly available from the PDK data. Using Rsoft and the details about the waveguide sizes and materials, we automatically perform a set of simulations to estimate the effective refractive index at the design central wavelength, and the group refractive index. For this calculation, the FemSIM module is used, which is able to efficiently calculate the refractive index of the waveguide fundamental mode using a finite elements method. The input file is dynamically created using the parameters extract from the PDK info and PDK materials files; the computational domain is a rectangle which width and height are set to six times the width and height of the simulated waveguide.

In a similar way, the coupling coefficient of the directional coupler is estimated using Rsoft BeamPROP module, based on the Beam Propagation Method, to simulate the propagation in two waveguides, as a function
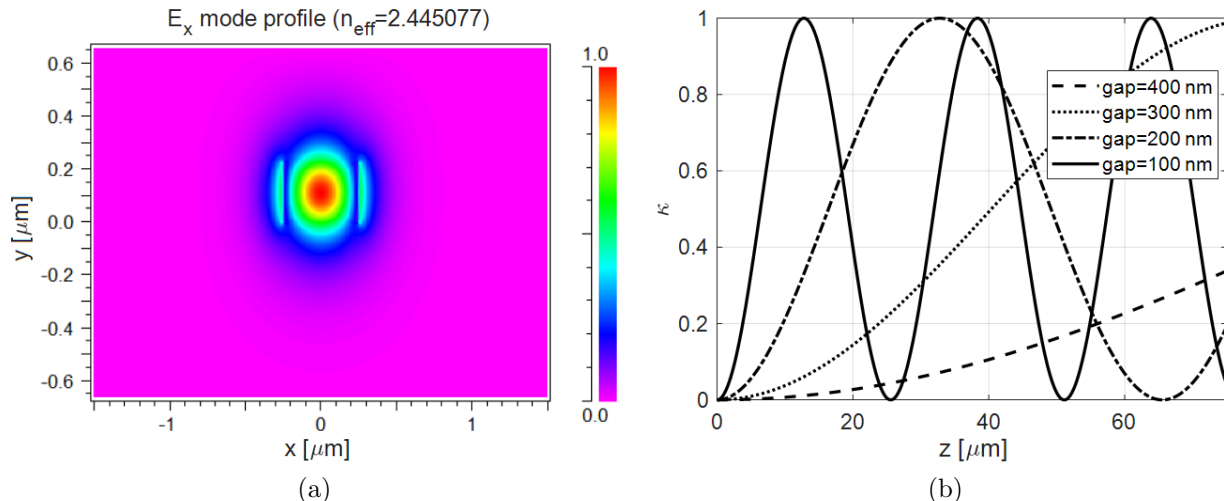
Figure 4: Example of automatically generated Rsoft results. (a) Fundamental transverse mode profile for the straight waveguide of the selected DEMOFAB PDK and (b) coupling coefficient as a function of the gap and the coupler length.

of their mutual distance and length. Effective index method (2.5D) is used to speed up the calculations and reduce the memory consumption; one of the waveguide is initially excited with the fundamental transverse mode.

Another important step which is preliminarily performed is the estimate of the refractive index dependence with respect to voltage externally applied to the heater section. Using the previously calculated effective index, group index, and coupling coefficient, we design in Optsim an add-drop ring resonator: we use $\kappa = 0.8$, and use the minimum specified radius for the curved parts. Thermally controlled sections are used to complete the resonator, whose schematic is shown in Fig. 5b. In the automatically generated Optsim simulation (Fig. 5c), a broadband white source is applied at the input, and the signal is extracted at the through port using a spectrum analyzer; simulations are repeated stepping the voltage from 0 V to 18 V with step 3 V. The variation of refractive index in the phase control regions is calculated observing the shift in the transfer function peaks, as shown in Fig. 5c.

3.3 2×2 Crossbar switch design

The OSE is generated using a first order add-drop ring filter as template (Fig. 5a), and the previously evaluated coupling coefficient is substituted inside the Optsim schematic (Fig. 5b).
The overall round-trip length of the ring element is evaluated starting from the design frequency f_1 and the FSR set by the user.

To guarantee resonance at the design wavelength, the equation

$$\frac{2\pi}{\lambda} n_{eff} L = 2\pi k \tag{1}$$

must hold; setting $k = 1$ we can obtain the length step required to compute the final length:

$$L_{step} = \frac{\lambda}{n_{eff}} \tag{2}$$

From the FSR using the relationship (3) we can evaluate the approximated round-trip length of the ring:

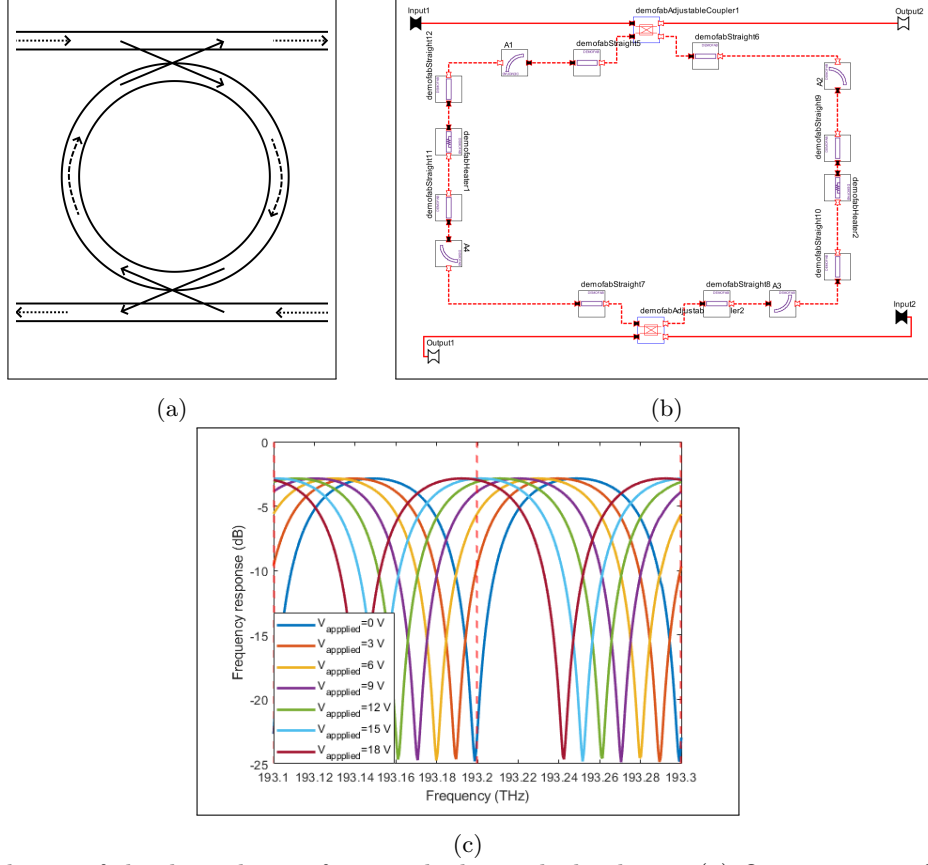$$FSR \approx \frac{\lambda^2}{n_{eff} \cdot L_{app}} \tag{3}$$

6

(a)

(b)

(c)

Figure 5: Calculation of the dependence of n_{eff} with the applied voltage. (a) Optsim circuit for the add-drop resonator,(b) Topological model of the first order switch (c) Transfer function at the through port as a function of the applied voltages.

From the preliminary and step length, k can be evaluated, leading to the final ring length, as shown in (4):

$$k = \left\lfloor \frac{L_{app}}{L_{step}} \right\rfloor$$

$$L_{final} = k \cdot L_{step}$$

(4)

As stated in the previous section, the Optodesigner implementation of many PDK elements doesn't allow for length customization through Optsim variables: the evaluated length is correct for the Optsim simulation of the ring elements, shown in Fig. 5b, although manual adjustments must be made to maintain the correct length in the mask design process.

The second design parameter for the OSE is the switching voltage V_{on}: this value represent the electrical signal needed to heat the phase shifting components as to move the frequency response of both ports by $\Delta_\phi = \frac{FSR}{2}$. This shift can be simulated for various applied voltages, as shown in Fig. 5c, and displays a linear dependency. In order to reduce the amount of simulations needed to find the correct V_{on}, an initial step signal is applied, in order to approximate it through a linear equation. From this initial approximation further simulations are carried out in order to tailor the accuracy of the value up to a certain degree, based on the position of the central frequency of the stop-band.

Following these process the model for the OSE is fully functional and can be implemented in the network.
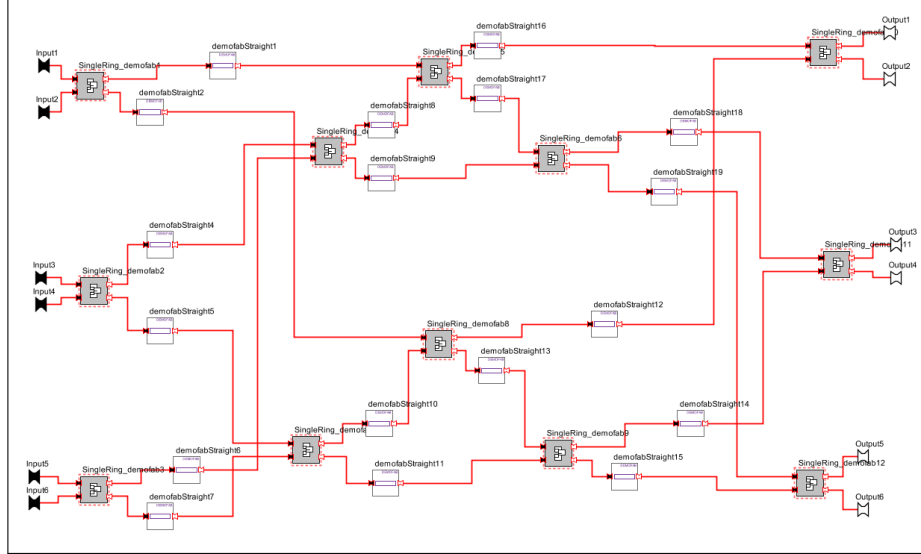
Figure 6: 6×6 Beneš schematic generated by the tools

3.4 Complete Beneš switch construction

As explained in Section 2, and shown in Fig. 1c, the Beneš topology can be built recursively, for an arbitrary number of input channels N.

The first step is the construction of an abstract mathematical model, which will serve both for the generation of the Optsim schematic, as well as for the routing analysis.

The mathematical model is composed of a cascade of switching stages, acting as permutation matrices depending on the control vector, and interconnect stages, linking the OSEs following the Beneš topology template.

From the mathematical model, the Optsim implementation can be achieved by automatically modifying a *.moml* Optsim schematic, adding the OSE entities designed in the previous phase, as well as the interlinks, dictated by the recursive generation algorithm.

The control voltages in the OSEs are parametrized in order to allow for further simulation through the external coding utility.

The 6×6 Beneš example used in the paper is shown in Fig. 6: the network is automatically designed as a compound component, meaning that implementations in more complex systems can be handled by using the Optsim feature, allowing to import the whole network as a single-block model, similarly to the OSE elements.

3.5 Mask generation

Once the complete Beneš has been assembled, the last step of the design procedure consist in taking advantage of the mask generation capabilities of OptoDesigner to automatically generate a script which, once compiled, creates the GDSII layout. Due to the complexity of the addressed circuits, a manual fine tuning of the position of the components in the script can be required, but the automatically generated script is indeed an excellent starting point for the creation of a production ready layout.

The layout obtained from the Optsim circuit in Fig. 6 is shown in Fig. 7; minimal manual adjustments where required to properly align the blocks. The actual mask would use a more dense layout in order to reduce PIC footprint; in the proposed image we increased the spacing to improve readability.

3.6 Control states evaluation

In order to test the actual performance and behavior of the Beneš switch, it is necessary to evaluate and test different control configurations, in order to ensure the correct input-output link formation, as well as the QoT. Given the multiple equivalent paths available to obtain the same output configuration, as discussed in 2, the
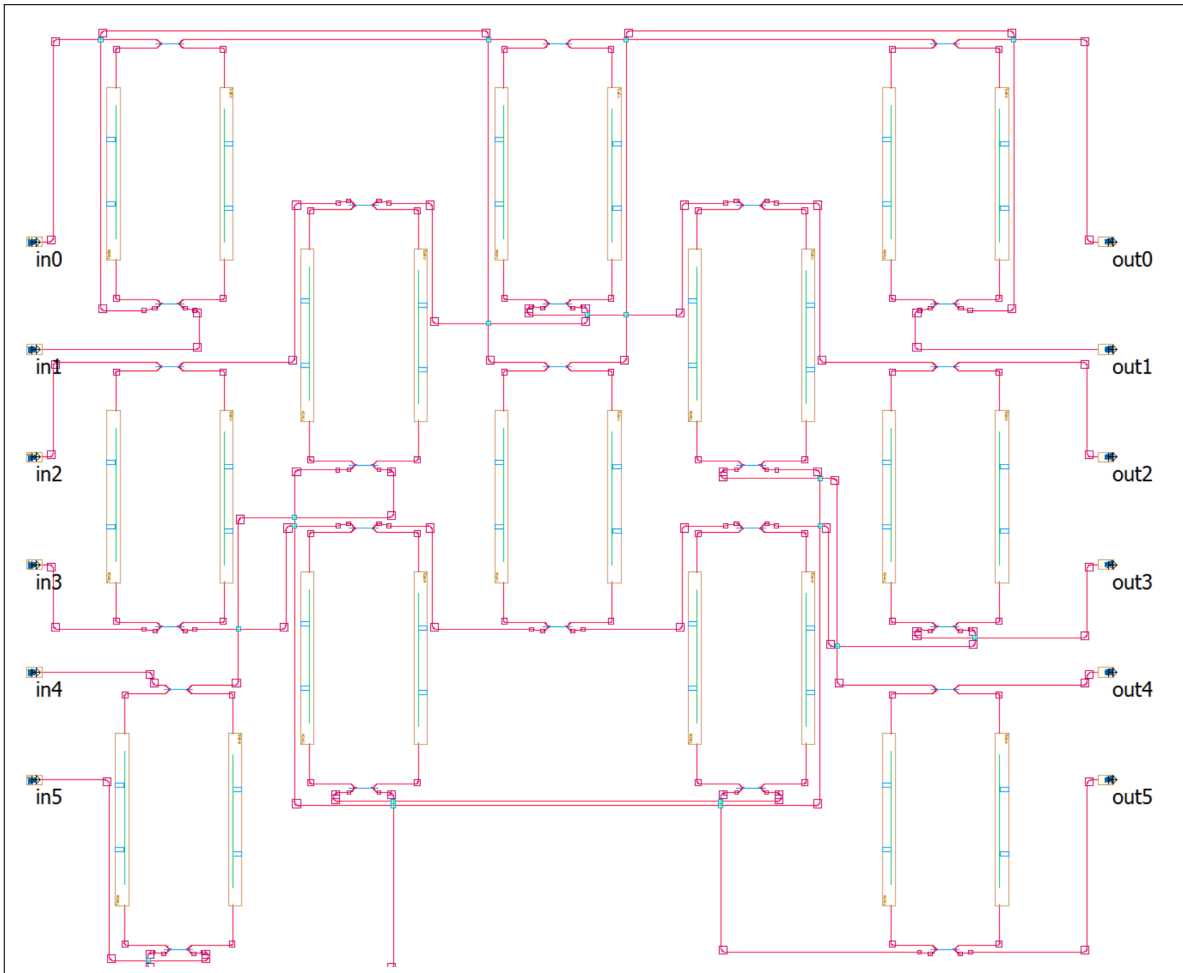
8

Figure 7: Mask layout for the 6×6 Beneš shown in Fig. 6
.

user might also require an evaluation of the best routing to guarantee a specified metric.

Various routing algorithms are described in the literature, concerning the Beneš switch, starting from the historical looping algorithm,[10] up to more modern matrix-based implementations.[11]

We opted to implement the matrix-based routing, with a custom-made generalization for any arbitrary-sized Beneš (the proposed one only works for strict-sense Beneš structures).

The algorithm works on a mathematical abstraction of the network, with the control of each internal switch represented by a single bit: the final control vector is a $1 \times M$ binary array, which can then be multiplied by the previously evaluated on-off voltage, leading to a correctly sized control signal.

The end result is an algorithm able to evaluate not only a single control, but the full set of control vectors routing the same output configuration.

It is important to note that, due to the rapid computational growth of the Beneš network, following the factorial and exponential dependency on the parameters, the complete routings evaluation may not be feasible: this is not a limitation of the tools or the algorithm, but an intrinsic bottleneck due to the non-polynomial increase in the size of the solution space.

For reasonably sized networks, the user can specify the metric for the selection of the best routing, with approaches at different level of simulation complexity:

- A simple metric can be the evaluation of the ON-OFF states in the control vectors, choosing the best solution under a simplified assumption on the loss and the power consumption of the switches.

- With a more in-depth simulation on the frequency response, it's possible to target the control for the maximization of the power transfer to the target port, or the minimization of the adjacent channels crosstalk.

- The last, more time consuming metric, is the full evaluation at system-level of the BER and GSNR, allowing a more realistic simulation of the routings effects on the QoT.

It's important to observe that the different metrics have considerably different computational times, especially the system level simulation, as such it's unfeasible to test all $N!$ unique combinations, for a high order switch. The simulation of the frequency response offers an intermediate metric to infer the actual performance of the network, without requiring the lengthy analysis based on the modulation and transmission parameters

4. CONCLUSIONS

We presented an automatic procedure to design and simulate a user-defined customizable Beneš crossover switch. Through the workflow the user can easily produce and test different implementation of this class of switches, under a unified and consistent procedure, allowing comparison between the resulting performances.

Given the flexible and generalized approach used at each step, the capabilities of the tools can be expanded to cover broader topologies and component implementations, as such can be considered a case-study for a coherent and reliable design process for PICs.

The described procedure could be easily adapted and extended to the vertical design of many other PICs, significantly speeding up the development of this kind of integrated components, while offering a standardized benchmark for simulating performances.

5. ACKNOWLEDGEMENTS

REFERENCES

[1] Curri, V., "Software-defined wdm optical transport in disaggregated open optical networks," in [*2020 22nd International Conference on Transparent Optical Networks (ICTON)*], 1–4 (2020).

[2] Tu, X., Song, C., Huang, T., Chen, Z., and Fu, H., "State of the art and perspectives on silicon photonic switches," *Micromachines* **10**(1) (2019).

[3] Helkey, R., Saleh, A., Buckwalter, J., and Bowers, J., "High-performance photonic integrated circuits on silicon," *IEEE Journal of Selected Topics in Quantum Electronics* **25**(5) (2019). cited By 10.

[4] "Synopsys photonic solutions." https://www.synopsys.com/optical-solutions/rsoft.html. Accessed: 2021-1-31.

[5] "Pic design suite." https://www.synopsys.com/photonic-solutions/pic-design-suite.html. Accessed: 2021-1-31.

[6] Giannuzzi, G. and Bardella, P., "Semiconductor racetrack resonator coupled to an s-bent waveguide: Influence of the coupling coefficients on the unidirectional operation," *IET Optoelectronics* (2021).

[7] Ghillino, E., Virgillito, E., Mena, P. V., Scarmozzino, R., Stoffer, R., Richards, D., Ghiasi, A., Ferrari, A., Cantono, M., Carena, A., and Curri, V., "The synopsys software environment to design and simulate photonic integrated circuits: A case study for 400g transmission," in [*2018 20th International Conference on Transparent Optical Networks (ICTON)*], 1–4 (2018).

[8] Giannuzzi, G., Ghillino, E., and Bardella, P., "Geometry optimization of unidirectional integrated ring laser," in [*Integrated Optics: Devices, Materials, and Technologies XXIV*], García-Blanco, S. M. and Cheben, P., eds., **11283**, 295 – 301, International Society for Optics and Photonics, SPIE (2020).

[9] Chang, C. and Melhem, R., "Arbitrary size Beneš networks," *Parallel Processing Letters* **07** (05 1997).

[10] Opferman, D. C. and Tsao-Wu, N. T., "On a class of rearrangeable switching networks part I: Control algorithm," *Bell System Technical Journal* **50**(5), 1579–1600 (1971).

[11] Chakrabarty, A., Collier, M., and Mukhopadhyay, S., "Matrix-based nonblocking routing algorithm for Beneš networks," in [*2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns*], 551–556 (2009).