



POLITECNICO DI TORINO

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATIONS

—  
MASTER'S THESIS IN ELECTRONIC ENGINEERING

A.Y. : 2020/2021

---

# Integrated Beneš optical switches: an automated bottom-up design implementation

---

*Author:*

Lorenzo Tunesi

*Student Number:*

S267572

*Supervisors:*

Professor Paolo Bardella

Professor Andrea Carena

# Contents

---

<b>Abstract</b>	<b>vi</b>
<b>Acknowledgements</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Optical networks and fiber technology . . . . .	1
1.2 Network elements . . . . .	2
1.3 Optical switches . . . . .	4
1.4 General approach and goals . . . . .	6
1.5 Transmission & simulation parameters . . . . .	8
<b>2 Optical switching elements</b>	<b>10</b>
2.1 Microring resonators filters . . . . .	11
2.1.1 First order filter analysis . . . . .	12
2.1.2 Second order filter analysis: directly coupled MMRs . . . . .	17
2.1.3 Second order filter analysis: crossing coupled MMRs . . . . .	23
2.1.4 Control signal . . . . .	27
2.2 Simulation models . . . . .	29
2.2.1 RSoft waveguide technologies . . . . .	29
2.2.2 OSE modelling . . . . .	30
2.2.3 Crossing . . . . .	33
2.3 MATLAB scripts . . . . .	35
<b>3 Beneš Switching Network</b>	<b>38</b>
3.1 Multistage switching network . . . . .	38
3.1.1 Banyan switches . . . . .	38
3.1.2 Clos networks . . . . .	40
3.2 Beneš topology & properties . . . . .	43
3.2.1 Arbitrary size Beneš . . . . .	45
3.3 MATLAB: implementation . . . . .	48
3.3.1 Graph-based approach . . . . .	48
3.3.2 Logical equations approach . . . . .	50
3.3.3 Matrix-based approach . . . . .	53
3.3.4 Computational costs . . . . .	55
3.4 MATLAB: Brute Force approach . . . . .	58
3.4.1 Cross-Bar states optimization . . . . .	58
3.4.2 Switch optimization . . . . .	59

3.5	MATLAB: routing . . . . .	63
3.5.1	Matrix-based routing evaluation . . . . .	64
3.5.2	AS-Beneš generalization . . . . .	69
3.6	Advanced routing control: machine-learning approach . . . . .	71
3.7	Optsim Circuit implementation . . . . .	74
<b>4</b>	<b>System level simulations</b>	<b>76</b>
4.1	Single input frequency response . . . . .	77
4.2	Filtered channels transmission . . . . .	81
4.2.1	Passive state . . . . .	82
4.2.2	Active state . . . . .	84
4.2.3	Mixed state . . . . .	86
4.2.4	Alternative paths . . . . .	88
4.2.5	Filtered channels: reduced bandwidth . . . . .	90
4.3	Future expansion . . . . .	94
<b>5</b>	<b>Layout Mask</b>	<b>95</b>
5.1	PDK Libraries . . . . .	95
5.2	OptoDesigner Mask implementation . . . . .	98
<b>6</b>	<b>Conclusions</b>	<b>100</b>
	<b>References</b>	<b>101</b>

# List of Figures

---

<b>Introduction</b>	<b>1</b>
1 Attenuation of optical fibers, with highlights of the three main propagation windows . . . . .	1
2 Simple topological example of a generic network . . . . .	2
3 Block model of an optical link between two nodes . . . . .	3
4 Generic representation of an $N \times N$ switch . . . . .	4
5 Black-box model for a $2 \times 2$ crossbar switch . . . . .	4
6 Different topologies of multistage crossover switching networks . . . .	5
7 Proposed workflow for the design of the Beneš switch . . . . .	7
8 Example of a WDM spectral representation . . . . .	8
9 Constellation diagrams for different size QAM formats . . . . .	9
<b>Optical switching elements</b>	<b>10</b>
10 $2 \times 2$ crossbar switch based on the Mach-Zehnder Interferometer . . .	10
11 First order MRR-based add-drop optical filter . . . . .	11
12 Signal routing for both states of the first order OSE . . . . .	12
13 Effect of the coupling coefficient $k$ on the response of the first order OSE (default state) . . . . .	15
14 Response of the first order OSE chosen as benchmark ( $k = 0.6$ ) . . .	16
15 Signal routing for both states of the second order DC OSE (direct coupling) . . . . .	17
16 Effect of the coupling coefficient $k_1$ on the response of the second order DC OSE (default state) . . . . .	19
17 Effect of the coupling coefficient $k_2$ on the response of the second order DC OSE (default state) . . . . .	20
18 Degenerate second order DC OSE ( $k_2 = 1$ ) . . . . .	21
19 Second order DC OSE performance: response and first order OSE comparison . . . . .	22
20 Signal routing for both states of the second order CC OSE (crossing coupled) . . . . .	23
21 Effect of the coupling coefficient $k$ on the response of the second order CC OSE (default state) . . . . .	24
22 Crossing effect on second order CC OSE frequency response ( $k=0.4$ ) .	26
23 Effect of crossing phase mismatch in the second order CC OSE . . . .	27
24 MRR frequency response phase shift due to the applied bias . . . . .	28
25 Material simulation for the chosen waveguide technology . . . . .	30
26 First order OSE Optsim model . . . . .	31



27	Optsim schematic for both second order OSEs under analysis . . . . .	32
28	Second order CC OSE: PDK implementation of the crossing . . . . .	33
29	Simple perpendicular crossing; Left: geometry, Right: Output power .	34
30	Parabolic intersection crossing; Left: geometry, Right: Output power	35
31	Drop port frequency response: control voltage automatic evaluation .	36
32	Frequency responses for both states of the second order CC OSE (Cross/Bar) . . . . .	37
<b>Beneš Switching Network</b>		<b>38</b>
33	Examples of $8 \times 8$ Banyan-based switches . . . . .	39
34	Examples of blocking in $8 \times 8$ Banyan-based switches . . . . .	40
35	Crossbar switch black-box model . . . . .	41
36	Generic Clos Network with $r \cdot n$ signals and $m$ intermediate switches	42
37	Recursive $8 \times 8$ Beneš generation . . . . .	43
38	Beneš and crossbar switches comparison: number of $2 \times 2$ OSEs required	44
39	Recursive AS-Beneš generation . . . . .	45
40	$5 \times 5$ AS-Beneš: highlight of the recursive blocks for the generation .	46
41	AS-Beneš and Beneš comparison : number of OSEs required for an $N \times N$ implementation . . . . .	47
42	Comparison between BFS and DFS exploration methods . . . . .	49
43	$9 \times 9$ Beneš topological and blocks model . . . . .	50
44	Direct cascade of $2 \times 2$ crossbar switching elements . . . . .	52
45	Permutation vector stages of a $5 \times 5$ Beneš switch . . . . .	54
46	Timing comparison between the proposed network descriptions: time required to evaluate a single configuration of a $N \times N$ device . . . . .	56
47	Number of iterations required to find the unique routings of a $N \times N$ Beneš . . . . .	58
48	$6 \times 6$ Beneš configuration: highlight on redundant elements . . . . .	61
49	Comparison between the number of active switches (cross state) for each routing in the optimized vs unoptimized case . . . . .	62
50	$8 \times 8$ Beneš switch . . . . .	63
51	$8 \times 8$ Beneš routing evaluation: first stage . . . . .	65
52	$8 \times 8$ Beneš routing evaluation: second stage . . . . .	67
53	$8 \times 8$ Beneš routing evaluation: final stage . . . . .	68
54	AS-Beneš routing exceptions . . . . .	69
55	AS-Beneš vs Beneš routing cost: time required to find an available path	71
56	Example of a $6 \times 6$ Beneš schematic automatically generated by the script . . . . .	75
<b>System level simulations</b>		<b>76</b>
57	$4 \times 4$ Beneš configuration . . . . .	76

58	Single port response model - Flat spectral input . . . . .	77
59	Network equivalent model for the two control configurations Single input scenario (Flat spectral signal) . . . . .	78
60	4×4 Beneš output response in passive state - Flat spectral input . . .	79
61	4×4 Beneš output response in active state - Flat spectral input . . .	80
62	Filtered channel simulation environment - Optsim schematic . . . . .	81
63	Filtered channels simulation - Launch power . . . . .	82
64	Network equivalent model for the two control configurations Complete input set (filtered channels) . . . . .	82
65	Comparison between device implementations - Passive state ( $S_{in} =$ $[1\ 2\ 3\ 4] \rightarrow S_{out} = [1\ 2\ 3\ 4]$ ) . . . . .	83
66	Comparison between the three device implementations Active state ( $S_{in} = [1\ 2\ 3\ 4] \rightarrow S_{out} = [3\ 4\ 1\ 2]$ ) . . . . .	85
67	4×4 Beneš configuration: mixed routing ( $V_{bias} = [1\ 0\ 0\ 1\ 0\ 1] \cdot V_{on}$ ) . . .	86
68	Comparison between the three device implementations Mixed routing ( $S_{in} = [1\ 2\ 3\ 4] \rightarrow S_{out} = [2\ 4\ 1\ 3]$ ) . . . . .	87
69	4×4 Beneš configuration: alternative [2 4 1 3] routing paths . . . . .	89
70	Alternative paths comparison: frequency response for the 4×4 second order DC device . . . . .	90
71	Filtered channels simulation - Launch power (reduced bandwidth) . .	91
72	Comparison between the three device implementations Mixed routing ( $[1\ -\ -\ -] \rightarrow [-\ -\ 1\ -]$ ) for reduced bandwidth single input signal	92
73	Comparison between the three device implementations Mixed routing ( $[1\ 2\ 3\ 4] \rightarrow [2\ 4\ 1\ 3]$ ) for reduced bandwidth input signals . . . . .	93
<b>Layout Mask</b>		<b>95</b>
74	PDK-based model - First order OSE . . . . .	96
75	PDK-based model - Second order DC OSE . . . . .	97
76	6×6 Beneš switch - Single ring MRR implementation . . . . .	99

# ABSTRACT

---

The trend in increase in bandwidth consumption, as well as low latency applications, is pushing the network technology in the direction of a completely transparent optical transport layer.

Optical switching elements and topologies are fundamental to enable the routing needed in a flexible, or even software-defined network. Due to their transparency they allow optical routing without requiring a more expensive and power consuming optical-to-electrical conversion of data packets.

The advantages are numerous due to their small footprint, low energy-consumption and latency, as well as silicon photonic compatibility, making this class of component cardinal in the implementation of general purpose photonic transmission networks, as well as data centers applications.

Their implementation as photonic integrated circuits (PICs), compatible with Silicon Photonics processes, is crucial in making this components affordable and practical for implementation in common structures.

The availability of applications for the numerical simulation and the design of PICs has allowed me to analyze, through a bottom-up approach, a class of this switches, namely the Beneš multistage crossover switch, following a generalized method, from physical design of the internal components, up to the production mask, with element-wise and system level performance simulations.

The use of the photonic simulation and design suite (Synopsis, Inc.), in conjunction with ad-hoc external scripts I wrote, allowed the generation of a general developing tool for the Beneš optical switch. These additional tools offer the possibility of developing and designing custom made optical switches under a single unified workflow, allowing the user to customize the internal components of the PIC, as well as the transmission parameters, such as the number of channels, bandwidth, and central frequency.

To assist the user in the evaluation of the performances, I implemented a generalized deterministic algorithm to evaluate routing optimization for any arbitrary sized Beneš networks. This algorithm operates on a mathematical abstraction of the switch, and could be applied as a control driver for the switching element.

The generalized approach and workflow was finally used to compare and test different implementations of the basic switching elements inside the network, showing the effect on the transmission performances, such as bit error rate, inter-channel crosstalk, as well as power efficiency.

In conjunction to the practical results on the performances of this class of switching networks, the generalized bottom-up approach is a testament to the strength of the interaction between PICs simulation suites and external scripts compatibility, allowing for in-depth analysis of different sets of photonic devices.

The developed tools offer a reliable and coherent platform to compare different Beneš optical switches, but acts also as a case-study of a unified design process, from material and technological specifications, up to system levels parameters.

# ACKNOWLEDGMENTS

---

During the work and development of the thesis I had the support and assistance of many wonderful people, under an enriching and stimulating environment.

I want to express my deepest gratitude towards my supervisors, Professor Paolo Bardella and Professor Andrea Carena for the help and guidance offered along the course of the project, with complete availability and kindness, overcoming the difficulties of remote working.

My sincere appreciation goes toward Professor Vittorio Curri, PhD Ihtesham Kahn and PhD Muhammad Umar Masood, for the wonderful opportunity of working in a research-focus and welcoming environment, expanding my knowledge and the applications of my work toward topical and modern subjects.

I would like to thank my family: my mother Marisa, my brother Michele, Piero, Bruno, Marco, as well as all the other members of my family. They provided support and encouragement during my academic experience, as well as guidance over my whole life, especially in the toughest of times. I owe them who I am today and no words can do it justice.

To all my friends, thanks for bringing joy and cheerfulness into my life. Happiness is only real when shared.

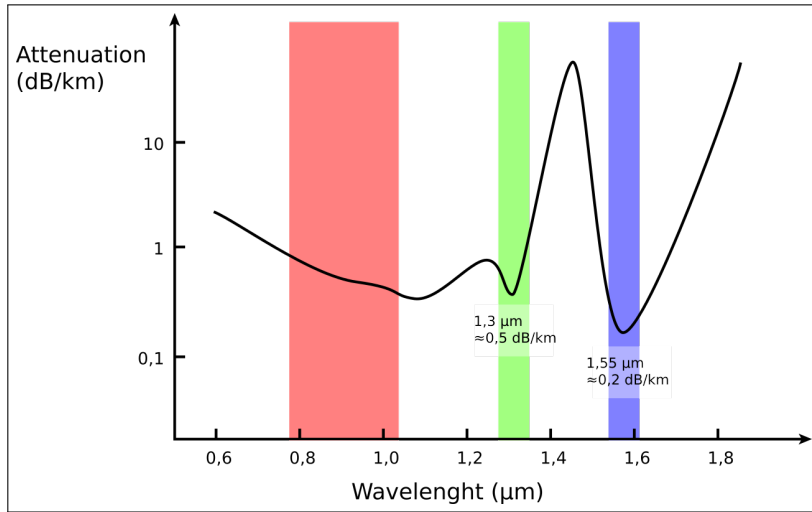
# 1 INTRODUCTION

---

## 1.1 OPTICAL NETWORKS AND FIBER TECHNOLOGY

In today's landscape optical networks are becoming more and more commonplace, in order to accommodate the larger bandwidth consumptions and low latency requirements, under the expansion of the user-base for IP services, as well as the cloud-migration of content providers, both for video streaming and as other entertainment applications.

Optical networks are able to satisfy these requirements due to improved transmission performances of optical fibers with respect to traditional electric cables, making them the backbone technology of this shift in propagation medium. Optical fibers show the lowest attenuation over distance with respect to other forms of propagation, particularly in the third communications window (**Fig. 1**), with an ideal  $\alpha = 0.2\text{dB/km}$ , leading to the possibility of long haul spans of fibers with negligible loss, with respect to the electric counterpart.



**Figure 1:** Attenuation of optical fibers, with highlights of the three main propagation windows

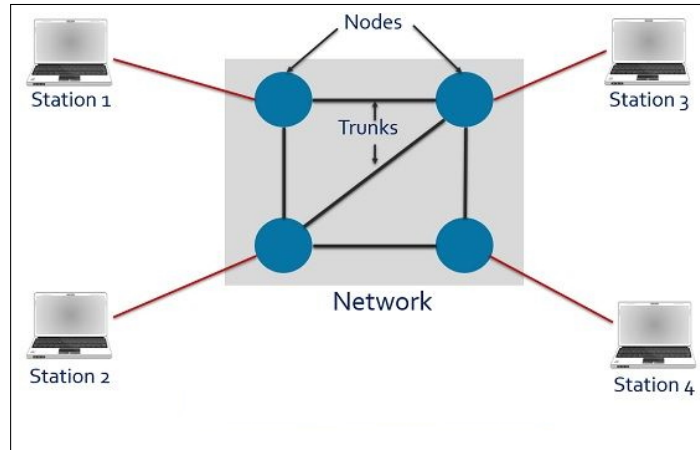
The high carrier frequencies available also lead to the second main advantage of this technology, namely the increased available bandwidth for the channels, allowing for faster transfer rates along the network.

## 1.2 Network elements

The C-band, centered around 1550nm (191 THz-196 THz), is chosen for its compatibility with the Erbium-Doped Fiber Amplifiers (EDFA) technology as well. This class of optical amplifiers are commonly used to regenerate the signal, as they allow transparent amplification, without needing a more taxing O-E-O conversion (Optical to Electrical to Optical).

Amplification and signal regeneration are needed due to the input power limit of the links, which are well below the damaging power levels of the fiber ratings. Due to non-linear effects depending on the intensity of the propagating field, mainly the Kerr effect, propagation parameters of the fiber, such as the refractive index, are modified. This lead to potentially severe non-linear distortions of the signal, losing the transmitted information, and so imposes a constraint on the maximum input power, limiting the maximum length of non-amplified fiber spans.

## 1.2 NETWORK ELEMENTS

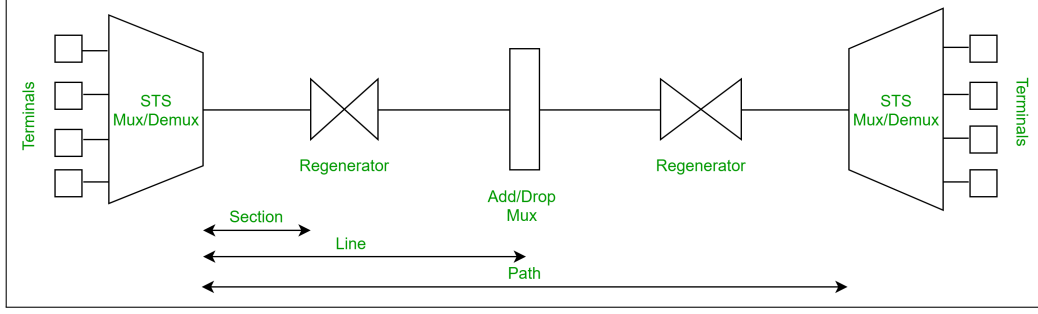


**Figure 2:** Simple topological example of a generic network

From a topological standpoint an optical network is not at all different from the electric counterpart: the connections between terminals and the routing are handled through a conjunction of nodes and physical links, allowing for communication between the end-point stations connected to the structure (**Fig. 2**).

The link between two adjacent nodes can be modelled as shown in **Fig. 3**, through the use of EDFA to regenerate the signal, as previously discussed, as well as using wavelength division multiplexing and demultiplexing (WDM), in order to improve the capabilities of the link.

WDM is one of the most common multiplexing strategies in optical networks, and



**Figure 3:** Block model of an optical link between two nodes

consists of allocating a subsection of the complete bandwidth for each channel, allowing the simultaneous streaming of multiple signals, which can coexist in the same link, and can be separated at the receiver side, to correctly route the data to the specified target.

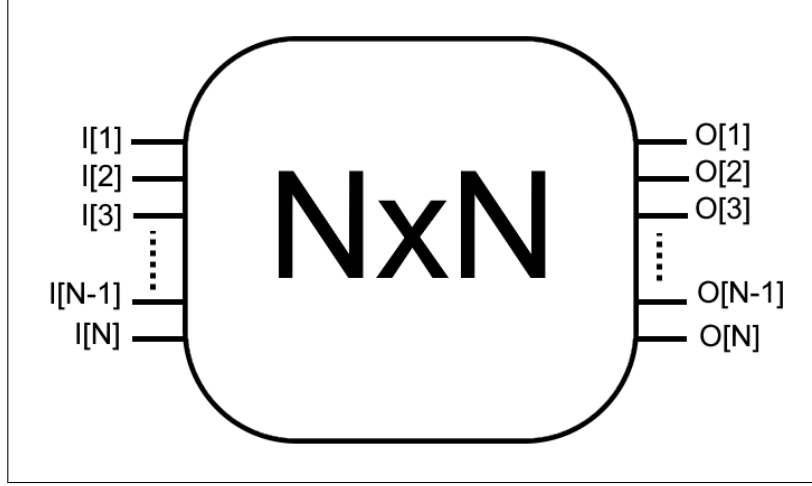
Each signal has a different central frequency, representing the transmission channel, and is separated from the adjacent channels to avoid excessive cross-talk and loss of data due to non-linear interference.

The nodes of the network are tasked with handling the routing of the incoming traffic, and redirecting it without conflict to the next destination node.

This is achievable through the use of optical switches, which enable transparent routings, as well as flexible control of the input-output permutation. Complex software defined networks may require Reconfigurable Optical Add Drop Multiplexers (ROADM) as node elements, enabling the management of each channel independently with respect to the others, while straightforward channel routing may be achieved through simpler and more compact switching devices. These components lack the ability to route multiple inputs to the same output port, showing instead a bijective relationship between inputs and outputs. This types of  $N \times N$  switches can be defined as a black box element with  $N$  input and output ports, as shown in **Fig. 4**. Each input signal can be routed to any unoccupied output port, without signal superposition or demultiplexing features.

For the purpose of this analysis the ROADM and WSS implementations (Wavelength Selectable Switch) are not considered, while using the term "optical switches" in reference to only this  $N \times N$  bijective model.



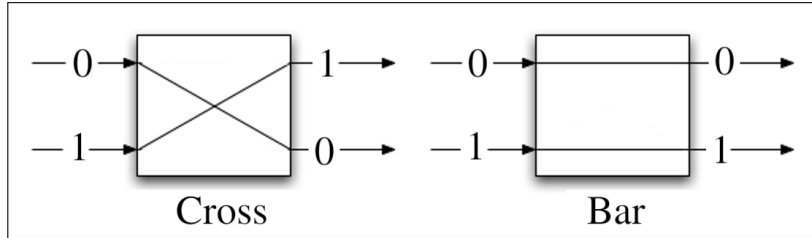


**Figure 4:** Generic representation of an  $N \times N$  switch

### 1.3 OPTICAL SWITCHES

Optical switches can be implemented through a variety of topologies, with different target applications and internal properties. One of the main class of switches is the cross-over switch, or matrix switch, which can be defined as a multistage device made of a set number of basic switching elements: these switching elements are organized in multiple stages with the interconnections defined by the given topological implementation.

The fundamental element is usually defined as a  $2 \times 2$  crossbar switch, shown in **Fig. 5**. This device can operate in two states, set by a control signal  $M$ , with the CROSS state for  $M = 1 \implies \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ , and the BAR state for  $M = 0 \implies \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ .



**Figure 5:** Black-box model for a  $2 \times 2$  crossbar switch

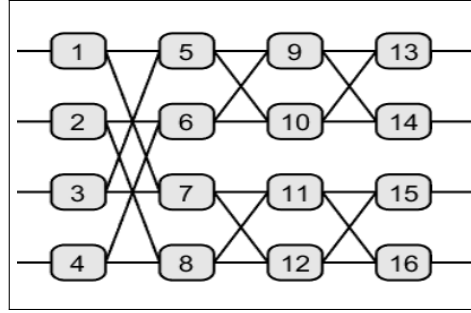
In an optical switch this element is referred as Optical Switching Element (OSE), and the physical implementation will be discussed in **Section 2**.

Once the  $2 \times 2$  element is set, different network topologies can be implemented to generate a  $N \times N$  switch, with varying properties and transmission performances. The characterization of these networks, on a mathematical abstraction level, can be done through the following attributes:

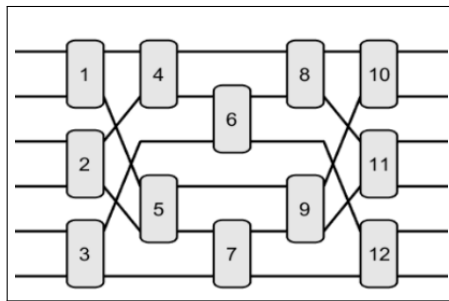
- Scalability: the switch can be generalized for any  $N \times N$  size.
- Non-blocking: all input permutations are routable without any conflict inside the switch.
- Number of OSE: the number of basic  $2 \times 2$  used for a given network size  $N$ .
- Planarity: absence of inter-stage crossing.

The three networks shown in **Fig. 6** offer a demonstration of the variety of attributes of different topologies. The Butterfly network **Fig. 6a** is defined only for size  $N = 2^x$   $x \in \mathbb{N}$ , does not avoid conflict and inter-crossings, as well as having higher number of OSE with respect to the other topologies.

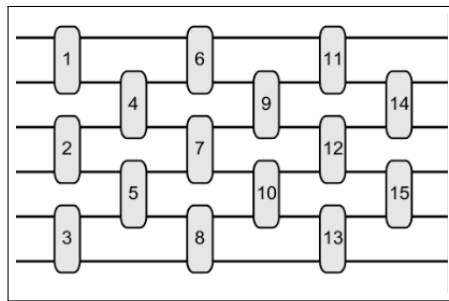
The Spanke-Beneš network (**Fig. 6c**) supports arbitrary size, planarity and is non-blocking, although with a higher number of OSE with respect to the Beneš network, shown in **Fig. 6b**.



(a)  $4 \times 4$  Butterfly network



(b)  $6 \times 6$  Beneš network



(c)  $6 \times 6$  Spanke-Beneš network

**Figure 6:** Different topologies of multistage crossover switching networks

For the following analysis the topology of choice is the Beneš switch, and a more in-depth analysis of its features, as well as its generation is discussed in **Section 3**. It is important to remark that a multitude of alternative topologies exists, with different sets of attributes, although the Beneš network offers a scalable and optimal structure in terms of number of elements, while avoiding routing conflicts.

Higher order switches, based on a different fundamental element with respect to the  $2 \times 2$  crossbar switch, are available, although the device footprint and complexity increase drastically.

### 1.4 GENERAL APPROACH AND GOALS

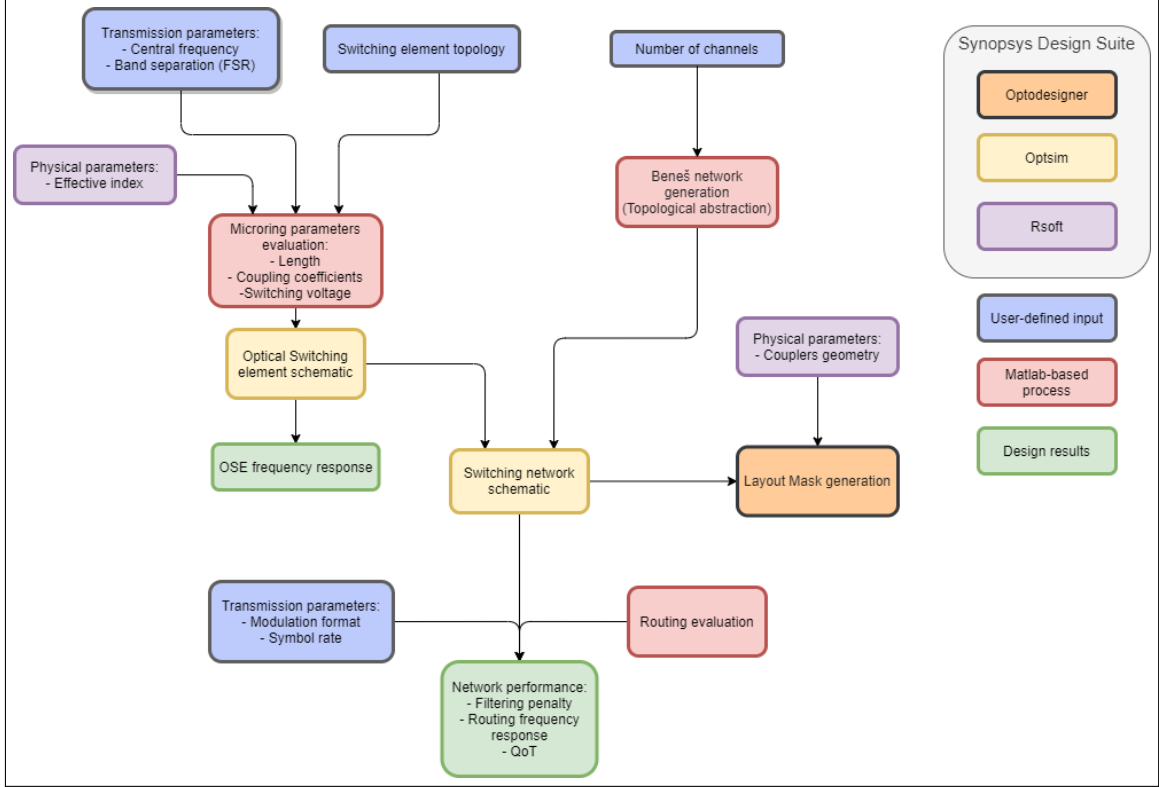
The goal of this work is the creation of a general, user-defined, bottom-up development tool for this class of optical switches. The overall workflow of the project is shown in **Fig. 7**.

The device and transmission simulation process is carried out through the use of the *Synopsys*®[1] photonic design suite, using MATLAB® scripts as coding bindings between the multiple design layers. In each of the following sections the simulations and their respective results are explained in depth. The process can be divided in multiple steps at a different design scope, as well as an increasing abstraction toward system level performances.

The software used in the workflow are the following:

- RSoft®: the simulation is based on integration of Maxwell's equations, in order to obtain accurate information regarding the physical layer, such as the waveguide group and effective index, and the coupling coefficient.
- Optsim®: the simulation is based on a circuit model of the wanted device, based on customizable block implementing the wanted functions. This allows the simulation of a complete structure in order to evaluate Quality of Transmission (QoT).
- Optodesigner®: this tool allows the generation of a GDSII production mask from an Optsim schematic, allowing the final step of the design.
- MATLAB: used for the generation of the schematics, as well as the evaluation of the internal parameters from the user-defined attributes.

## 1.4 General approach and goals



**Figure 7:** Proposed workflow for the design of the Beneš switch

Once the user has selected the main parameters, such as the size  $N$  of the network, as well as waveguide technology and central frequency, the OSE is generated (**Section 2**) and implemented into the chosen Beneš network (**Section 3**). The resulting schematic can successively be used for system level simulations of transmitter/receiver systems, with the evaluation of the routings performances inside the device (**Section 4**).

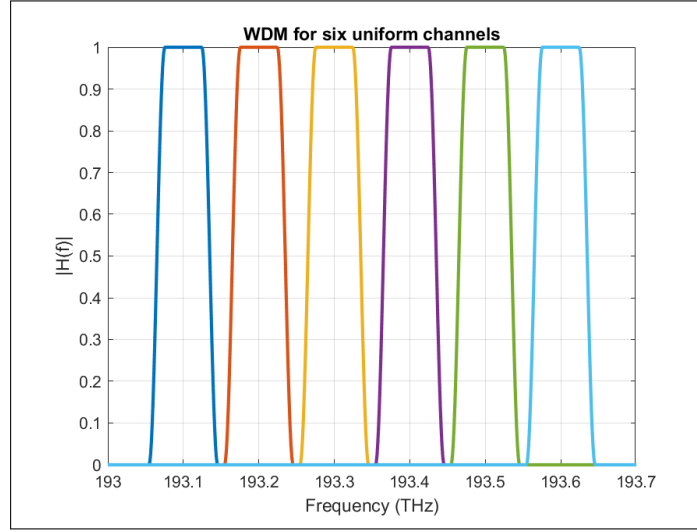
Starting from the Optsim circuit schematic, the production mask is generated, through the use of Optodesigner (**Section 5**), although manual adjustments are needed to comply with production standard, as well as to verify the correct implementation of the device.

Simulations results are available at each step of the design, to asses at a different abstraction level the performance of the implemented models.

## 1.5 TRANSMISSION & SIMULATION PARAMETERS

Although the design parameters can be easily modified through the developed tool, for the results shown in the later sections, standardized values have been chosen, in order to allow clear comparison and results uniformity.

In terms of central frequency of operation the C-band has been chosen (191 THz - 196 THz), with the first channel frequency set as  $f_1 = 193.1$  THz and the free spectral range (FSR) between channels equals to  $FSR = 100$  GHz. An example of the frequency spectrum of the WDM comb is shown in **Fig. 8**, for six uniform channels.



**Figure 8:** Example of a WDM spectral representation

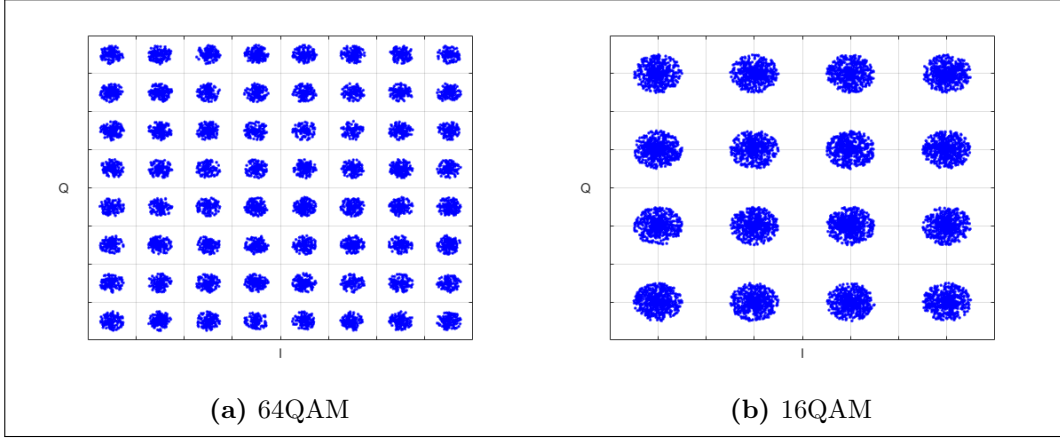
The frequency of the channels and the FSR are enough for the design up to the system level, where different transmission (TX) and receiver (RX) blocks can be implemented.

The system level simulations on the Quality of Transmission (QoT) are based on two main approaches. The first consists of simplified analysis of the frequency response, considering a flat broad-band source as input signal for one of the ports. The second implemented analysis is the evaluation of the transmission, attenuation and side-channel crosstalk when spectrally separated filtered channels are propagated through the structure, similarly to a WDM frequency comb.

Further studies would require the implementation of a coherent TX/RX structure, with Quadrature Amplitude Modulation (QAM), shown in **Fig. 9**. QAM is a standard modulation format in modern communication networks and is widely accepted as a benchmark for the performance of the device. The frequency occupation of a

properly shaped QAM modulation format is compatible with the WDM comb introduced previously, as such the filtered channels simulation can be considered as a simplified precursor to the more complex and time consuming QoT evaluation.

The results and analysis of the transmission penalties, as well as a more in-depth explanation of the possible expansions are explored in **Section 4**.



**Figure 9:** Constellation diagrams for different size QAM formats

The physical parameters and the waveguide technology are evaluated in **Section 2**. These attributes, as well as the central frequency and FSR are customizable through the developed tool. On the contrary, due to the focus on the Beneš topology, the switch class cannot be modified without introducing a new generation or routing algorithm, compatible with other switch implementations.

The general compartmentalization paradigm employed in the study allows modifications of the various layers of the code while maintaining the functionality of the other layers unchanged.

The three main design blocks are:

- OSE: material simulation and generation/simulation of the basic  $2 \times 2$  element.
- Switching network: generation of the  $N \times N$  topology, based on the previously evaluated OSE.
- TX/RX system: evaluation of QoT for the designed switch, using the topology-dependent routing algorithm.

The advantages of the bottom-up approach are clear, as modifications at a lower layer are automatically propagated up the simulation chain, while maintaining the previous steps independent and compatible.

## 2 OPTICAL SWITCHING ELEMENTS

---

The Optical Switching Elements (OSEs) are defined as the fundamental building blocks for any multistage crossover switch. For a Beneš switching circuit this element is the  $2 \times 2$  crossbar switch. In a generic Clos network, of which Beneš networks are a subset, the switch is a  $M \times N$  device with non-blocking properties.

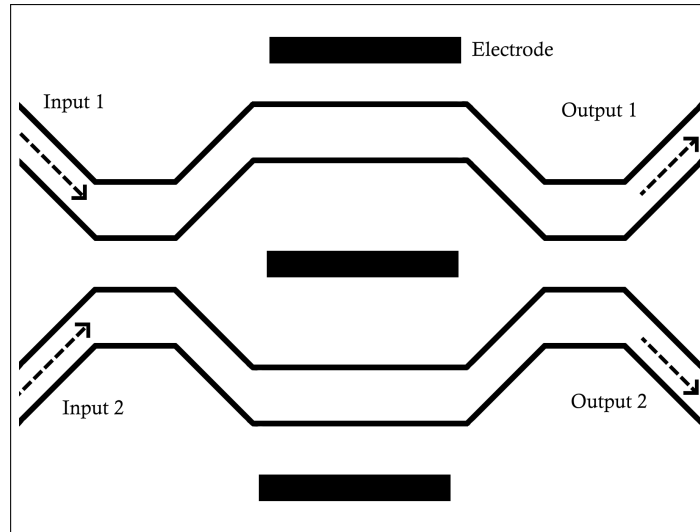
This device is an ideal black box model with the capability of routing each of the  $N$  input signals to a given different  $M$  output port.

In the Beneš network subset, the parameters  $M$  and  $N$  are defined as

$M = N = 2$ , so the OSE becomes a simple  $2 \times 2$  element (**Fig. 5**), with two possible configurations:

- Bar state: the switch acts as a transparent element and the inputs are simply routed to their respective output port
- Cross state: the inputs are routed to the opposite output port ( $1 \rightarrow 0$  &  $0 \rightarrow 1$ )

The state of the switch is defined through an appropriate control signal, depending on the switch physical implementation and topological description.



**Figure 10:**  $2 \times 2$  crossbar switch based on the Mach-Zehnder Interferometer

In an optical switching circuits such devices are built through the use of Micro Ring Resonator filters (MRR) or Mach-Zehnder Interferometers (MZI) [2].

Both devices (**Fig. 10-Fig. 11**) operate through the principle of wave interference, and allow switching between the Bar and Cross configuration, by changing the phase shift of the two propagating arms in the MZI-based approach, or the phase shift of the ring round-trip, in MRR solutions.

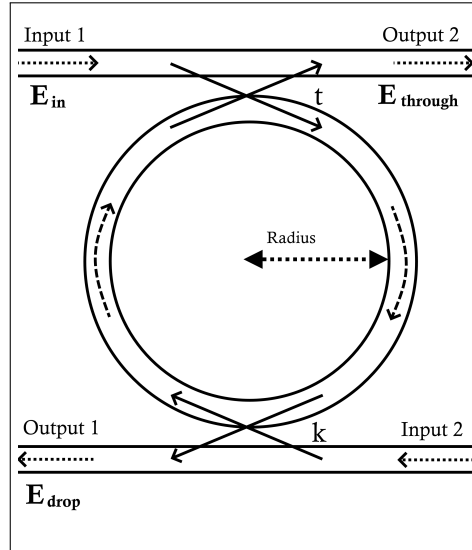
They are suitable for a WSM comb covering a wide range of uniformly spaced frequencies, enabling the design for a specific center channel and FSR, in order to tailor the device to the wanted communication parameters[3].

Due to the MRR allowing the implementation of multiple distinct structures, with different filtering orders, they are chosen as basis for the design procedure, to highlight the performance effect of the different OSEs, as well as showcasing the extensibility of the method to different implementations.

## 2.1 MICRORING RESONATORS FILTERS

MRR devices are based on the principle of constructive interference and optical coupling. The simplest MRR configuration used to implement a  $2 \times 2$  crossbar device is the first order add-drop filter[4], shown in **Fig. 11**.

The radius of the ring is designed in order to obtain resonance at the wanted frequency, which leads to the recombination through constructive interference at the coupled waveguide, allowing the removal of specific wavelengths on the input waveguides.



**Figure 11:** First order MRR-based add-drop optical filter

Given the goal of generating a  $2 \times 2$  crossbar switch, the device must be controllable through an external input, in order to allow the switching between the cross and bar states.



MRR-based devices can be controlled through the heating of the ring component, generated by an electrical current, which has the effect of shifting the resonant frequencies, allowing to switch the states, as well as calibrating the device.

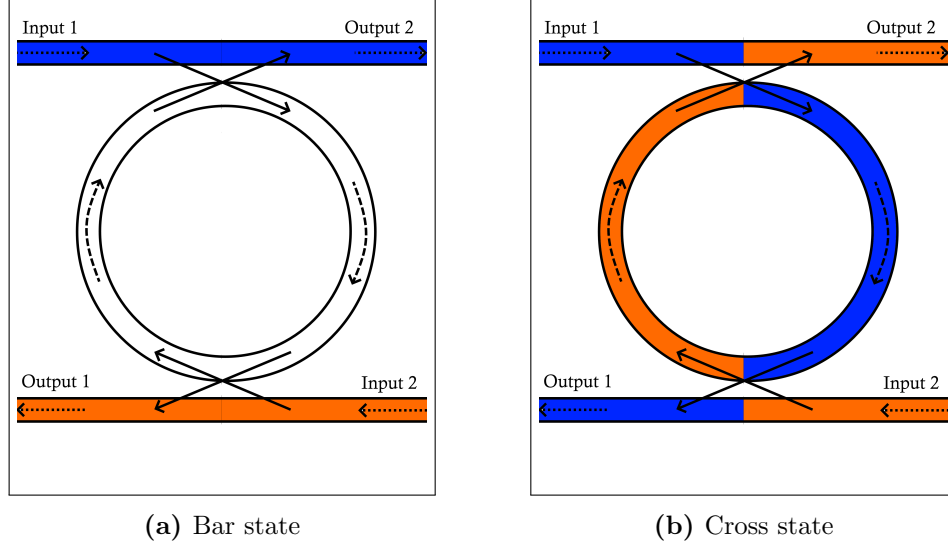
It must be taken into account that due to manufacturing limitations and material impurities, a realistic MRR device might not be centered at the wanted frequency, making the temperature control of the system fundamental for both transmission states.

### 2.1.1 First order filter analysis

The first order MRR filter, as shown in **Fig. 11**, is composed of two parallel waveguides, coupled to a waveguide ring placed in between.

This simple OSE topology has multiple advantages, due to its simplicity and small footprint, although one clear disadvantage, as shown in **Fig. 12**, is the position of the output ports with respect to the black-box model of a generic  $2 \times 2$  switch.

The presence of both input and output ports on the same side of the device leads to more complicated interconnects in the construction of the final  $N \times N$  Beneš structure. Nonetheless, the first order OSE serves as basis for the understanding of the design procedure for a generic MRR based device, so the performance comparisons and considerations will be explored after the introduction of the alternative topologies.



**Figure 12:** Signal routing for both states of the first order OSE

## 2.1 Microring resonators filters

---

The main physical parameters concerning the geometry of the device are:

- $L_{rt}$  is the roundtrip length of the waveguide ring, whose size determines the resonance frequency and FSR.
- $t, k$  which are the coupling coefficients between the waveguide and ring, shaping the filter behavior.

The round-trip is obtained from:

$$L_{rt} = 2\pi r + 2L_c \quad (1)$$

with ring radius  $r$  and coupling length  $L_c$ , considered zero in the ideal analysis. The point-coupling coefficients  $t$  and  $k$  between the waveguides and the ring are considered equal for both of them, due to symmetry, yielding:

- $t' = t''$  and  $k' = k''$
- $|t|^2 + |k|^2 = 1$

The response of the system can be evaluated through:

$$\frac{E_{thr}}{E_{in}} = \frac{-t_1 - t_2^* \sqrt{A} e^{i\phi_{rt}}}{1 - \sqrt{A} t_1^* t_2^* e^{i\phi_{rt}}} \quad (2)$$

$$\frac{E_{drop}}{E_{in}} = \frac{-k_1^* k_2 A^{\frac{1}{4}} e^{i\phi_{rt}}}{1 - \sqrt{A} t_1^* t_2^* e^{i\phi_{rt}}} \quad (3)$$

with round-trip optical phase  $\phi_{rt} = \beta L_{rt}$ , power attenuation  $A = e^{\alpha L_{rt}}$  and propagation constant  $\gamma = \alpha + i\beta$ .

In order to obtain resonance at the wanted frequency, the following relationship must apply:

$$\frac{2\pi}{\lambda} n_{eff} L_{rt} = 2\pi k \quad (4)$$

where  $k$  is an integer value representing the period of adjacent resonant peaks: given the periodic nature of the frequency response of the device, the roundtrip length  $L_{rt}$  must be tailored to enforce the FSR.

For  $k = 1$  the resulting length can be interpreted as the minimum step required to have resonance centered at the design frequency: the final length must be an integer multiple of this step in order to guarantee resonance at  $\lambda_1$ .

This multiple can be evaluated through the use of:

$$k = \text{floor} \left( \frac{2\pi}{\lambda} n_{eff} \frac{L_{rt}}{2\pi} \right) = \text{floor} \left( \frac{n_{eff} L_{rt}}{\lambda} \right) \quad (5)$$

$$\frac{L_{rt}n_{eff}}{\lambda_1 + \delta\lambda} = \frac{L_{rt}n_{eff} + \lambda_1}{\lambda_1} L_{rt} = \frac{\lambda_1^2}{\delta\lambda} \quad (6)$$

Following the design specification, with center frequency  $f_1 = \frac{c}{\lambda} = 193.4$  THz, the length step value ( $k = 1$ ) corresponds to  $L_{step1} \approx 0.619916 \mu\text{m}$ .

The next adjacent channel is centered in  $f_2 = 193.3$  THz, leading to  $L_{step2} \approx 0.620237 \mu\text{m}$ . The mean value between  $L_{step1}$  and  $L_{step2}$  is a critical point: any value  $L = k \cdot L_{step}$  for the median  $L_{step}$  will lead to a device with the correct FSR (if  $k$  is evaluated correctly) but with opposite frequency response.

From a mathematical point of view this simply inverts the behavior of the cross-bar switches, shifting the response by half the FSR, but two main problem should be highlighted.

- The frequency response of the ring is not symmetrical between the pass-band and stop-band: this could lead to sub-optimal performance with respect to the design goal
- Uncertainty in the implementation of the OSEs could lead some element to behave differently and non-uniformly inside the network: in this case the non-blocking property of the network could be negated, and routing conflicts could arise.

As a consequence the design should account for the needed accuracy and compatibility with the technological processes available.

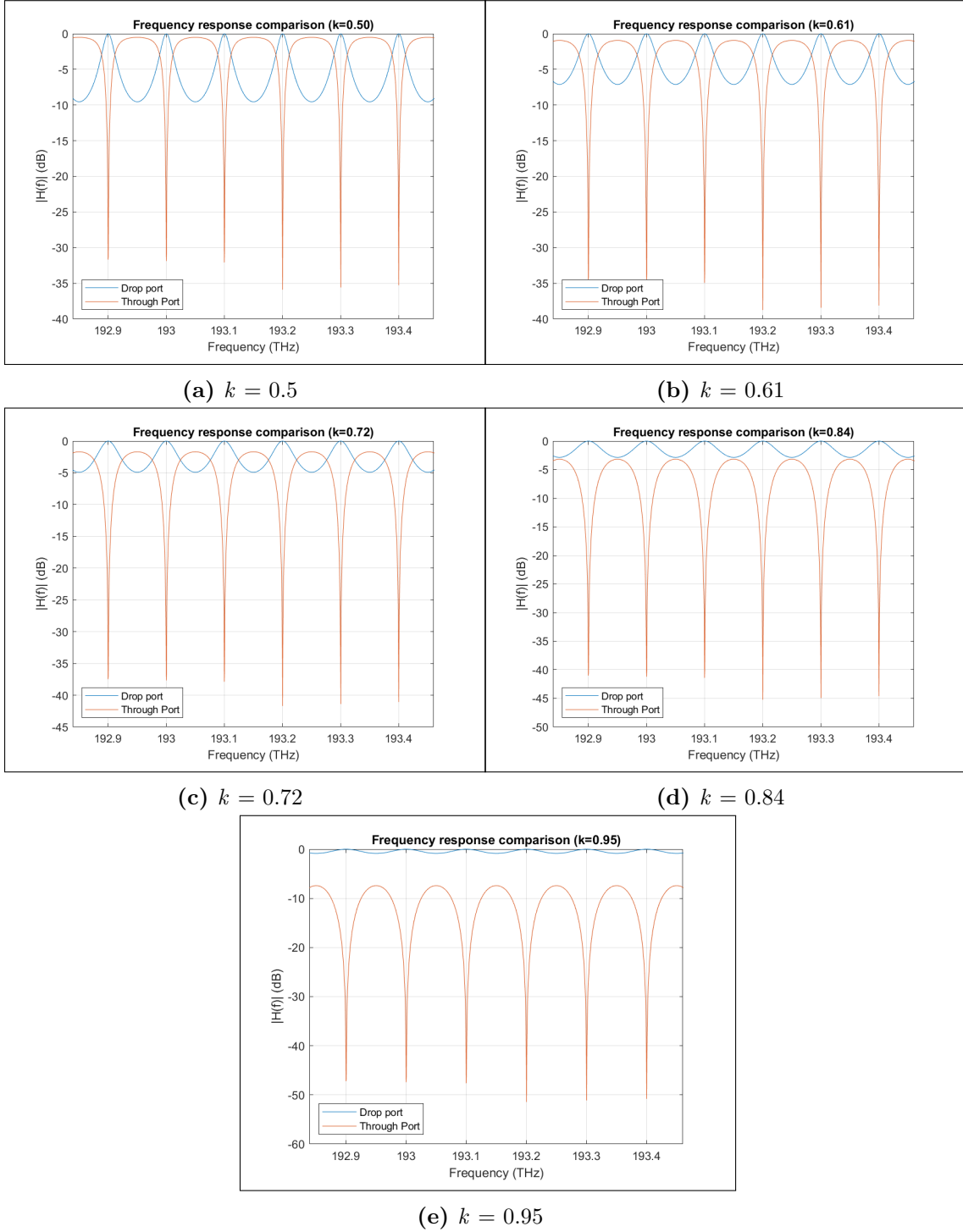
Due to the temperature control applied to the device, the correct design for the FSR can be enough to ensure the correct behavior of the switch. While the OFF state of the device (Bar or Cross depending on the design) should be achieved without any control signal, in case of an erroneous central frequency, the OFF signal can be calibrated to provide the required shift in frequency response.

Having set the radius of the MRR, the second design parameter is the coupling coefficient between the waveguide and the ring element. While the length is derived directly from the frequency of operation and the FSR, the effect of the coupling parameter  $k$  on the frequency response of the device is less straightforward.

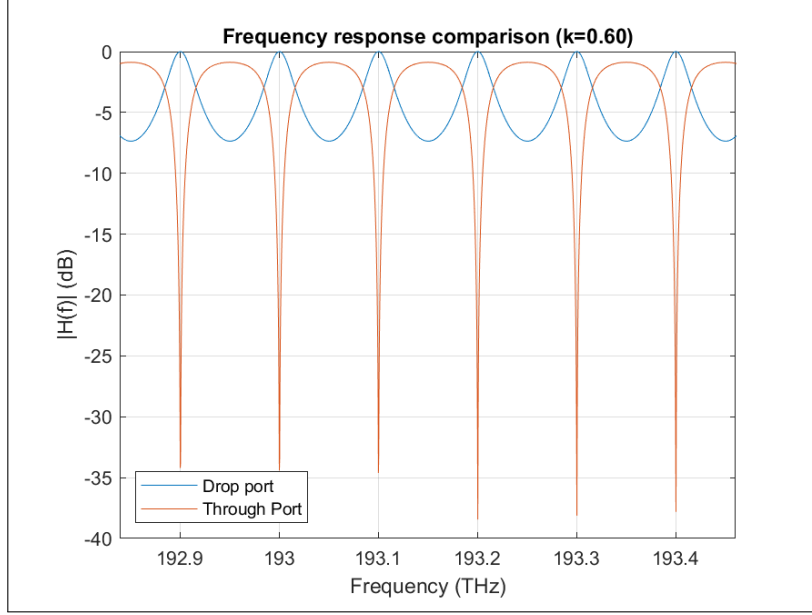
The coupling is responsible for the overall shape of the frequency response, altering the attenuation in the stop-band and the transmission of the pass-band. This phenomenon changes the steepness of the filter, modifying the overall available bandwidth for the signal.

This bandwidth is different from the FSR used in the evaluation of the radius, which defines the periodicity of the response. Depending on the order of the filter and the coupling  $k$  a certain amount of penalty is unavoidable, as the ideal square-wave response is not practically feasible through this device implementation.

## 2.1 Microring resonators filters



**Figure 13:** Effect of the coupling coefficient  $k$  on the response of the first order OSE (default state)



**Figure 14:** Response of the first order OSE chosen as benchmark ( $k = 0.6$ )

The asymmetry leads to a trade-off between the available bandwidth at the drop port with respect to the attenuation of the pass-band at the through port, as shown in **Fig. 13**. The response is evaluated for the off state of the device, which corresponds to a cross configuration, and due to the  $\frac{FSR}{2}$  frequency shift of the responses in the on state, similar consideration can be drawn for the bar configuration.

The importance of the correct design of the coupling section can be seen especially in **Fig. 13d** and **Fig. 13e**: the minimum in the stop-band of the drop port is higher than the transmission peak of the through port. This would drastically impact the performance of the OSE in the bar state, as the input power is equally split between the two port, with dramatic consequences if considered inside a cascade of similar devices.

At the same time the behavior shown in **Fig. 13a** raises concerns regarding the filtering penalty in the cross state: the available bandwidth of each channel is reduced, with severe impairment of the transmission spectral efficiency.

Due to the necessity of distinct power outputs in the passband and stopband, while taking into account the cascading effect in the final switch, the more suitable solution for the coupling coefficient is in the range  $0.55 \leq k \leq 0.7$ .

The value for this configuration is  $k = 0.6$ , with frequency response shown in **Fig. 14**.

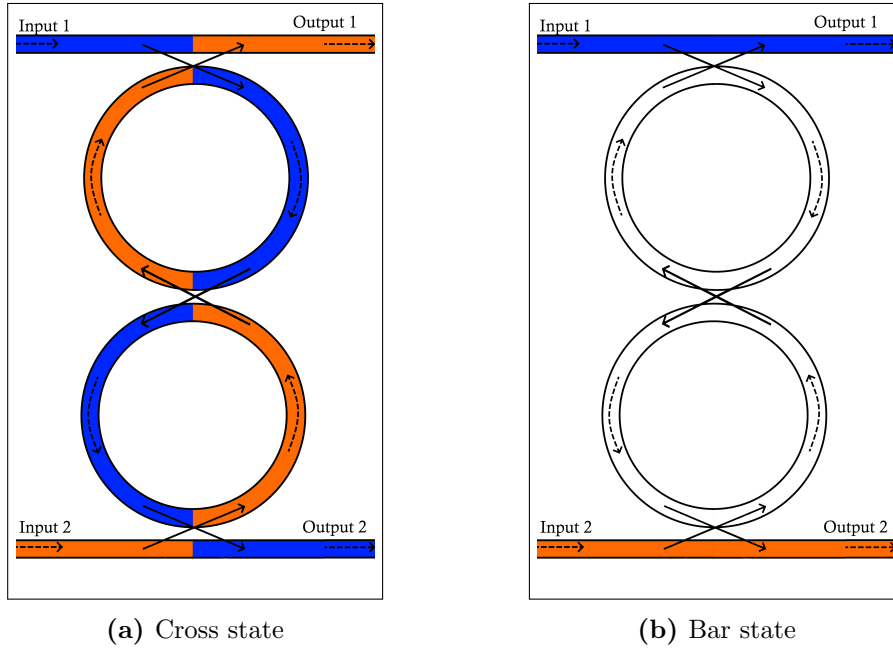
### 2.1.2 Second order filter analysis: directly coupled MMRs

In order to overcome the performance issues of first order filters, higher order filters can be created by including additional MRRs, with multiple available configurations. In this section the analysis of device shown in **Fig. 15** is explored. The OSE is implemented as a second order MMR filter, with direct coupling between the two rings.

The two available state of the OSE are:

- Cross state (**Fig. 15a**): this corresponds to the off state of the device, with the resonance at the design frequency, leading to the routing of the input signals to the drop ports of the device.
- Bar state (**Fig. 15b**): when the device is switched on, the resonance is shifted as to allow propagation of the signals to their respective through ports.

The design of the ring length is the same as the first order filter case, as the resonance of both MMRs must be centered at the same design frequency, but the inter-ring coupling  $k_2$  must be correctly sized to optimize the performances of the OSE. The coupling coefficient between the ring and waveguide  $k_1$  is equal for both the top and bottom ring, in order to maintain symmetry between the frequency response seen by both input ports.



**Figure 15:** Signal routing for both states of the second order DC OSE (direct coupling)

To understand the effect of these two coupling parameters on the behavior of the device, two set of simulations are carried out, fixing one of the variables and evaluating the frequency responses as a function of the coupling coefficient under analysis. In **Fig. 16** the effect of the waveguide-ring coupling coefficient  $k_1$  is shown, with a fixed value of  $k_2 = 0.45$ .

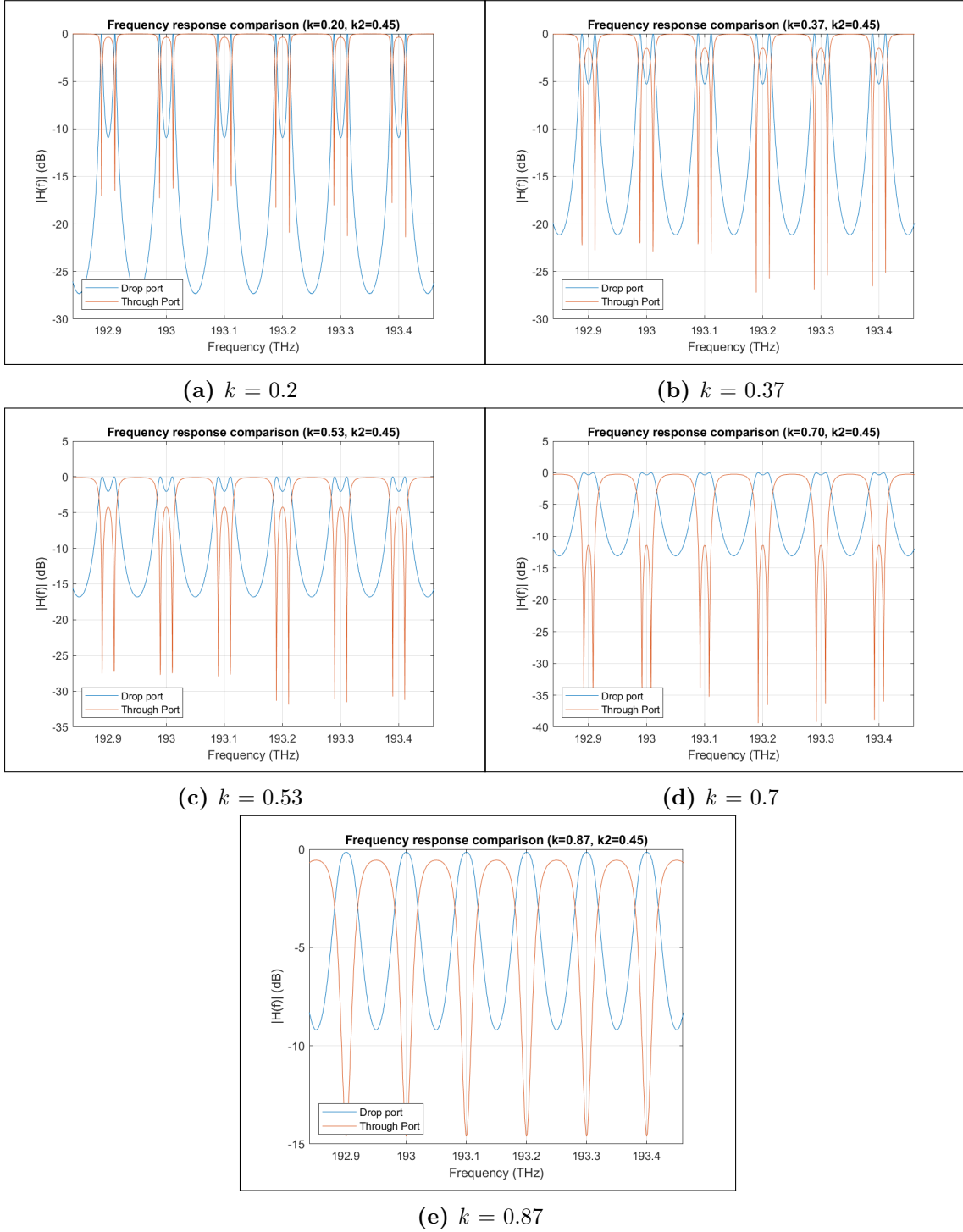
Similarly to the first order OSE, the coupling influences the shape of the overall response, lowering the transmission of the pass-band, as well as increasing the attenuation in the stopband.

Considering the cascading effect of the OSE inside a multistage switch topology, the design should focus on minimizing the losses in the pass-band, as well as providing a reasonable attenuation as to avoid excessive crosstalk: to achieve this goal, the responses shown in **Fig. 16a-Fig. 16c** should be avoided, due to poor attenuation of the stop-band, as well as **Fig. 16e**, which leads to an excessively low transmission of the pass-band.

For the fixed value of  $k_2 = 0.45$ , a reasonable design choice is  $0.6 \leq k_1 \leq 0.85$ .

The filter response as a function of the inter-ring coupling  $k_2$  is instead shown in **Fig. 17**, for a fixed value of  $k_1 = 0.84$ .

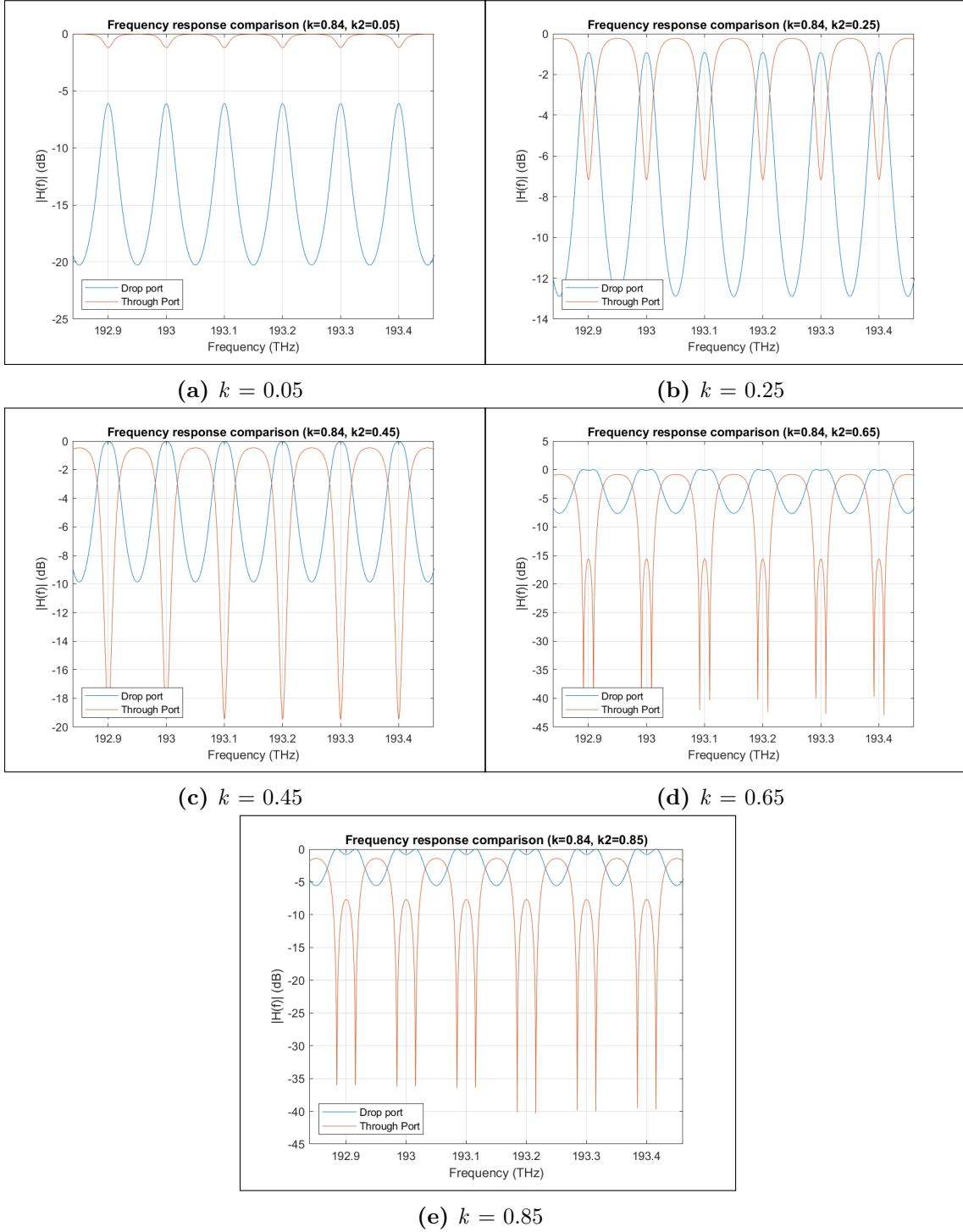
## 2.1 Microring resonators filters



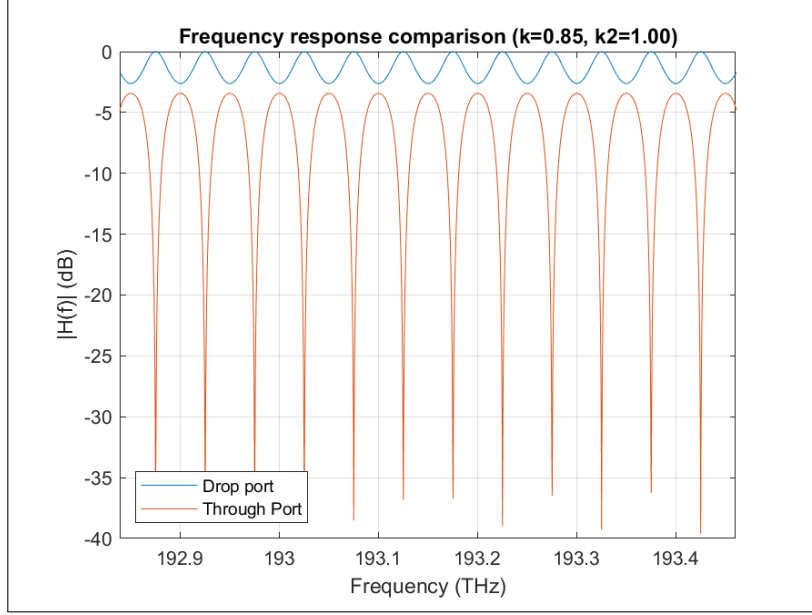
**Figure 16:** Effect of the coupling coefficient  $k_1$  on the response of the second order DC OSE (default state)



## 2.1 Microring resonators filters



**Figure 17:** Effect of the coupling coefficient  $k_2$  on the response of the second order DC OSE (default state)



**Figure 18:** Degenerate second order DC OSE ( $k_2 = 1$ )

The limiting case for  $k_2 = 1$  is useful to understand the peak present in the stop-band: with an ideal coupling between the MRRs, the effective roundtrip length is double with respect to the designed one, leading to a behavior equal to a first order OSE, with halved FSR (**Fig. 18**).

In non-degenerate cases, while an increase in  $k_2$  leads to a more symmetric filtering bandwidth between the through and drop ports, at the same time the shifts in attenuation and transmission peaks lead to a degradation of the performances. Considering the fixed value  $k_1 = 0.84$ , a reasonable design choice is  $0.45 \leq k_1 \leq 0.65$ . The two approximated boundaries for the coupling parameters are useful to highlight an optimal region for  $k_1$ ,  $k_2$ , allowing further simulations to estimate a reasonable value.

The final values chosen for the second order OSE are:  $k_1 = 0.85$   $k_2 = 0.5$ , leading to the frequency response shown in **Fig. 19a**.

In **Fig. 19c** and **Fig. 19b** the performances of the second and first order OSE are compared, represented in linear units, for readability purposes.

Due to the fixed frequency of operation and FSR, the length of the MRRs is equal in both cases, and all differences are due to the effect of the second ring and the modification in the optimal choice of the coupling parameters.

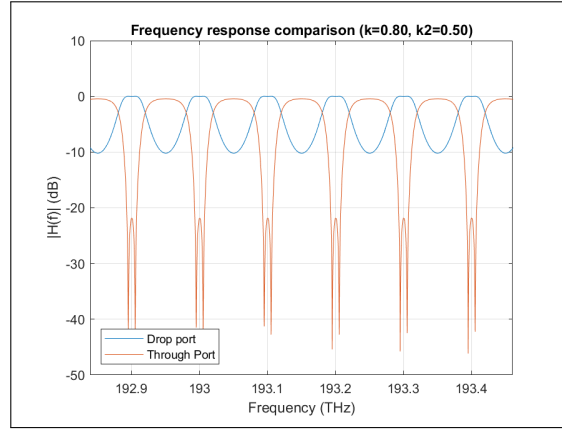
The directly coupled MMR OSE has two main improvement with respect to the single MMR device, namely the increase in transmission and filter bandwidth.

The pass-band of the through port shows a 10% increase in transmission, thanks to

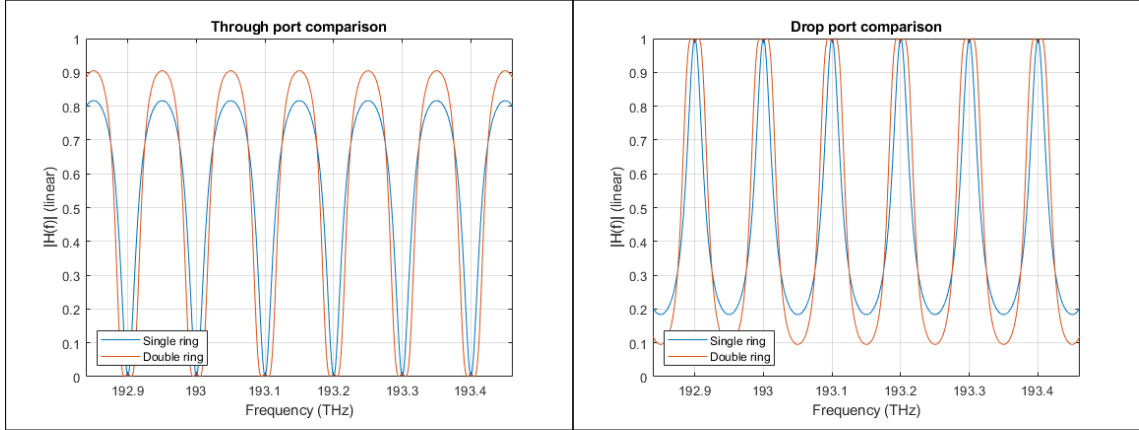
## 2.1 Microring resonators filters

the higher attenuation introduced by the filter: this is especially important in the bar state, when the pass-band of the through port is aligned to the transmission frequencies, leading to a decrease of the signal power loss. Consequently this leads to a reduction of the cross-talk, attenuating the amount of signal power directed to the drop port.

The improvement is not limited to the bar state, but thanks to the widening of the pass-band in the drop port frequency response, the filtering penalty in the cross state is mitigated. Due to the narrow resonant peak of the MRR structure, multiple drop configurations in a cascade of OSE can quickly degrade the QoT, by reducing drastically the bandwidth of the transmitted channel.



(a) Through-Drop frequency response



(b) Through port comparison

(c) Drop port comparison

**Figure 19:** Second order DC OSE performance: response and first order OSE comparison

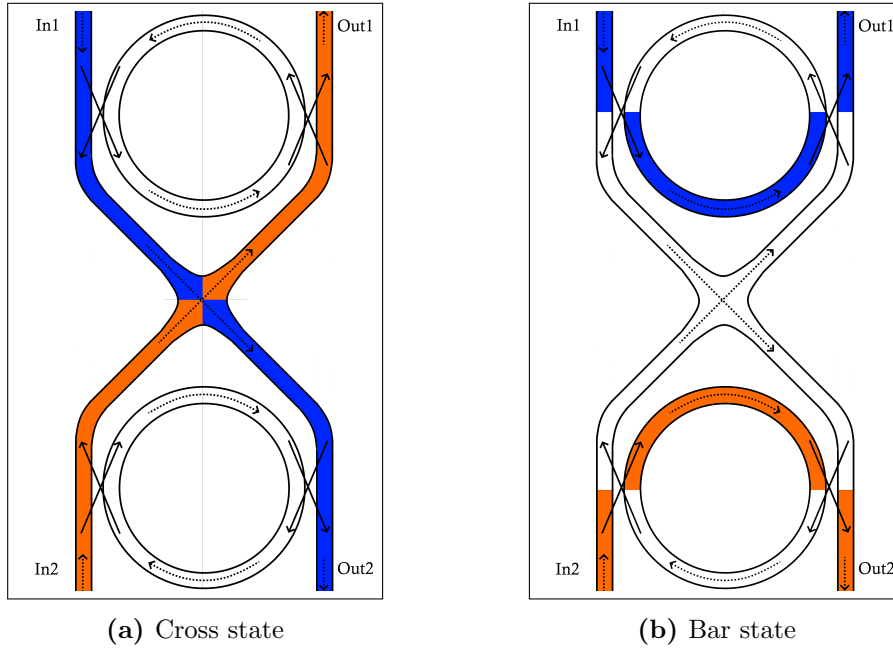
### 2.1.3 Second order filter analysis: crossing coupled MMRs

The second order MRR filter can be implemented without the direct inter-ring coupling seen in the previous section.

The configuration, shown in **Fig. 20**, has each MRR coupled to both waveguides, with an orthogonal crossing between the two rings.

The waveguide-MRR coupling is limited to the parallel waveguides regions, while the crossing is present for two main reasons:

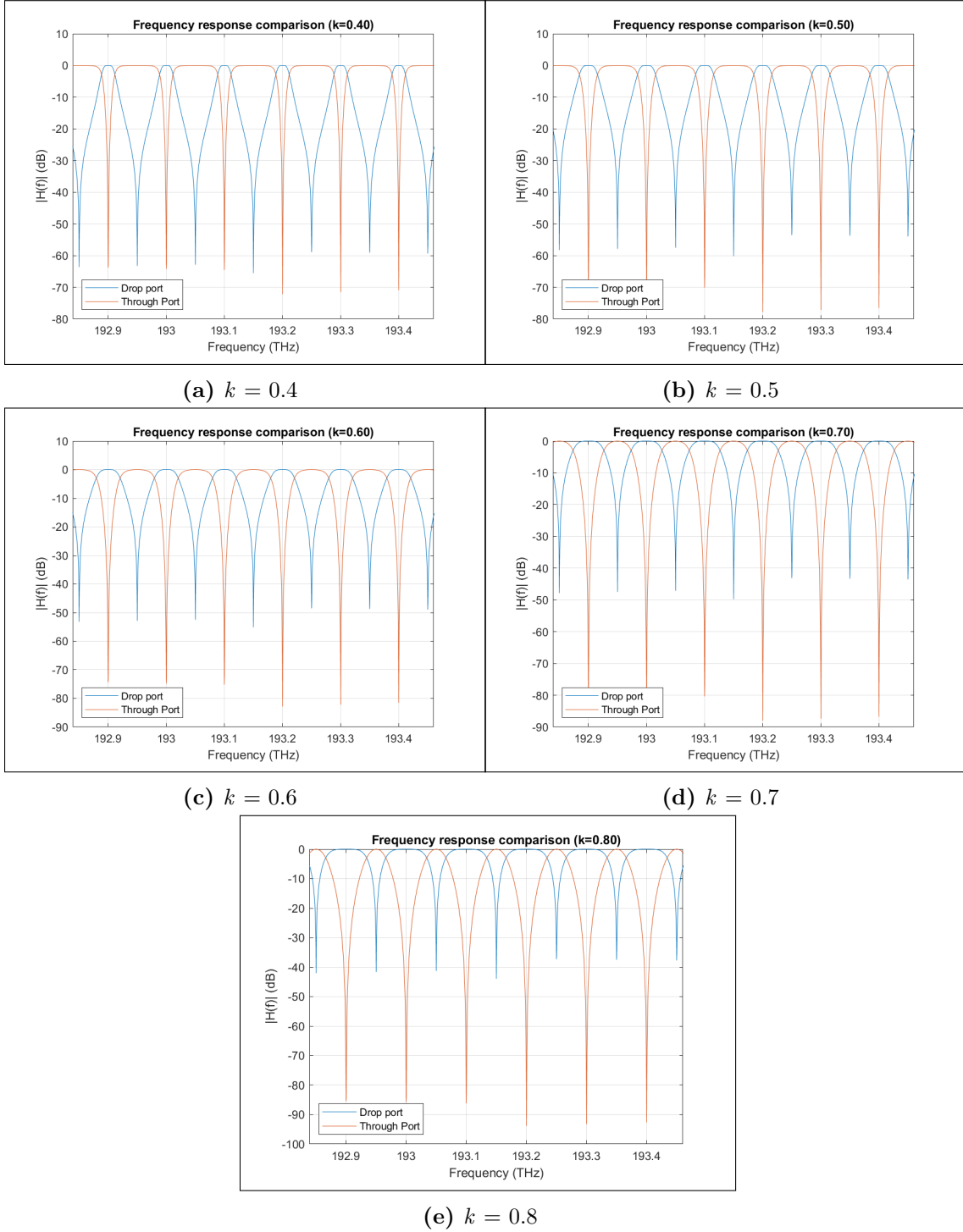
- By overlapping the input waveguides, the output ports can be placed on the same side of the device, enabling the cross state routing (**Fig. 15a**).
- Considering the coupling of the MMRs to the parallel waveguides, the crossing acts as a topological loop, with both ring working in cascade to one another.



**Figure 20:** Signal routing for both states of the second order CC OSE (crossing coupled)

The length of each MMR is evaluated through the same formulas discussed in the analysis of the first order OSE, due to the channel center frequency and FSR being constant. Similarly, the coupling coefficient  $k$  is equal for all four waveguide-ring couplers, in order to guarantee symmetry between the frequency responses seen by both input ports.

## 2.1 Microring resonators filters



**Figure 21:** Effect of the coupling coefficient  $k$  on the response of the second order CC OSE (default state)

The effect of the parameter  $k$  on the frequency response of the device, shown in **Fig. 21**, is quite straightforward. Although there is a variation of the stop-band minimum, the effect is almost negligible, due to the high attenuation provided, and as a direct consequence the pass-band transmission is nearly ideal. The main effect of the increase of the coupling coefficient is the widening of the pass-band at the through-port of the device, leading to a narrowing of the pass-band for the drop-port (**Fig. 21a-Fig. 21e**).

As seen for the previous implementations of MRR OSEs, a reasonable design goal for the transmission of a wavelength comb is the optimization of the parameters such as to obtain a symmetric frequency response at both ports.

The coupling parameter is chosen as  $k = 0.65$ , leading to a highly symmetric response, with negligible channel attenuation.

The design of the directly coupled OSE allows an apparent higher degree of freedom, due to the inter-ring coupling, with respect to the uniform coupling parameter present in the crossing coupled OSE. Up to this point of the analysis the phase shifts introduced by the waveguides outside the rings have not been considered, but in this device structure they become fundamental in order to maximize the performances. The reason for this simplification in the previous sections can be easily justified:

- 1<sup>st</sup> order OSE: the phase shift introduced by the input waveguides doesn't affect the MMR performances, as the signal is recombined after the round-trip, with the designed frequency shift.
- 2<sup>nd</sup> order directly coupled OSE: the same concept is applied for the input waveguides, and for the additional coupling between the MMR is clear that both round-trip lengths are equal, considering the design specifications, leading to a symmetric phase shift for both signal travelling inside the device.

The main difference for the crossing coupled OSE, is that the signal is travelling multiple loops inside the device. **Fig. 15b** can be misleading, as it represent the "logical" path of the bar configuration, while in reality, part of the input signal is propagated through the crossing to the second MMR, due to the non unitary coupling with the first MMR. Similarly, due to the symmetric resonance frequency, the majority of this crosstalk signal will be coupled to the second MMR and back to the second waveguide, traversing the crossing in the opposite diagonal direction, leading the coupling region between the first MMR and the first output port. This generates a loop between the two MMR, but it must be remembered that at each propagation cycle, around the whole device, the crosstalk signal directed to the incorrect output port diminishes drastically.

This global roundtrip is the primary cause for the high transmission efficiency of this device, acting as a cascade of filters for the crosstalk.

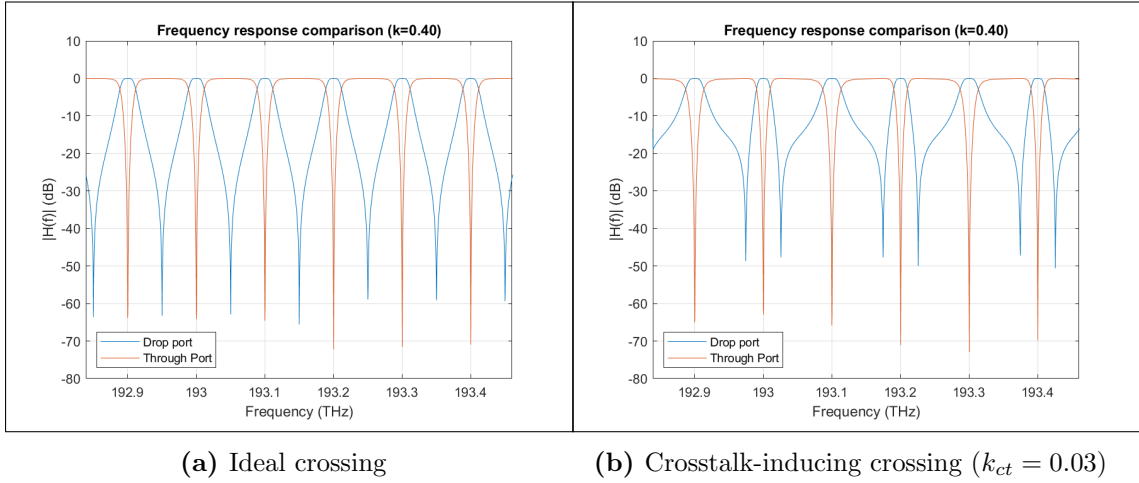
## 2.1 Microring resonators filters

Due to the strong dependence of the OSE performances on the waveguide crossing, two main non-ideal phenomena must be taken into account:

- A realistic waveguide crossing might be affected by a certain amount of crosstalk, leading to propagation of the input power to the incorrect port.
- Phase mismatch between the signal travelling through the two main diagonal pathway may arise from material impurities or length non-uniformity.

The effect of the crosstalk is shown in **Fig. 22**, comparing two OSE with equal parameters: **Fig. 22b** has been obtained by introducing a non-zero crosstalk ratio, meaning that each input of the crossing propagates to the adjacent ports a fraction of the input power, instead of ideally propagating to the opposite diagonal port. The OSE has been designed with  $k = 0.4$  and crosstalk ratio  $k_{ct} = \frac{P_{ct}}{P_{out}} = 0.03$ , in order to better illustrate the effect. The frequency response is severely distorted, doubling the period of the Drop port transfer function.

The resonance frequencies and FSR are still correct, but even and odd channels are filtered with a different response, potentially leading to severe distortion, once propagating through a cascade of OSEs.



**Figure 22:** Crossing effect on second order CC OSE frequency response ( $k=0.4$ )

The second non-ideal phenomenon that can arise in a realistic device is the frequency mismatch between the two branches of the crossing.

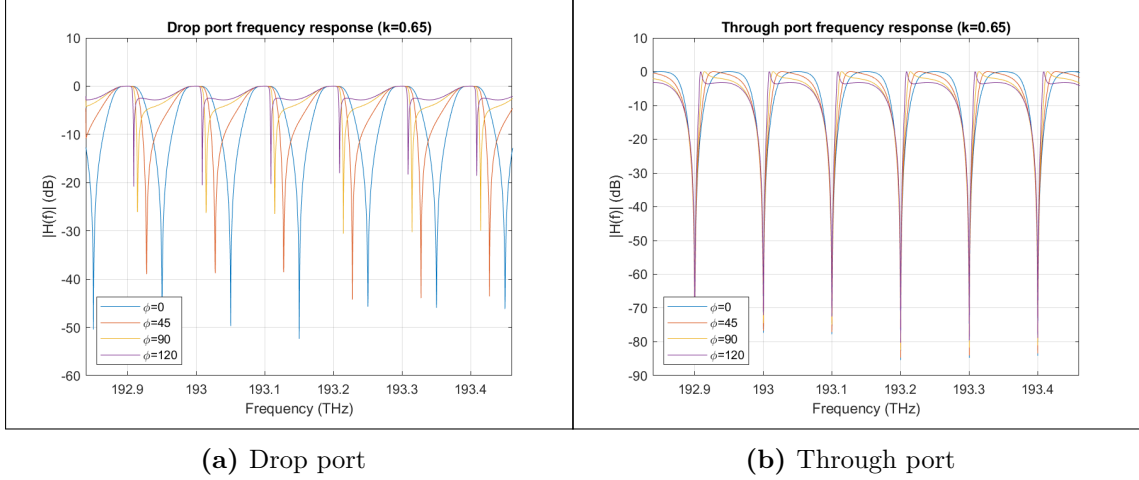
Because the MMR OSEs are based on interference, phase mismatch can have a catastrophic effect on the behavior of the device.

As previously stated, for first order or second order directly coupled OSEs, no phase mismatch can occur between the MMRs, while for the crossing coupled OSE the

## 2.1 Microring resonators filters

MRRs are coupled through additional waveguides segments, possibly resulting in signal mismatch in the overall round-trip through the device.

The phase mismatch has been simulated for a wide range of values, in order to highlight the behavior, and is shown in **Fig. 23** for both output ports of the OSE.



**Figure 23:** Effect of crossing phase mismatch in the second order CC OSE

The performance degradation for the Drop port response (**Fig. 23a**) clearly shows the underlying phenomenon: the resonance is shifted in frequency, as the MMRs output is combined with a mismatched signal, leading to severe attenuation in the design pass-band, as well as an overall asymmetry with respect to the central frequency of the period.

In the Through port response the effect is less apparent (**Fig. 23b**), as the distortion seems less severe, nonetheless for the case  $\delta\phi = 120^\circ$  the inversion of the designed behavior is clear: the unfiltered transmission peak is centered in the designed stop-band, while the whole pass-band is severely attenuated, with respect to the ideal case.

These high values of mismatch are presented to highlight the phenomenon, while a certain amount of mismatch can be handled by the structure ( $\delta\phi < 20^\circ$ ) without severe degradation of the performances.

### 2.1.4 Control signal

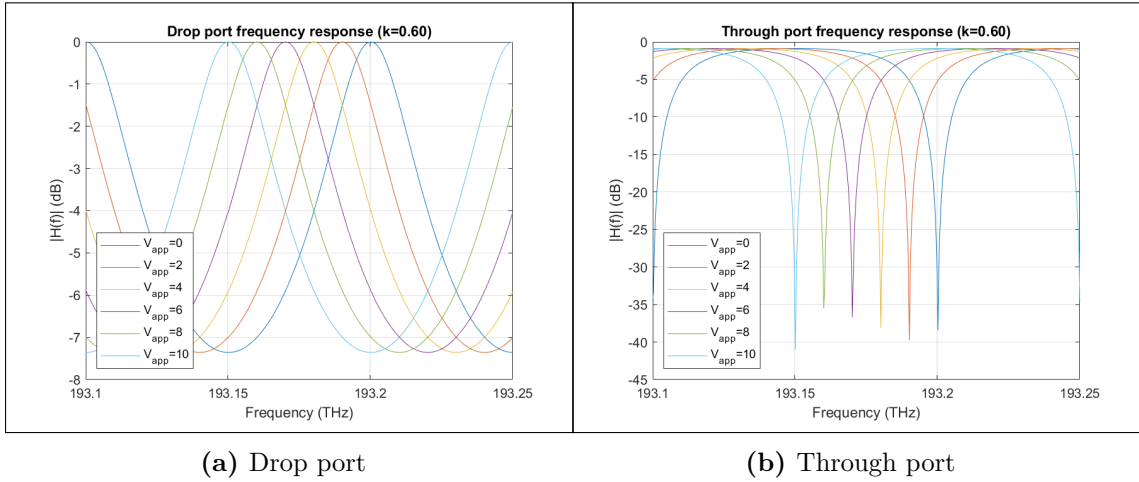
Up to now the analysis of the frequency response has been carried out for the default, or "OFF" state of the OSE. This design strategy is possible due to the symmetry of the transfer functions between the cross and bar states, determined by the physical phenomenon underlying the switching operation.



## 2.1 Microring resonators filters

While the design accounts for both the central resonant frequency, as well as the FSR, the only fixed parameter, once the length is correctly design is the channel spacing: the resonant peaks can be shifted by modulating the phase shift over the ring round-trip. This allows to pilot the device through the temperature control of the ring element.

By applying an electrical current, the MMR is heated, leading to a controlled phase shift of the resonant peaks.



**Figure 24:** MRR frequency response phase shift due to the applied bias

The details of the formulas for the simulation models are explained in **Section 2.2**, depending on the mathematical model behind the phase shift block, while the evaluation of the switching voltage  $V_{on}$  is carried out through script-assisted simulation, explained in **Section 2.3**.

The evaluation of  $V_{on}$  is independent from the OSE topological structure, as the ring element is constant between the three implementation discussed in the previous sections.

The effect of the applied signal can be seen in **Fig. 31**, for an equally spaced voltage span. The scale of the graph has been reduced to a single resonant peak, as to improve the readability: apart from the chromatic dispersion introduced by the waveguide, the simulations shows that no significant distortions are generated, with an almost ideal frequency shift. The evaluation of  $V_{on}$  is carried out under the assumption that, in the considered bandwidth and for the design resonance frequency  $f_r = f_1$ , **Eq. 7** applies:

$$V = V_{on} \implies f_r = f_1 + \frac{FSR}{2} \quad (7)$$

### 2.2 SIMULATION MODELS

The device modelling is carried out through the use of two software in the Synopsys design suite, with a primary circuit-scale approach, as to obtain a simulation model with a reasonable accuracy and computational cost.

The analysis of the waveguide properties and parameters is achieved in the RSoft Photonic Component Design Suite, operating at the component level. The simulation is done by integration of the Maxwell equations, by using a variety of available algorithms, such as beam propagation method (BPM), finite difference time domain (FDTD), eigenmode expansion (EME). This allows a realistic degree of accuracy in the evaluation of the material parameters, granting the selection of the most appropriate simulation method for the designer goal.

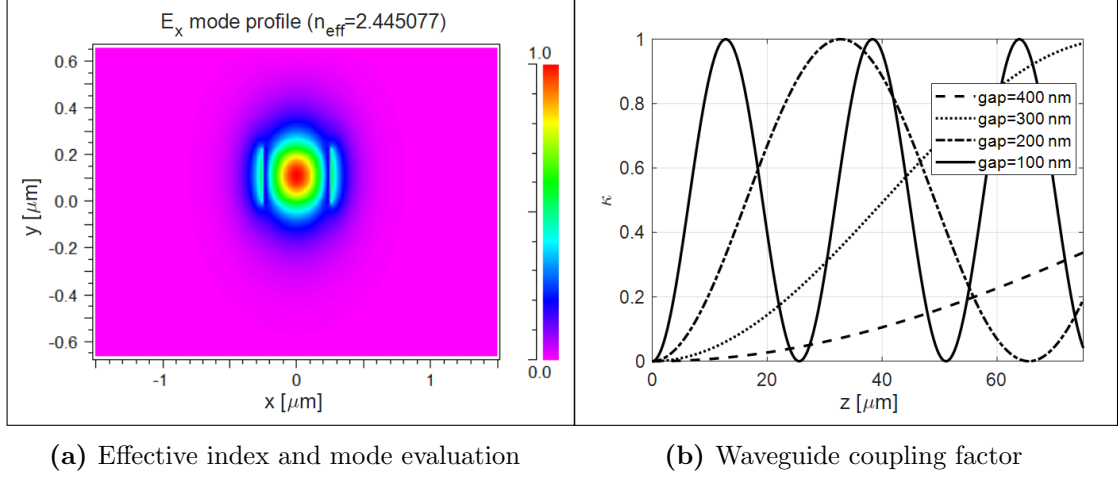
Due to the higher computational cost of the simulations based on the solution of Maxwell's equations, as well as the overall lesser accuracy requirement, at the device level the simulation is achieved in Optsim Circuit.

Optsim Circuit operates on a link and system level, evaluating the given circuit schematic as mathematically-modeled block elements, leading to a faster and more flexible parametrization and analysis of the device attributes.

#### 2.2.1 RSoft waveguide technologies

The starting simulations, carried out in the RSoft environment, are targeted at evaluating the chosen waveguide technology parameters, which are used in the later stages of the design.

Most of the elements available in Optsim account for the material effect on the block by using a model based on the effective refractive index  $n_{eff}$  of the physical waveguide. In order to extract this value a segment of the waveguide can be simulated in RSoft, evaluating the fundamental mode profile, as well as its effective index (**Fig. 25a**). This value can then be inserted in the MATLAB binding scripts, as to set the Optsim variable in all of the simulations shown in the following sections.



**Figure 25:** Material simulation for the chosen waveguide technology

Another important parameter that must be calculated is the coupling coefficient  $k$ , as a function of the coupling length  $L_k$  and waveguides gap  $W_k$  (**Fig. 25b**). While these geometrical parameters can be ignored in the Optsim simulation, due to the availability of a simpler mathematical model, they impose a constraint of the physical implementation of the OSE, which must be taken into account during the design and verification of the production-ready layout mask.

Through RSoft the simulation of the complete OSE can be carried out, although for the scope of this analysis, the faster and more easily characterized Optsim simulation is preferred.

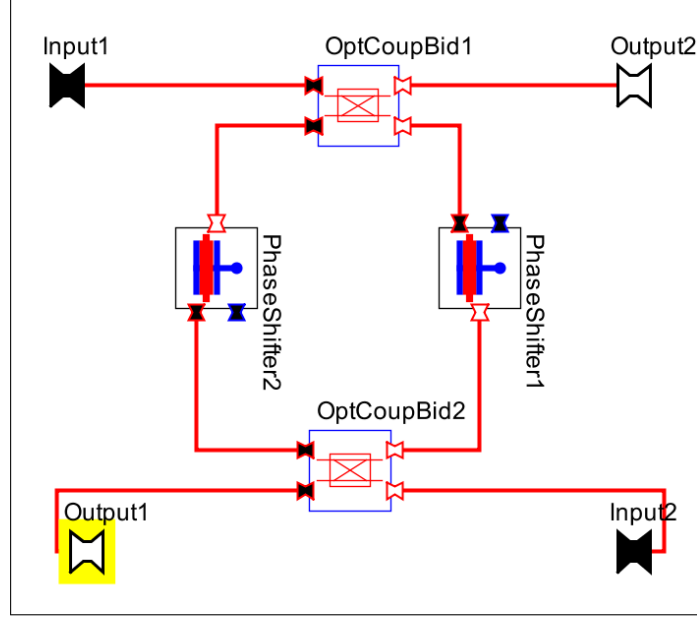
Due to intrinsic length and computational cost of the physical simulations, the rest of the analysis will be carried out in Optsim, regarding the photonic circuit, and MATLAB, concerning the topological evaluation of the device.

### 2.2.2 OSE modelling

Optsim allows the construction of complex circuits through a block-like description of their elements, with virtual links used as connection. These blocks can be selected from the components library and their parameters customized to tailor them to the desired specifications.

The inter-components links acts as logical connections, without introducing any modification on the propagating signal.

The equivalent circuit for the first order OSE is shown in **Fig. 26**.



**Figure 26:** First order OSE Optsim model

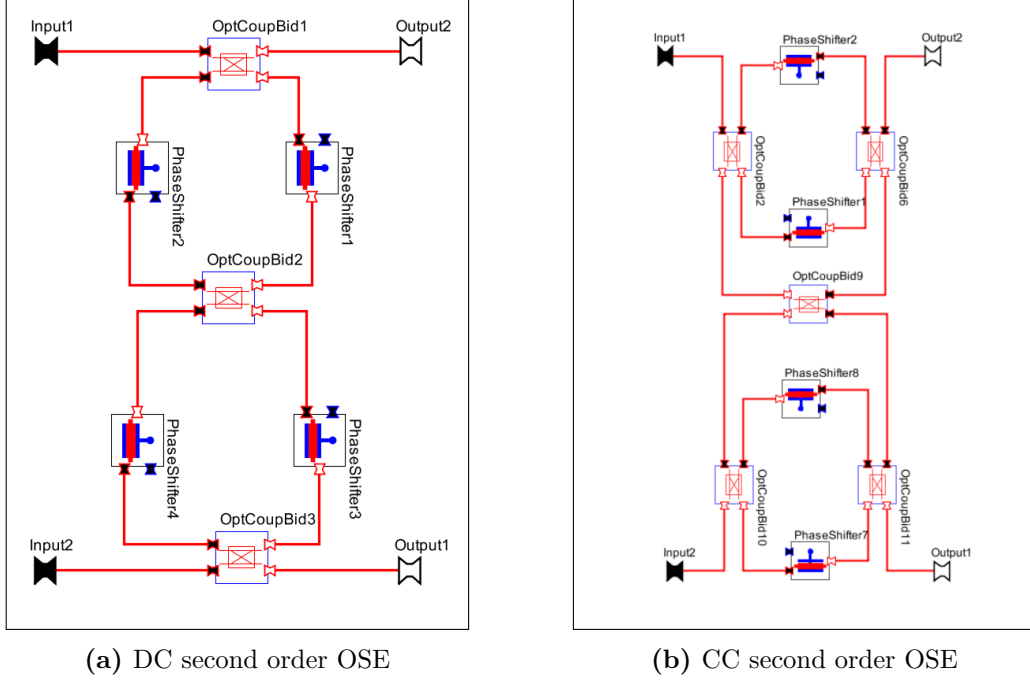
The basic elements used in the modelling of the device are:

- Bidirectional couplers: these blocks represent the coupling regions between the waveguides, and are customized with the previously evaluated coupling coefficients  $k_1$ .
- Voltage-controlled phase shifters: these components are used to set the designed length  $L_{rt}$ , as well as acting as a model for the phase shift introduced by the increase in temperature of the ring, due to the electrical control signal applied.

This basic model for the first order MMR filter is used for the construction of both second order devices, with slight modification to account for the higher degree of freedom.

For the directly coupled OSE (**Fig. 27a**) the inter-ring coupler is modified in order to account for the parameter  $k_2$ , while the the standard waveguide-MRR coupling is defined by the parameter  $k_1$ .

The crossing coupled OSE, shown in **Fig. 27b**, uses an additional bidirectional coupler as a ideal model for the orthogonal waveguide crossing. While in the ideal case the coupling factor is unitary, leading to a uniquely diagonal power transfer, the crosstalk can be manually tuned as well as the phase of each output, by modifying the complex matrix describing the power flow to each port.



**Figure 27:** Optisim schematic for both second order OSEs under analysis

Due to the coupling regions being implemented as ideal zero length point-couplers, the phase shifters length must account for the designed resonance length  $L_{rt}$ . These components also model the frequency response shift due to the heating of the MRR waveguides, through a voltage-dependent equation. The output signal  $O(t)$  is evaluated from the input signal  $I(t)$  through **Eq. 8**, with the following parameters:

- The waveguide loss is expressed through  $a_i(t)$ , defined in **Eq. 9**, and takes into account the influence of the driving voltage on the signal attenuation.
- The voltage dependent phase shift  $\phi(t, \omega_0)$  is described as the linear function in **Eq. 10**, with the applied voltage  $V(t)$ , the waveguide length  $L$  and the scaling factor  $V_\pi L$  used to define the needed voltage to introduce a phase shift  $\Delta\phi = \pi$

$$O(t) \cong a_i(t)e^{j\phi(t, \omega_0)}I(t - \tau) \quad (8)$$

$$a_i(t) = \alpha(V(t))L \quad (9)$$

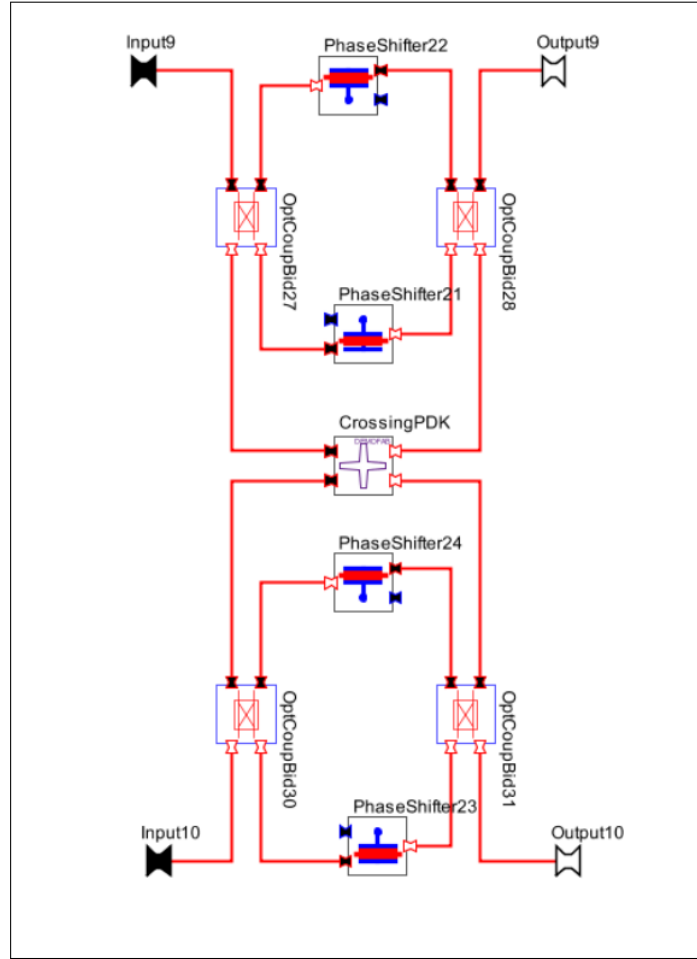
$$\phi(t, \omega_0) = \pi \frac{V(t)}{V_\pi L(V(t))}L \quad (10)$$

## 2.2 Simulation models

It must be remarked that for the purpose of this analysis the component have been considered as lossless, in order to focus on the filtering effect of the OSE implementations.

The results displayed in the previous sections are obtained under this assumption, due to the negligible effect of waveguides attenuation with respect to the filtering effects of the device.

### 2.2.3 Crossing



**Figure 28:** Second order CC OSE: PDK implementation of the crossing

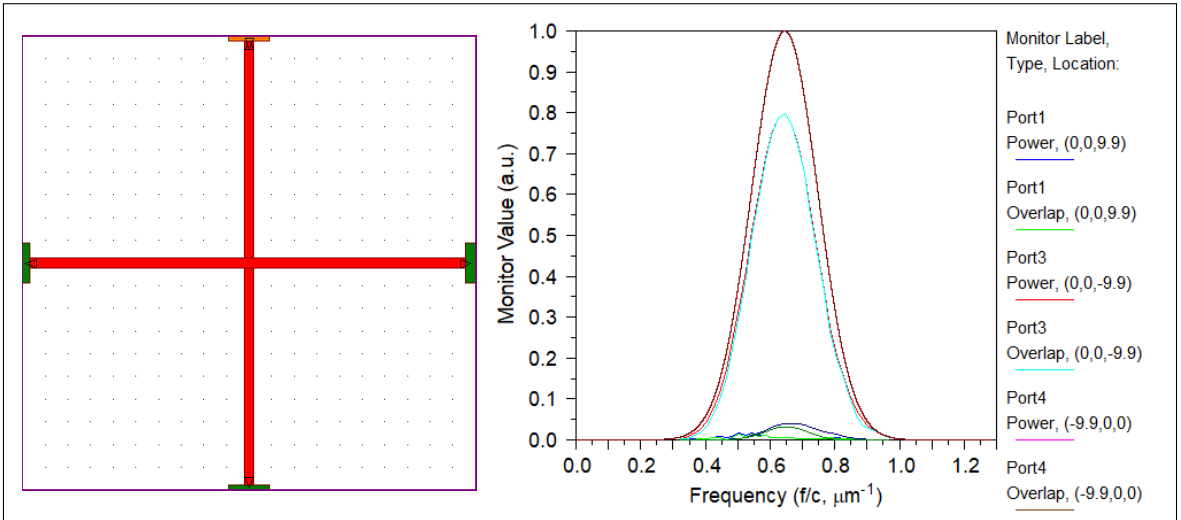
The perpendicular crossing structure is the critical component for the optimized design of the second order OSE, and as such, additional simulations have been run to obtain a deeper insight on the component behavior.

The crossing implemented in **Fig. 27b** act as an ideal mathematical block, allowing full customization of the internal parameter, but a comparison with realistic physical structure is useful to grasp the realistic response of the element.

Optsim allows the implementation of custom-elements (**Fig. 28**), generated through the RSoft Photonic Component Design Suite, in order to test more complex and realistic physical devices while maintaining the block-oriented abstraction, fundamental for a faster simulation of multi-elements circuits.

RSoft has been used to build a crossing following the waveguide specification for the central frequency of operation, with silicon ( $Si$ ) on silicon dioxide ( $SiO_2$ ) as reference materials.

The first crossing model under analysis consists of a simple uniform waveguide crossing, seen in **Fig. 29**. The performance of this model can be taken as the lower bound benchmark of the geometrical optimization that can be applied to this class of structure.

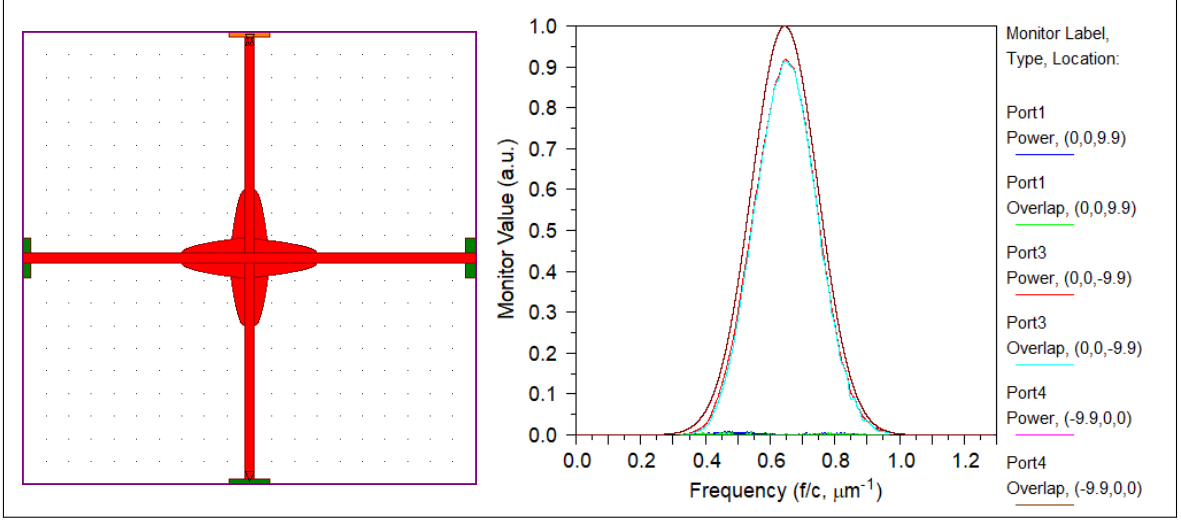


**Figure 29:** Simple perpendicular crossing; Left: geometry, Right: Output power

Due to the symmetry of the structure, the evaluation of a single input launch is sufficient to determine the component behavior.

The main disadvantage of the simple crossing with two orthogonal waveguide is the power loss at the transfer port, as well as a non-negligible amount of crosstalk power, whose distortion effects have been analyzed in the previous sections (**Fig. 22b**).

An improvement on the performances is obtained by introducing the parabolic structures shown in **Fig. 30**.



**Figure 30:** Parabolic intersection crossing; Left: geometry, Right: Output power

The power transfer at the transmission port is improved from  $\eta_{tr} = 0.78$  to  $\eta_{tr} = 0.93$  with  $\eta = \frac{P_O}{P_I}$ .

This results is important as it's clear that further optimization could be employed to obtain an almost ideal transmission, nonetheless the fundamental results for the correct behavior of the device is the almost complete absence of crosstalk. The parabolic regions have two main design parameters that can be tuned, namely the final width  $W_t$  and the overall length of taper  $L_t$ . These values were chosen as to minimize the crosstalk, and were selected after running a scan simulation: RSoft allows a parametric scan of a design variable in a user-defined range, which was used to pinpoint a reasonable geometric size for the parabolic waveguides.

The obtained results show that the device performance are strongly dependent on the width of the taper, while a looser constraint is applied to the length. The final result obtained for this specific structure is  $W_t = 3.2 \cdot W_{wg}$  and  $L_t = 2.5 \cdot W_{wg}$ , with waveguide width  $W_{wg}$ .

## 2.3 MATLAB SCRIPTS

The design process for the OSE is partially automated, as to allow a faster evaluation of the alternative structures and parameters combination.

The initial generation of the topological description of the switch, the placement of the couplers and phase shifters, must be done manually, but due to the Synopsys suite compatibility with custom MATLAB scripts, the evaluation and representation of the results, as well as the design of some attributes can be done automatically.



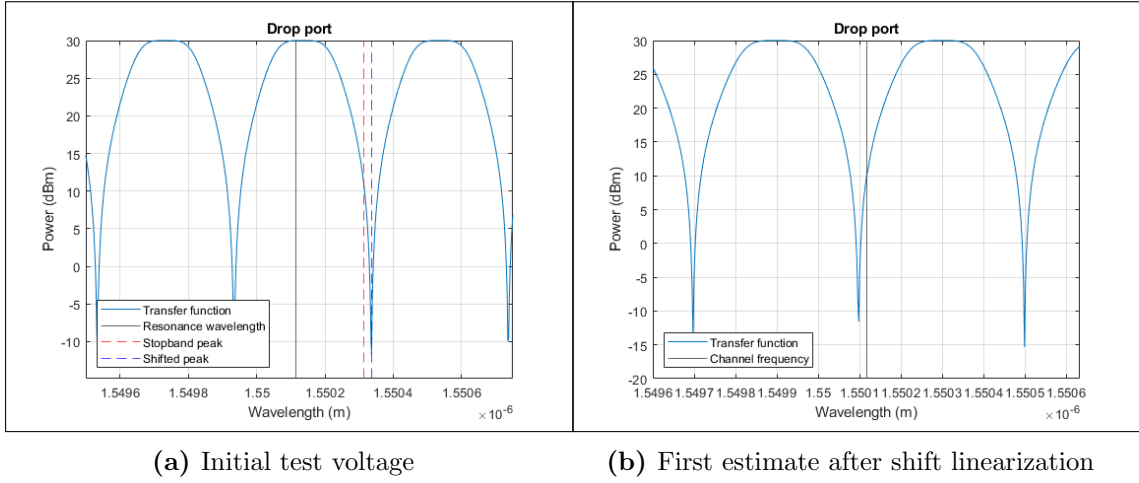
The material and transmission parameters, such as the waveguide effective index  $n_{eff}$  and coupling coefficient  $k$ , as well as the central frequency and FSR must be provided manually, as they depend on the technological implementation, or they cannot be easily optimized mathematically.

The coupling coefficients are selected manually, due to the intrinsic trade-off derived from the choice, as a topologically agnostic optimization algorithm is not feasible. The roundtrip length  $L_{rt}$  of the MRR is evaluated as explained in the first order OSE analysis, due to the availability of a physical formula, while the derivation of the switching voltage  $V_{on}$  is less straightforward.

The algorithm implemented for the evaluation of the control signal is based on the linear shift in frequency of the port responses, shown in **Fig. 31**.

The MATLAB script launches a simulation with zero applied bias, in order to obtain the off-state transfer function of the OSE under analysis. This starting simulation is used to evaluate the minimum peak of the drop port, as a reference for the following step.

After the off state simulation, a small test voltage is applied to the model, and the frequency response is evaluated in order to estimate the frequency shift as a function of the applied signal.



**Figure 31:** Drop port frequency response: control voltage automatic evaluation

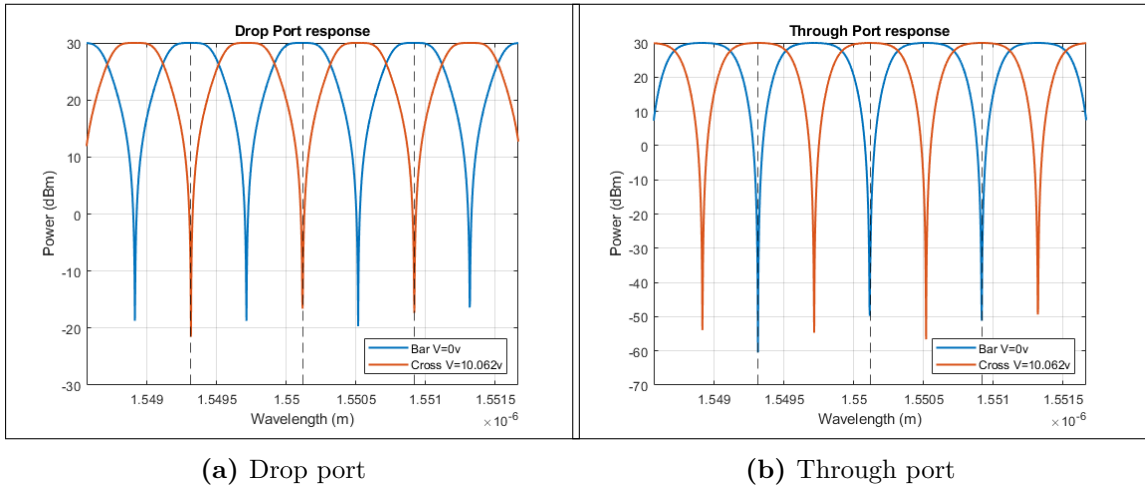
The script then evaluates, under a linear shift assumption, the bias needed to align the minimum peak with the design frequency, obtaining a rough estimate for the switching voltage (**Fig. 31b**).

After these two initial steps, through the use of an iterative procedure, the correct voltage is found, with the degree of precision required. This final search implement an halving step descent following the Newton's method: once in the neighborhood

of the target minimum, the bias is increased or decreased by a step function, which is halved each time the derivative of the frequency response changes sign.

The final effect and alignment of the spectral responses are shown in **Fig. 32**: under the assumed model, no significant difference is present in the Cross or Bar frequency responses, as the frequency shift is ideal apart for negligible discrepancies in the attenuation peaks.

The voltage is topologically independent of the OSE implementation, as the shift must be applied equally to each ring, leading to a dependency exclusively on the resonance peaks, which is defined by the central frequencies and FSR.



**Figure 32:** Frequency responses for both states of the second order CC OSE (Cross/Bar)

All the previously described parameters are parsed by the MATLAB script to the Optsim environment, due to the parametric description used in the creation of the OSE templates. The external script modifies the *.moml* Optsim schematic by substituting the manually selected, or automatically generated, values into the placeholder parameters used in the blank schematic.

This allows for future customization and user-based expansion of the studied templates, as to maintain the general paradigm of this study.

## 3 BENEŠ SWITCHING NETWORK

---

The OSE evaluated in the previous section serves as a basis, or fundamental element, for the design of a generic  $N \times N$  switch.

These switches can be constructed based on different topologies and structures, with a wide range of properties, as introduced in **Section 1**.

The analysis of the complete device is carried out in a similar manner with respect to the  $2 \times 2$  crossbar switch, with automation of the design through MATLAB scripts and performance simulations through the Synopsys suite.

In this chapter the main focus is the mathematical approach to these classes of networks, concerning the generation and routing control, while in **Section 4** the simulation results are shown in more details.

### 3.1 MULTISTAGE SWITCHING NETWORK

Higher order switching networks operate through different implementation paradigms, depending on the use of a different elementary switch[5]. Through a cascade of multiple stages of  $n \times n$  switches various topologies of multistage  $N \times N$  switching networks (MSN) can be generated.

These circuits have different target applications, with some structures oriented to a best-effort blocking approach, while other offer various degrees of reliability in the routing of each signal in the requested order permutation.

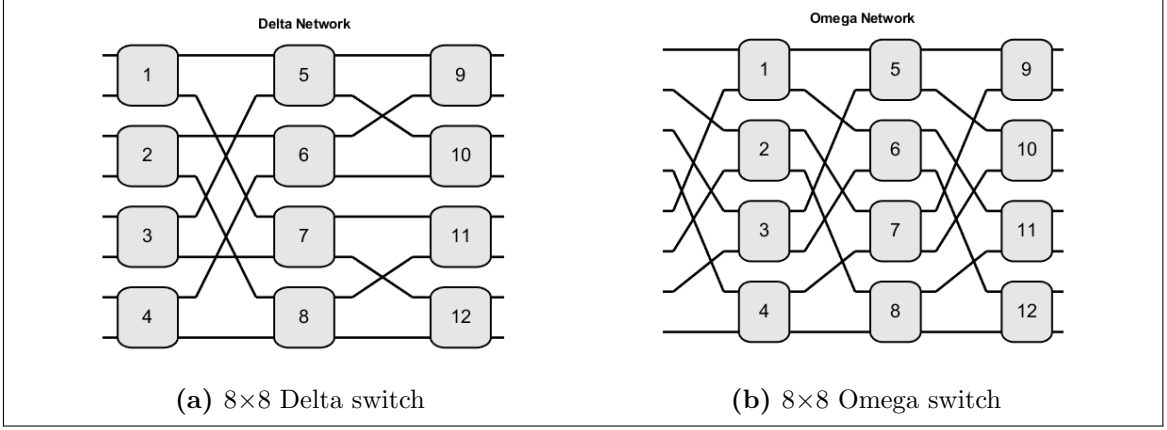
The basic principle underlying MSNs is the placement of multiple switching stages composed by the elementary device, acting as controlled permutation arrays, with a variable interconnect between each stage, acting as a fixed permutation array.

The resulting circuit can be expressed as a permutation matrix, evaluated as the cascade of all the stages of the network.

#### 3.1.1 Banyan switches

The Banyan switch is a class of such circuits, with different topological implementations in terms of the interstage connections, but generally the same structure in terms of switching stages. Two examples of Banyan-based switches are shown in **Fig. 33**, demonstrating the overall similarity between different implementations of the Banyan paradigm: three stages structures, with  $\frac{N}{2} \cdot \log_2(N)$  basic switching elements.

It must be noted that the Banyan class is somewhat undefined in terms of implementation, leaving the possibility of scaling up to additional switching stages, although these cases are usually referred under different topological names.



**Figure 33:** Examples of  $8 \times 8$  Banyan-based switches

Both the Delta and Omega networks show extremely small foot-prints with respect to other topological implementations, albeit allowing conflict in the routing of specific output permutations.

As introduced in **Section 1**, the non-blocking property is fundamental in the application scenario under analysis, and more in general in optical switches, which is not guaranteed through the three stage Banyan-approach.

These circuits are employed when the switching speed is critical, like in electronic processing components, with the availability of buffers and memories to allow storage and forwarding of conflicting packets.

In the optical networking scenario under study, the implementation of buffers and memories is not feasible, while maintaining a low-cost, fully transparent approach, so these configurations have been discarded.

An example of a conflict in both topologies is depicted in **Fig. 34**, with the two relevant signals explicitly drawn.

It's clear that no alternative solution can be routed, as the only routing guaranteeing the propagation of the first input signal to the first output port is the one depicted in blue.

Conversely, no switching state in the whole device can lead to the fifth signal being routed to the second port, while the first constraint is active.

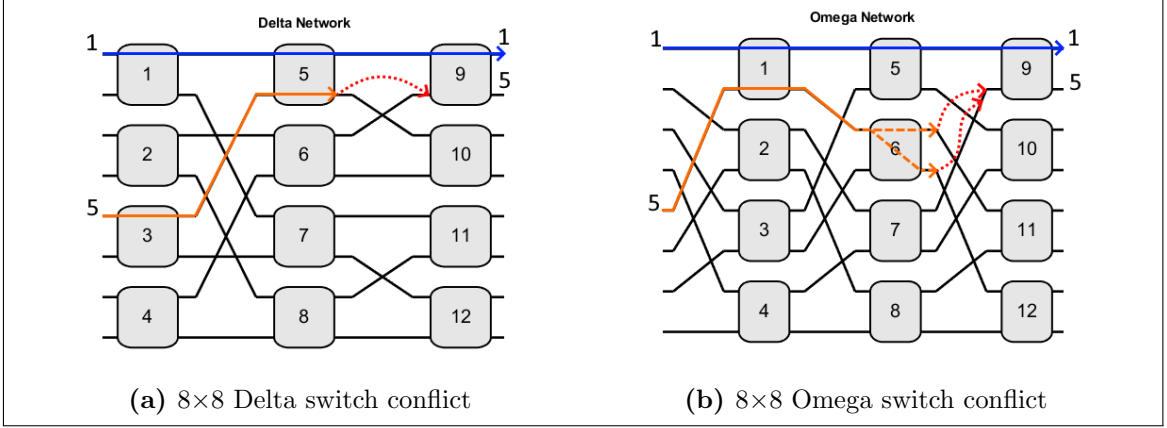
This is an example of a blocking network, which can be clearly demonstrated considering the combinatorics size of the circuit. Considering a switch with size  $N = 8$  and number of switching element equals to  $M = 12$ :

- The number of unique output permutations can be evaluated as  $N_{out} = N! = 40320$ .
- The number of different control state of the network is  $N_{st} = 2^M = 4096$ .

### 3.1 Multistage switching network

Therefore, in the best case scenario, with no control state routing the same output permutation, the circuit can guarantee  $N_{st} = 2^{12}$  unique output permutations. This simple mathematical analysis is insufficient to demonstrate if a circuit is non-blocking, but must hold true for any conflict-avoiding switching network (**Eq. 11**).

$$\text{if } 2^M \leq N! \implies \text{Conflict} \quad (11)$$



**Figure 34:** Examples of blocking in  $8 \times 8$  Banyan-based switches

#### 3.1.2 Clos networks

The Clos network[6] is a widely implemented solution for the design of MSN, with compatibility with optical applications due to its use of the crossbar switch as fundamental element.

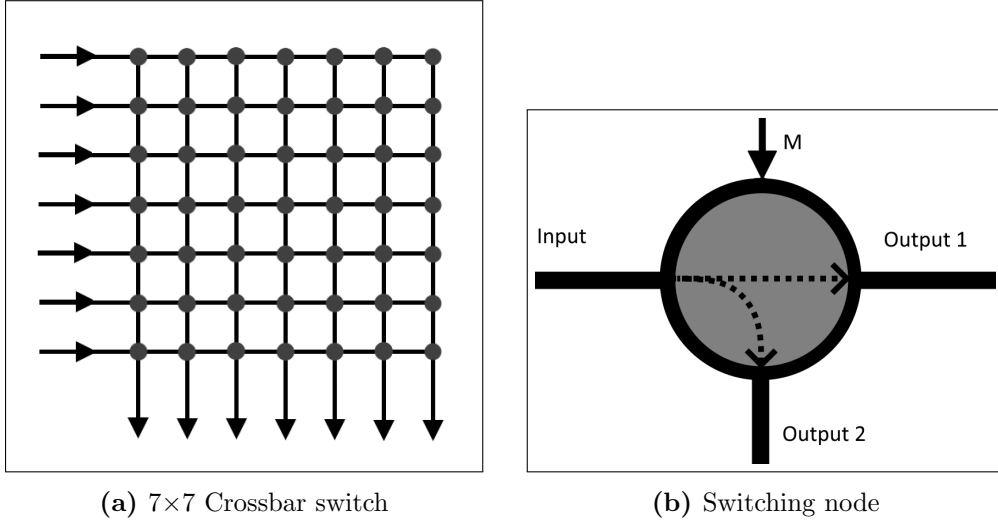
The general Clos switch is not based on the  $2 \times 2$  element seen in the previous sections, while relying instead on a generic  $n \times m$  crossbar structure.

An arbitrary size crossbar switch can be constructed as a rectangular mesh between  $n$  inputs links and  $m$  orthogonal output links, such that each input link intersect all outputs connections. Every intersection can act as a switching device, propagating the signal from the link to its orthogonal connection.

An example for a  $7 \times 7$  crossbar switch is shown in **Fig. 35a**, with each gray node representing a  $1 \times 2$  black-box switch, depicted in **Fig. 35b**.

The state of each node is determined through a control signal  $M$ , similarly to the driving signal of the OSE seen in **Section 2**.

These switches have trivial routing algorithms, as well as allowing the independent propagation of each signal to any port, even already occupied ones. The trade off is due to the high number of switches needed, which grows as  $n \cdot m$ , as well as physical concern, due to the potentially high crosstalk and power dissipation.

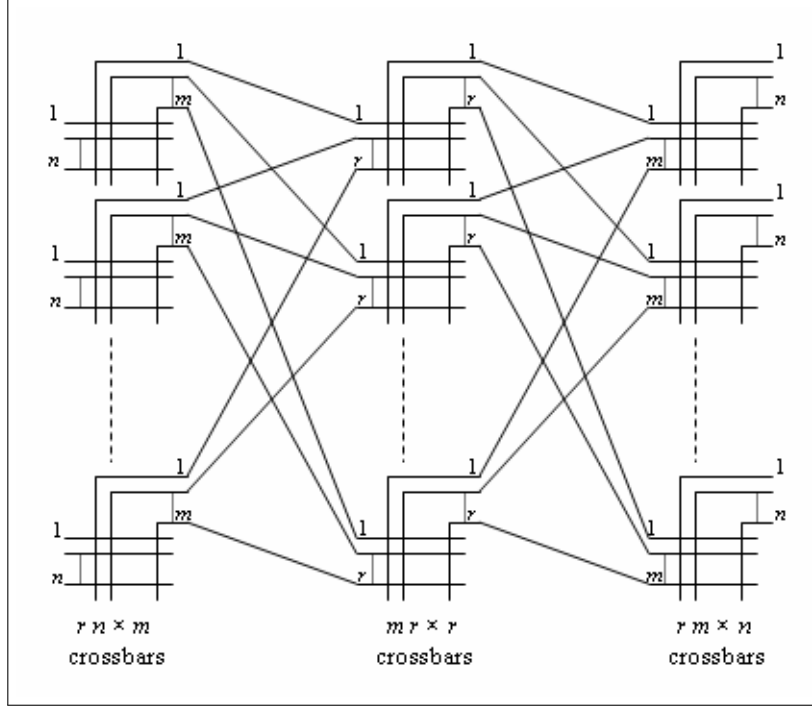


**Figure 35:** Crossbar switch black-box model

The Clos network (**Fig. 36**) is a widely implemented solution to the scalability problem, generating a multistage network with arbitrary sized crossbar elements, as to reduce the quadratic increase in footprint and number of crossings.

The topology is divided into three stages, which can be described through three main parameters that determine the sizes and number of crossbar switches in each section of the circuit.

- The ingress stage is composed by  $r$  crossbar switches with  $n \times m$  size
- In order to maintain bijective connections, the middle stage must be defined as  $m$  switches with size  $r \times r$ .
- The last stage, similarly to the first one, must be set as  $r$  switches with  $m \times n$  dimension.



**Figure 36:** Generic Clos Network with  $r \cdot n$  signals and  $m$  intermediate switches

While conflict cannot be avoided for the three stages Banyan switches, the Clos network allows various degrees of freedom in the design, as to ensure a stricter or wider conflict-avoidance.

For  $m \geq 2n - 1$  the switch is strict sense non-blocking, meaning that the input output routing can always be established for any request, even with traffic already occupying parts of the network. This property is the more robust form of conflict-avoidance, although it comes with an intrinsic cost in terms of number of switches. If the Clos network is designed with  $m \geq n$ , a weaker non-blocking property is obtained, defined as *rearrangeable*.

Rearrangeable non-blocking networks can route any output permutations, although the previously established connections may need to be changed or re-routed: this solution compromises between the performance and flexibility of the device with respect to the footprint and complexity.

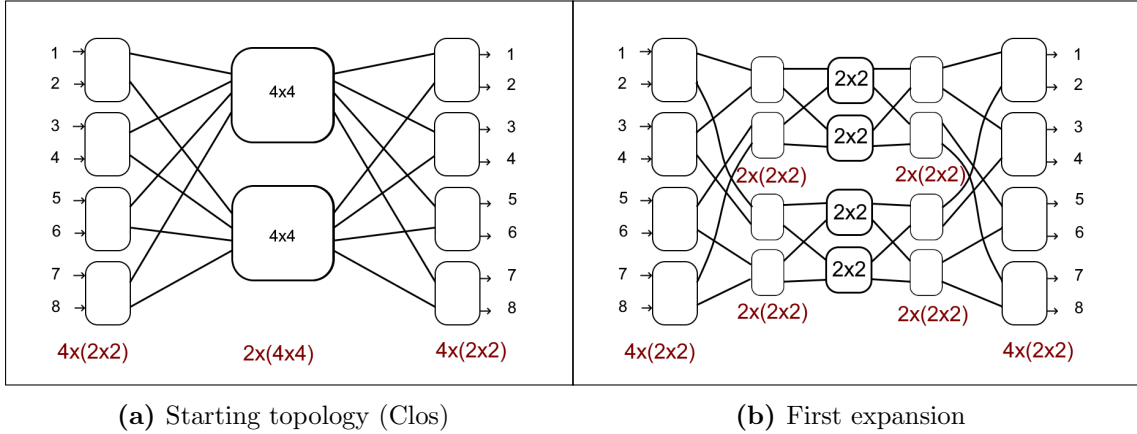
The wide-sense non-blocking hold true for values of  $m = n$ , which allows the implementation of this class through the  $2 \times 2$  crossbar switch discussed in the previous sections. This configuration is referred to as Beneš network, whose properties and general construction are explored in the following section.

### 3.2 BENEŠ TOPOLOGY & PROPERTIES

The Beneš network topology, as explained previously, can be considered as a specific configuration of the more general Clos network. While typically the Clos topology is depicted as a three stage structure, it can be generalized to any odd number of stages, by altering the sizes of the intermediate stage switches: the general approach for this process is the recursive expansion of the middle stage into an equivalent three-stage Clos. The process can then be repeated, until the size of the device allows the expansion.

These steps are shown for an  $8 \times 8$  Beneš in **Fig. 37**:

- Starting from the basic Clos definition (**Fig. 37a**) the parameters are set as  $m = n = 2$ . As a consequence of the choice of the switch,  $r = 4$ , and consequently the middle-stage is defined as two  $4 \times 4$  crossbar switches.
- The middle stage is expanded into a standard three stage Clos (**Fig. 37b**), with each  $4 \times 4$  switch represented by a Clos with  $m = n = 2$  and by consequence  $r = 2$ .



**Figure 37:** Recursive  $8 \times 8$  Beneš generation

Once the basic element size is obtained, the recursion stops, and the Beneš device is generated.

This method is straightforward when the size of the sub-network under analysis is even: if every step of the expansion must be even, the overall network size is constrained to  $N = 2^x \quad x \in \mathbb{N}$ .

The rule for the interconnections between the stages follows a similar pattern to the central stage OSE expansion, due to the presence of the two sub-networks.

For the generic Clos topology, each output of the crossbar blocks is connected to a different crossbar element in the next stage, with the symmetric connection being



### 3.2 Beneš topology & properties

applied from the third stage, backwards.

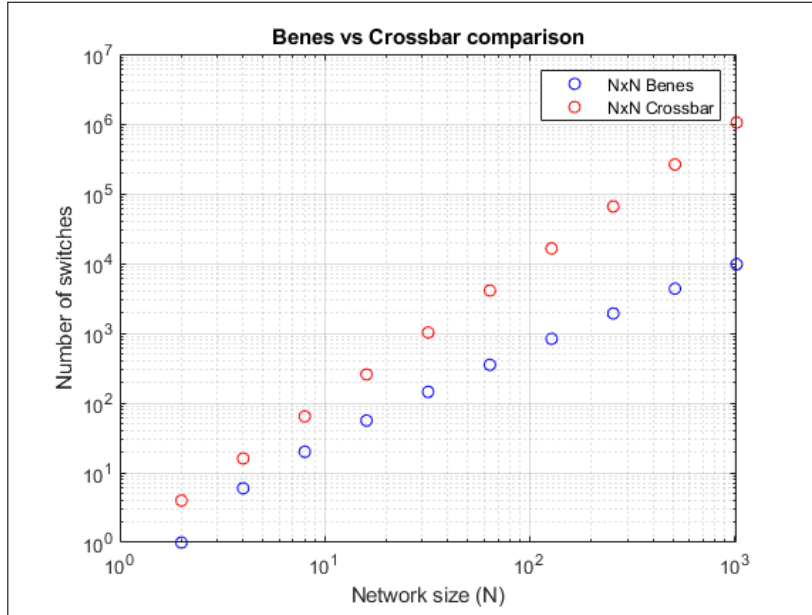
Under the constraint of a  $2 \times 2$  basic element, this property remains unchanged, leading to the following relationship: considering the  $i^{th}$  switch in a sub-stage  $I$  of size  $N$ , connected to the two successive stages  $J_1, J_2$  of size  $\frac{N}{2}$ , the two output ports of the switch are connected to the  $i^{th}$  element of both  $J_1$  and  $J_2$ .

Alternatively considering the successive sub-network  $J = [J_1 \ J_2]$  of size  $N$ , the outputs of the  $i^{th}$  element are connected to the input ports  $[i, i + \frac{N}{2}]$  of the sub-network  $J$ .

Due to the recursive generation of both switches and interconnects, the Beneš is an highly regular network, with both horizontal and vertical symmetry with respect to the central stage.

The device complexity can be evaluated from the input size  $N$  as:

- The number of unique output permutations is equal to  $N_{out} = N!$ .
- The number of stages is  $N_{stages} = 2 \log_2(N) - 1$ , each containing  $N_{sw} = \frac{N}{2}$  crossbar switches of  $2 \times 2$  dimension.
- The total number of switches in the network is therefore  $N_{SW} = N_{sw} \cdot N_{stages} = N \cdot \log_2(N) - \frac{N}{2}$ .
- The total number of control states available is  $N_{st} = 2^{N_{SW}} \geq N_{out}$ .



**Figure 38:** Beneš and crossbar switches comparison: number of  $2 \times 2$  OSEs required

### 3.2 Beneš topology & properties

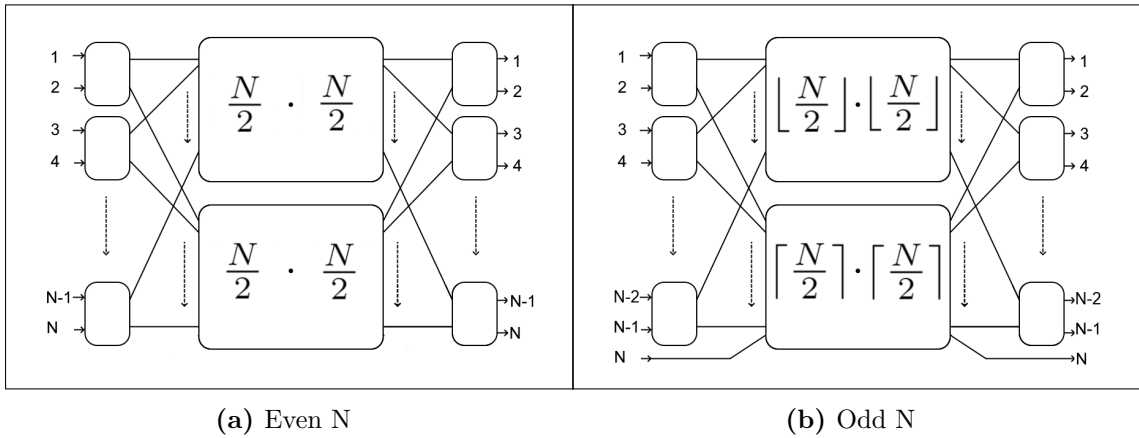
It's clear that the complexity of the device grows rapidly as the number of input channels  $N$  increases, due to the number of switch following  $O(N \cdot \log(N))$  growth, although keeping in mind that the straightforward implementation of the  $N \times N$  crossbar switch as a  $O(N^2)$  dependency.

The comparison between the Beneš and the generic  $N \times N$  crossbar is shown in **Fig. 38**, with logarithmic scales, clearly demonstrating the improvement of using a Clos-based approach, concerning the number of OSE needed.

As previously stated, the results displayed in the graph, as well as the evaluation of number of stages and switches, is constrained to input sizes which are power of two. This is due to the intrinsic strict-sense definition of Beneš switches, based on the Clos stage expansion, although the topology can be generalized to any arbitrary unconstrained size  $N$ .

This generalization is referred as Arbitrary Sized Beneš, or AS-Beneš [7], and follows the overall same principle of the strict-sense structure, albeit with slight modifications to account for the odd sized sub-network arising from the unconstrained size.

#### 3.2.1 Arbitrary size Beneš



**Figure 39:** Recursive AS-Beneš generation

In order to construct the AS-Beneš switch through the same recursive procedure, some exceptions must be set for odd-sized network sizes, which can occur at any stage of the recursive expansion.

Considering a stage with input size  $N = 2k + 1$   $k \in \mathbb{N}$ , using  $k$  switches leaves one input port of the stage "uncoupled", while using  $k + 1$  elements leads to one of the switches having only one distinct input signal.

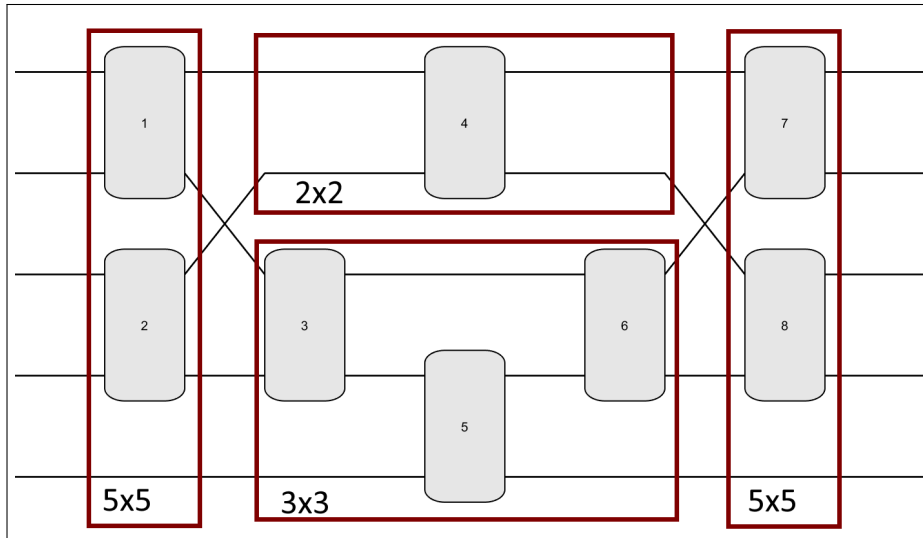
Maintaining a similar black-box assumption on the middle stages, already used in the expansion of the Clos network, we can obtain the recursive structure seen in

**Fig. 39.** The even case sub-network can be assumed to be structured as the traditional Beneš, thus maintaining the correct non-blocking property, while the odd-sized sub-network, shown in **Fig. 39b**, must be analyzed to ensure the correct behavior. As a standard notation the "uncoupled" signal is placed as the last port of the stage, and is propagated to the lower sub-network, which has size  $\lceil \frac{N}{2} \rceil$ , while the top sub-network has sizes  $\lfloor \frac{N}{2} \rfloor$ : this causes the symmetry break with respect to the constrained Beneš.

Following this strategy, the two sub-networks of the odd case are again even sized structures, albeit with different sizes.

The main benefit of using a recursion-based structure is the analysis of the behavior by observing the final step in the expansion of the circuit: the structure at each phase must be able to route the signals from any possible input to any output port, which imposes a constraint over the successive recursive networks.

The odd sized ingress stage can route each signal to the top or bottom networks, except for the uncoupled port, which is hard wired, while the two black-box networks are connected similarly to all switches at the egress stage, shifting the routing requirement to the middle stage.



**Figure 40:** 5×5 AS-Beneš: highlight of the recursive blocks for the generation

An example for the 5×5 network is shown in **Fig. 40**, with an highlight of the recursive blocks. The simple 2×2 top network can clearly route the two required combinations, while the 3×3 bottom network can be easily tested to demonstrate the possibility to route all six required combinations.

Without entering into a deeper justification of the 3×3 model, it can be considered

### 3.2 Beneš topology & properties

---

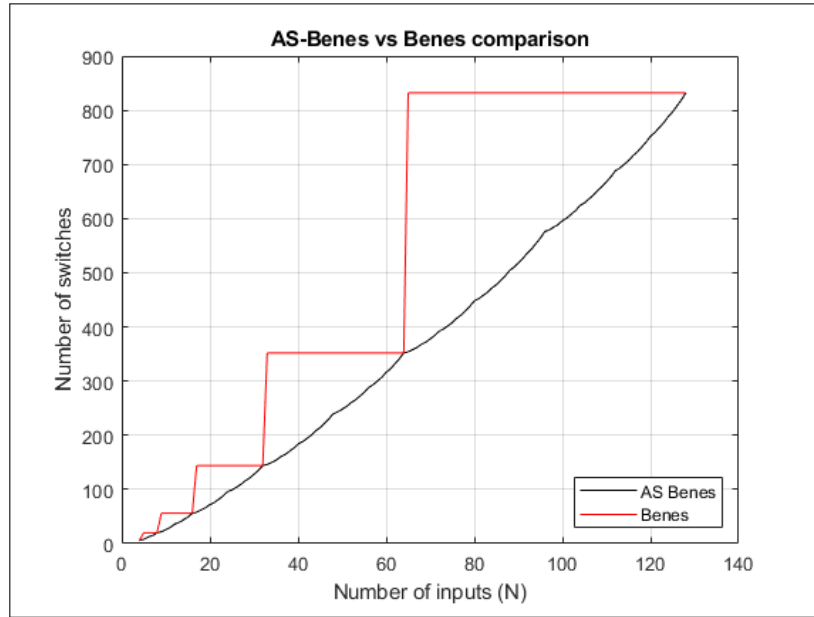
as an additional stopping block for the recursion, together with the  $2 \times 2$  crossbar structure.

Having demonstrated the non-blocking routing capabilities of the minimum size blocks obtainable in the recursion, the justification of the non-blocking property for the complete structure is omitted, as being outside the scope of the current analysis, but it follows closely the non-blocking proof for the Clos network.

This method allows the generation of any AS-Beneš, with a higher degree of flexibility in the implementation of specifically sized-devices, without requiring the design of a much bigger structure based on the next available  $2^x \times 2^x$  Beneš size.

In **Fig. 41** the number of elements needed to design the AS-Beneš is compared to the number of switches in the smaller suitable Beneš network.

Nonetheless some consideration for this generalization are in order, as to introduce some analysis which will be shown in the following sections. The absence of symmetry in the device, may lead to degradation of performances in a realistic device, as the path travelled by different signals may contain a non-uniform number of switch hops and inter-stage crossings. This can lead to uneven filtering penalties over the whole device, requiring a more careful evaluation of the routing strategies in the network.



**Figure 41:** AS-Beneš and Beneš comparison : number of OSEs required for an  $N \times N$  implementation

### 3.3 MATLAB: IMPLEMENTATION

Having set the mathematical model and background for the generation of an AS-Beneš, different strategies can be used for the actual construction of the logical simulation model, based on the data-structure used to store the network description. The notation of AS-Beneš will be avoided from this section forward, referring to both structure as Beneš switch, unless explicitly specified for comparison purposes.

At this stage the approach is still purely mathematical, without any consideration on OSE structures, signal modulations, or any other physical or transmission parameter, considering only the topological description and the different logical implementations that have been analyzed.

In addition to the network size  $N$  an additional parameter is established to characterize the state of the network, representing the switching signal applied to the  $2 \times 2$  elements of the device.

The state can be fully characterized by a binary vector  $V$ , with  $V_i = 0$  representing the default bar state, and  $V_i = 1$  the switching, or crossing state, for  $i \in [1, N_{SW}]$ . Concerning the logical description of the network, three different strategies have been tested and implemented, with different results and applications:

- The network can be evaluated through a matrix-based description, with each stage represented by a permutation vector, leading to the evaluation of the output through multiplication of a cascade of vectors.
- The structure can be represented as a unilateral unweighted graph, obtaining the output vector by exploration of the structure.
- The structure has also been evaluated mathematically, describing the switches as logical equations, as to obtain an analytical formula for the output signal of each port.

The vector-based approach has been chosen as the final method for evaluating the routing and the device topology, due to its faster computational time, although the other two approaches are explained as they offer insights in the behavior and mathematical complexity of the analysis of this class of circuits.

All three methods have been implemented in MATLAB, with the capability of generating the description for any  $N \times N$  Beneš network.

#### 3.3.1 Graph-based approach

In the graph-based description of the network each switching element is implemented as a node, with the unweighted edges acting as the inter-stage links. Given that each OSE is connected to two other elements, the data-structure for each nodes includes

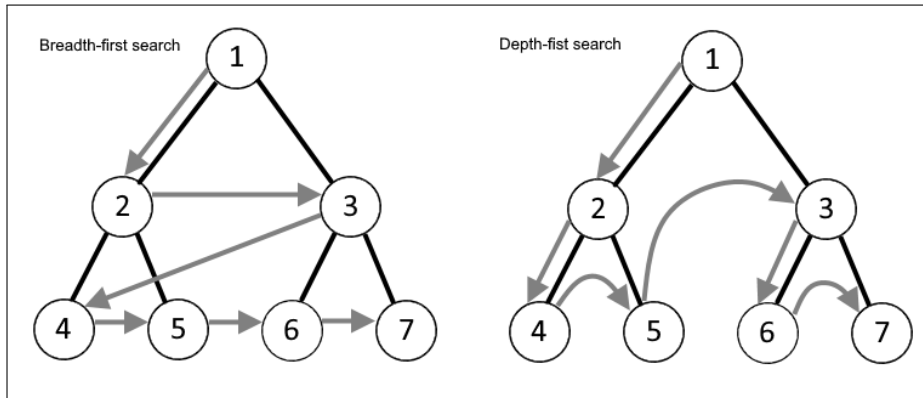
the two output links, with their respective destination ports.

Following the recursive algorithm for the Beneš generation, each group of nodes is initialized and connected during the recursive call, while the horizontally symmetric second-half of the network is generated during the return calls. During the generation of the network, a depth-first search approach is used (*DFS*), instead of the breadth-first search (*BFS*) method employed in the generation of the last stages.

During the recursion stages, which terminate at the previously discussed stop-blocks ( $2 \times 2$ ,  $3 \times 3$ ) configurations, the algorithm uses a DFS technique, shown in **Fig. 42**. This is mainly due to the possibility of building the next blocks starting from the next parent sub-network: both the top and bottom recursive networks depends only on the parent structure.

The DFS cannot be employed for the whole network, as once the second half of the network is reached, the opposite property applies: both parent sub-networks are needed to generate the successive switching block.

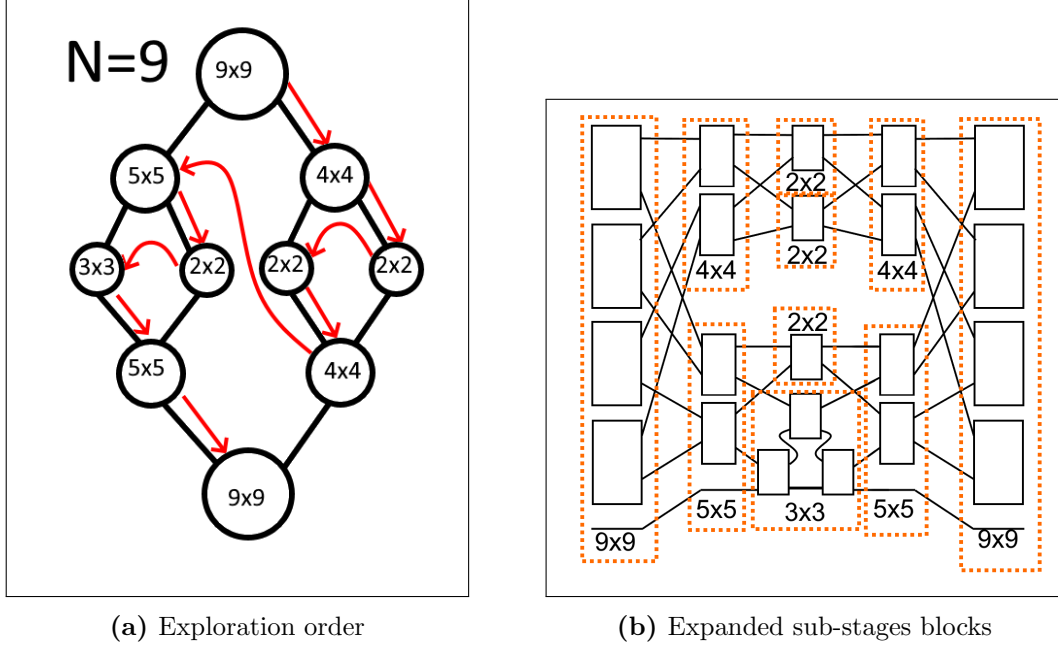
This lead to the necessity of implementing a hybrid traversal algorithm, due to the creation of the left network during the call, and the right network during the return. This heterogeneous method of exploring the device is mainly due to the model chosen to represent it.



**Figure 42:** Comparison between BFS and DFS exploration methods

The structure can be seen as two binary trees, linked through their leaves, which represent the middle stage and the return from the recursion.

In **Fig. 43a** the order of search through the structure is shown. Each node contains multiple OSE and each branch is comprised in reality of multiple connections: the tree approach is simply to visualize the overall process and hierarchy on which the script relies on.



**Figure 43:** 9x9 Beneš topological and blocks model

The circuit topology with switches placement and interconnections is shown in **Fig. 43b**: each element inside a given group is evaluated and connected before passing to the successive block. This process is carried out up to the final stage, using the hybrid *BFS* and *DFS* to generate the correct links between the cascading stages.

Although intuitive considering the graph structure of the network, this method suffers from a clear disadvantage. Due to the generalized graph-oriented approach, the exploration of the graph requires a similar recursive analysis of the dependencies of the rings, in order to evaluate the cascading effect correctly. This means that the graph generation and the test of a specific control vector have equal computational and time costs.

The main advantage of this description is the re-usability of the codes for the Optim implementation, due to a similar node and link approach, and the possibility of carrying out analysis on paths and specific routes of the network more easily, due to path-finding oriented exploration of the circuit.

### 3.3.2 Logical equations approach

The second approach under analysis is the expression of the basic switch as a logical equation, with a binary-constrained variable.

This is akin to the modelization and parametrization carried out in mathematical

optimization, by formulating the constraints and behavior of a system through the use of decision variables, on top of a simpler mathematical model.

Considering the two states of the switches, the output of each port can be described based on the control signal, as shown in **Table 1**.

	Control ( $M$ )	
	0	1
Output <sub>1</sub>	Input <sub>1</sub>	Input <sub>2</sub>
Output <sub>2</sub>	Input <sub>2</sub>	Input <sub>1</sub>

**Table 1:** Switching matrix of a 2×2 crossbar switch

This allows the switching element to be defined analytically through the use of an *if* logical constraint, which represents the control signal  $M$  (**Eq. 12**).

$$Out_1 = \begin{cases} In_1 & \text{if } M=0 \\ In_2 & \text{if } M=1 \end{cases} \quad Out_2 = \begin{cases} In_1 & \text{if } M=1 \\ In_2 & \text{if } M=0 \end{cases} \quad (12)$$

The definition of each output port can be rewritten as a single formula, by exploiting the binary nature of the control variable, obtaining a logical equation for the outputs of a generic switch  $i$ , as shown in **Eq. 13**

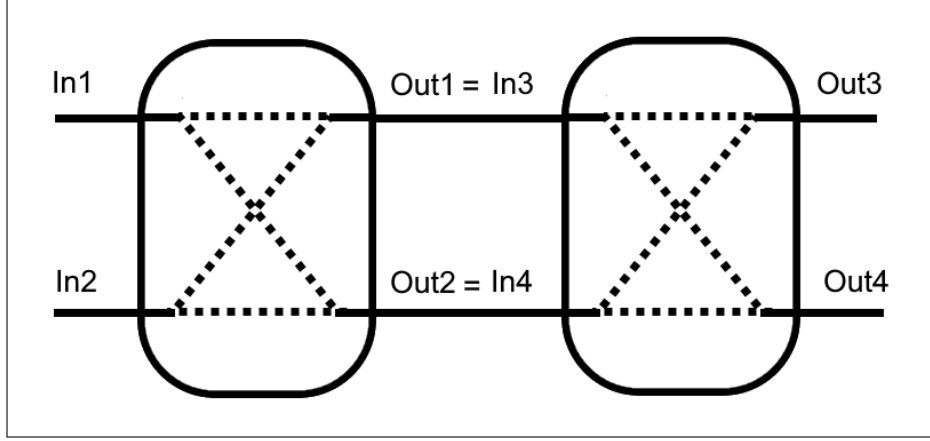
$$\begin{aligned} Out_1 &= In_1 \cdot (1 - M_i) + In_2 \cdot (M_i) \text{ for } M_i \in \{\mathbb{N} : [0, 1]\} \\ Out_2 &= In_1 \cdot (M_i) + In_2 \cdot (1 - M_i) \text{ for } M_i \in \{\mathbb{N} : [0, 1]\} \end{aligned} \quad (13)$$

With the definition of the switch in place, cascading elements can be evaluated by traditional substitution.

Considering the example shown in **Fig. 44**, the evaluation of the output ports for each switch is straightforward, as demonstrated in **Eq. 14**.

$$\begin{aligned} Out_1 &= In_1 \cdot (1 - M_1) + In_2 \cdot (M_1) \text{ for } M_1 \in \{\mathbb{N} : [0, 1]\} \\ Out_2 &= In_1 \cdot (M_1) + In_2 \cdot (1 - M_1) \text{ for } M_1 \in \{\mathbb{N} : [0, 1]\} \\ Out_3 &= In_3 \cdot (1 - M_2) + In_4 \cdot (M_2) \text{ for } M_2 \in \{\mathbb{N} : [0, 1]\} \\ Out_4 &= In_3 \cdot (M_2) + In_4 \cdot (1 - M_2) \text{ for } M_2 \in \{\mathbb{N} : [0, 1]\} \\ In_3 &= Out_1 \\ In_4 &= Out_2 \end{aligned} \quad (14)$$





**Figure 44:** Direct cascade of 2x2 crossbar switching elements

In order to have a clear equation for the input-output relationship of the whole device, the formulas of the first switch element must be explicitly expanded in description of the second element. Using  $Out_1$  as a notation for the first output port of the whole device, the expansion is shown in **Eq. 15**, with the binary constraint omitted.

$$Output_1 = ((In_1 \cdot (1 - M_1) + In_2 \cdot (M_1)) \cdot (1 - M_2)) + ((In_1 \cdot (M_1) + In_2 \cdot (1 - M_1)) \cdot (M_2)) \quad (15)$$

This method allows the generation of a clear algebraic expression for the definition of any 2x2 crossbar switch based structure, although the size and complexity of the expression may prove to be impractical for any solution search.

These expressions have been evaluated in MATLAB for a number of different  $N \times N$  Beneš structures, and are not reported due to the size and overall complexity.

The main problem in the evaluation and resolution of this kind of equations is due to the nested nature of the multiplications, which can be cumbersome even for a mathematical oriented programming language like MATLAB. Some test have been carried out concerning the resolution and optimization of the formulas, although severe limitation halted the progress of this analysis. Concerning the evaluation of the output, with a suitable control vector provided, the script relies on the MATLAB *eval* function, which is non-optimal and heavily time-consuming, making this method inefficient.

Theoretically, mathematical minimization could be applied to the Beneš switch, thanks to this notation, but the problem falls under the class of *Mixed Integer Non Linear Programming* (MINLP). Unfortunately, no general solver is available for this class of problems, as both the non-linearity and integer nature of the model doesn't allow the application of the traditional resolution strategies.

Although this description has been proven to be ineffective in any practical analysis of the device the overall method may be of interest for future expansion in the analysis of topological optimization of these structures. Due to the high computational complexity for these structures, which will be analyzed in the later sections, some insight into mathematical optimization and non-linear programming could be beneficial for tackling the complexity of this whole class of devices, without relying on topology-dependent algorithms, or brute-force analysis.

#### 3.3.3 Matrix-based approach

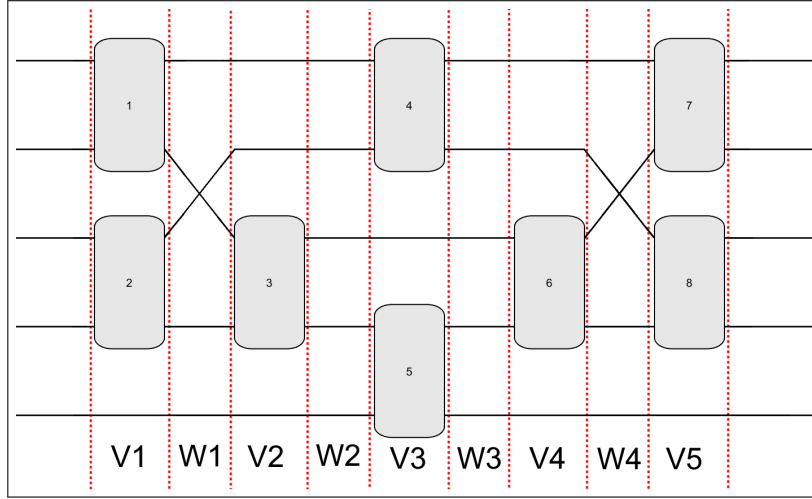
The method typically applied in the description of the switching structures is the matrix approach. The circuit is divided into switching stages, made from the elementary switching elements, and interconnect stages, with the links defined by the topology under analysis.

This approach to the description of the device is preferred as it also allows faster implementation of routing algorithms, discussed in **Section 3.5**.

Through this approach each switching stage is defined as a vector  $V \in \mathbb{N}^{N_{sw} \times 2}$ , with  $N_{sw}$  representing the number of basic element in the given stage, while the inter-stage links are defined as  $W \in \mathbb{N}^{N \times 1}$ .

The division of the various stages is depicted in **Fig. 45**, with the equivalent matrix format shown in **Table 2**. The notation used in the link vectors is straightforward, as the content of each cell defines the output port for each input signal, with the rows representing the input stage ports, and the column representing the different crossing stages.

The majority of the link stages simply propagate the signals without any modifications, so the two crossing present in the  $5 \times 5$  switch have been highlighted.



**Figure 45:** Permutation vector stages of a 5×5 Beneš switch

The notation used in the switching stages is slightly different, to account for the variable number of elements present in each stage. For an  $N \times N$  device the maximum number of switches per stage is equal to  $N_{sw} = \lfloor \frac{N}{2} \rfloor$ , so in the case under analysis only two rows are necessary to describe the switch. The cells contain the two input signals that can be switched in each point of the circuit, if the appropriate control signal is active.

	Link stages (W)			
	1	2	3	4
Port <sub>1</sub>	1 → 1	1 → 1	1 → 1	1 → 1
Port <sub>2</sub>	<b>2 → 3</b>	2 → 2	2 → 2	<b>2 → 3</b>
Port <sub>3</sub>	<b>3 → 2</b>	3 → 3	3 → 3	<b>3 → 2</b>
Port <sub>4</sub>	4 → 4	4 → 4	4 → 4	4 → 4
Port <sub>5</sub>	5 → 5	5 → 5	5 → 5	5 → 5

	Switching stages (V)				
	1	2	3	4	5
Switch <sub>1</sub>	1 ↔ 2	3 ↔ 4	1 ↔ 2	3 ↔ 4	1 ↔ 2
Switch <sub>2</sub>	3 ↔ 4	-	4 ↔ 5	-	3 ↔ 4

**Table 2:** Vector representations of switching and crossing stages

In MATLAB the two description can be implemented through matrices and expressed in a more compact way by combining them in a single data-structure, with the description of each vertical section of the network. The switching state of the network can be obtained by inverting the content of the cells for the stages  $V$ , when a non-zero signal is applied to the switch.

The control state is expressed as described in the previous sections, with a binary control vector  $M$ , providing the cross or bar configuration for each  $2 \times 2$  element. The evaluation of the device output is carried out by applying the permutations of each stage sequentially, which is the less computationally intensive process between the previously explained approaches.

The main advantage of this notation is the separation between the generation of the network and the evaluation of the outputs for a given state: this is not possible in the graph-based approach, due to the need of exploring the structure with the same recursive algorithm used for creating the connections. Overall this method is preferred due to its faster evaluation, as well as overall simpler and less costly structure used to store the information.

#### 3.3.4 Computational costs

Following the generation of the network description, a fundamental step is obtaining the control signals in order to route all possible combination output permutations. The device, as previously introduced, suffers of a scalability issue concerning the growth of the available configuration states, as well as the unique signals permutations at the output ports.

The combinatoric size of the problem can be expressed through two parameters:

- The number of possible output permutations is evaluated as  $N_{out} = N!$ , with  $N$  representing the number of input ports.
- The number of possible control states of the network is evaluated as  $N_{st} = 2^{N_{sw}}$ , as each switch has been constrained to a  $2 \times 2$  crossbar model.

Before tackling the non-polynomial growth of the solution space, defined as the set of all output configurations with their respective control states, the timing cost of each description model must be analyzed, to ensure the feasibility of the analysis, as well as choosing the most suitable method.

The analysis has been carried out by testing the time elapsed in the evaluation of a single control state of the network, for each model, as a function of the size of the device  $N$ .

The results are shown in **Fig. 46**, confirming the assumption of the previous sections. It must be noted that the range of sizes used for the test is extremely wide,

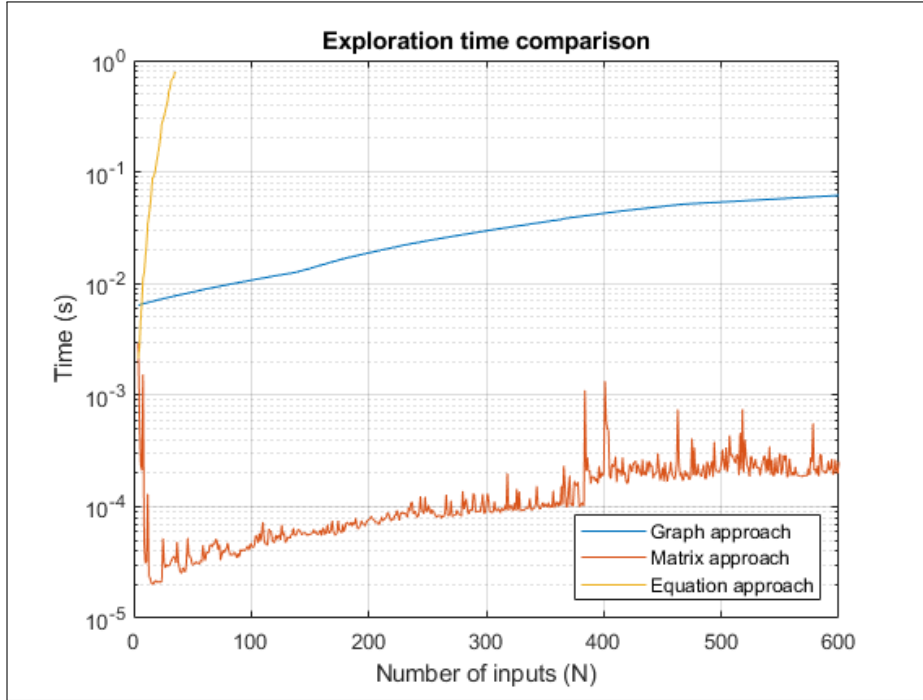
### 3.3 MATLAB: implementation

---

over any reasonably sized Beneš switch: this was done to test the robustness of the method, as well as highlighting the overall trend, although a realistic upper limit for an optical switching network may be around  $N \approx 10^1 \sim 10^2$ , with an already severe degradation of performances.

Concerning the three methods previously analyzed:

- The equation-based approach is completely unfeasible, as the cost increases orders of magnitudes even for a small increment of the network size.
- The graph-based description suffers from a higher computational time, due mainly to the costly recursive description, but in terms of increase as a function of the network size, although sub-optimal, could be used for larger network analysis.
- The matrix-based description, ignoring some spurious noise due to the measuring error, shows the best behavior, with an average difference of two order of magnitudes with respect to the graph approach.



**Figure 46:** Timing comparison between the proposed network descriptions: time required to evaluate a single configuration of a  $N \times N$  device

This linear growth in computational time might look promising on its own, although to understand the computational cost it's useful to consider the previously discussed combinatoric constraints.

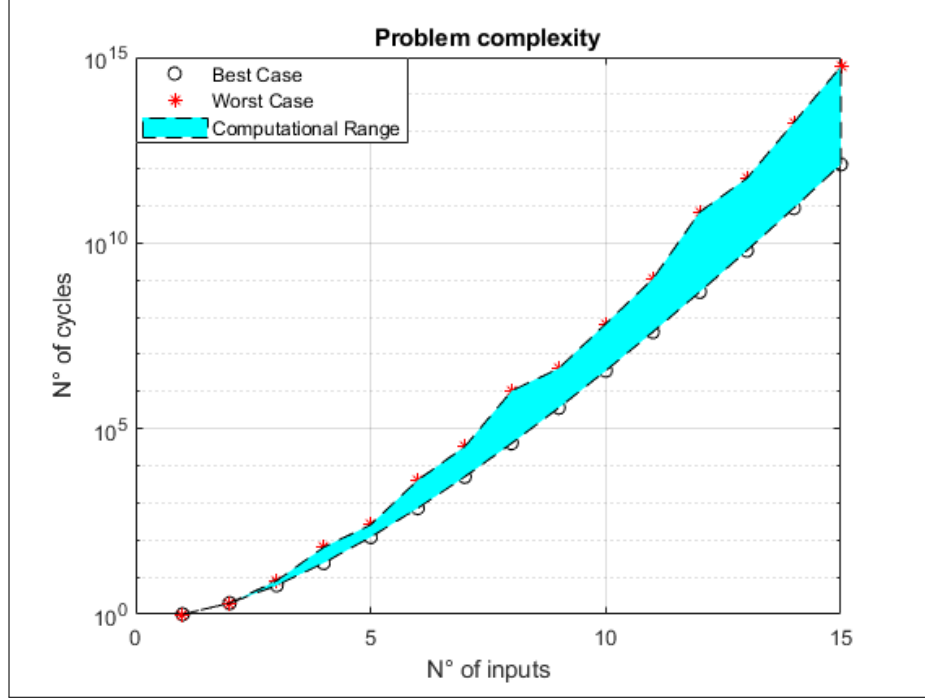
In order to find a complete set of routing, the minimum number of needed evaluation are  $N!$ : this is under the assumption that all state configurations tested lead to a unique output configuration. In reality the total number of states of the network depends on the number of switches used, with multiple routings generating the same output configuration.

The exponential growth of the available control vectors  $N_{st} = 2^{N_{sw}}$ , as well as the overall dependency of the number of elements on the size  $N$ , leads to an approximated model for the increase in the complete solution set for the Beneš under analysis. It must be remembered that for AS-Beneš devices the number of switching elements  $N_{sw}$  still follows the overall trend  $O(x \cdot \log(x))$ , although with some local variations, as shown in **Fig. 41**.

As a consequence the generation of the complete solution is a clear non-polynomial (NP) problem in nature, as the minimum number of needed iterations grows as a factorial, while the worst case as the exponential.

This NP behavior is shown in **Fig. 47**: for a given number of input ports  $N$  the upper and lower bound for the number of iteration are shown, with the computational range between the two limits highlighted.

Considering the rapid increase in the solution space, the linear growth of the device evaluation, as a function of its size, has a negligible effect, over the main NP increase. This leads to a clear bottleneck in terms of the generation of a look-up table for the control of the device. Brute-force approaches and even topology-dependent routing algorithms cannot tackle the problem complexity, as such the full evaluation is possible only as a case-study for smaller size devices.



**Figure 47:** Number of iterations required to find the unique routings of a  $N \times N$  Beneš

### 3.4 MATLAB: BRUTE FORCE APPROACH

The first approach implemented consists of the brute force evaluation of the complete solutions set. This method suffers from a strong scalability issue, as the NP complexity strongly limits the cases that can be studied through this approach.

This complete set generation has been carried out up to a limit of  $N = 8$ , which corresponds to  $N_{st} \approx 1 \cdot 10^6$  and  $N_{out} = 40320$ , although the described optimization processes have been analyzed for a  $6 \times 6$  Beneš, as to reduce the computation time and provide more clear results.

#### 3.4.1 Cross-Bar states optimization

The control vectors are generated as integer values  $x \in \{\mathbb{N} : [0, 2^{N_{sw}}]\}$ , which are then converted to their binary representations, obtaining the binary signals driving each switch.

The script evaluates the cascade of permutation vectors, applying the cells inversions based on the given control signal, saving both the control and output configuration in the data-structure containing the solution space.

Two different scripts can be used, in order to obtain the partial or complete data-set. By choosing random control vectors and saving only the unique outputs, the

process is faster, as the scripts interrupts the analysis once an instance of each output configuration is found: this leads to an incomplete data-sets, which can be used to route all possible configurations as a look-up table, although the driving signals are random and may not be optimized.

The second scripts evaluates sequentially every possible control vector, generating the complete data-set, with all the alternative routings that can lead to any output configuration.

The second approach requires a number of iterations equal to  $N_{st}$ , while the unique solution set can be evaluated in a smaller time: considering that only  $N!$  unique solutions are present in the network, as the size increases, the difference in order of magnitude between the two quantities is more significant, as already shown in **Fig. 47**. This complete data-set will be used for the optimization proposed in the next section.

For the generation of the incomplete solution space, two different approached have been tested to select the driving signals:

- Priority can be given to control signals activating the minimum number of switches ( $V = V_{on}$ ): all possible permutations will be evaluated for a single active switch, increasing the number of active elements one at a time, until all  $N!$  unique outputs are obtained.
- The opposite target, with respect to the previous one, can be selected: the control signals are tested in order to minimize the number of passive switches ( $V = V_{off}$ ), following the same approach.

The goal behind this selection is due to the properties of the OSEs seen in **Section 2**. Depending on the physical implementations of the switching elements, different filtering penalties may be associated to the two states of the switch. This evaluation of the look-up table allows the generation of a set minimizing the transmission power loss, or more in general, considering the power required to enable the switching of the states  $V = V_{on}$ , minimizing the bias power dissipation due to the heating of the MRRs.

#### 3.4.2 Switch optimization

The recursive algorithm behind the Clos expansion into a general  $N \times N$  Beneš network offers a reliable and quick way to generate a rearrangeable non-blocking structure, although the topology offers no guarantee that the number of implemented OSE corresponds to the minimum required elements for ensuring the routing of all output permutations.

Ad-hoc solutions may exist for a given  $N$ [8], although non applicable to the bottom-up approach followed in this project, due to the manual optimization carried out to



obtained these custom networks.

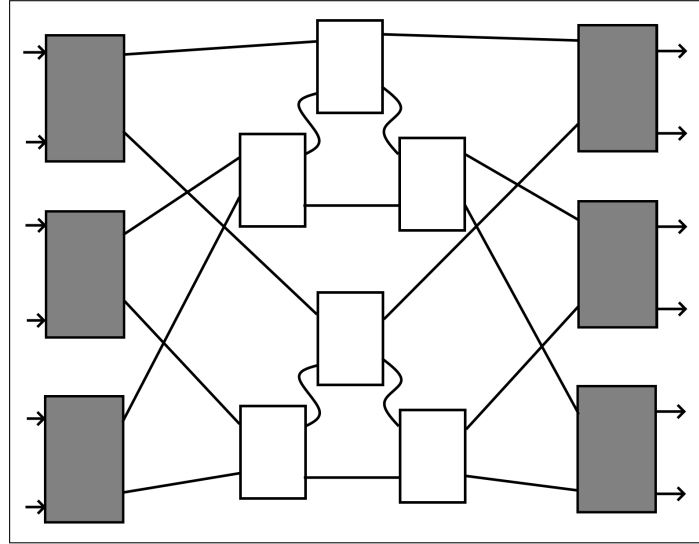
In order to test the optimality of the number of switches used by the network, while maintaining a generalized point of view, it is necessary to evaluate the matrices  $Out_{all}, M_{all} \in \mathbb{R}^{2^{N_{sw}} \times N}$  previously described, which correspond to the complete data-set of all output configurations and driving states.

To verify if every switch is required to maintain all unique routings, it's enough to test the output matrix  $O_{all}$  against the control matrix  $M_{all}$ . The search can be described as follows:

- $O_{all}$  and  $M_{all}$  rows are ordered so that all duplicate rows of  $O_{all}$  are clustered in sequence.
- The indexes of the subgroups are saved in order to access each one without needing a complete read of  $O_{all}$  at each call.
- Starting from the elements in the first column of  $V_{all}$ , i.e. the state of the first switch, the algorithm searches if in each subgroup at least one row vector of  $V_{all}$  has the same value, i.e. there exist a complete routing with the first switch always set in the active/passive state.
- If this search is successful, the switching element can be replaced by an hard-wired crossing, or removed and substituted by two straight-forward links.
- The search is repeated for each column index.

The switches discovered in the search are flagged depending on the type of substitution supported (removal, permanent crossing or both).

It's important to highlight that this procedure is not guaranteed to produce a network with the absolute minimum number of switches: as previously stated ad-hoc solutions may exist. This search allows to reduce the number of switching elements for a network configured under the recursive Beneš structure.

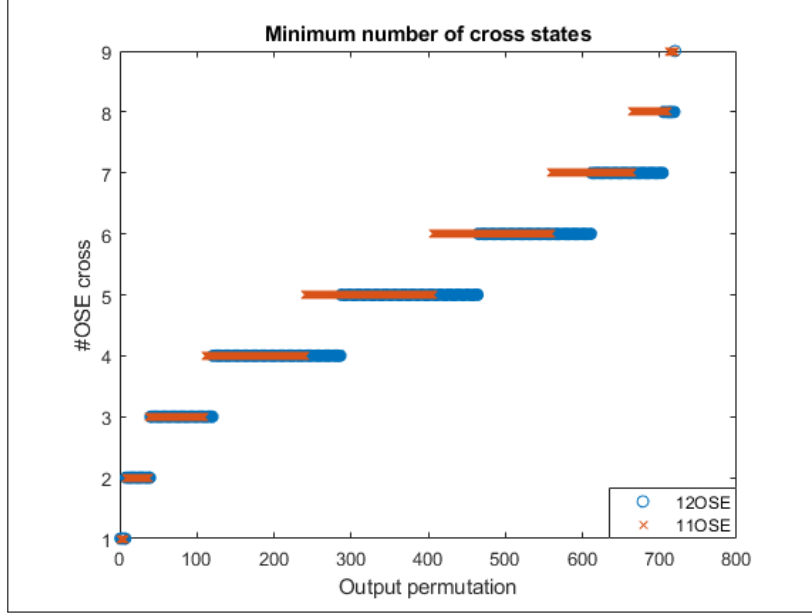


**Figure 48:** 6x6 Beneš configuration: highlight on redundant elements

**Fig. 48** shows an example of the results obtained through this procedure, for a  $6 \times 6$  network: the grey elements are redundant and can be removed by inserting a hard-wired bar state (parallel straightforward links). This search is not able to evaluate co-dependencies between redundant switches, so the results have to be interpreted correctly: after the removal of one element, the search algorithm has to be run again, to check for the presence of another component still removable, in the updated structure.

The  $6 \times 6$  network was proven to be still rearrangeable non-blocking even after the removal of one of the grey elements, through evaluation of the complete, updated, set.

This is a clear example that the recursive procedure does not guarantee optimality for the number of switches implemented.



**Figure 49:** Comparison between the number of active switches (cross state) for each routing in the optimized vs unoptimized case

In order to better understand the trade-off between the number of switching elements and the minimization of the active switches (cross state), the results for the  $6 \times 6$  are shown in **Fig. 49**, as a case study for the possible effects of this procedure. The graph shows the minimum number of switches in the cross state required to route a given unique output permutation.

The effect of the removal of one of the elements is clear: given the reduction of the redundancy over the network, an higher number of elements must be switched in the crossed state to allow the routing of the permutations depending on the removed element.

This optimization can be implemented to reduce the number of OSE in the physical device, with a drawback of an higher power dissipation, due to the need to switch more OSE in the cross state for routing certain permutations. Similarly to all the previously shown functionalities, this procedure can be toggled by the user, allowing the simulation of the device under different goals and levels of complexity.

### 3.5 MATLAB: ROUTING

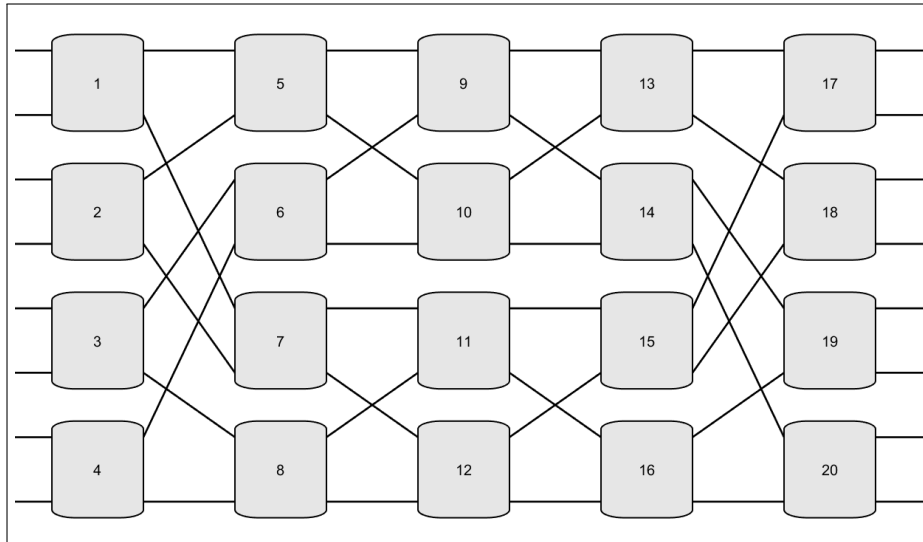
The circuit complexity and scalability issues presented in the previous section highlight the limitation of a brute force approach, and even the general concept of a signal look-up table, in order to implement the control unit for this class of devices. Furthermore, all method introduced relies on the control vector being provided, evaluating the output configuration, while in a practical implementation scenario the output permutation is given, while the problem consists in the evaluation of the driving signals.

In the literature various routing algorithms for the Beneš structures have been proposed and analyzed [9] [10], although constrained to the strict-sense definition, without a clear generalization for a AS-Beneš structure.

In this section a matrix based approach is proposed [11], with the generalization to allow its application to any  $N \times N$  AS-Beneš structure.

The algorithm operation is demonstrated for an  $8 \times 8$  Beneš structure, shown in **Fig. 50**, as to introduce the general concept under a regular symmetric structure, while the generalization for any arbitrary size is introduced afterwards, with the required modifications.

Both algorithms have been implemented in MATLAB, with the capabilities of evaluating a single routing for the required output goal, as well as the full set of equivalent routings for the same output configuration.



**Figure 50:**  $8 \times 8$  Beneš switch

### 3.5.1 Matrix-based routing evaluation

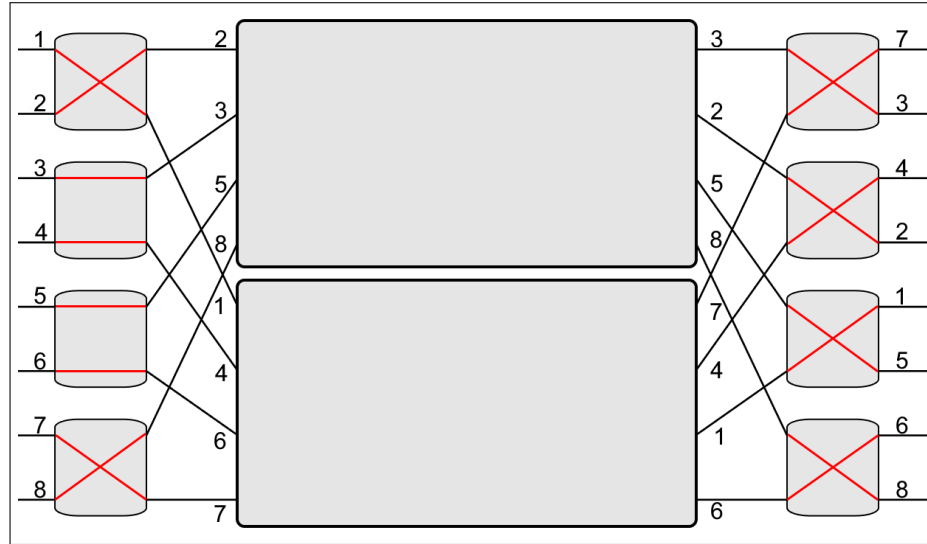
The routing algorithm is based on the interconnection rules defining a general Clos network, and as a consequence, the Beneš switch. In these topologies each element is connected to both successive sub-networks, allowing at each successive stage the possibility of routing the signal through the top or bottom sections, while maintaining a possible path to any element of the symmetric output stage.

The algorithm relies on the multistage structure having horizontal symmetry with respect to the center section, and at each iteration the stages under analysis are selected closer to the central region of the device.

Let's consider a specific output request for the  $8 \times 8$  switch under analysis: the input signals are described as the integer vector  $V_{in} = [1, 2, 3, 4, 5, 6, 7, 8]$ , while the output signals required at end of the last stage are  $V_{out} = [7, 3, 4, 2, 1, 5, 6, 8]$ .

The steps of the algorithm are as follows:

- Considering the  $8 \times 8$  device has  $N_{stages} = 5$  the first and last stages are selected as  $S_{in} = 1$  and  $S_{out} = 5$ .
- Generate an empty  $N_{sw} \times N_{sw}$  matrix, where  $N_{sw} = 4$  is the number of elements in the stage, with each row representing a switching element in  $S_{in}$  and each column a switching element in  $S_{out}$ .
- Starting from the input signals of the first element in  $S_{in}$  evaluate the target switches in the output stage  $S_{out}$ .
- Insert "1" or "0" in the first row of the matrix at the columns of the target switches, with "1" representing a path through the bottom sub-network, and "0" through the top sub-network. Each row can contain only one instance of "1" or "0".
- Iterate through all the switching elements of  $S_{in}$  (rows), filling the matrix with the rule of the previous steps.
- Once all rows have been set, verify that no repetitions of elements occur in any rows or column. If one of the column contains two instance of "1" or "0", flip the content of one of the row causing the conflict.
- Once no conflict is present in the matrix, evaluates the new  $V_{in}$ , after the propagation through  $S_{in}$ , and  $V_{out}$ , before the propagation through  $S_{out}$ .
- Select the new stages  $S_{in} = S_{in} + 1$  and  $S_{out} = S_{out} - 1$  and iterate until completion.



		Output Stage			
		1	2	3	4
Input Stage	1	-	$1 \rightarrow 0$ (2)	$0 \rightarrow 1$ (1)	-
	2	0 (3)	1 (4)	-	-
	3	-	-	0 (5)	1 (6)
	4	$0 \rightarrow 1$ (7)	-	-	$1 \rightarrow 0$ (8)

**Figure 51:** 8×8 Beneš routing evaluation: first stage

A visual representation of the first iteration of the procedure is shown in **Fig. 51**, together with the full routing matrix, already balanced. In the matrix, the arrows mean that the value initially set has been swapped during the balancing phase, in order to remove a conflict. The signal routed through that cell is shown in brackets in order to improve clarity, as well as keeping track of the switching operation.

The meaning behind this process is quite straightforward. Each signal from a switch in the input stage can be routed through one of the following sub-networks, although this imposes that the other signal originating from the switching element must traverse the opposite network: this sets the constraint on the rows content.

The columns operate on the same principle, as they indicate the sub-network origin for each signal reaching the final stage. By ensuring that both rows and columns are balanced, while assuming that the sub-networks are able to route the signals correctly, the routing evaluation can be compartmentalized into a much smaller problem, tackling one layer of the network at a time.

The figure shows an example of the signal paths ensuring the specified goal, keeping

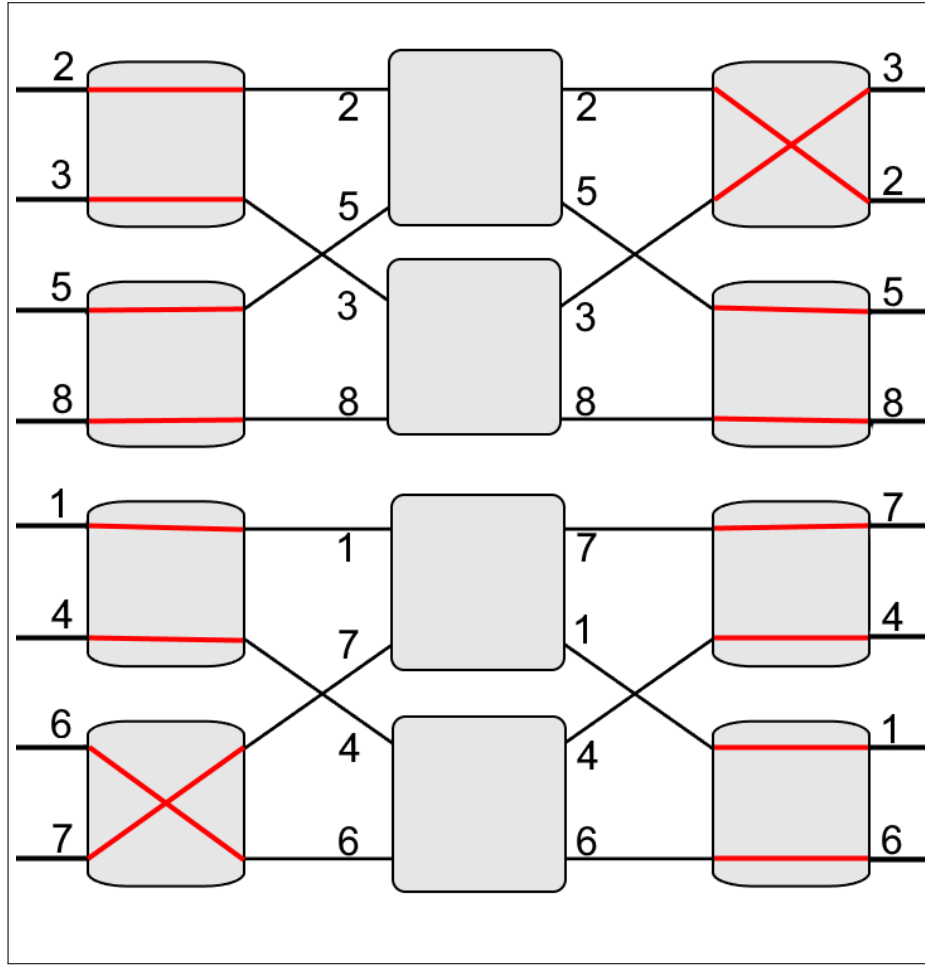
in mind that different balanced equivalent matrices may be achievable, so multiple paths can be obtained through this method, by choosing different rows to resolve the columns conflicts.

With the updated  $V_{in} = [2, 3, 5, 8, 1, 4, 6, 7]$  and  $V_{out} = [3, 2, 5, 8, 7, 4, 1, 6]$ , the second iteration can be carried out, considering the second layer, with  $S_{in} = 2$  and  $S_{out} = 4$ . At this point the  $N_{sw} \times N_{sw}$  matrix can be generated again, although observing the structure depicted in **Fig. 52**, it's evident that the matrix dimension can be simplified.

Given that the two top elements of both  $S_{in}$  and  $S_{in}$  cannot interact with signals from the bottom two elements, the switching matrix can be formulated as two sub-matrices of size  $\frac{N_{sw}}{2} \times \frac{N_{sw}}{2}$ .

This general rule is applied at each iteration of the algorithm, with the matrix size of the previous cycle halved.

The top or bottom path is chosen in a similar way to the previous step, considering the target output switch for each signal. Observing the top two switches of the stage  $S_{in}$ , a new notation is introduced: as the target of both signals is the same switch in the output stage  $S_{out}$ , only one cell is occupied in both the rows and the corresponding columns. This also highlights the two equivalent paths available, with signal "2" routed through the top network or the bottom network. This choice is arbitrary, although coherence must be enforced on the output stage, such that the signals are correctly routed.



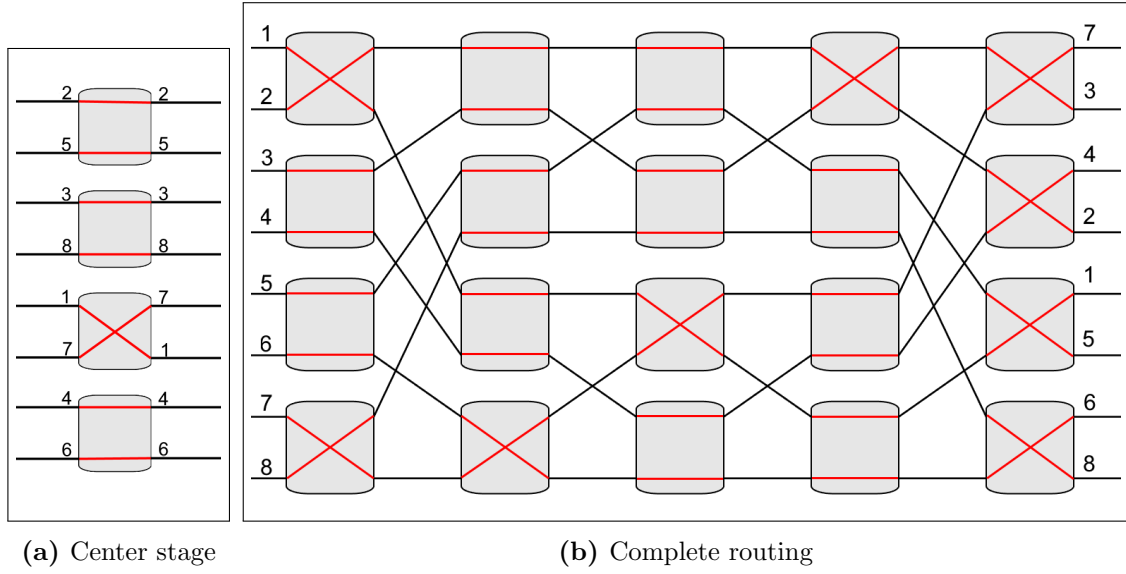
		$S_2$	
		1	2
$S_1$	1	0-1 (2-3)	-
	2	-	0-1 (5-8)

		$S_2$	
		3	4
$S_1$	3	1 (4)	0 (1)
	4	1 $\rightarrow$ 0 (7)	0 $\rightarrow$ 1 (6)

**Figure 52:** 8x8 Beneš routing evaluation: second stage

Once the paths in the second layer are routed only the central section must be addressed. There is no requirement for any matrix notation in the evaluation of the state for this layer, as the needed configuration is straightforward, as shown in **Fig. 53b**. The final path evaluated for the given target goal is shown in **Fig. 53b**.





**Figure 53:** 8×8 Beneš routing evaluation: final stage

The vector containing the states of the switches is obtained by analyzing the signals  $V_{in}$  and  $V_{out}$  before and after each switching stage, determining which elements must be set in the active state and which elements are in the default passive state.

This method was expanded to allow the evaluation of the whole set of equivalent routings ensuring the correct target output.

This can be achieved without major modifications to the overall algorithm, by simply generating all possible balanced matrices for the target switches of the configuration under analysis.

After the generation of all the matrices describing the first layer, the algorithm is iterated for each remaining stage, and the process is carried on until all possible alternatives are exhausted.

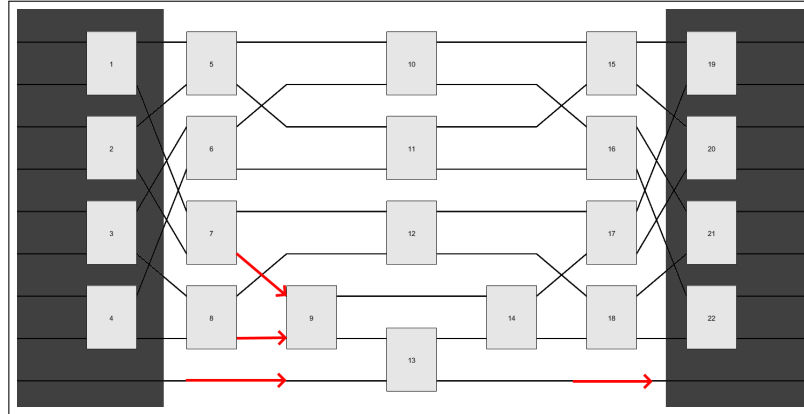
This is handled through a recursive script, as generally, each matrix of a previous layer leads to a completely different set of matrices available in the layer under analysis.

This in turn can cause a similar problem encountered with the brute-force methods, as the scalability of the device increases in an NP fashion the number of alternative paths routing the same output combination.

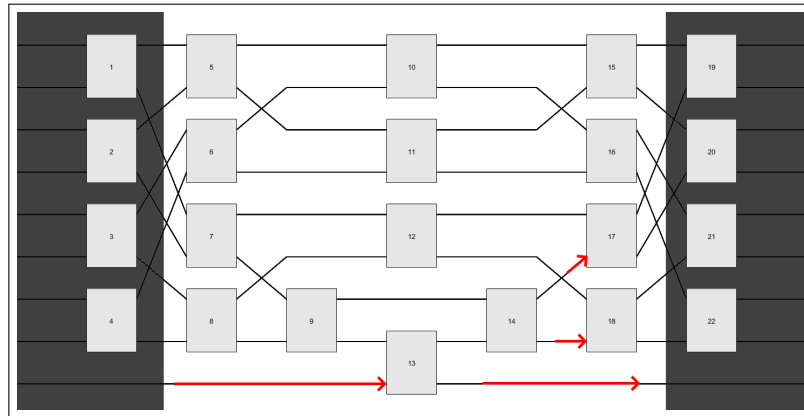
To this end, the script can be limited to obtain a user-defined number of alternative routings, if the size of the equivalent set is above the threshold.

### 3.5.2 AS-Beneš generalization

Considering that the project targets and implements the automatic generation and analysis of AS-Beneš networks, the constraint on the routing algorithm is not coherent with the generalized approach. To this end the routing algorithm and scripts have been modified to handle the evaluation of a single solution, as well as the whole set of alternative paths, for a given target output in AS-Beneš structures.



(a) Element to wire request



(b) Wire to element request

**Figure 54:** AS-Beneš routing exceptions

From a conceptual point of view the approach is identical, exploiting the top-bottom recursive networks through a matrix-based notation, while evaluating each layer as an independent structure, under the same compartmentalization paradigm. The main difference in the arbitrary size case is the presence of "uncoupled" signals and uneven sub-networks, which contradicts the simple matrix notation based on the constant number of switches per stage of a traditional Beneš network.

The presence of these wires must be taken into account in the switching matrix: considering only the switching elements of each stage, fundamental information regarding the origin or destination of a signal may be lost.

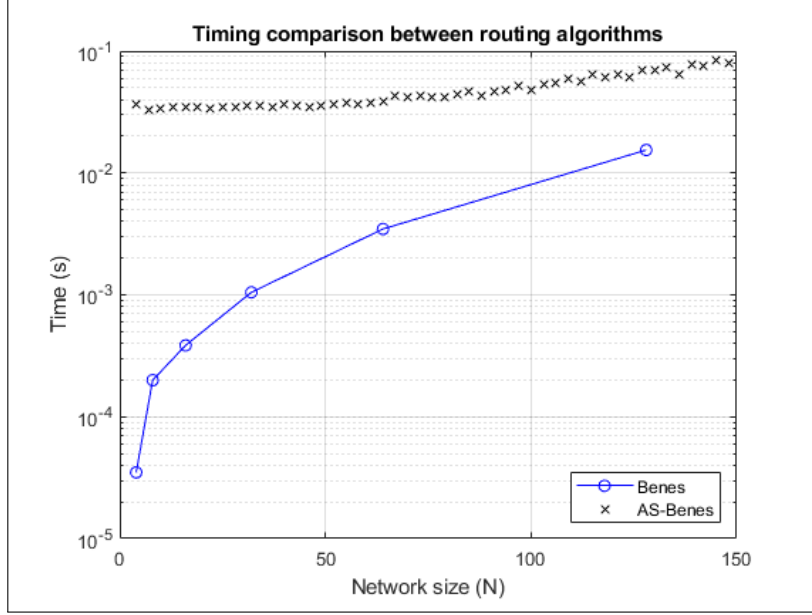
This requires the addition of special rows and columns, representing the loose links, which are constrained as to contain a single state. Considering the notation used in the previous sections, these additional vectors may be ignored in the case of a signal originating from a wire element in the input stage  $S_{in}$  and directed to a wire element in the output stage  $S_{out}$ .

On the contrary the effect of these elements must be taken into account in two specific scenarios, which are depicted in **Fig. 54**. Due to the design decision of inserting the uncoupled elements always as the last element of the bottom network, the exceptions to the standard rules established in the routing algorithms are easily imposed.

Considering a signal originating from an ingress switch  $i$  and targeted to an output link  $j$  (**Fig. 54a**), its path must travel always in the bottom network, which is represented by a "1" in the cell  $(i, j)$  of the respective switching matrix, for which the  $i^{th}$  row represent a standard switch element. Similarly, as shown in **Fig. 54b**, the signal originating on the wire element  $i$  and targeting an egress switch  $j$ , imposes the value "1" in the matrix cell  $(i, j)$ , this time with the  $j^{th}$  column representing a standard switching element, as the signal cannot be received from the top network. This lead to a modification in the conflict resolution algorithm, which cannot swap the conflicting row values, as this would represent an unfeasible configuration.

As a consequence, the routing algorithm for the AS-Beneš case can be seen as a constrained version of the original algorithm, which must take into account the asymmetry introduced by the odd sized recursive blocks.

Similarly to the Beneš case, a script was also created to allow the evaluation of all the equivalent routings for a given output goal, with the same recursive structure considered in the strict-sense case.



**Figure 55:** AS-Beneš vs Beneš routing cost: time required to find an available path

Due to the additional steps required to ensure the correct routing, the time cost of the procedure is expected to be higher with respect to the simpler symmetric case. This difference is shown in **Fig. 55**, which highlights the additional complexity of the algorithm, introducing a severe increase in the order of magnitude of the time required to evaluate a complete path.

### 3.6 ADVANCED ROUTING CONTROL: MACHINE-LEARNING APPROACH

Although the routing algorithm can be used for the evaluation of the complete set of signals paths for the required configuration, this approach still suffers from a fundamental scalability issue.

In the simulation environment, the use of the deterministic routing algorithm is fundamental to test and verify the behavior and performances of the switch, although this solution cannot be easily implemented in a realistic performance-aware control of the device.

Under the reasonable assumption of dependency between the transfer function, or signal degradation, and the state of the switches along the transmission paths, it's clear that an optimal solution exist in this set, which maximizes Quality of Transmission (QoT).

### 3.6 Advanced routing control: machine-learning approach

---

$N$	$N_{SW}$	Complete set ( $\approx$ )	Unique combinations ( $\approx$ )	Equivalent paths ( $\approx$ )
8	20	$1 \cdot 10^6$	40320	26
10	26	$67 \cdot 10^6$	$3.6 \cdot 10^6$	18
12	36	$68.7 \cdot 10^9$	$479 \cdot 10^6$	143

**Table 3:** Solution set sizes and parameters

**Table 3** shows the combinatoric size for three instances of Beneš network. For the network size  $N = 8$ , the number of unique combinations is reasonably small to still allow a look-up table implementation of the routing space, although the meaning of the complete set size must be clearly understood.

The generation of a control unit of this kind would require the simulation at the system level of  $10^6$  cases, as to evaluate the best achievable QoT for each configuration. This is clearly an unfeasible task in terms of computational time, as a realistic system simulation is intensive.

The implemented routing algorithm, namely the fast computation of the available paths, could be considered a real-time control strategy, if the trade-off on the optimality is accepted. In this control scenario an assumption can be made on the device performance, selecting the routing through a fixed metric, like the minimization of the active switch states, or other constraint based on the accessible data for the available paths.

A second solution, which is becoming an ever growing topic of interest in current-day research landscape, is abandoning the deterministic algorithms in favour of more advanced stochastic approaches, namely machine-learning (ML) control agents.

The ML-based strategies are outside the design scope of the thesis project, as such the focus is placed only on the requirements for the applications of such methods.

The underlying basis of these approaches is the reliance on large data-sets which are used to train ML-based algorithms to solve complex problems, without requiring the evaluation of the complete set of solution for the problem under analysis.

The virtualization and abstraction created for the Beneš networks allows such operation: through a mathematical model of the network, large sets of configurations can be evaluated, as to allow the training of stochastic ML agents.

The ML-based approach has one key advantage with respect to the deterministic algorithms, especially concerning the generalized and expandable structure of the developed design environment.

While expansions of the OSE templates and models, the material analysis, as well as the network topology is supported under the compartmentalized paradigm followed in the project, the routing algorithm proposed in the previous section can only be applied to Beneš topologies. The implementation of a different architecture,

such as the Spanke-Beneš, or Banyan-based switching structures, would require the implementation of a topology-depended control script, which is not applicable to custom-structure or optimized multi-stage networks, due to the absence of specific routing methodologies.

The general topology-agnostic algorithms present in the literature do not provide a scalable solution for this classes of networks, as the problem size exhibits a non-polynomial growth.

On the contrary ML methods, due to their stochastic nature, do not require specific network description models concerning the structure under analysis, as the training procedure is relying on data-sets obtained from the device abstraction, as such the framework is intrinsically topology-independant.

The preliminary results of such application are presented in [12], highlighting the accuracy in the prediction of the control states, after training on data obtained through the network model proposed in the previous sections.

This offers another insight into the usefulness and importance of vertical design procedures and component abstractions, especially in the optical communication environment. The compartmentalization of the developed codes allow high reusability in performing analysis which are not directly involved with the simulation and evaluation of the component performances, as already introduced with the network analysis carried out through the brute-force methodology.

The Beneš case-study illustrates the benefits of working in a multi-layer aware workflow, which allows the designer to tackle aspects and design problems that might be ignored when focusing on the optimization of a single design-step.

Considering the modern application of the software-defined networking paradigm (SDN) the device design requires insight into the operation and management of today's infrastructure[13]. Under this application, the routing control and configuration of the device must be compatible with customized APIs and scripts, as to avoid a closed proprietary control system, allowing management flexibility for the network operator.

The virtualization and abstraction of the component, as well as its control method, is fundamental for the implementation of these modern paradigm, which belong to an active and topical area of research for the optimization of optical communication systems.

## 3.7 OPTSIM CIRCUIT IMPLEMENTATION

Considering the proposed workflow for the design and simulation of the device, the steps already described are the following.

1. After the topological selection for the template of the OSE, the length, coupling coefficient and control voltages are automatically evaluated through analytical formulas and simulations, unless specified by the user.
2. The Optsim circuit template for the OSE is compiled with the evaluated values, and the compound component model is generated.
3. The Beneš topology is generated for the requested size  $N$  as a virtual data-structure, allowing the mathematical evaluation of the routing data-set, or the generation of the control combinations for the target output permutations.

The next step in the process is the generation of an Optsim circuit for the network, using the previously compiled OSE schematic as the fundamental switch.

This is carried out through a MATLAB script, which generates the Beneš switch mesh including the simulation-ready compound components.

This can be achieved thanks to the format used for the description of the Optsim circuits ".moml" files, which can be read and modified as "xml" files.

The Optsim OSE file is used as a template and copied inside a new compound structure, with parametrization of the driving voltages, such that the switching operation can be controlled at the system level. The wiring for the interconnections are evaluated through a similar approach to the graph-based network description, due to the overall similarity in the structure type.

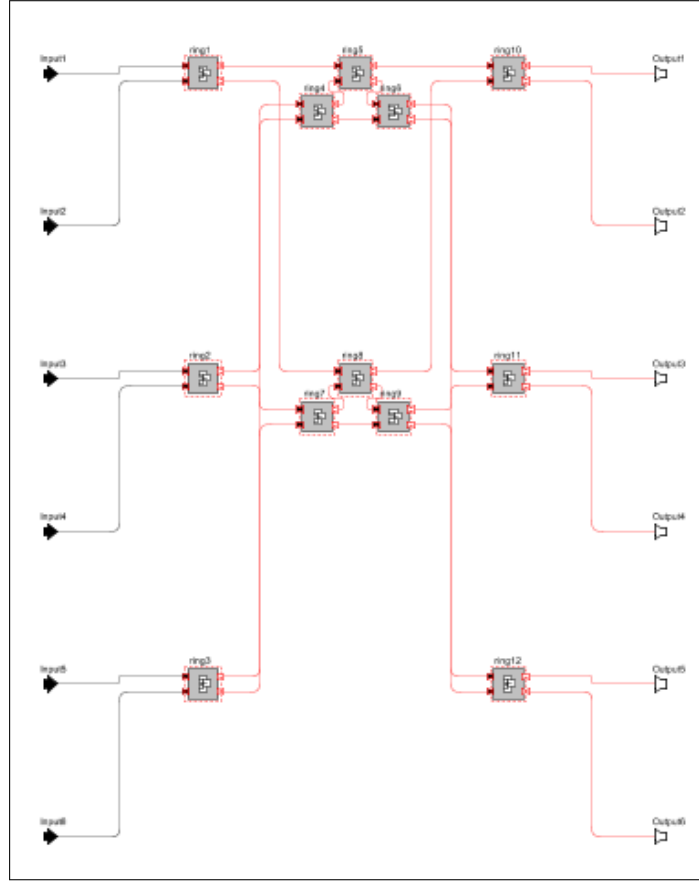
An important factor that must be taken into account is the generation of the OSEs with respect to the approach used for the generation of the solution data-set.

Due to the matrix-based approach acting on the stages as a whole, while the graph-oriented description treats the exploration of the structure under a hybrid DFS (**Section 3.3.1**), the numbering of the switches may be incoherent if two approaches are mixed: this would lead to an error in the evaluation of the look-up table or the routings, as the controls are applied to the incorrect switches.

Due to the design choice of implementing the matrix-based approach for the definition of the routing algorithm, the same order of the OSEs must be ensured, which is automatically carried out by the external scripts.

The final circuit, shown for a  $6 \times 6$  switch, is represented in **Fig. 56**. The links between the elements are treated as logical instead of actual waveguide interconnections, so they do not introduce any phase distortion or loss, as if the OSE were directly connected to each other.

This compound component can be placed in more complex simulation projects without requiring any compatibility change, as it acts as a stand-alone device defined by



**Figure 56:** Example of a 6x6 Beneš schematic automatically generated by the script

the structures designed in **Section 2.2**.

This automatic generation is compatible with the three  $2 \times 2$  switches model studied, and more in general can be expanded to any model using a similar port structure. The change of the fundamental OSE is done by targeting a different template for the generation of the circuit, as such, the generalized approach is maintained in the construction of the complete device.

The simulations results and environments are shown in the following section, while, concerning the generation of the  $N \times N$  device, **Section 2** and **Section 3** include the complete workflow for the design of the general AS-Beneš switch, as for the goal of the project.



## 4 SYSTEM LEVEL SIMULATIONS

---

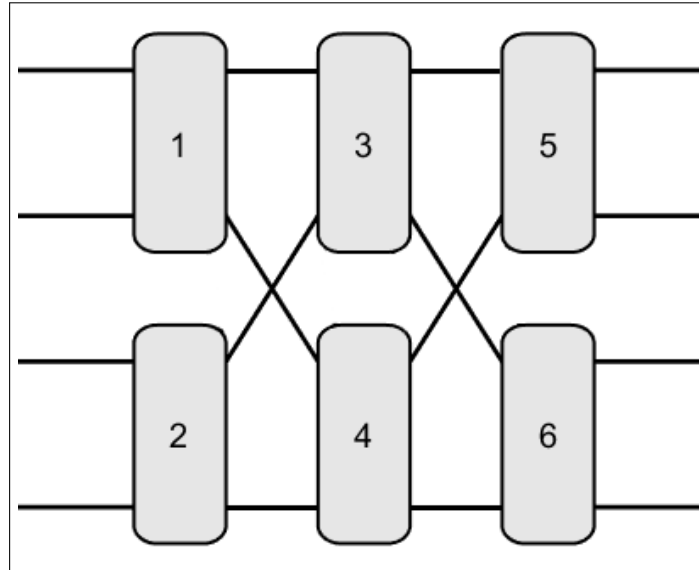
The Optsim Circuit schematic generated in the previous section can be simulated in a more complex environment, in order to perform a similar analysis to the one shown in **Section 2** for the optical switching elements.

Although the frequency response of the OSEs can be fully characterized for the chosen parameters, the Beneš switch simulation has the advantage of highlighting the filtering effects over multiple cascaded elements, which can quickly lead to degradation of performances, through signals cross-talk and pass-band attenuation.

The simulation of the quality of transmission (QoT) has been evaluated in different simulation scenarios, which offer a trade-off between the computational cost requirements and the degree of accuracy and usefulness of the results.

In this section three main methods of analysis are proposed, in order to characterize the switches performance based on the implemented OSE, as well as the control state piloting the network.

For the purpose of improving the readability of the results, a  $4 \times 4$  Beneš (**Fig. 57**) structure has been chosen as the simulation template, although the process can be carried out for any  $N \times N$  AS-Beneš, due to the modularity and generalized approach used in the device generation scripts, which allow the generation and simulation of any arbitrary sized device.

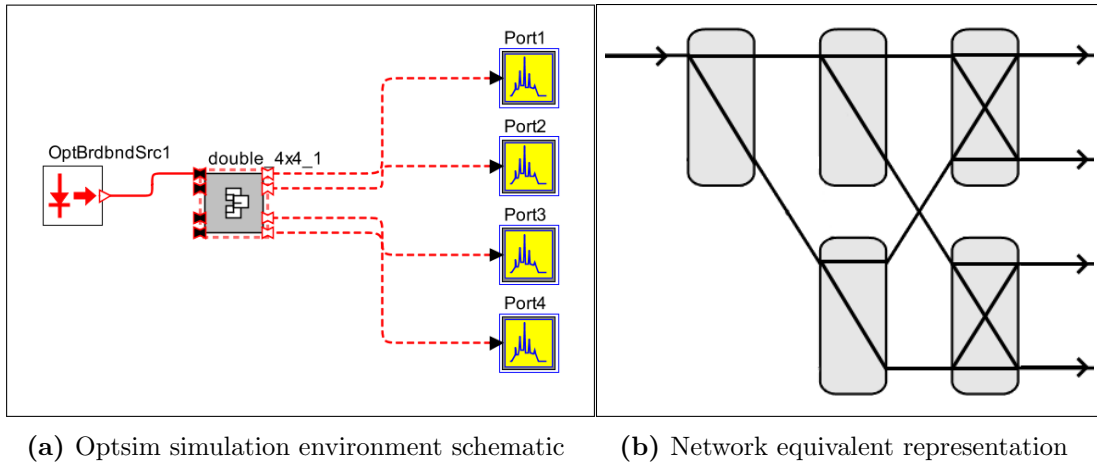


**Figure 57:**  $4 \times 4$  Beneš configuration

## 4.1 SINGLE INPUT FREQUENCY RESPONSE

One of the faster and basic analysis that can be carried out is the evaluation of the frequency response of the device, performed in a similar fashion to the OSEs response simulations.

Using the previously obtained compound component for the Beneš, a new schematic is created, as shown in **Fig. 58a**, with a broadband flat optical generator used as input in the first port. At the output of the network, for each port, a signal spectrum analyzer is connected, in order to plot and store the incoming signal. Considering that only one input port is active, the underlying Beneš network can be represented as shown in **Fig. 58b**, by removing the unfeasible paths that cannot be traversed by the input signal.

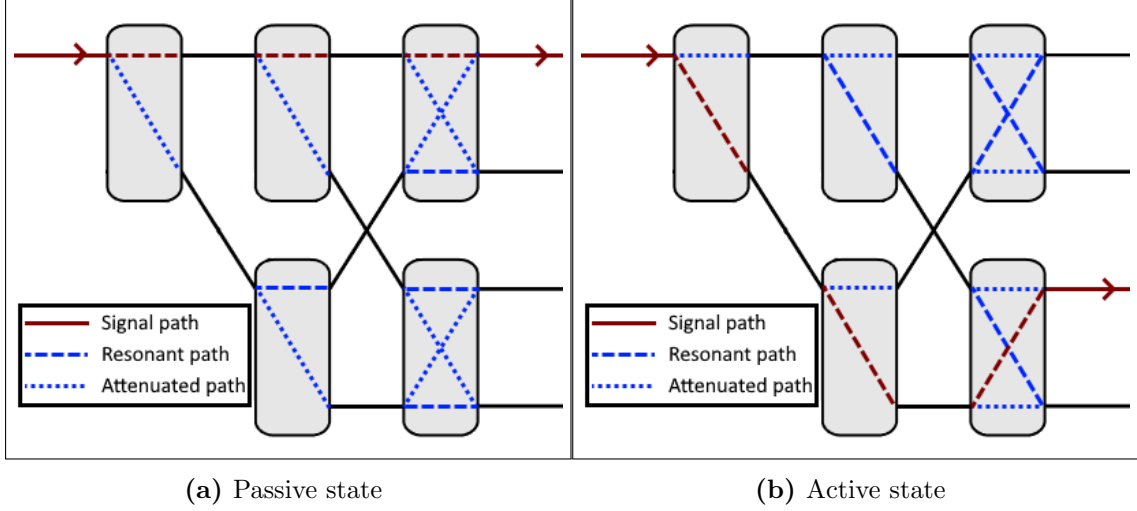


**Figure 58:** Single port response model - Flat spectral input

The target of the simulation is the analysis on the effect of the OSEs cascade, which should increase the global filter penalty with respect to the single element case, while ideally propagating the resonance frequencies, representing the channels, to the correct output port.

Due to the response asymmetry of the  $2 \times 2$  switching elements in the *Bar* and *Cross* states, as highlighted in **Section 2**, the two limiting cases correspond to:

- Passive state (**Fig. 59a**): all switching elements are in the default *Bar* state, with  $V_{bias} = 0$ , leading to the channels routing  $S_{in} : [1\ 2\ 3\ 4] \rightarrow S_{out} = [1\ 2\ 3\ 4]$ .
- Active state (**Fig. 59b**): all switching elements are in the *Cross* state, with  $V_{bias} = V_{on}$ , leading to the channels routing  $S_{in} : [1\ 2\ 3\ 4] \rightarrow S_{out} = [3\ 4\ 1\ 2]$ .



**Figure 59:** Network equivalent model for the two control configurations  
Single input scenario (Flat spectral signal)

The two simulation scenarios shown in **Fig. 59** highlight the effects present in any multistage switching network. The control signals applied to the OSEs create a set of attenuation and propagation paths, generating a complex mesh of interconnections.

While in the analysis of the elementary switch, the attenuation of the transmission band can be easily measured in the non-target output port, in a more complex structure each non-ideal characteristic of the switches combines depending on the control states along the whole device.

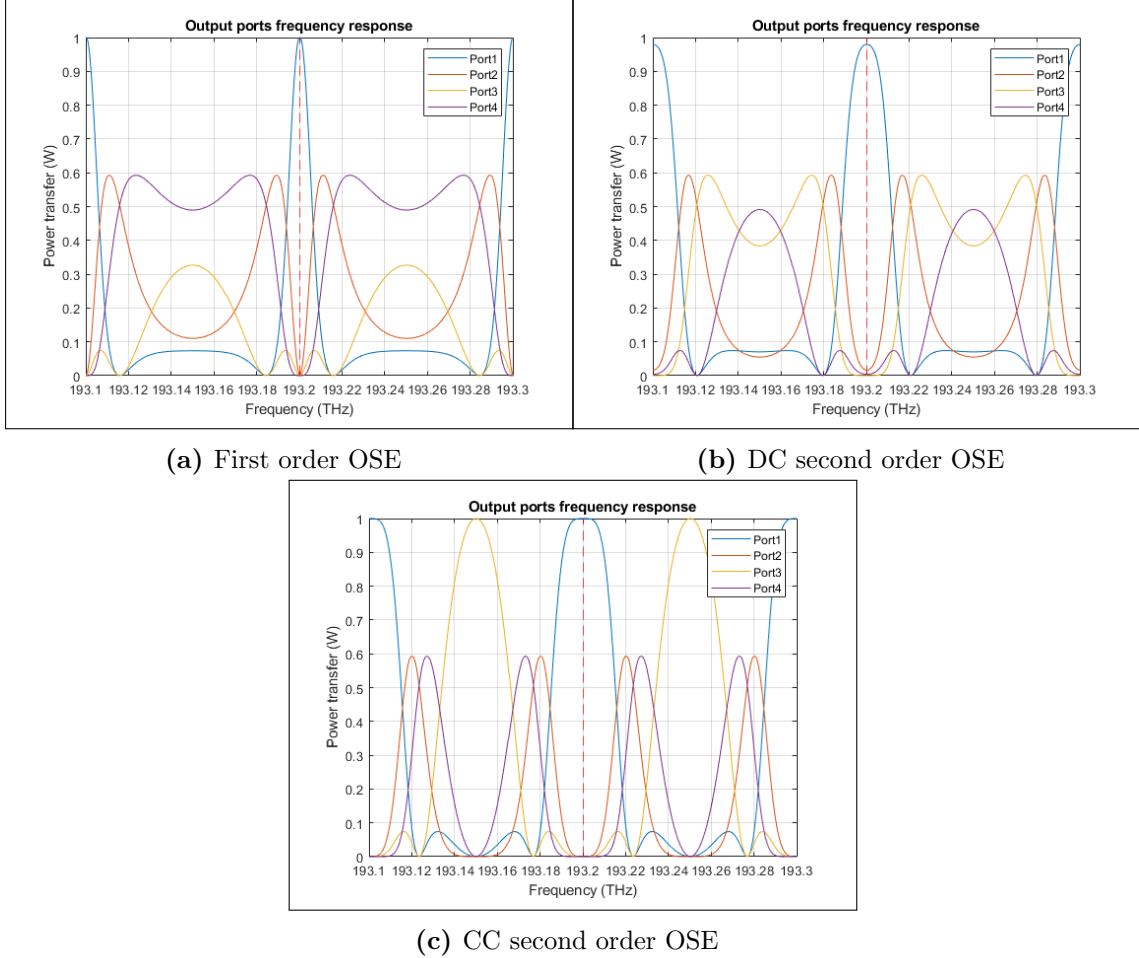
While switching the state of the OSEs leads to a shift in the frequency response of each port, in a multistage device the effect is less straightforward: due to the final response depending on the product and combination of the transfer functions along all paths to the considered port, any change to the states of the elementary switches lead to drastic modifications in the device behavior.

The exception to this general behavior is seen in the active and passive states previously described, as all the components introduce the same shift, leading to a symmetric modification of the device response. Concerning the passive state response, shown in **Fig. 60**, the effect of the  $2 \times 2$  element choice is mainly related to the transmission bandwidth. The implementation with the first order OSE (**Fig. 60a**) leads to a severe reduction of the transmission band, while the more complex device implementations (**Fig. 60b-Fig. 60c**) allow for a larger bandwidth, as expected from the analysis carried out in **Section 2**.

It's important to remark that the stop-band response cannot be directly analyzed with this simulation method, as it doesn't correctly represents the general effect of

#### 4.1 Single input frequency response

the switching states, as previously described. Nonetheless it should be clear that the responses of the non-target ports in the transmission bandwidth quantify the amount of crosstalk present in the system.



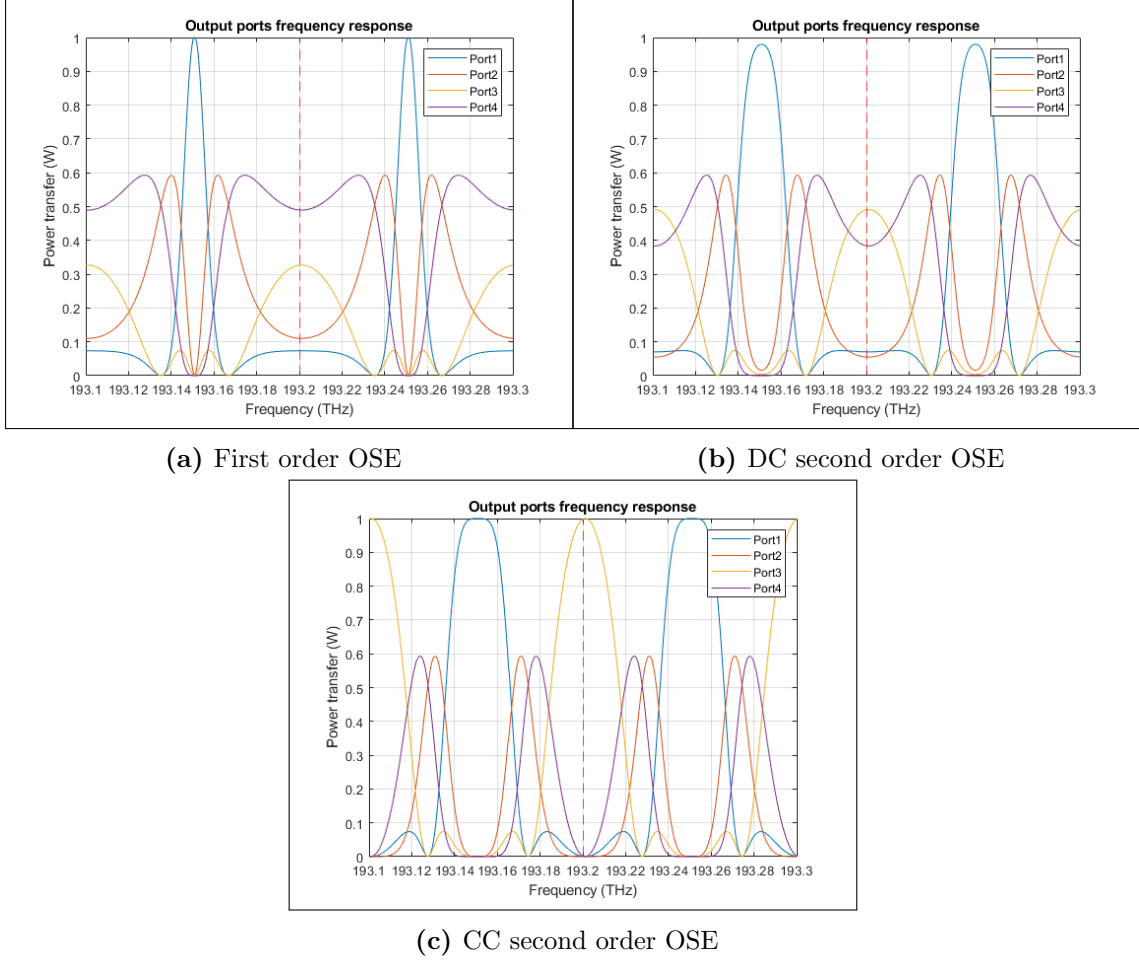
**Figure 60:** 4×4 Beneš output response in passive state - Flat spectral input

In the active state (**Fig. 61**) the behavior of the first order and directly coupled second order OSE leads to a severe impairment of the QoT. The transmitted signal should reach the third output port of the device, while the simulations clearly show that the input power is transmitted mainly to *Port*<sub>4</sub> in the first order OSE (**Fig. 61a**), and almost equally divided with respect to the target port, as seen in the DC-second order OSE (**Fig. 61b**).

Unfortunately, this is expected, as the simulations of the OSEs showed already the large attenuation of the pass-band in the active state.

The performance of the crossing-coupled second order element is instead comparable

#### 4.1 Single input frequency response



**Figure 61:**  $4 \times 4$  Beneš output response in active state - Flat spectral input

to the passive state case (**Fig. 60c-Fig. 61c**), as the response of the  $2 \times 2$  switch is largely symmetrical, with negligible attenuation in both transmission states.

Overall, this method of analysis is problematic for two main reasons, although sufficient in order to verify the approximated filtering penalty for a given input port.

In a more realistic simulation scenario the effect of the multiple signals must be taken into account to correctly quantify the amount of crosstalk and filtering for each routing solution.

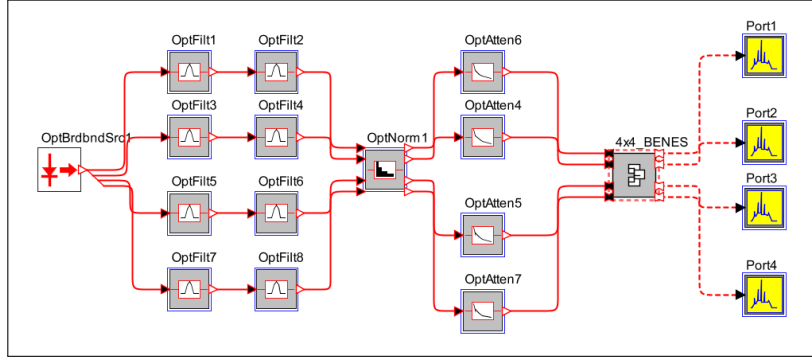
By considering only one input signal, with a wide frequency range as to cover all channels, the result is a periodic response, which actually fails to describe the different penalties of the path encountered by each channel.

In order to address this concern, a more in-depth simulation scenario has been devised, as to extract a more useful characterization of the device performance.

## 4.2 FILTERED CHANNELS TRANSMISSION

In order to obtain a more accurate representation of the OSE performances, in the  $4 \times 4$  Beneš switch under analysis, the simulation need to take into account the propagation of all four channels.

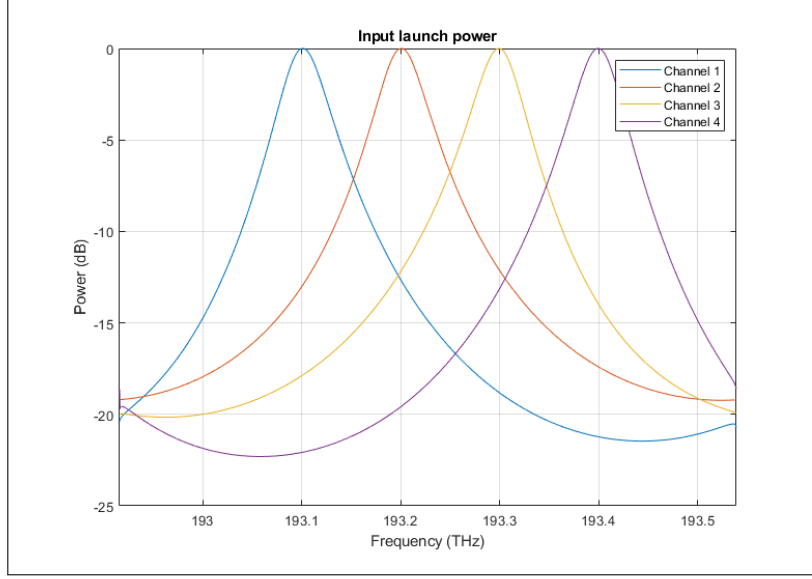
The new simulation environment is depicted in **Fig. 62**: the broadband source of the previous simulation is used to generate four filtered channels, centered around the resonance frequencies  $f_{ch} = [193.1 \ 193.2 \ 193.3 \ 193.4]$  THz. The optical power normalizer and attenuators are introduced to control the peak of each channel independently, compensating for some non-idealities introduced by the filtering blocks. The input signals of the Beneš structure are depicted in **Fig. 63**: the bandwidth of each channel is considerably smaller with respect to the target FSR, nonetheless the target of this simulation is the evaluation of the band distortion and side-channels propagation, which **Fig. 61** shows being already present close to the channel center frequency.



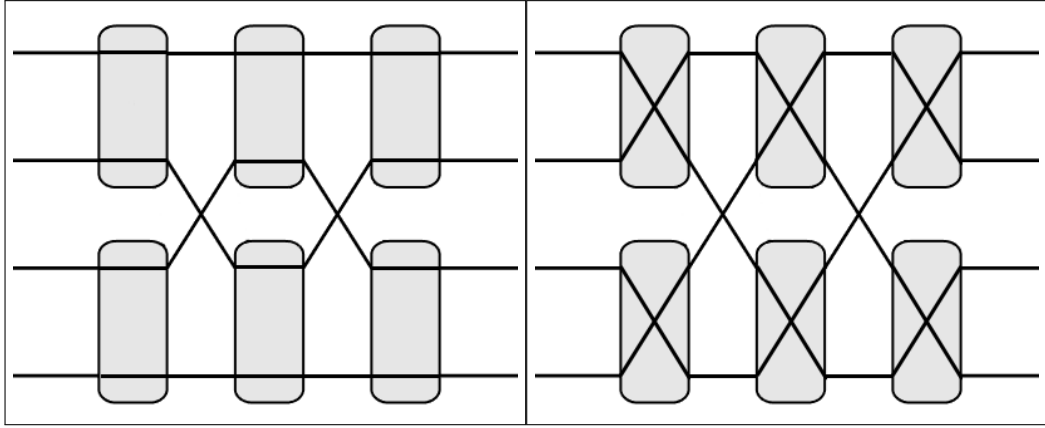
**Figure 62:** Filtered channel simulation environment - Optsim schematic

The analysis has been carried out for the active and passive cases, seen in the previous section, as well as for a mixed routing, which represent a more typical operational state for a multistage switch, with some elements in the *Cross* state, while others in the idle *Bar* state.

The main difference between this simulation environment, with respect to the single-input flat response, is the improved readability of the measured outputs: while in the previous section the results highlights a periodic transfer function, overlapping the response of all four channels, in this simplified transmission model each figure represents the power transfer from all the input ports to the measured output.



**Figure 63:** Filtered channels simulation - Launch power

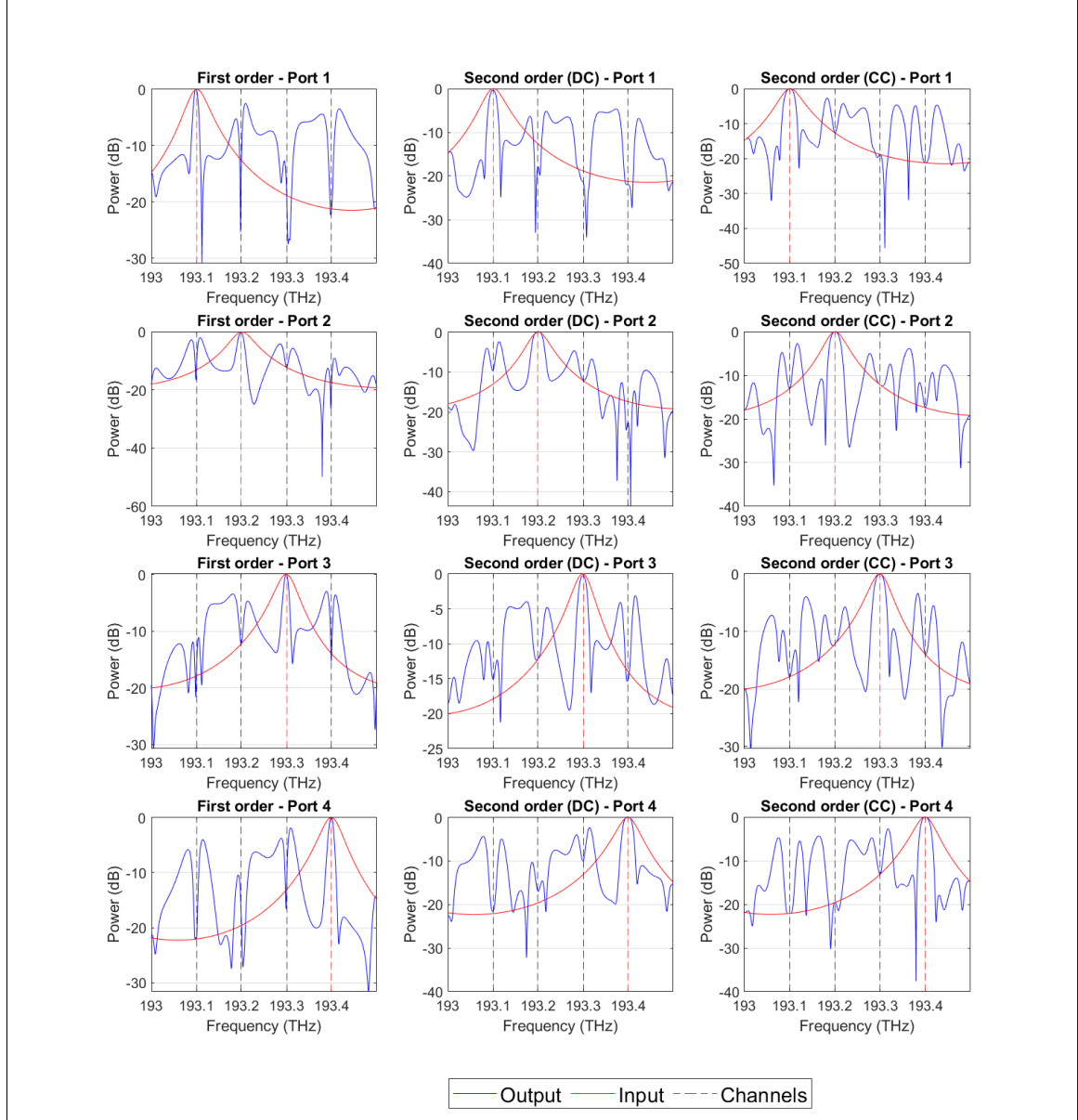


**(a)** 4×4 Beneš configuration: passive state    **(b)** 4×4 Beneš configuration: active state

**Figure 64:** Network equivalent model for the two control configurations  
Complete input set (filtered channels)

### 4.2.1 Passive state

The network model under analysis is depicted in **Fig. 64a**: no simplification like **Fig. 59a** can be applied as all four input ports are used to insert a different channel in the device. The expected output permutation is equal to the one analyzed in the previous environment ( $S_{in} = [1\ 2\ 3\ 4] \rightarrow S_{out} = [1\ 2\ 3\ 4]$ ) with  $V_{bias} = 0$  for all



**Figure 65:** Comparison between device implementations - Passive state  
 $(S_{in} = [1\ 2\ 3\ 4] \rightarrow S_{out} = [1\ 2\ 3\ 4])$



switching elements. The results for all three devices are depicted in **Fig. 65**. The performance of each OSE topology is in line with the expected results and can be characterized as follows:

- First order OSE: for this implementation the filtering penalty is the highest, with a reduction of the bandwidth to a narrow slice of the original transmitted signal. The effect is also noticeable on the blocked channels of each port, where the filtered bandwidth absent from the target port is present, distributed in all other output ports.
- Second order DC OSE: a larger portion of the signal bandwidth is transmitted, although with the attenuation of some of the peaks, visible in  $Port_1$ .
- Second order CC OSE: similarly to the previous section results, this configuration leads to the best performance, as no attenuation of the peaks is present, as well as the maximum transmitted bandwidth with respect to the other cases.

Overall the results are in line with the flat frequency response test, although with a clearer characterization of the crosstalk present at each output port.

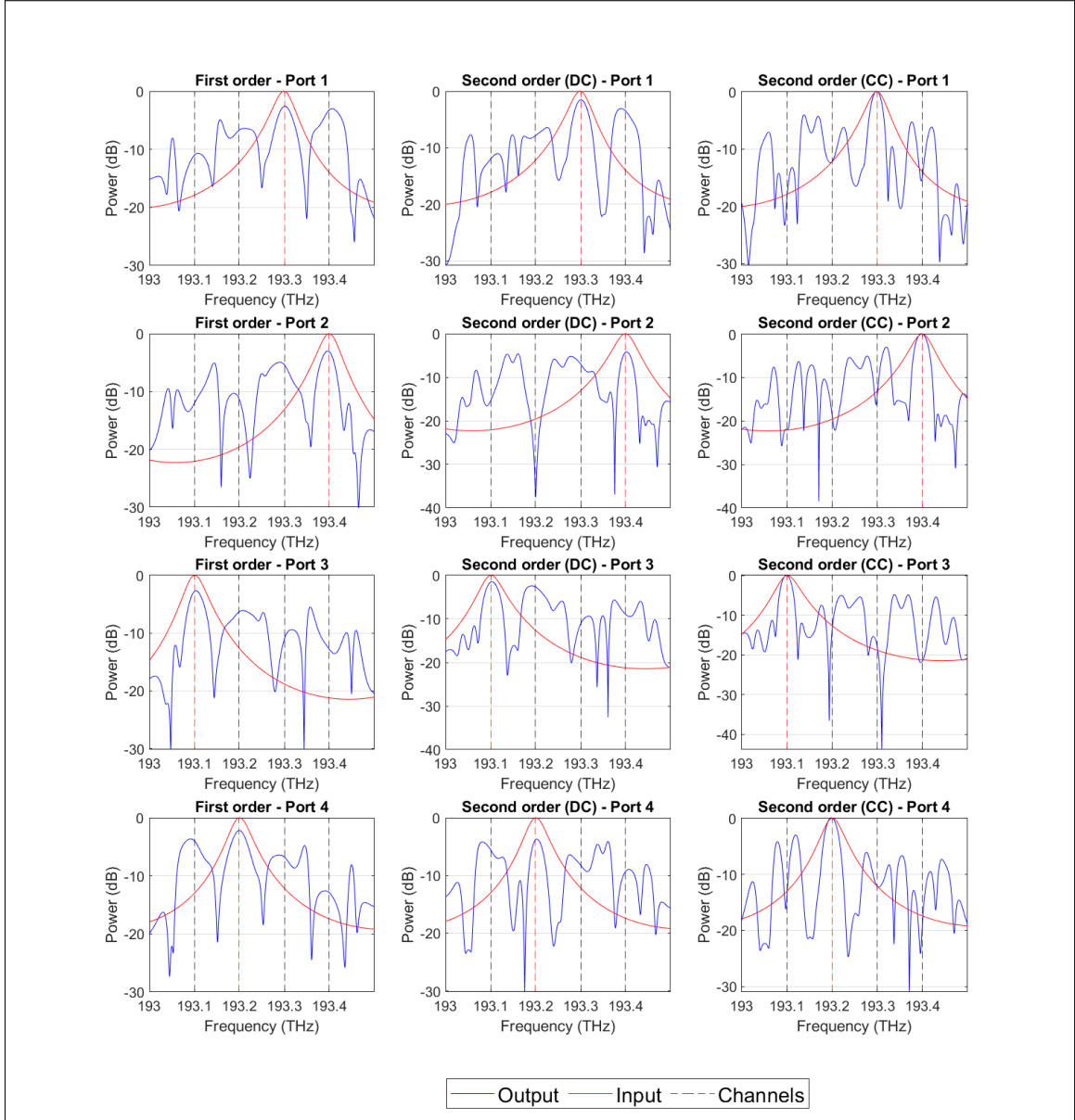
### 4.2.2 Active state

The active state caused a strong degradation of the performances for the flat response analysis, in particular for the first and second order DC devices, as such a similar result is expected in the filtered signals propagation.

The network paths are shown in **Fig. 64b**, with the output goal ( $S_{in} = [1\ 2\ 3\ 4] \rightarrow S_{out} = [3\ 4\ 1\ 2]$ ) and  $V_{bias} = V_{on}$  for all switching elements.

Considering the first two device implementations, the difference with respect to the previous case can be easily understood in reference to the responses shown in **Section 2-Fig. 19**: in the default passive state the main degradation is due to the severe bandwidth reduction, while in the active state the cascade of devices amplifies the small attenuation present in the frequency response of the OSE.

The effects are illustrated in **Fig. 68**, where the side-channels propagation and signal attenuation leads to comparable peaks for multiple resonant frequencies in the same port, for example  $Port_1$  of both the first order and the second order DC OSE. This effect is absent from the crossing-coupled second order device, as the frequency response of the  $2 \times 2$  switch is highly symmetrical, with negligible loss in the active channel.



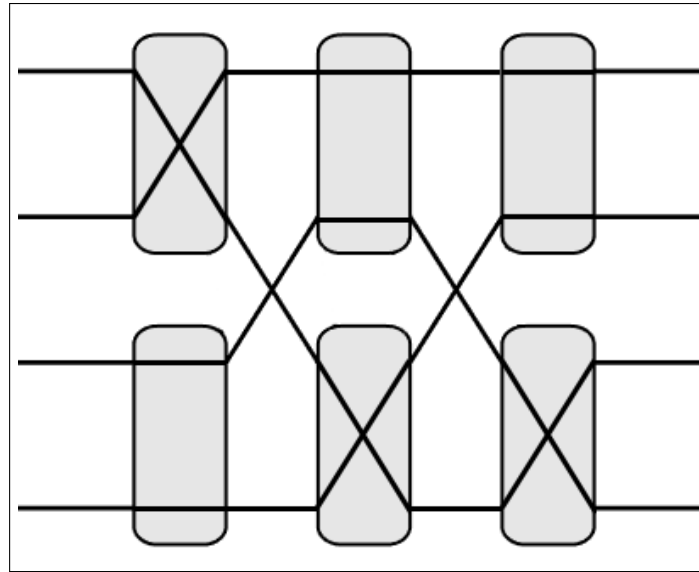
**Figure 66:** Comparison between the three device implementations  
Active state ( $S_{in} = [1\ 2\ 3\ 4] \rightarrow S_{out} = [3\ 4\ 1\ 2]$ )

### 4.2.3 Mixed state

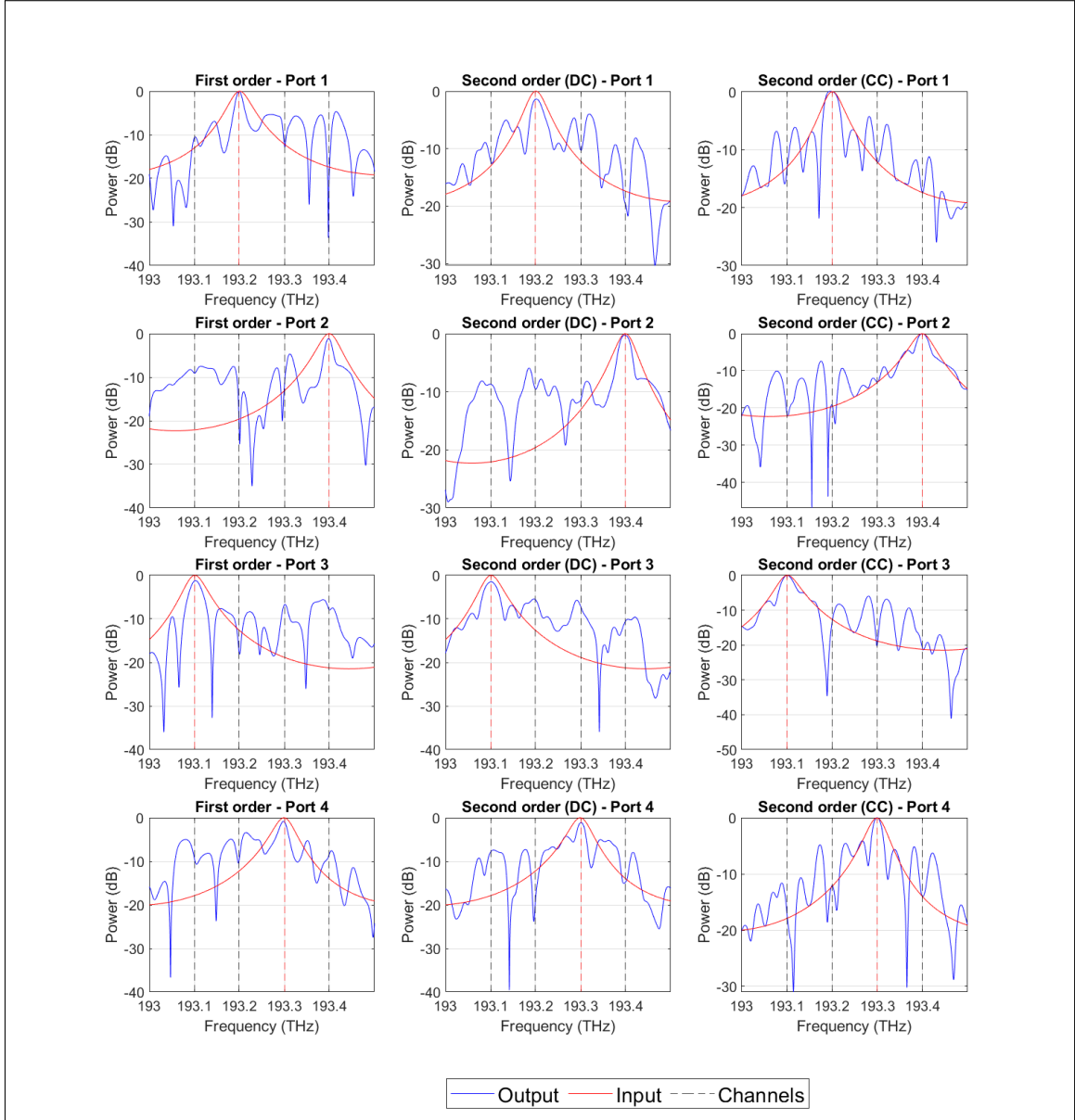
The active and passive configurations have been defined as the limiting cases for the device operation, as no degradation compensation happens along the propagation paths. Considering the trade-off behind the  $2 \times 2$  elements design, the asymmetry always leads to one propagation state having a higher bandwidth and lower transmission, while the other exhibits the opposite behavior.

Under this simple assumption, valid for the two devices with the highest degradation, a random control state would imply a certain degree of bandwidth reduction, equal or inferior to the passive case one, while at the same time side-channels propagation and transmitted signal attenuation, with a lesser or equal effect than the one displayed in the active case.

An example of this mixed configurations is proposed, defined as ( $S_{in} = [1\ 2\ 3\ 4] \rightarrow S_{out} = [2\ 4\ 1\ 3]$ ) with control vector  $V_{bias} = [1\ 0\ 0\ 1\ 0\ 1] \cdot V_{on}$ , which has been represented in **Fig. 67**.



**Figure 67:**  $4 \times 4$  Beneš configuration: mixed routing ( $V_{bias} = [1\ 0\ 0\ 1\ 0\ 1] \cdot V_{on}$ )



**Figure 68:** Comparison between the three device implementations  
Mixed routing ( $S_{in} = [1\ 2\ 3\ 4] \rightarrow S_{out} = [2\ 4\ 1\ 3]$ )

The results depicted in **Fig. 68** confirm the assumption on the compensation of the two main effects of the OSEs. While the best performances are achieved with the CC second order OSE, as expected from the previous cases, a noticeable improvement is present in both of the other two devices.

This effect cannot overcome the limitations of both cases, as such the first order OSE still suffers from a severe bandwidth reduction, although less critical than the one achieved in the passive case. Similarly the overall pass-band attenuation is mitigated, leading to a smaller central peaks in the side-channels, which was the main contribution to the degradation of the QoT introduced in the active state.

This example highlights a more reasonable expected result from this class of devices, while stressing the importance of evaluating the routing performances, whose control vectors and paths are automated by the algorithms implemented in **Section 3**.

### 4.2.4 Alternative paths

The previous analysis are useful to depict the effect of the different OSEs, as well as showing the penalty balancing between the attenuation and bandwidth reduction, in mixed state configurations.

Nonetheless the three cases correspond to three different target output configurations, as such they cannot show directly the difference in penalties for the equivalent paths of the same requested output permutation.

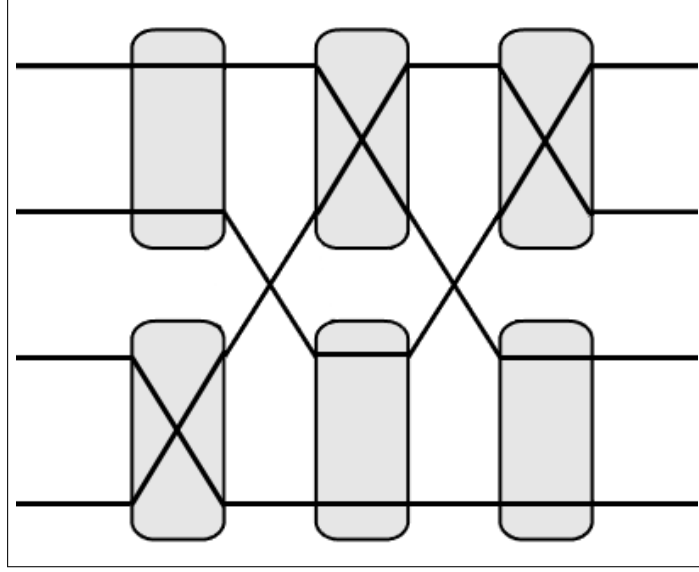
The generalized routing algorithm, as previously described, can evaluate all the paths and routing states leading to the target configuration. Imposing again  $([1\ 2\ 3\ 4] \rightarrow [2\ 4\ 1\ 3])$  as the switch output state, the equivalent network configurations are  $V = [1\ 0\ 0\ 1\ 0\ 1] \cdot V_{on}$ , considered in the previous scenario (**Fig. 67**), and  $V = [0\ 1\ 1\ 0\ 1\ 0] \cdot V_{on}$ , depicted in **Fig. 69**.

This configuration is vertically symmetric to the *mixed-case*, although it's important to state that:

- The symmetric equivalence is not a general property of the network, even for the strict-sense Beneš topologies.
- Although the number of *Bar* and *Cross* states in the network is identical, this cannot infer identical transmission for the considered input signals.

This analysis is carried out on the second order DC Beneš network, due to its average performance between the severely filtered signals of the first order structure, and the high transmission achieved by the second order CC device.

The power received at each output port is depicted in **Fig. 70**: the same four channels of the previous analysis are used as input signals, with the expected output peak of each port highlighted by the magenta lines.

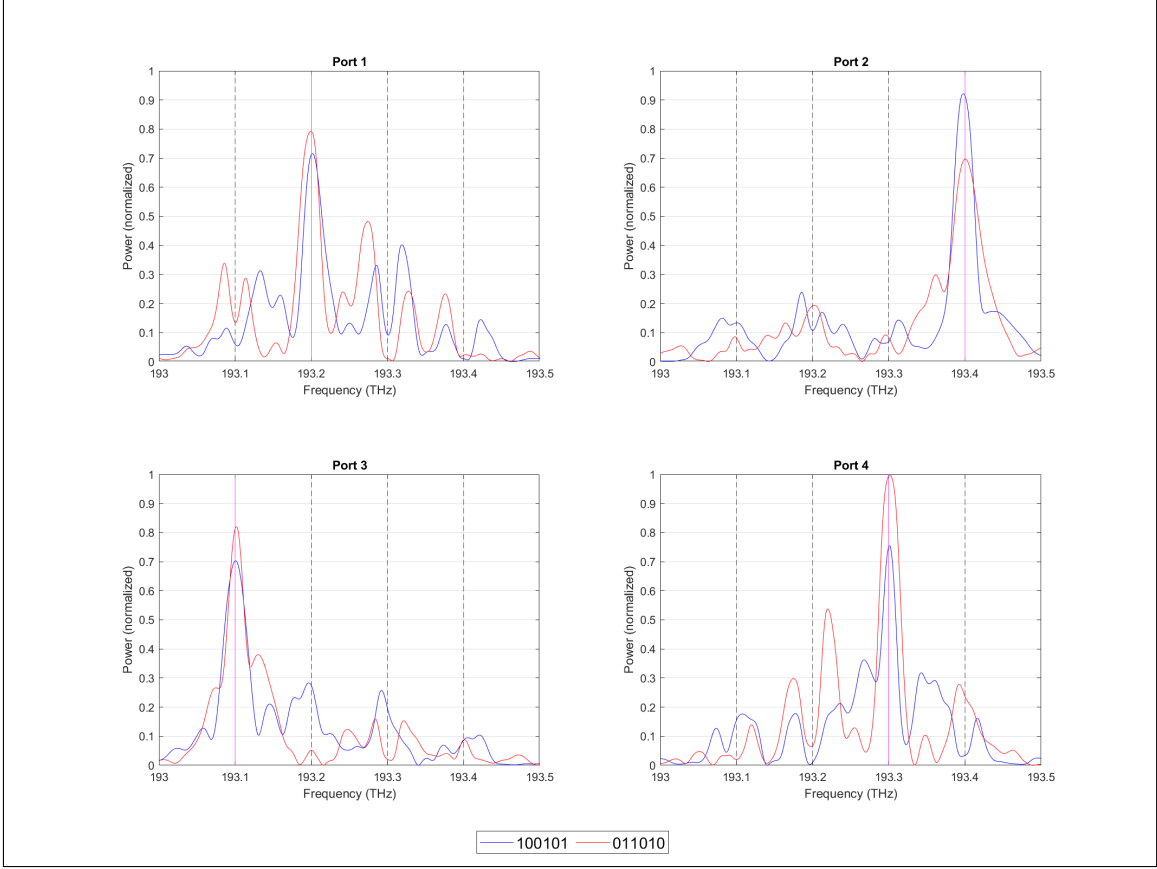


**Figure 69:** 4×4 Beneš configuration: alternative [2 4 1 3] routing paths

This example is indicative of the complex effect of the OSE states on the device behavior. The transmission improvement is clearly visible in three of the output ports, with only  $Port_2$  exhibiting the opposite behavior.

This result is not directly important for the numerical values of the transmission bandwidths and peaks, instead it highlights the presence of path-dependency in the QoT between alternative routes of the same target configuration.

Furthermore, this stresses the importance of the routing control unit for an optimized implementation of these devices, as introduced in **Section 3.6**. The evaluation of the optimal path, instead of any generic acceptable routing state, can drastically improve the performances, and must be implemented in a realistic networking scenario.



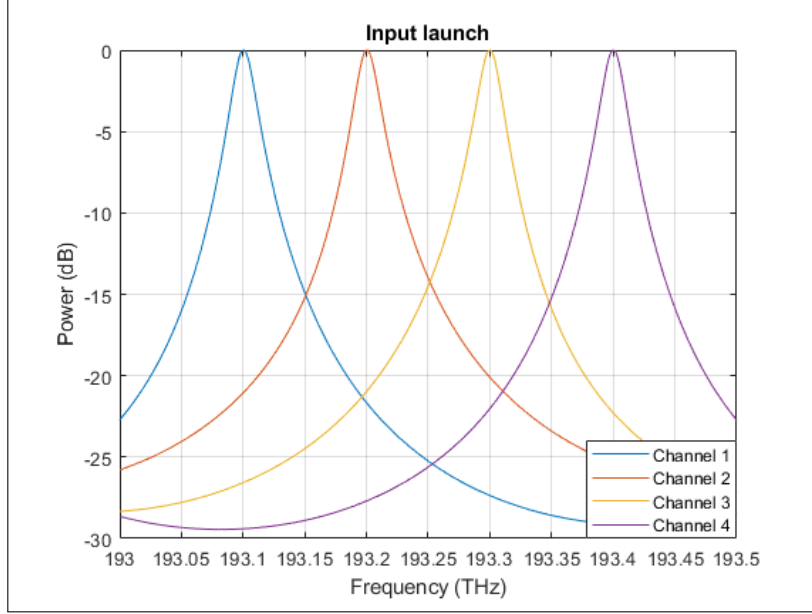
**Figure 70:** Alternative paths comparison: frequency response for the  $4 \times 4$  second order DC device

### 4.2.5 Filtered channels: reduced bandwidth

Due to the severe filtering penalties observed in the previous analysis, the implementation through the simpler first and second order DC OSEs is clearly unsuitable to handle a dense WDM comb, with large bandwidth channels.

In order to improve readability, as well as eliminate the adjacent channel interference in the transmitted bandwidth, the simulation environment has been modified to generate narrower filtered channels (**Fig. 71**). The device configuration has been set to the mixed routing already explored, with target goal  $[1 \ 2 \ 3 \ 4] \rightarrow [2 \ 4 \ 1 \ 3]$  and switching state  $V = [1 \ 0 \ 0 \ 1 \ 0 \ 1] \cdot V_{on}$ .

**Fig. 72** displays the simulation results with only a single input signal provided to the device: while maintaining the previously described configuration, only the first channel is propagated through the device with the third output port as routing target.



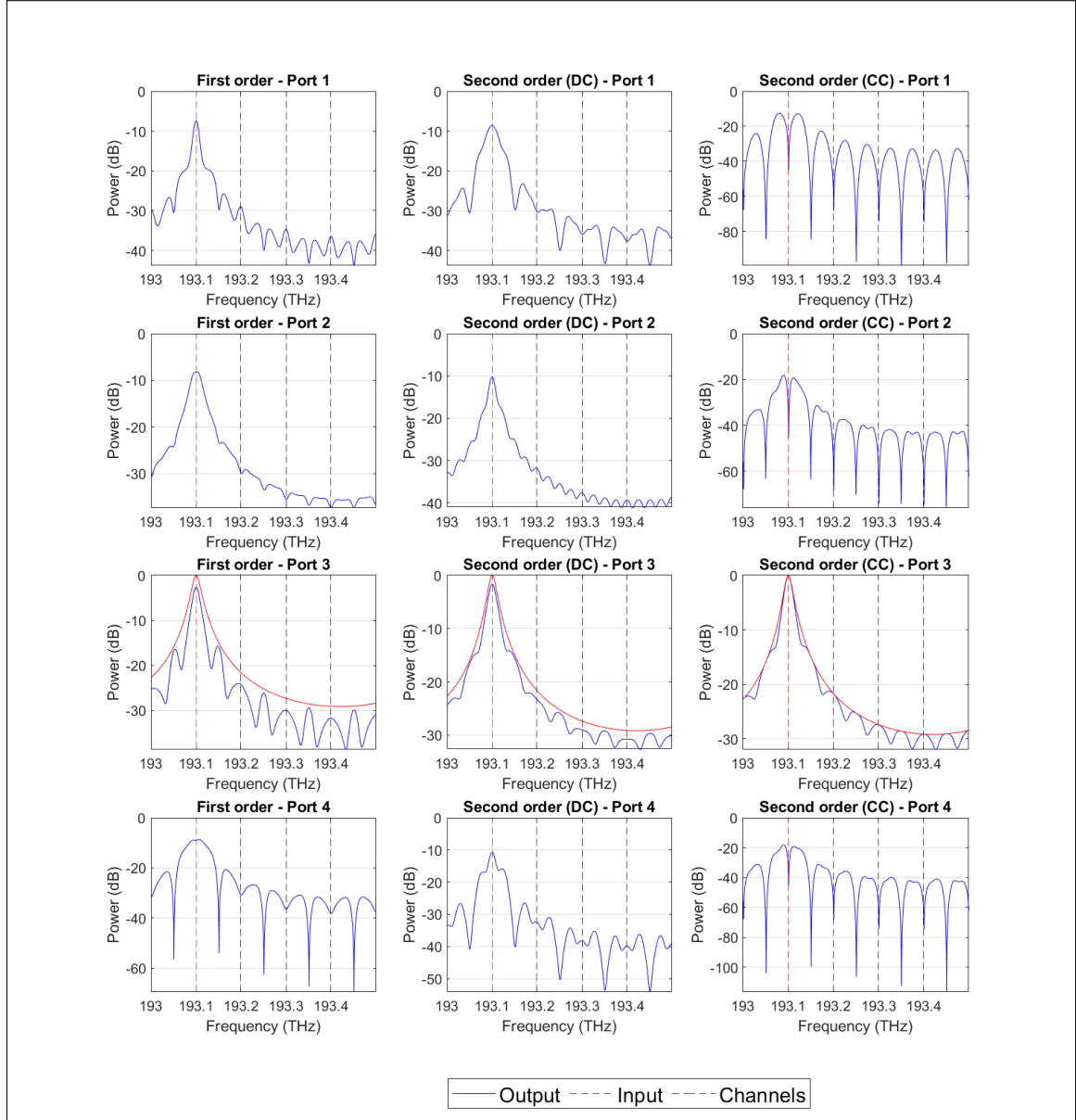
**Figure 71:** Filtered channels simulation - Launch power (reduced bandwidth)

This allows a clearer representation of the misrouted input power, without overlap of the side-channels interference. Comparing this result to **Fig. 73**, where all four channels are transmitted through the device, it's clear that the signal interference adds a significant distortion, like the strong attenuation peaks present at both sides of the transmitted channel in  $Port_3$  of the first order device.

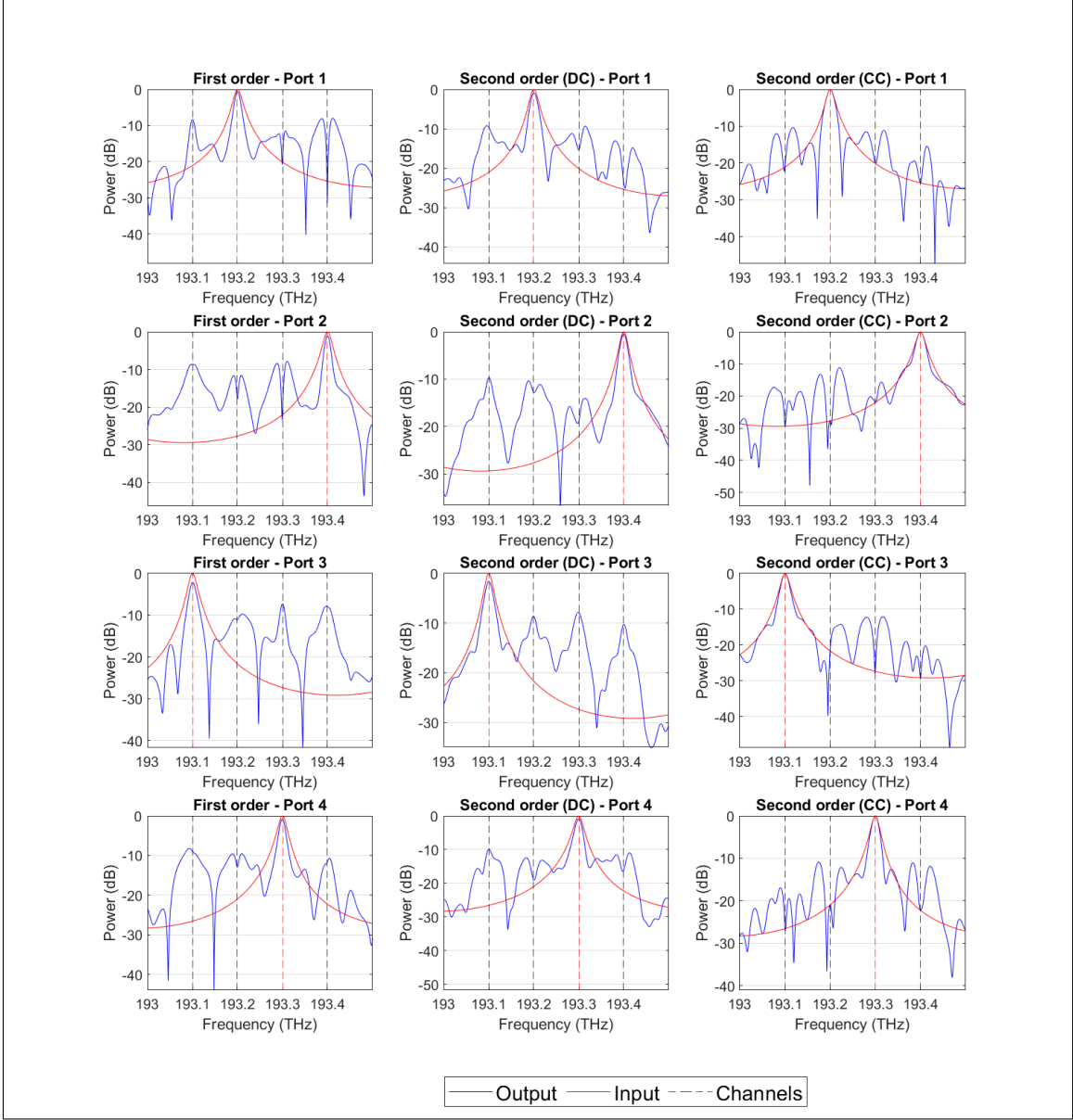
Overall, the same behavior and trade-offs previously analyzed apply to this simulation scenario, highlighting the strong degradation introduced by the cascade of these components.

The second order CC OSE is still the best solution for the implementation of scalable and larger bandwidth devices. This results is not unexpected considering the device analysis presented in **Section 2**. Both first and second order DC switches have an intrinsic loss of 20% and 10% respectively, which is seen both in the attenuation of the transmitted channel, as well as the propagation and interference to the adjacent ports: while this phenomenon can be acceptable in smaller networks, it clearly introduce severe distortions unsuitable for a larger implementation. The effect of the misrouting is amplified by the network structure, as the interconnect of the Beneš topology lead to the propagation of this signal to all following paths, which affects an increasing number of switching elements as the network size increases.





**Figure 72:** Comparison between the three device implementations  
Mixed routing ( $[1 \ - \ -] \rightarrow [- \ - \ 1 \ -]$ ) for reduced bandwidth single input signal



**Figure 73:** Comparison between the three device implementations Mixed routing  $([1\ 2\ 3\ 4] \rightarrow [2\ 4\ 1\ 3])$  for reduced bandwidth input signals

This requires further analysis in order to characterize the device, as the evaluation of the signal degradation on the filtered channels cannot yield realistic and useful data for the analysis of the best path, or the suitability of the simpler OSE implementations. Although indicative of the expected behavior and penalty, these results can be analyzed only as an implementation comparison, and are not adequate in offering an absolute performance evaluation of the device.

### 4.3 FUTURE EXPANSION

As previously stated, in both this section and in **Section 1**, a more realistic simulation environment is necessary to evaluate the QoT penalties in any meaningful way. The broadband scenario is useful to verify the correct channels alignment and the initial estimate for the crosstalk, although it is not suitable for multi-channel simulation, due to its single input port under analysis.

The filtered channel method is a precursor to more complex evaluations of the system performances: the shape of the input channels can be customized to simulate the expended bandwidth occupation of the channels, evaluating the transmitted power and the overall losses due to side-channel crosstalk.

The main issue with this method is the results format, which cannot be easily used to compare alternative configurations. While it can visually represent the propagation distortions and attenuation, the extraction of useful quantities for further analysis is difficult and heavily dependent on the arbitrary choices concerning the input signals spectrum.

The clear way to obtain a metric for the performance of these devices, while maintaining the generalized scope, is the implementation of a coherent transmitter and receiver system (TX/RX). This can be implemented with external scripts through Digital Signal Processing blocks (DSP), allowing the control of the transmission parameters with high accuracy, which is not possible in the filtered channels scenario. Under this full system model, the metric under analysis would be the Bit Error Rate (BER) which can be taken as the normalized efficiency of each configuration of the devices.

The generalization to any arbitrary  $N \times N$  systems is not yet implemented, although this methodology has been tested on a case-study concerning a  $4 \times 4$  Beneš structure, under a similar design scenario as explored in this work [14]. This simulation environment is compatible with the generalized vertical procedure presented, thanks to the compartmentalized scripts underlying the design process.

Active research and expansion is being carried out in both the stochastic ML-based approach, as well as the implementation of more complex system-level performance scenarios, while the automated process can deliver an initial estimate on the quality of the available routings, based on the number of  $ON$  states, or through the filtered channels simulation, by evaluating the overall signal loss, as well as the effective bandwidth of the output signals ( $BW_{3dB}$ ).

## 5 LAYOUT MASK

---

The last step in the design of any device, after the model evaluation and simulation, is the generation of a production layout, which must be compatible with the technological processes and limitations available at the target production foundry.

This step can be automated in a similar fashion to the switch simulation and design, through the built-in functionality available in the Synopsys design suite.

A slight modification to the OSEs templates used previously is required, in order to account for the physical dimensions of each section of the device, such as the coupling regions, curved waveguides and heating sections.

### 5.1 PDK LIBRARIES

While the simulations have been carried out using the built-in Optsim Circuit block models, which allows the evaluation of the behavior of the device, these components are not defined with a clear technological implementation, as such they cannot provide any information regarding the geometrical structure of the waveguide, as well as its placement in a layout mask.

In order to guarantee a standardized approach to the design of advanced and complex PICs, Photonic Design Kits (PDK) have been developed over the years, as to simplify and account for the foundries capabilities and technological limitations in the production steps of the desired component.

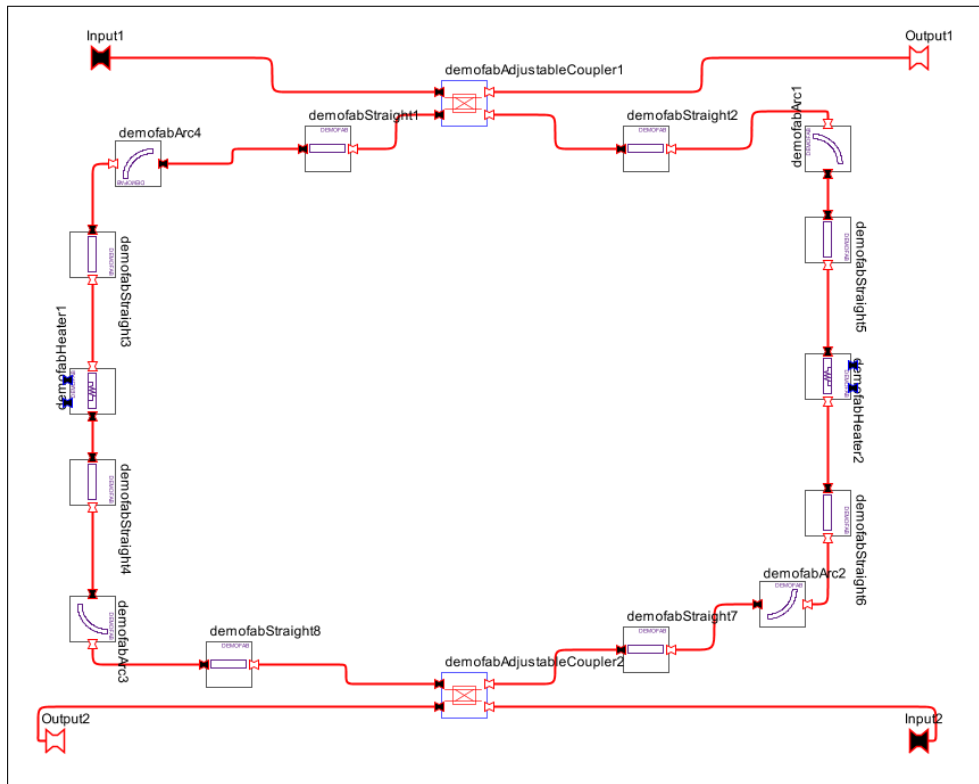
These libraries contain block elements similar to the mathematical structures used by Optsim in the evaluation of the circuit, while taking into account the constraints of the chosen technological process.

The OSEs templates used in the device simulation are based on *unconstrained* blocks, and as such they cannot be used for the automatic generation of a physical layout mask.

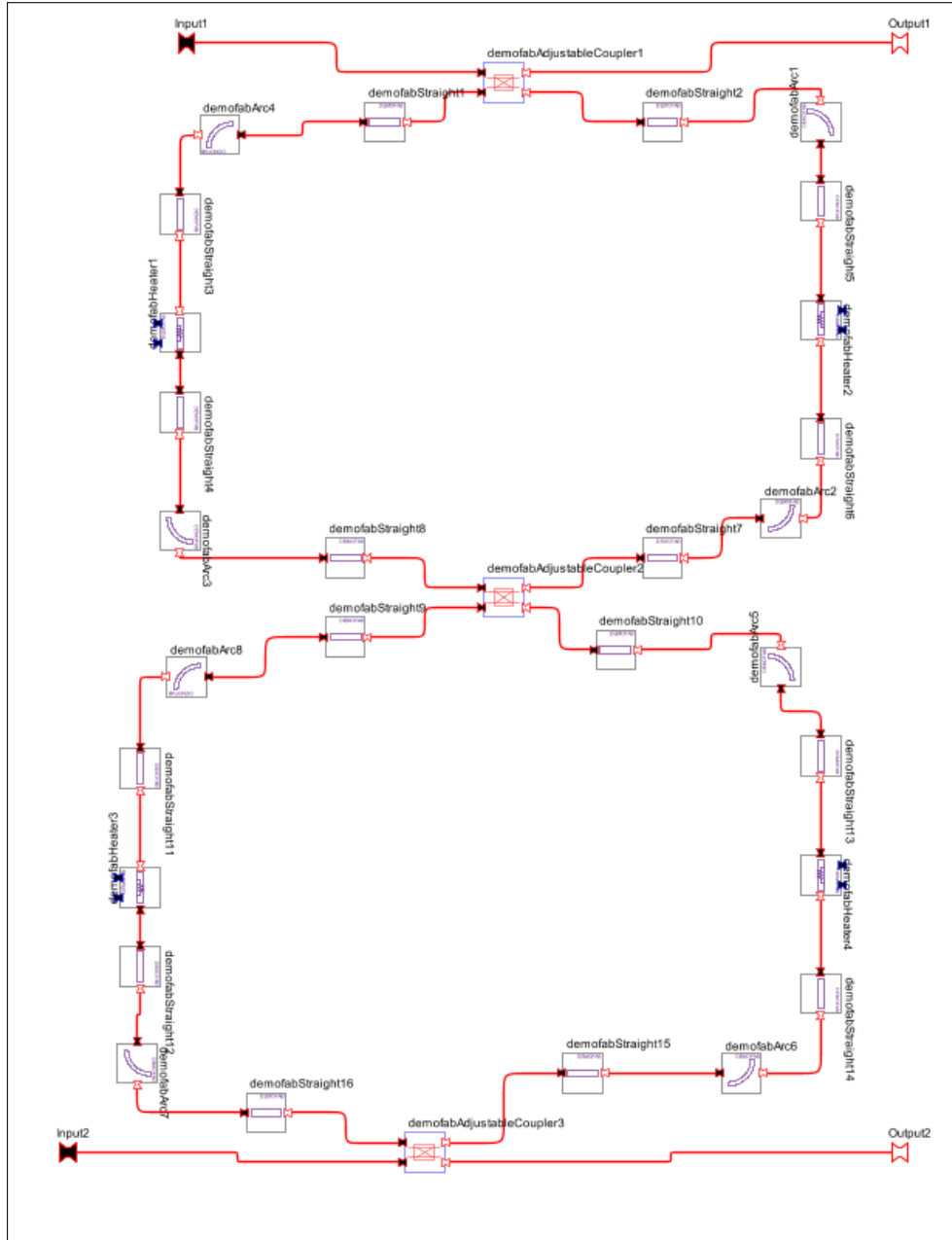
To overcome this issue, ad-hoc MATLAB scripts have been used to handle the generation of these PDK-based templates, which show no significant difference in terms of simulated behavior, although requiring a deeper specification of the geometrical parameters. The equivalent structures for the first order and second order DC OSE are shown in **Fig. 74** and **Fig. 75**. The increased number of components, especially waveguide elements, is used to constrain the geometry of the ring element, allowing the precise design of the waveguide radius in the curved sections, as well as the overall length of each section: the total round-trip length is equal to the one evaluated in **Section 2**, while allowing the user to alter the shape of the structure.

The scripting capabilities developed in this project allow for the substitution of these elements with user-defined custom components, although a complete automation is not yet achieved, as the parametrization of each block cannot be completely arbitrary, due to compatibility issues with the implemented codes.

Nonetheless this can be considered as a benchmark of the strength that script-assisted design introduce in the typical workflow for a component analysis: while the automation of some design steps may yet to be achieved, the compartmentalization and modularity of the proposed method is well suited for expansion, while maintaining the already implemented functionalities.



**Figure 74:** PDK-based model - First order OSE



**Figure 75:** PDK-based model - Second order DC OSE

### 5.2 OPTODESIGNER MASK IMPLEMENTATION

Through the same network generator script described in **Section 3**, a complete Beneš switch schematic can be obtained, using the PDK-based elements instead of the simpler Optsim block-based models. This schematic can be exported directly in Optodesigner, taking advantage of the automatic layout generator implemented in the Synopsys suite.

This initial layout mask does not represent the final implementation, as the Optodesigner compiler might not be able to generate the layout under the imposed constraints. User intervention is required to ensure the correct placement of the elements and structures, as well as verifying the correctness of the interconnects.

For complex design with numerous crossings and overlapping paths, the automatic handling of the waveguide placements can be problematic.

Although this is a software limitation, the preliminary mask still offers a valid starting point for the user-defined tailoring of the layout. Another limitation is the placement and handling of the electric part, needed in this class of devices for the heating control of the MRR elements, which can lead to conflict in the preliminary generation.

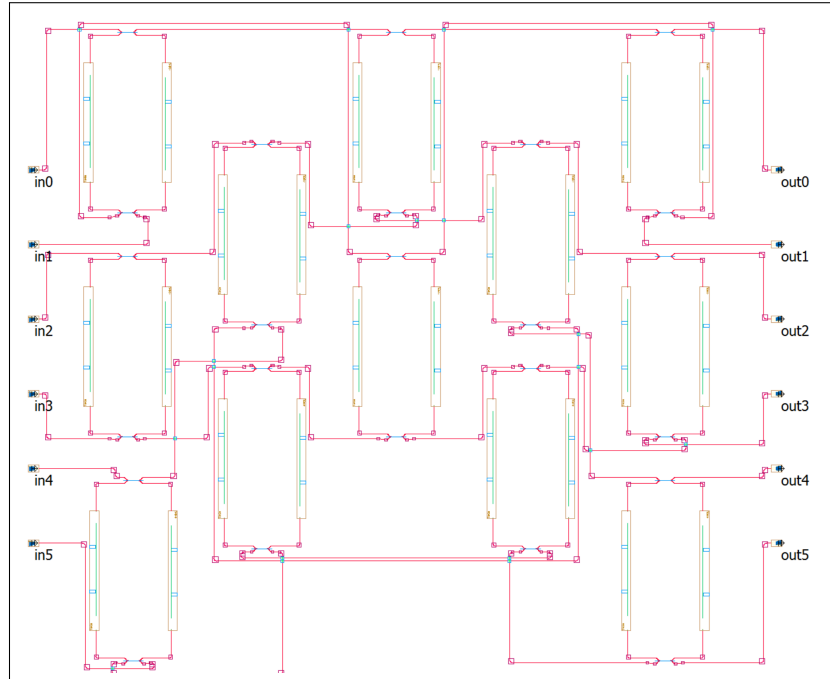
Considering that the manual adjustments and verification would be needed even with a more sophisticated starting layout, this step offers a reasonable conclusion for the automated process discussed in the project.

An example of a  $6 \times 6$  Beneš layout is shown in **Fig. 76a**, with the equivalent topological description in **Fig. 76b**.

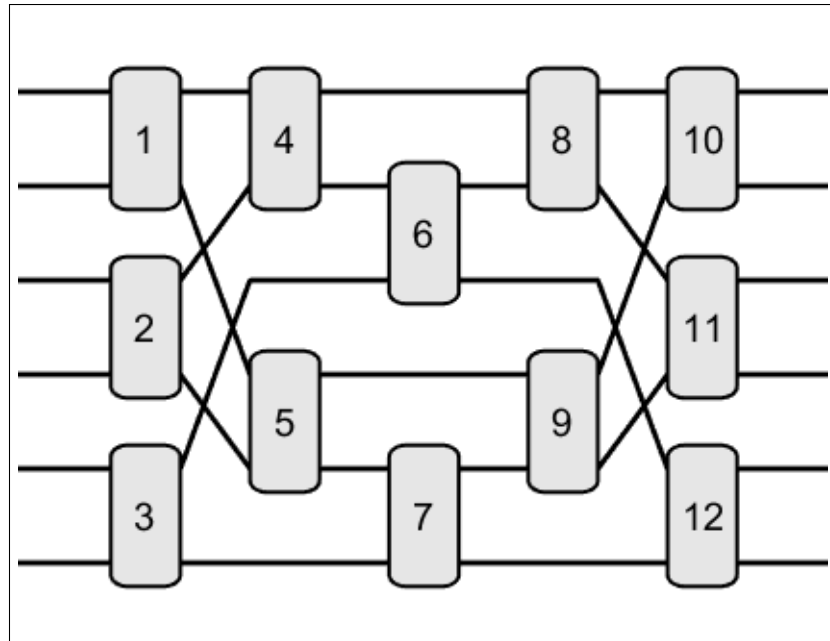
Considering the PDK implementation discussed in the previous section, the MRR have been designed with a very narrow curvature radius, in order to both account for the longer straight heating sections, as well as the limitation on the maximum curvature under the foundry specifications.

Ad-hoc design of the component would include the heating section on the entire round-trip of the MRR, which cannot be easily implemented through the block-based PDK libraries available in the Synopsys suite.

This reinforces the underlying concept regarding automation in the mask generation step: although useful for a case-study or the generation of a simpler component, it cannot completely substitute the manual design of the physical structure, which can improve performances and allows the implementation of the desired constraints and optimizations.



(a) Production-ready layout mask



(b) Network topology

**Figure 76:** 6×6 Beneš switch - Single ring MRR implementation



## 6 CONCLUSIONS

---

The thesis was mainly focused on the development of a generalized, scalable and expandable workflow for the design and simulation of Beneš switches.

Starting from the optical switching elements implementations and design strategies, three main structures have been proposed, based on micro ring resonator add-drop filters. These three alternatives have been used to test the cascading effect which underlines the operational behavior of the multistage structure.

The simulation results, proposed throughout the thesis, have been chosen as to represent the format and results expected from the automated process, as well as serving as a comparison on the physical and modelling templates developed for the analysis.

The central goal of the project is the generalization of the analysis for this class of switches, guiding the user through a step by step development and characterization of each of the elements and layers necessary to generate the final structure.

Optsim Circuit has been used as the backbone of the simulation process, with MATLAB acting as the binder, allowing the generation of the schematics, as well as the evaluation of parameters such as the switch controls, obtained through simulation and data post-processing, to maintain the generalized scope of the project.

The network topology has been evaluated through custom-developed MATLAB scripts, which allow an in-depth mathematical and logical analysis of the properties of the network, such as routing costs, number of switches in the allocated paths, as well as the generation of a routing table for the target configurations, or the complete set, when scalability allows it.

The system level simulations to evaluate the filtering penalty are proposed as a case-study for specific output goals, to show the capability of inserting the generated schematic in more complex environments.

The described approach and topic has been chosen based on the evolution of today's landscape concerning optical telecommunications, whose expansion and upgrade will require a shift toward the Software-Defined Networking paradigm (SDN). Under this reasonable assumption based on the current trends, design and integration of optical switches as PICs is fundamental, especially while maintaining a logical networking abstraction, as the one developed in this work, to characterize the multistage structure. The proposed workflow can be expanded to cover more devices for the generation of a Beneš structure, as well as serving as a starting template for the analysis of different switching topologies, or PICs implementations.

## REFERENCES

---

- [1] E. Ghillino, E. Virgillito, P. V. Mena, R. Scarmozzino, R. Stoffer, D. Richards, A. Ghiasi, A. Ferrari, M. Cantono, A. Carena, and V. Curri. “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)*. 2018.
- [2] M. S. Islam and M. J. Barsha. “Mach Zehnder Interferometer (MZI) as A Switch for All Optical Network”. In: *2018 International Conference on Innovation in Engineering and Technology (ICIET)*. 2018.
- [3] X. Tu, C. Song, T. Huang, Z. Chen, and H. Fu. “State of the art and perspectives on silicon photonic switches”. In: *Micromachines* 10.1 (2019).
- [4] Q. Li, D. Nikolova, D. M. Calhoun, Y. Liu, R. Ding, T. Baehr-Jones, M. Hochberg, and K. Bergman. “Single Microring-Based  $2 \times 2$  Silicon Photonic Crossbar Switches”. In: *IEEE Photonics Technology Letters* 27.18 (2015).
- [5] Q. Chen, F. Zhang, R. Ji, L. Zhang, and L. Yang. “Universal method for constructing N-port non-blocking optical router based on  $2 \times 2$  optical switch for photonic networks-on-chip”. In: *Optics express* 22 (May 2014).
- [6] C. Clos. “A study of non-blocking switching networks”. In: *The Bell System Technical Journal* 32.2 (1953).
- [7] C. Chang and R. Melhem. “Arbitrary Size Benes Networks”. In: *Parallel Processing Letters* 7 (1997).
- [8] M. Yahya, N. Wu, G. Yan, T. Ahmed, J. Zhang, and Y. Zhang. “HoneyComb ROS: A  $6 \times 6$  Non-Blocking Optical Switch with Optimized Reconfiguration for ONoCs”. In: *Electronics* 8 (July 2019).
- [9] S. Arora, T. Leighton, and B. Maggs. “On-Line Algorithms for Path Selection in a Nonblocking Network”. In: *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*. STOC ’90. Baltimore, Maryland, USA: Association for Computing Machinery, 1990.
- [10] D. C. Opferman and N. T. Tsao-wu. “On a class of rearrangeable switching networks part I: Control algorithm”. In: *The Bell System Technical Journal* 50.5 (1971).
- [11] A. Chakrabarty, M. Collier, and S. Mukhopadhyay. “Matrix-Based Nonblocking Routing Algorithm for Bene[U+009A] Networks”. In: *Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, Computation World* 0 (Nov. 2009).
- [12] I. Khan, L. Tunesi, M. Chalony, E. Ghillino, M. U. Masood, J. Patel, P. Bardella, A. Carena, and V. Curri. “Machine-learning-aided abstraction of

- photonic integrated circuits in software-defined optical transport”. In: *Next-Generation Optical Communication: Components, Sub-Systems, and Systems X*. Ed. by G. Li and K. Nakajima. Vol. 11713. International Society for Optics and Photonics. SPIE, 2021.
- [13] V. Curri. “Software-Defined WDM Optical Transport in Disaggregated Open Optical Networks”. In: *2020 22nd International Conference on Transparent Optical Networks (ICTON)*. 2020.
- [14] P. Pasella. “Assessing the Impact of an Optical Switch Physical Design in Network Routing Impairments”. In: *Corso di laurea magistrale in Ingegneria Elettronica (Electronic Engineering)*. Politecnico di Torino. 2019.