Implementation of an optimization model and interface for an optimized operation of Semiconductor Optical Amplifier cascades in Optical Switches

Desalegn Wolde Feyisa, Salim Abdi, Zhaowei Chen, Yuqing Jiao, Nicola Calabretta and Ripalta Stabile

Institute for Photonic Integration, Eindhoven University of Technology, PO Box 513, 5600MB Eindhoven, the Netherlands.

Abstract

Optical switches are becoming crucial in optical communication networks to scale switching capacity to meet continuously growing traffic demand. Semiconductor optical amplifier (SOA)-based switches could play such roles as they can meet current optical network capacity and speed scaling requirements. To scale up an SOA-based switch, we need to cascade several SOAs in a multistage configuration, which increases noise figures and hinders scalability, exacerbated by the difficulty in setting the optimal parameters to mitigate the noise building up. Therefore, we develop a technique for optimizing parameters in SOA cascades, specifically bias current for each SOA, depending on SOA length, the losses between consecutive SOAs, and the input optical power, which is imperative. To this end, we experimentally measure the parameters of a single SOA and create its model, including gain dynamics and noise figures in the VPI photonics modeling tool, to build an SOA cascade circuit model. Then we develop an optimization algorithm that can automatically find the optimum operating condition for each SOA in SOA cascades. Our approach is based on carefully mapping the input power and bias currents to the path performance. We then validate the method experimentally with a chain of integrated SOA devices.

1. Introduction

Optical signals are known for their fast signal transportation, high bandwidth capacity, agnostic to transmission format, and consumption of less energy per transmitted bit [1]. While this feature has already been widely used in signal transportation, switching in the optical domain is crucial to take full advantage of these desirable features of communication in the domain. Large port count optical switches needed to cope with a growing global network and the number of nodes that need to be interconnected require complex control layer implementation. Specifically, when SOAs are cascaded in a multistage configuration [2], controlling the switch becomes more challenging. Switch modeling and control layer automation are necessary for the design and circuit operation phase due to the following reasons: -

1. The design and design validation stages are crucial in improving the performance of complex optical circuits. Implementing robust and hierarchical modular modeling of the circuit in advance is vital for a fast design cycle and parameter optimization. For instance,important SOA design parameters such as waveguide width and SOA length impact signal integrity differently in a standalone SOA and SOAs in a chain. The gain, extinction ratio, OSNR, and gain saturation depend

not only on the performance of the individual SOA but also on the signal quality and level from the preceding SOA.

2. The control layer development and testing are relatively more manageable, accessible, flexible, and cheap in the model than working on the actual circuit. It also allows the development of the switch control algorithm and testing at the design stage than waiting for fabrication. The development of the control layer on an actual circuit is sometimes costly as some devices are pricy. For instance, measuring eight paths simultaneously for an 8×8 switch requires eight laser arrays and 8 BERT, which is expensive. However, this can be done on a software basis quickly and in parallel, cutting a lot of time. It allows one to evaluate the component parameter effect on the circuit level.
3. In blocking or rearrangeable non-blocking switch configuration, path scheduling and control are complex. Scheduling concepts are easy to develop on models and then deploy into the circuit

This paper will show strategies to model a switch module that emulates a realistic switch and develop a switch control method and user interface for switch design and control automation. We will discuss a process for selecting the optimal currents that give the best performance for a path.

2. Modular optical switch modeling

VPI photonics component maker tools allow component models based on measured data or transfer functions from other design tools. In this work, we model each switch component and emulate the optical switch in the VPI photonics design tool. After making the component model based on our design, we validate whether it works as intended and use it for control layer implementation, automation, and optimization.

Fig 1 shows the modular switch modeling. Fig 1a contains the elementary building blocks customized according to our design. We model the waveguide, waveguide crossing, waveguide bends, multimode interference splitter/combiner, and SOA and transfer the

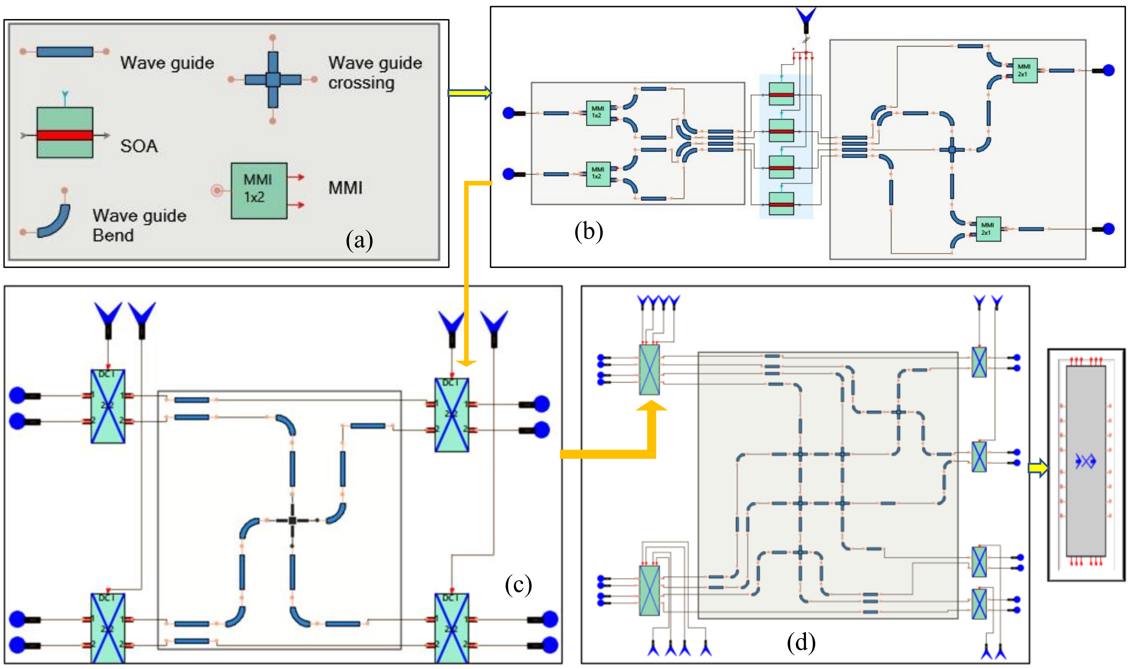

Figure 1: Modular optical switch modeling for development of design and control automation methods

model to VPI to build the smallest switch-building block module as in Fig 1b. Then the compact model of the 2×2 switch is used to construct 4×4 and 8×8, as shown in figs 1b and 1 c, respectively. The switch control and optimization layers are then applied to the compact 8×8. This modular approach is crucial both in design and actual implementation. It enables validation of the design process on a small scale and scales it quickly by simplifying the design process and localizing error diagnosis.

3. Switch control layer implementation for control automation

The schematic of a switch is created in the VPI photonics design tool, as in Fig 1, and connected to MATLAB. The switch parameters to be controlled and the switch output are monitored by programs developed in MATLAB. The software provides suitable currents to the switch depending on optimization rules. Several files are created to facilitate the control and convey the circuit information to the software. The SOA correspondence file stores mapping information between SOAs in the circuit and associated drivers; the SOA id file contains the ids of SOAs. Additionally, the flag files, such as the SOA flag file, contain if an SOA is working or inactive, a path flag has status information about the path, and the port flag indicates the status of ports.

The algorithm for implementing the control layer is shown in figure 2. It starts by reading the flag and SOA information files and assigning them to global variables declared within MATLAB. It then displays the current status of the circuit on the graphical user interface created in MATLAB for user information. The control layer continuously probes if there is a path connection request from a user. If a connection request is detected for a path, the software checks if all flags are clear and the path can be established. If the path is free, the control layer will set the current to the optimal value found from the current tuning algorithm, updates the related flag status, and updates the displays accordingly. If the path is currently connected, the request is counted as a path clear request, the path is disconnected, and the associated flags are cleared. If the two conditions are not fulfilled, the path is busy, and the busy flag is updated. In this case, either one/more SOAs in the path are occupied by other connections, or the input/output port is busy.

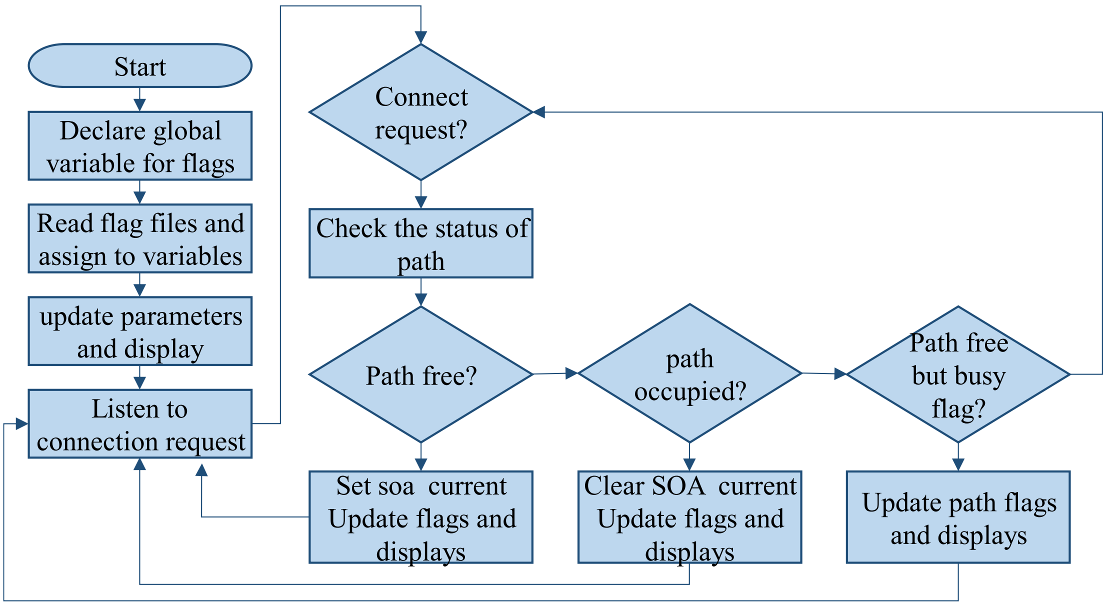

Figure 2: Algorithm for automated switch control implementation

4. Automatic optimal current selection implementation

The optimal current selection is time-consuming and challenging to achieve if done manually. However, it is possible to mathematically establish the relation between optimal currents at a particular input power and the optimal signal integrity from sample data sets generated from the circuit. This can be done in two separate approaches, shown in fig 3. The first is by creating a parametric equation relating input power and currents to the BER and then applying optimization algorithms to find currents that give minimal BER. The second approach uses a black box approach where data sets are used to train the system and generate a look-up table that shows the optimal BER at a given input power. The former method is suitable when the relationship among the parameters follows a pattern and can be described by equations, while the former can describe complex relations

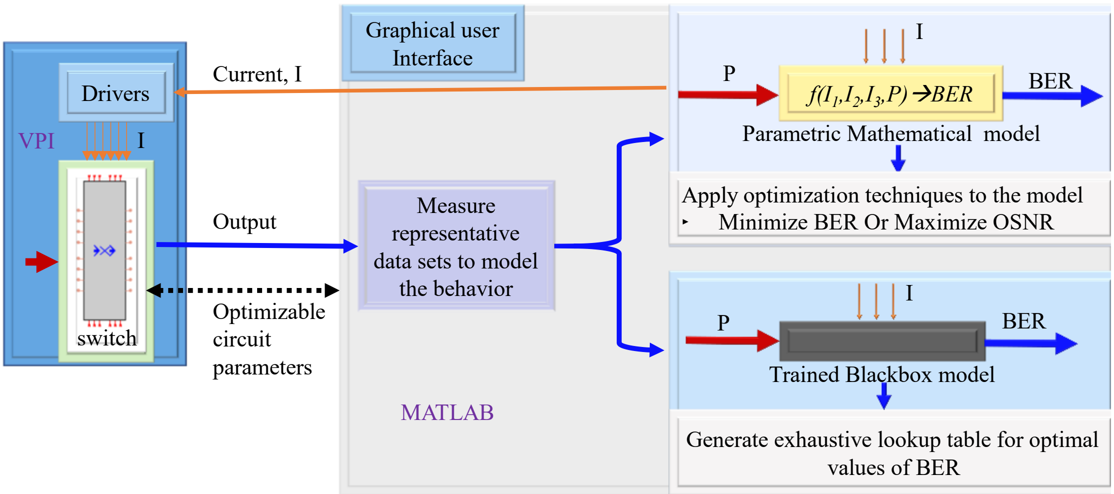

Figure 3: schematic indicating optimal switch drive current selection techniques

5. Conclusion and future outlook

We show a model of an optical switch. We implement a control layer that controls the switch without knowing how the switch is designed by using files that convey information about the on-chip parameters to the control layer software. We discussed the method to find the currents that give the optimal values for the signal integrity. We will apply Both forms of selecting optimal currents to the model in the future to make a performance comparison analysis.

Acknowledgment

This work was supported by the H2020 ICT TWILIGHT Project (contract No. 781471) under the Photonics PPP

References

[1] J. Strand and A. Chiu, "Realizing the Advantages of Optical Reconfigurability and Restoration with Integrated Optical Cross-Connects," *J. Light. Technol.*, vol. 21, no. 11, pp. 2871–2882, 2003, doi: 10.1109/JLT.2003.819539.

[2] R. Stabile, A. Albores-Mejia, and K. A. Williams, "Monolithic active-passive 16 × 16 optoelectronic switch," *Opt. Lett.*, vol. 37, no. 22, p. 4666, 2012, doi: 10.1364/ol.37.004666.