Programmable Silicon Photonic Circuits

Wim Bogaerts*, Xiangfeng Chen, Hong Deng, Lukas Van Iseghem, Mi Wang, Iman Zand, Yu Zhang, Yichen Liu, K.P. Nagarjun, Umar Khan
Ghent University – IMEC, Department of Information Technology, Ghent, Belgium
Center of Nano and Biophotonics (NB-photonics), Ghent University, Belgium
*wim.bogaerts@ugent.be

Abstract: *Programmable photonics configure the flow of light on a photonic chip using software and electrical actuators. Such flexible chips are useful for prototyping new applications. We discuss the underlying technologies for hardware and software.*
Keywords: *Silicon Photonics, Photonic Integrated Circuits, Programmable Photonics*

I. Introduction

Photonic Integrated Circuits (PICs) are rapidly growing in complexity as the number of building blocks on a chip and their connectivity increases. This evolution is largely powered by Silicon Photonics technology, which enables drastic miniaturization of photonic building blocks that can be manufactured using industrial CMOS tools. But even with this growing maturity of technology, we still see that it is difficult to design and manufacture complex silicon photonics devices such that they work first-time-right[1]. The sensitivity of silicon waveguides to nanometer-scale variations is very high, and typical fabrication tolerances that are acceptable for electronics prove to be too stringent for photonic circuits. This can be partially compensated by active tuning, and today we see more and more silicon photonic circuits that are actively paired with electronic driver and control circuits to adjust their functionality.

Programmable photonics takes this an important step further, in the sense that electronic control is not just used to fine-tune or adjust the circuit, but to fully configure it [2]. This makes the photonic circuit reconfigurable, in a similar way as programmable electronic hardware such as field-programmable gate arrays (FPGAs) [3]. This flexibility presents interesting opportunities, where it is no longer necessary to design full-custom circuits, but rather make use of generic photonic chips that can be programmed to perform a variety of functions, and can therefore accelerate development and prototyping for new applications [4].

To make a photonic circuit programmable, you need to control the flow of light, both in amplitude and phase. This is achieved by creating a mesh of waveguides that are interconnected using tunable power couplers, interspersed with phase shifters. The tunable couplers not only allow to switch the light path from a cross state to a bar state, but also to split and combine beams along different paths, making it possible to implement multi-path interference devices.

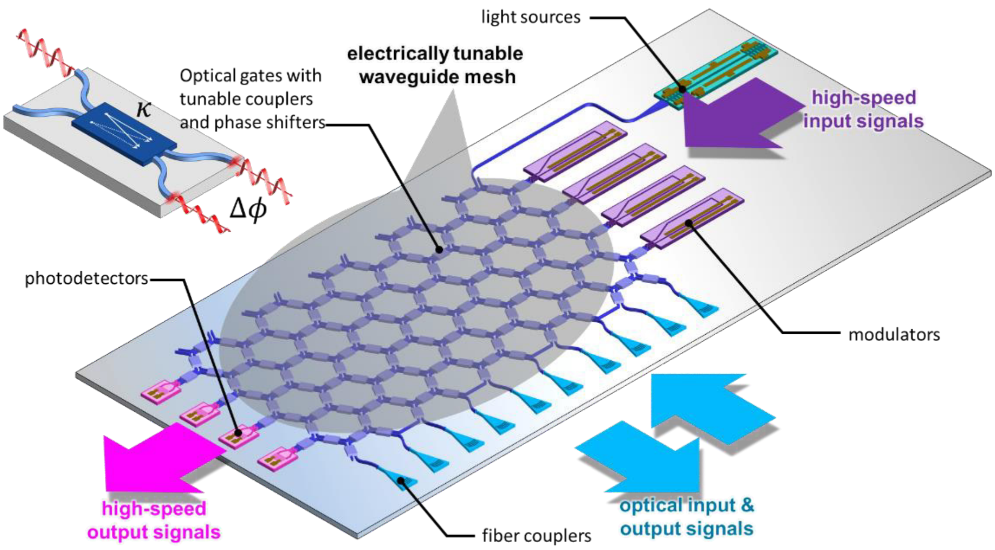

Fig. 1. Schematic illustration of a programmable photonic chip with a hexagonal waveguide mesh.

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 780283 (MORPHIC) and from the European Research Council under grant agreement No 725555 (PhotonicSWARM).
Part of the work was supported by the Flemish Research Foundation (FWO-Vlaanderen) by the grants G020421N (GRAPHSPAY) and G031421N (MALEPHICENT)

Making all elements in a large photonic circuit tunable brings along a number of steep challenges. Where a custom designed tunable circuit might require a few tens of electronic drivers, generic programmable photonic circuit rapidly scale into hundreds of individually controlled elements that each need an independent electronic control. This raises the question of how the photonics and electronics need to be co-integrated, and if the tuners are based on heater, careful management of power consumption and thermal crosstalk is needed.

Because the functionality of a programmable photonic circuit is defined by configuring the tunable elements, additional logical layers are needed to govern the configuration and the control of the photonic circuits. First of all, a user should be able to define the desired functionality, and translate this into the ideal connectivity of waveguide and functional building blocks on the chip, which can then be addressed by the corresponding electronic drivers. However, because all elements on the photonic chip are in essence analog devices (including the electronic driver) the response curve for each tunable coupler and phase shifter will vary slightly from the reference value. This will require calibration steps to identify these nonidealities and pre-compensate them in software. On top of that, perturbations at run time (including the aforementioned thermal crosstalk) need to be compensated by active control routines, which require monitoring and feedback loops embedded inside the waveguide mesh.

In this paper we will discuss the overall technology stack of programmable photonic circuits, going from the physical hardware of the silicon photonics chips and its building blocks, over the packaging all the way to the software layers needed to make the chip configurable.

II. Programmable Waveguide Meshes

Fig. 1 shows a schematic drawing of a programmable photonic chip. We see at the core a programmable waveguide mesh consisting of optical gates that can mix two input waveguides with an arbitrary power coupling factor and phase delay. Such a waveguide mesh can be connected in different ways. In a forward-only mesh, also called a multi-path interferometer or a universal unitary circuit, light propagates in only one direction between a set of input ports and a set of output ports [5]. Such circuits effectively implement an arbitrary linear transformation between the input and outputs, which makes them interesting for specific applications in quantum optics of artificial intelligence, where such matrix-vector products are very common operations [6], [7].

An alternative to these forward-only meshes are recirculating meshes, where the optical gates are arranged in loops, allowing the waves to propagate in any direction in the mesh [8]. Depending on the unit cell, it is now possible to implement optical delays with different increments to create interferometers and even resonators that can be used as wavelength filters [9].

The waveguide mesh itself is only limited to passive optical functions. Therefore, it is important that other optical functions are incorporated on the chip. This can be done inside the mesh, but a more practical approach is to intrgrate the custom building blocks on the periphery of the mesh, such as indicated for the modulators and the photodetectors in Fig. 1. These high-speed devices also allow the encoding of high-speed analog and digital signals, which can then be spectrally processed into the programmable waveguide mesh [8].

III. The Technology stack for Programmable Silicon Photonics

To realize a fully programmable silicon photonic chip, many aspects need to come together. The resulting technology stack is more complex than that of conventional photonic circuits because of the extra requirements of software layers to control and configure the chip.

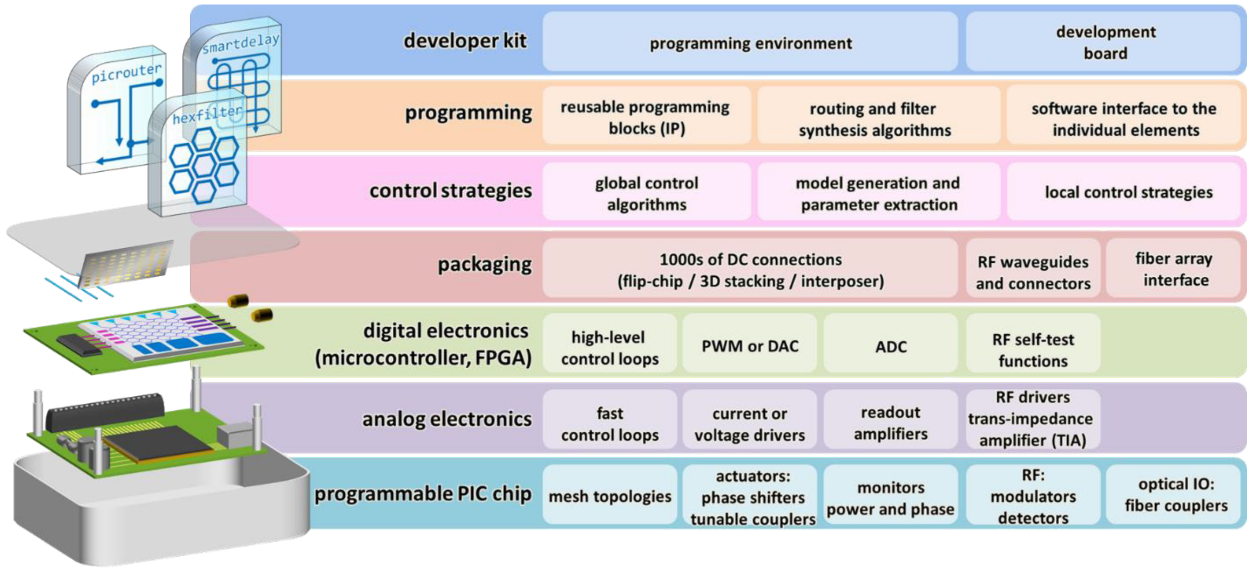

Fig. 2. Technology stack for programmable photonic circuits.

We already mentioned the electronics as an additional layer of complexity in the technology stack. The large-scale photonic circuit needs scalable drivers with high precision, that can be easily co-integrated with the photonics. This in itself poses a packaging challenge, to connect hundreds or even thousands of electrical wires to the drivers. The packaging should also take into account the optical connections, and if necessary, high-speed RF connections. Because the photonic chip is quite sensitive, even with all the tuning capabilities, thermal control of the package might also be desirable.

These packaging requirements are challenging because they scale up considerably from traditional PICs, but the overall approach can still be considered traditional. It is different for the software layers that are needed. There, we see the need for new custom algorithms and procedures specifically designed for programmable photonic waveguide meshes. First of all, every element in the waveguide mesh should be calibrated. While basic algorithms have already been proposed to iteratively characterize every tunable coupler and phase shifter in the mesh by measuring the output at the edge of the mesh [10], there is a lot of room for improvement, for instance by incorporating additional monitoring detectors inside the mesh. This might be especially necessary for recirculating meshes which can suffer from severe interference effects with even small deviation from the nominal behavior [11]. Additional monitoring detectors cold also allow for parallel calibration that can speed up the testing for larger circuits. The result of the calibration can be stored in software or firmware settings as transducer curves for the phase shifters and tunable couplers to speed up the configuration of the mesh during operation.

A second layer of software is needed to control the mesh during its operation. Because photonic circuits are very sensitive to small variations, it is not inconceivable that the response of the circuit elements will drift over time. Therefore, methods are needed to stabilize and control the set points of the actuators. While such routines have been described for specific cases, especially in forward-only meshes [12], this is a non-trivial unsolved problem in recirculating meshes that also need to have controlled wavelength dependence for filter functions. Use of dithering or pilot tones could help here, as it would allow to disentangle the contributions of individual actuators to an output signal.

A third layer of software should allow the user to configure the photonic chip for specific functionality, for instance as a switch matrix, rerouting layer, or optical filter function. It will support the user with autorouting functions [13] and filter synthesis, and provide the necessary tools to visualize and debug the configuration of the chip. In that respect, it is very similar in purpose as the developer toolkits accompanying today's electronic hardware such as FPGAs.

IV. SUMMARY

Programmable photonic circuits offer a potential to revolutionize the adoption of photonic integrated circuits, but drastically shortening the development time for new applications. However, there are still significant challenges to be addressed to complete the full technology stack of hardware and software.

REFERENCES

[1] W. Bogaerts and L. Chrostowski, "Silicon Photonics Circuit Design: Methods, Tools and Challenges," *Laser Photonics Rev.*, vol. 1700237, pp. 1–29, 2018, doi: 10.1002/lpor.201700237.

[2] W. Bogaerts *et al.*, "Programmable photonic circuits," *Nature*, vol. 586, no. 7828, pp. 207–216, 2020, doi: 10.1038/s41586-020-2764-0.

[3] D. Pérez, I. Gasulla, and J. Capmany, "Field-programmable photonic arrays," *Opt. Express*, vol. 26, no. 21, p. 27265, 2018, doi: 10.1364/oe.26.027265.

[4] W. Bogaerts and A. Rahim, "Programmable Photonics: An Opportunity for an Accessible Large-Volume PIC Ecosystem," *IEEE J. Sel. Top. Quantum Electron.*, vol. 26, no. 5, p. 1, 2020, doi: 10.1109/JSTQE.2020.2982980.

[5] D. A. B. Miller, "Self-aligning universal beam coupler.," *Opt. Express*, vol. 21, no. 5, pp. 6360–70, 2013, doi: 10.1364/OE.21.006360.

[6] N. C. Harris *et al.*, "Linear programmable nanophotonic processors," *Optica*, vol. 5, no. 12, pp. 1623–1631, 2018, doi: 10.1364/optica.5.001623.

[7] Y. Shen *et al.*, "Deep learning with coherent nanophotonic circuits," *Nat. Photonics*, vol. 11, no. June, pp. 441–447, 2017, doi: 10.1038/nphoton.2017.93.

[8] J. Capmany, I. Gasulla, and D. Pérez, "Microwave photonics: The programmable processor," *Nat. Photonics*, vol. 10, no. 1, pp. 6–8, 2016.

[9] D. Pérez, I. Gasulla, J. Capmany, and R. A. Soref, "Reconfigurable lattice mesh designs for programmable photonic processors," *Opt. Express*, vol. 24, no. 11, p. 12093, 2016, doi: 10.1364/oe.24.012093.

[10] S. Bandyopadhyay, R. Hamerly, and D. Englund, "Hardware error correction for programmable photonics," *Optica*, vol. 8, no. 10, p. 1247, Oct. 2021, doi: 10.1364/OPTICA.424052.

[11] I. Zand and W. Bogaerts, "Effects of coupling and phase imperfections in programmable photonic hexagonal waveguide meshes," *Photonics Res.*, vol. 8, no. 2, pp. 211–218, Feb. 2020, doi: 10.1364/prj.376227.

[12] D. A. B. Miller, "Self-configuring universal linear optical component [Invited]," *Photonics Res.*, vol. 1, no. 1, p. 1, 2013, doi: 10.1364/prj.1.000001.

[13] X. Chen, P. Stroobant, M. Pickavet, and W. Bogaerts, "Graph Representations for Programmable Photonic Circuits," *J. Light. Technol.*, vol. 38, 2020.