CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Software framework architecture for programmable photonic chips
Authors	Zand, I.;Antony, Cleitus;Chen, X.;Bogaerts, W.
Publication date	2022-08-22
Original Citation	Zand, I. Antony, C., Chen, X. and Bogaerts, W. (2022) 'Software framework architecture for programmable photonic chips', 2022 IEEE Photonics Society Summer Topicals Meeting Series (SUM), Cabo San Lucas, Mexico, 11-13 July, WB1.3 (2pp). doi: 10.1109/SUM53465.2022.9858236
Type of publication	Conference item
Link to publisher's version	10.1109/SUM53465.2022.9858236
Rights	© 2022, IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Download date	2026-10-30 03:46:15
Item downloaded from	https://hdl.handle.net/10468/13685

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Software Framework Architecture for Programmable Photonic Chips

I. Zand [1], C. Antony [2], X.Chen [1], W. Bogaerts [1]
[1] Ghent University - imec, Photonic Research Group, Department of Information Technology, Gent, Belgium.
Center of Nano and Biophotonics, Ghent University, Belgium.
[2] Tyndall National Institute, Lee Maltings Complex Dyke Parade, T12 R5CP Cork, Ireland
Email: wim.bogaerts@imec.be

2022 IEEE Photonics Society Summer Topicals Meeting Series (SUM) | 978-1-6654-3489-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/SUM53465.2022.9858236

***Abstract*—We introduce a software framework for design and programming of large scale programmable photonic chips. The framework facilitates design, simulation, configuration and measurements of reconfigurable waveguide meshes, managing hundreds of actuators and translate high-level user requirements into driving strategies of the electronic channels.**

***Index Terms*—silicon photonics, programmable photonic, software framework**

I. Introduction

Programmable Photonic Integrated Circuits (PICs) are an emerging concept that could be transformative for the field of photonics, just like FPGAs have transformed electronics [1]. The concept of programmable photonic circuits revolves around interconnected waveguides where phase relations and coupling coefficients can be individually tuned. Such tunability can be achieved by mechanical or electro-optical actuation of the chip tunable blocks (generally couplers and phase shifters), or incorporating electro-optic materials. And, thanks to the CMOS compatibility of silicon photonic platforms and steady increase in quality of their fabrication, it is expected that these circuits to be implemented in a large scale using hundreds or even thousands of tunable blocks with acceptable yield and circuit performance.

Final implementation of such photonic chips requires a multi-layer photonic system and a robust software framework to address users needs from designs and simulations to test and measurements. A programmable photonic system architecture usually consists of different layers of electronics, RF, and optics which enable various optical and high speed signal processing operations and calculations for a variety of applications [1]. Hence, such a system needs a framework that manages communication between different layers of the system and facilitates users decision making steps.

Here we elaborate the software framework, which we are developing as part of the MORPHIC project, to control and configure a large scale programmable photonic chips based on the silicon photonic MEMS platform [2]. Although such a framework can be used for variety of programmable photonic circuits, our focus will be on those based on the recirculating waveguide meshes.

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).

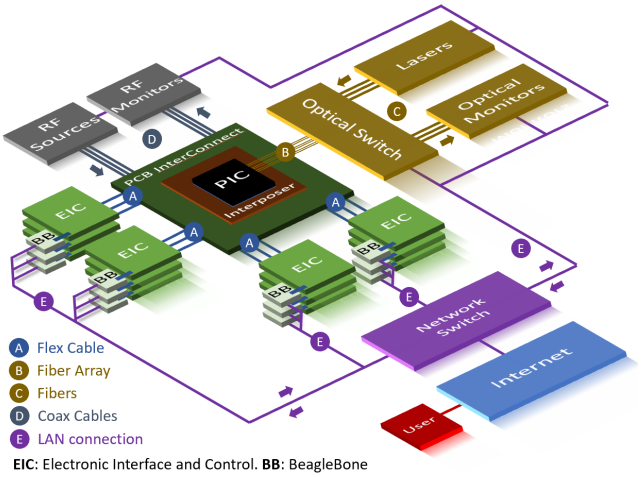


Fig. 1. Schematic of a programmable photonic system architecture.

II. Programmable Photonic System Architecture

To implement a software framework for a programmable photonic system we first need to understand the architecture of such a system (Fig. 1). The core of this system is a PIC which is attached to a custom-developed high-density interposer to manage the large number (1000s) of electrical connections for the MEMS actuators and the monitor photodiodes [2]. Input/output interfaces up to 24 RF connections (modulators and high-speed detectors) are also incorporated on this interposer, and optical I/O is accomodated by a 72-fiber array. The interposer is attached to a multi-layer PCB interconnect which is connected to the series of EIC (Electronic Interface and Control) boards using 40-pin flex cables with 32 driver channels. Each EIC board has its own dedicated single-board computer (Beaglebone Black) as controllers. For characterisation, the PIC fiber arrays are connected by a fiber switch to the lasers and monitors. For high speed communications, the PCB interconnect is connected to RF Sources/Monitors using 2×12 coax cables. And, all the controllable system components are accessible through a local ethernet network.

Authorized licensed use limited to: UNIVERSITY COLLEGE CORK. Downloaded on September 27,2022 at 20:36:05 UTC from IEEE Xplore. Restrictions apply.

III. SOFTWARE FRAMEWORK

The framework which is built based on the Python language consists of various classes representing physical components of the systems and set of functions to manage routing, calibrations, and feedback control loops. The framework also interfaces with the measurement and instrumentation. In the following, we describe its essential parts and features.

A. Netlists

Netlists represent connectivity between different elements. The framework includes a *master netlist* which defines all the connectivity from the abstract schematic (Fig. 2) to the DACs/ADCs channels. This netlist is created step by step during various design cycles of electronic and optical components. It should be mentioned that each element in the netlist (DAC, ADC, phase shifter, ...) is annotated with calibration data which can be accessed using a look-up table.

B. Process Design Kit (PDK)

To make effective use of the silicon-photonics MEMS platform an initial process design kit (PDK) have been created which extends the existing iSiPP50G PDK with the functionality for the new building blocks. This enables circuit-level design into the extended platform. Design freedom at the geometric level is intentionally restricted to improve the reliability and yield of the standard building blocks.

C. Mesh Configuration and Routing Algorithms

The framework also allow us to use various graph-based libraries to perform automatic routing for different mesh configurations [3]. Users can either use existing libraries or add their own libraries. Using standard data structures of the framework, they can simply extract graph data of the mesh, call routing algorithms functions, and pass the new configurations to the mesh. Then, all the parameters at the different levels from abstract schematic to electronics will be automatically updated.

D. Circuit Simulations

For the circuit simulations, we interface with the IPKISS tool set by Luceda Photonics. The circuit simulations are performed after mesh configuration and defining the mesh ports, using different abstraction levels at the node (fast and approximate, using a single model to describe a node) or the building block (slower but more accurate, where each component has its scattering model).

E. Feedback loops

For accurate control during the programming stage, feedback loops (electronic or in software) are implemented between the monitor photodiodes and the MEMS actuators. Such programming can be used to correct imperfect behaviour from fabrication imperfections, or to redefine the circuit's functionality.

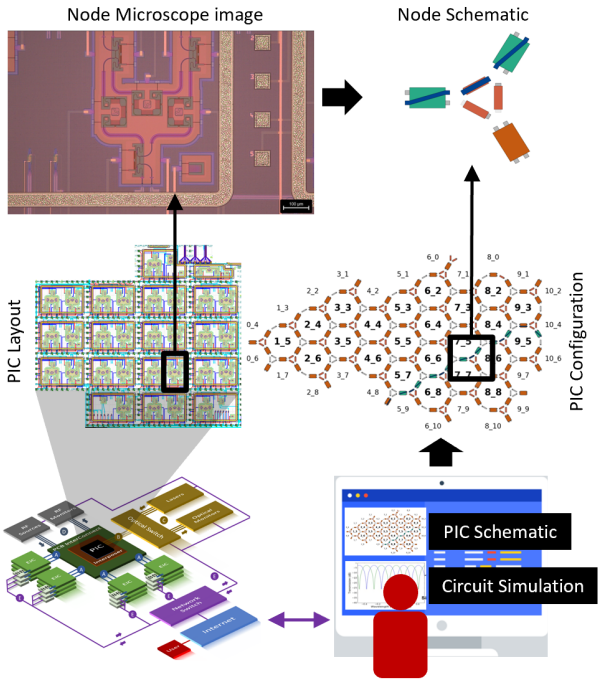


Fig. 2. Software layers for controlling and programming a photonic integrated circuit (PIC). From a single description of the circuit connectivity or the circuit mesh, users can define a PIC layout implementation, configure the tunable elements, perform circuit simulations and control the hardware.

F. Visualizations and Layouts

For the visualizations, users can interact with an abstract mesh schematic (Fig. 2) for various objectives such as checking mesh configuration and finding malfunctioned blocks. They also can inspect the equivalent graph view of the mesh. Another, feature of the framework is that users can instantly convert their mesh schematic to the actual layout of the mesh for fabrication and then make their own customization.

IV. CONCLUSION

We discussed a software framework for the programmable photonic chip based on the silicon photonic MEMS platform and recirculating meshes. The software facilitates chip configuration and calibration, provides automatic routing algorithms, enables various visualizations, support circuit simulations, and, also, allow for automation of the layout generation.

REFERENCES

[1] W. Bogaerts, D. Pérez, J. Capmany, D. A. B. Miller, J. Poon, D. Englund, F. Morichetti A. Melloni," Programmable Photonic Circuits," Nature, vol. 586, pp.207–216, 2020.

[2] W. Bogaerts, A.Y. Takabayashi, P. Edinger, G. Jo, I. Zand, P. Verheyen, M. Jezzini, H. Sattari, G. Talli, C. Antony, M. Saei, C. Lerma-Arce, J. Lee, S. Kumar, M. Garcia, T. Jonuzi, K.B. Gylfason, N. Quack, F. Niklaus, U. Khan, "Programmable Silicon Photonic Circuits powered by MEMS," SPIE Photonics West OPTO (invited), 12005, United States, pp.12005-21, 2022.

[3] X. Chen, P. Stroobant, M. Pickavet, W. Bogaerts, "Graph Representations for Programmable Photonic Circuits," IEEE Journal of Lightwave Technology, vol. 38(15), pp.4009-4018, 2020.

Authorized licensed use limited to: UNIVERSITY COLLEGE CORK. Downloaded on September 27,2022 at 20:36:05 UTC from IEEE Xplore. Restrictions apply.