Review Article

Twan Korthorst*, Remco Stoffer and Arjen Bakker

Photonic IC design software and process design kits

Abstract: This review discusses photonic IC design software tools, examines existing design flows for photonics design and how these fit different design styles and describes the activities in collaboration and standardization within the silicon photonics group from Si2 and by members of the PDAFlow Foundation to improve design flows. Moreover, it will address the lowering of access barriers to the technology by providing qualified process design kits (PDKs) and improved integration of photonic integrated circuit simulations, physical simulations, mask layout, and verification.

Keywords: design; design kits; integrated photonics; layout; photonics design flow automation; simulations.

DOI 10.1515/aot-2015-0004
Received January 11, 2015; accepted February 18, 2015; previously published online March 19, 2015

1 Introduction

The key to creating and verifying innovative products for any technology is a set of precharacterized building blocks, which a designer can use to create a complex device or system. For electronic design and manufacture, mature electronics design automation (EDA)-oriented flows require foundries to offer a set of building blocks, with corresponding verification rules, which a designer can integrate to create complex digital CMOS-based electronic integrated circuits to meet a desired need. This concept also applies to today's newer photonics design automation (PDA)-oriented flows; however, the PDA-oriented flow allows for a greater emphasis on novel implementation and analysis of the photonic building blocks (BBs) to develop new and innovative photonics products. A second key differentiation between EDA and PDA is the large parameter space typical photonic BBs have, where transistors are typically described by one or two parameters with a limited value range, basic photonics components can easily have five to 10, with large value ranges. This is mainly caused by having 'analog' rather than 'digital' design. A typical workflow for digital IC design contains several automated steps including logical synthesis and autogeneration of the circuit layout. The analog IC design flow is much more comparable with photonics IC design, with many manual steps, like component design and routing.

***Corresponding author: Twan Korthorst,** PhoeniX Software, Enschede, The Netherlands, www.phoenixbv.com, e-mail: twan.korthorst@phoenixbv.com
Remco Stoffer and Arjen Bakker: PhoeniX Software, Enschede, The Netherlands, www.phoenixbv.com

2 Photonics design flow automation

It was in the end of the 80s and early 90s of the previous century that integrated photonics research started to surface and become visible to a wider audience. Materials like polymers, doped glass, and dielectric thin film materials like silicon oxide and nitride were dominant at that time, and this new emerging field was called integrated optics, studying lightwave devices or planar lightwave circuits (PLC). As a result of these research activities, a need for proper design software emerged, focusing on the simulation of (mainly) passive optical structures at micrometer scale and the mask layout for the actual fabrication of the structures and devices. This is reflected, for example, in the start of an annual conference on Optical Waveguide Theory and Numerical Modelling (OWTNM) [1] in 1992 and the first commercial activities for design services and simulation tools (BBV in the Netherlands in 1991 and Photon Design [2] in the UK in 1992). Since then, photonic IC design software has developed into what is available today: a set of more or less integrated solutions from a variety of vendors covering different levels in what is called the 'photonics design flow' or PDA.

In the design process, a set of conceptual or abstraction levels can be defined. The lower levels are relevant

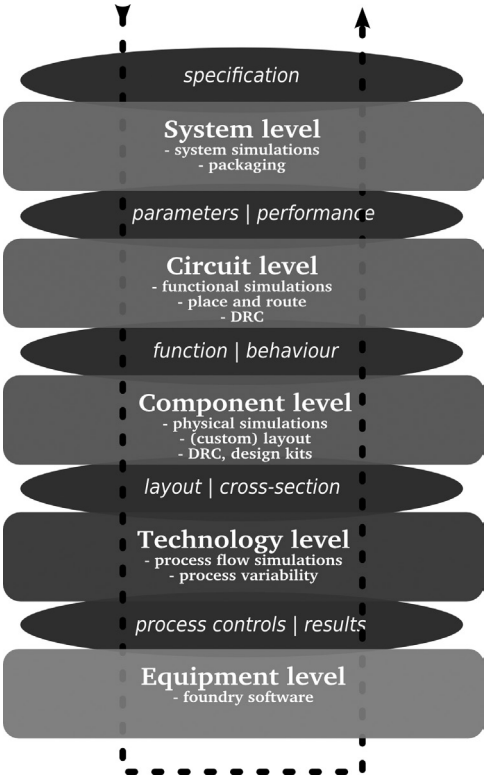

Figure 1: The five different abstraction layers of PDA [3].

for the design of the chip; higher up the tree are the design methodologies for hybrid devices and the final system. This subdivision in levels is shown in Figure 1.

On each level, simulation tools are required that can calculate the response of the components on that level, in terms of quantities that are required by the simulation tools of the next higher level. The inputs to these simulation tools are threefold: the results of the simulation tools or measurements of the next lower level, plus the parameters that are input into this level from the next higher level, and finally the layout choices that are made in this level to connect the components, from the next lower level, that are combined in this level. The current practice today is often a mix of measurements and simulations as input to higher levels.

The simulations do not yield values assuming perfect technology only but also the tolerances on all quantities, which are propagated through the layers from the technological accuracy in the lowest level.

From origin, the designers in the field of integrated photonics have been working with a bottom-up approach, starting from the fabrication technology and materials and taking these as a starting point to develop photonic-integrated devices. With the introduction of more standardized and generic fabrication processes since 2005 and the resulting creation of design kits (see Section 6), a mixed design approach has evolved in which a group of designers is developing the contents of the design kits, and (another) group of designers is using these design kits in a top-down approach starting from the system or circuit level. The required software tools to create a full photonics design flow include circuit simulators, mask layout tools, measurement databases, and design rule checkers, and also physical modeling tools such as mode solvers and propagation simulators.

3 Physical simulation tools

At the technology and component level, the design activity is mainly physically oriented. Components or building blocks are designed individually with a combination of component layout, physical simulation, and characterization of structures to extract valid compact models that could be used by a designer to create more complex circuits with simulation tools at the circuit and system level. Today, physical design for photonics and electronics is done in dedicated tools combining process, electro, electromagnetic, thermal, and mechanical simulations.

Software like *FIMMWAVE* [2], *FDTD Solutions* [4], and *OptoDesigner* [5] are commercially available and widely used by both industrial as well as academic organizations to support designers to simulate the behavior of light in small waveguiding structures. Such simulations are needed to obtain the optimal parameters for the mask layout process and are also used to explore new device concepts. The first simulation step in any design is usually a mode-solver calculation. Mode solvers calculate the field profiles (see Figure 2), propagation constants, and losses of light running through a nonchanging straight or bent waveguide. Such modes can be used directly, to design a length difference for a desired phase shift, for example, or as input for the more complicated propagation algorithms. Key (passive) photonics components can often be fully designed using these modal properties, using analytical modeling to derive the component layout properties and tend to avoid the need for other simulations apart from model verification and sometimes crosstalk analysis.

If the design is not uniform in one direction and no analytical model is available, one usually has to resort to propagation algorithms. Each algorithm has its own advantages and disadvantages, mostly to do with computational effort vs. accuracy or vs. applicability to a class of problems. For example, the finite difference time domain (FDTD) method (see Figure 3) can be applied to pretty much any structure, taking into account reflections and

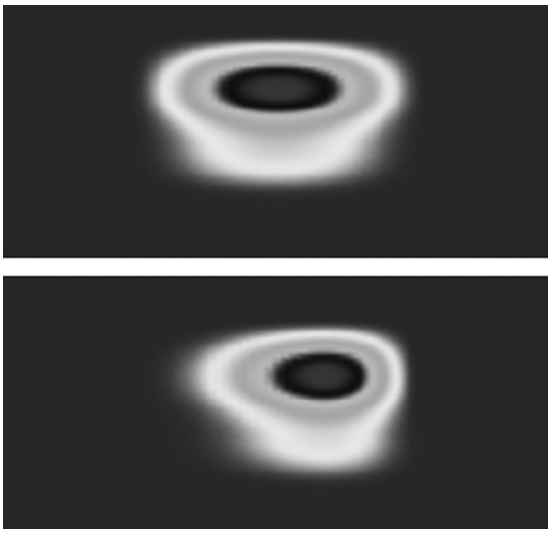

Figure 2: A 2D representation of the mode field of a straight (upper) and bent (lower) waveguide. It can be observed that the energy distribution of the electromagnetic wave is pushed toward the outside of the bent waveguide.

out-of-plane propagation, at a high computational cost. Many structures, however, are only slowly varying in one direction and do not suffer from significant reflections; then, the much 'numerically cheaper' beam propagation method (BPM) is a better choice; it can be applied to much longer and wider structures than the FDTD method can, but does not handle reflections, very high contrasts or high propagation angles well. Between those two extremes, the eigen mode expansion (EME) method is positioned, which uses many modes calculated by a mode solver to calculate the propagation through a device. It does take into account reflections and can typically deal with longer (but not wider) structures than the FDTD. Its computational cost lies somewhere between the BPM and the FDTD.

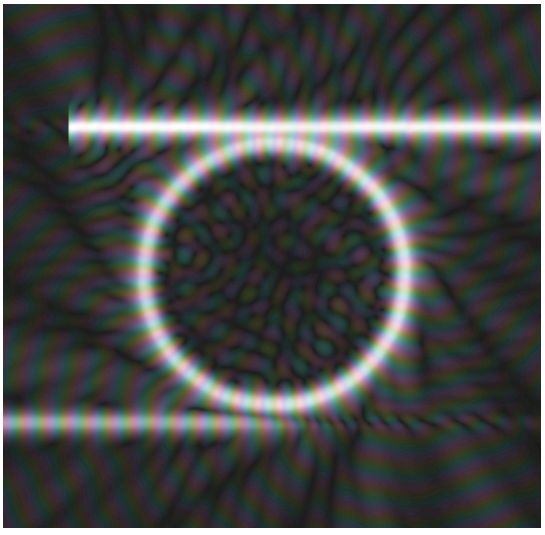

Figure 3: A result of a FDTD simulation of a ring resonator; part of the light coming in from the top left waveguide is coupled to the bottom left one. This is a very wavelength-dependent process, making the device a wavelength filter.

4 Circuit simulation tools

At the circuit and system level, the design activity is centered on the construction of photonic-integrated circuits in schematic form by linking components together and then implementing these circuits into larger networks including the detection or communication schemes. For the latter, software like *VPItransmissionMaker* [6] and *OptiSystem* [7] are applied to design and simulate transmission systems and networks. In contrast to the physical simulations as discussed above, the circuits are simulated on a higher level, using behavioral models. These models and tools make the assumption that the light is flowing in clearly defined waveguide modes, allowing to simulate much larger devices than a physical simulator could. This is common practice in electrical design, using Spice-based simulators. In photonics, this is also becoming more commonplace, and photonic circuit simulation tools are becoming more widespread. However, unlike Spice, there is no single standard simulation method for photonics, and depending on the needs, different tools and methods might be preferable. Several vendors offer dedicated solutions, like *Aspic* [5, 8] *INTERCONNECT* [4], *VPIcomponentMaker* [6], *PICWave* [3], and *Caphe* [9].

4.1 Simulating in the frequency domain: scattering matrices

For passive linear circuits, the most widely applied simulation method is based on scattering matrices. These 'S-matrices' embed the analytical model and numerical or measurement data for each component or building block. Models can be either derived from the theoretical behavior of the building blocks or can be extracted from electromagnetic (physical) simulations. They can also include experimental data. To perform a simulation, the software assembles the scattering matrices of all the applied building blocks of the circuit and solves the resulting sparse matrix equation providing amplitude and phase at the input/output port of each building block for both the forward and backward field. Assuming all building blocks provide their wavelength-dependent responses, the whole spectral response of the circuit can be calculated, and thus, parameters like the group delay and chromatic

dispersion of the entire circuit are also available as a function of wavelength.

Large flexibility in the choice of the building block parameters allows the user to do much more than a simple spectral analysis, enabling for instance 'what if' analysis, virtual experiments, tolerance analysis, case analysis, statistical analysis based on, for example, Monte Carlo simulations to evaluate the robustness of the circuit against fabrication uncertainties, and so on. Figure 4 shows a typical screen shot of the *Aspic* simulator. It shows a discretely tunable delay line based on a split and select technique with semiconductor optical amplifier (SOA) gates [10]. The state of each SOA is defined by its electrical current, and the spectral response is simulated for each state. The output power and the group delay at a given wavelength are shown in the plot. The simulation of this rather large circuit requires just a few seconds, a result not at all achievable with electromagnetic simulations.

4.2 Simulating in the time domain

Time domain-based photonic circuit simulators are capable of modeling both passive and active components. Passive (linear components) can be specified by simple parameters like the optical length of a waveguide, the coupling coefficient of a directional coupler or alternatively a wavelength-dependent S-matrix. For active components such as semiconductor optical amplifiers, laser diodes, and optical modulators, detailed models are required to simulate these in the time domain. In contrast to passive components, where good models like the S-matrix formulation exist, it is difficult to create compact models for active devices, like an SOA, for example, that match static and dynamic characteristics over a large operating range of temperature, electrical drive, optical input, etc., at all modulation rates from MHz to 10s of GHz. Therefore, most time-domain models for active BBs include a lot of relevant physics such as, carrier diffusion, spatial hole burning, and current spreading. Inherently, these are complex and, thus, cpu expensive models. Alternatively, these models are essentially 'smart' curve fits based on a mix of physical understanding derived from the simulations and optical performance measurements, which are much cheaper in cpu use, while still predicting circuit performance quite well.

5 Mask layout

In photonics, historically, most emphasis has been on full-custom layout of the individual components and combining these into (simple) circuits. In Figure 1 this is represented at the component level. Today, with the increasing

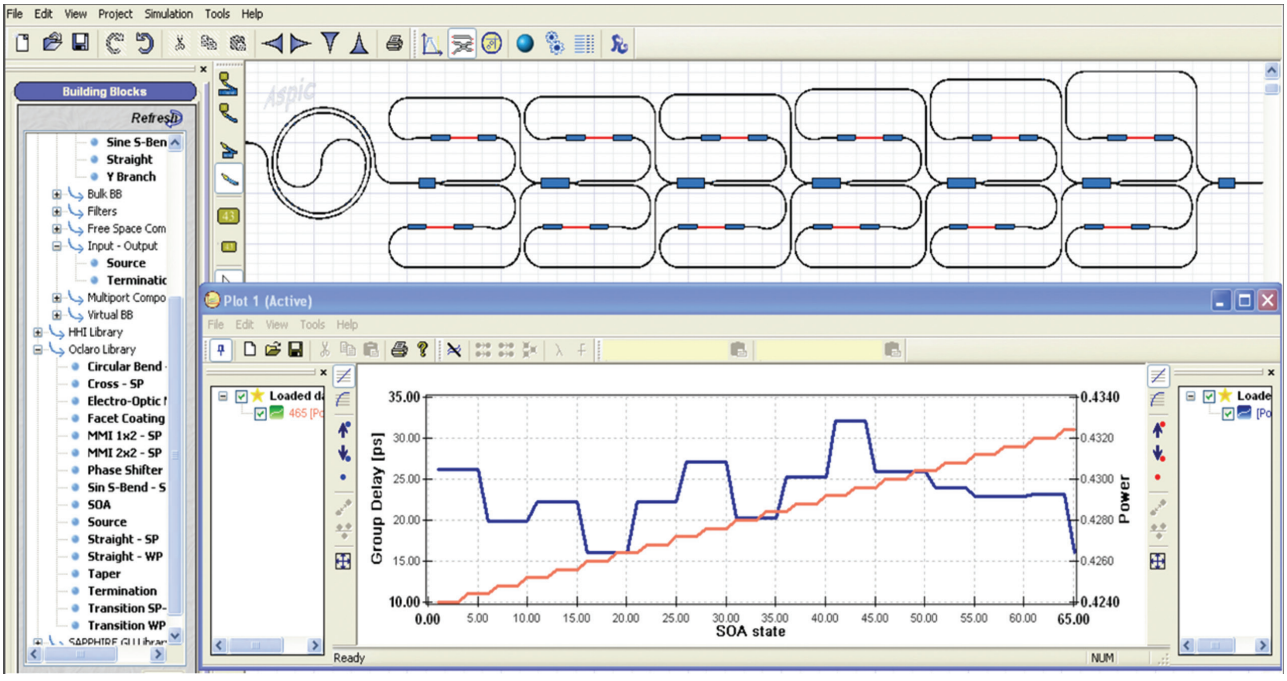

Figure 4: Screenshot of the Aspic simulator output with the circuit to simulate at the top and the plot window with the simulation result at the bottom.

complexity of the circuits, the mask layout ideally should be generated from the schematic as created with a circuit design tool. This top-down or schematic-driven layout (SDL) strategy is well developed in electronics, using semi-automated algorithms for placing 'functional pieces' and routing the connecting 'wires'. Electronic design is very suitable for this, as in most designs, the wires can be considered as 'just a connection', and they do not influence the overall design, for example, due to increased delay times. For many low(er) speed applications, the electric wires on the chip are just a low-loss way to transmit signals. Therefore, the placement of the functional parts with nonoverlapping wires between the different pieces is a purely geometrical problem. This, in concept simple requirement, is often solved using autorouting approaches of the wires, where the paths are typically vertical or horizontal. These are called Manhattan routing patterns. Nowadays, routing at angles other than 90° is sometimes also supported, but then at only a few fixed angles, like 30°, 45°, and 60° only.

For high-speed (RF) tracks, analog design, and high-speed (>10 GHz) digital designs, these assumptions are no longer valid as the transmission losses can become considerable and both impedance mismatches, voltage drops over the wire, and timing delays are becoming crucial as in photonics designs. For photonics, the 'wires' are, in most cases, not just simple connections. The physical properties are starting to play a role or are even determining the functionality of a component or of the whole photonic circuit. Therefore, the connecting 'wires' between building blocks or components are called 'waveguides' because the purpose of the connection is to guide an electromagnetic wave from one place to another. Remember that the telecom C-band comprises infrared wavelengths around 1550 nm, corresponding to a frequency of 193 THz. Very often, the functional pieces, themselves, consist mainly of waveguides and/or waveguiding structures with very specific requirements for individual waveguides or for combinations of waveguides. These specific requirements can vary from a very precise control over the length and width or length and width differences and even mathematically defined varying widths along the length of a waveguide (so-called tapering). This is also why a proper translation of the actual design intent for the waveguide structures into the final discretized mask file (GDS2) is very important, avoiding gridding and rounding errors.

Based on these boundary conditions, it is easily understood that a mask layout tool for photonics has some special requirements, not necessarily available in electronics mask layout tools. All angle capabilities, the ability to produce very smooth curves, and connectivity are the most important ones, and as the actual shape of the waveguide(s) are playing a dominant role, designers want to have full control over these shapes and how these shapes are connected. In 1992, the concept of parametric design was introduced for this purpose. Instead of drawing the shape, a designer sets some parameters, and software will then translate this design intent into a set of geometric shapes like polygons. Figure 5 below depicts this idea, showing a sine bend photonics waveguide primitive with the relevant parameters to be set.

Based on a library of predefined geometrical primitives, dedicated for integrated photonics, all required waveguide structures can be designed and used in larger structures or composite building blocks like a Mach-Zehnder interferometer (see Figure 6), an arrayed waveguide grating, and even full circuits. The crucial step of translating the 'design intent' into the final 'geometry' can be covered by manual coding in generic script languages like Python as used in *IPKISS* [9] or like Ample as applied in the *Pyxis* layout framework [11] used for the formulation of parametric cells. *dw2000* [12] and *OptoDesigner* [5] provide domain-specific scripting capabilities in addition to the build-in photonics-aware synthesizers as well as specific layout-preparing functionalities, removing this translation burden from the designer.

Connectivity of the individual parts of the waveguide structures and of the connections between the building

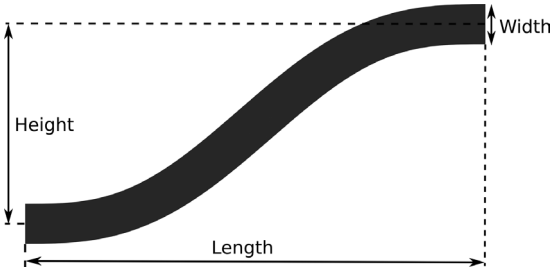

Figure 5: Example of a photonics layout primitive; a sine bend defined by the waveguide width, length, and height.

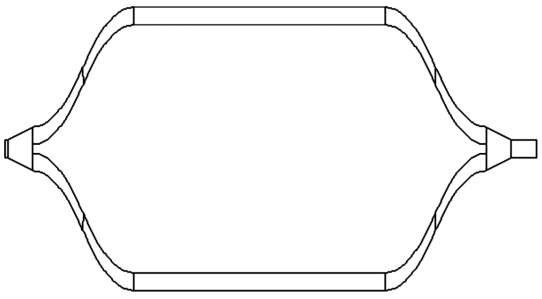

Figure 6: A (not to scale) MZI design, composed of 12 photonics layout primitives.

blocks is required to be able to make designs that contain multiple parts, without the need to manually adjust positions when changes are made to parts of the design. A good example of this is a Mach-Zehnder interferometer composite building block (Figure 6), which is constructed of several photonics primitives like junctions, bends, and straights. These individual waveguide parts all have their own parameters, depending on the actual waveguide shape or cross-section, the wavelength of interest, and the phase difference that is required. These individual waveguide parameters are normally strongly related to the composite building block parameters using often fairly simple equations, for example: the path length of one branch of a Mach-Zehnder interferometer should be a precise amount longer than the other branch. This length difference is determined by the waveguide materials, dimensions, and required filtering characteristics. When designing such a composite building block, it is very beneficial that all the individual smaller pieces stay connected when changes are made to the design based on simulation results or measurement data. The need for connectivity and the automatic translation of the design intent into the required layout instead of drawing or programming these complex polygons by hand is now easily understood.

5.1 Verification

Before a mask layout is sent to fabrication, the designed circuit needs to be verified. The initial stage is the functional verification of the applied composite building blocks (CBB), where the large parameter space (range and amount) requires a CBB designer to ensure that the combination of actual parameter values is within the scope of the implemented CBB design. An optical filter may work in theory given a perfect technology, but as a CBB user, you would expect warnings or errors when the resulting design becomes unproducible or has very low yield.

The second phase is layout validation, and this typically happens in two steps: a design-rule-check (DRC), which focuses mainly on physical features (line widths, overlays, clearance, etc.) and a layout-versus-schematic (LVS), which verifies whether the layout corresponds to the functional definition of the circuit. In electrical design, these techniques are widely established, and Mentor Graphics' *Calibre* [11] is the dominant tool, but for photonic design, LVS is currently nonexistent, and DRC capabilities in existing photonics software tools are limited compared to the state-of-the-art in EDA.

6 Process design kits

In a generic fabrication approach [13], a high-performance standardized process is made accessible together with a design kit. Such a process design kit, or in short PDK, contains a number of building blocks of which the performance and functional behavior is accurately known, and layout and functional design rules are established. Designers do not have to be concerned about how to design them; they can just take them from a library and start building a circuit and analyze and optimize it with a circuit simulator. Of course, a good knowledge about the operation of the building blocks is still important, but detailed knowledge about the process technology and the layer stack is no longer required. Therefore, the designer can concentrate on a higher abstraction level of circuit design, just like system designers who build their circuits from discrete optical components, of which they know the behavior, but not what is exactly inside the box. Photonic IC design is very similar, but now, the system is integrated on a single chip. Furthermore, the designers have some additional freedom because a number of the building blocks are parameterized, so that they can adapt their performance to specific requirements, which is not so easy with discrete components, of which only a limited number of different types will be available.

In practice, a PDK is a piece of software code(s) that can be plugged into a design environment and contains all relevant information to support the design activities. Not only the library of building blocks but also their associated compact models, layout information, simulation settings, IP information, design rules, and even measurement data can be provided in a PDK. Depending on the maturity of a design kit, the amount of information that is available to the designer will vary. Furthermore, the foundries that offer PDKs with their technology might select certain software tool-sets to be supported.

PDKs can be used internally to drive innovation in commercial organizations like IBM or STMicroelectronics and in institutes or universities to support photonics research. Today, CEA-Leti, IHP, IME, IMEC, and VTT offer Multi Project Wafer runs and PDKs for silicon photonics; FhG/HHI, Oclaro, and SMART Photonics for InP-based photonic integration; and LioniX for their TriPleX technology.

6.1 Packaging and die templates

An important and often initially overlooked aspect is the actual use of the fabricated integrated photonics chips. A

'bare die' is only practical for initial lab tests, but cannot be used outside such a special environment. Therefore, the packaging of photonics plays an important role, and dedicated and specialized packages for high performance were dominant until recent [13]. The large cost reduction of a generic approach for the fabrication of the chip is now followed by the introduction of generic and standardized packages, comparable to the electronics world where 'system in a package' (SIP) and 2.5D and 3D die integration becomes established. To enable photonics designers to design for packaging, 'package and die' templates have been introduced, which form a 2.5D integration with the high-speed electronic drivers and low-speed environmental control electronics typically within the package. To resolve the interdependent design rules between the package and the chip, package providers [14, 15] have developed PDKs with information about the placement of optical and electrical interfaces and physical form factors.

7 Standardization and collaboration

With the growing interest over the last 5 to 8 years in silicon photonics, manufactured in electronics facilities instead of dedicated photonics or multipurpose facilities, it became clear that these electronics-oriented facilities are using tools from the electronics domain. Especially for verification and sign-off, dedicated EDA tools are used. However, even in these environments, the photonics designers tend to apply specialized PDA tools, to overcome some of the limitations of the EDA tools as mentioned before. In a recent article about design challenges for silicon photonics [16], the authors conclude that the differences between photonics and electronics will require that customized solutions for photonics are further developed and integrated into existing electronics workflows. Mixed-signal simulation of photonic circuits and electronic circuits will require the integration of photonic-capable circuit simulators with existing electrical simulators. Similarly, interfaces to physical electromagnetic solvers will be needed, as photonic design cannot always be captured in an abstract model. EDA layout tools should also facilitate all angle design, and the verification tools need to be extended to support photonic concepts for functional verification. These requirements are not just true for silicon photonics; also the other photonics technologies impose the same kind of challenges on the design tools.

Driven by the identified needs to improve existing design solutions and create design flows, software vendors have started collaborating with each other and with foundries offering the fabrication processes. This resulted in several standardization and collaboration activities. First, there is the collaboration between Filarete, PhoeniX Software, PhotonDesign, and the Technical University of Eindhoven that started in 2007 and resulted in the creation of the PDAFlow Foundation [17] in 2013. This is not for profit organization for the development, support, and licensing of standards for photonic design automation. Figure 7 depicts an example of a design.

Today, the end of 2014, the PDAFlow Foundation has Lumerical, VPI Photonics, and WieWeb Software as members, in addition to the four founders. The main results of this collaboration are the development of a standard interface (API) to allow interoperability of software tools and the creation of a standard for defining PDKs. This has resulted in more than 300 designs being made and fabricated over the last 2.5 years based on these PDKs and compliant tools within multiple foundries around the world. In addition, the developed standards are being used by a wide variety of both commercial as well as academic organizations to streamline their internal design process.

Second, there is the Silicon Photonics TAB (SiTAB), organized by Si2 [18]. Si2 is a standardization organization mainly aiming at EDA solutions, to

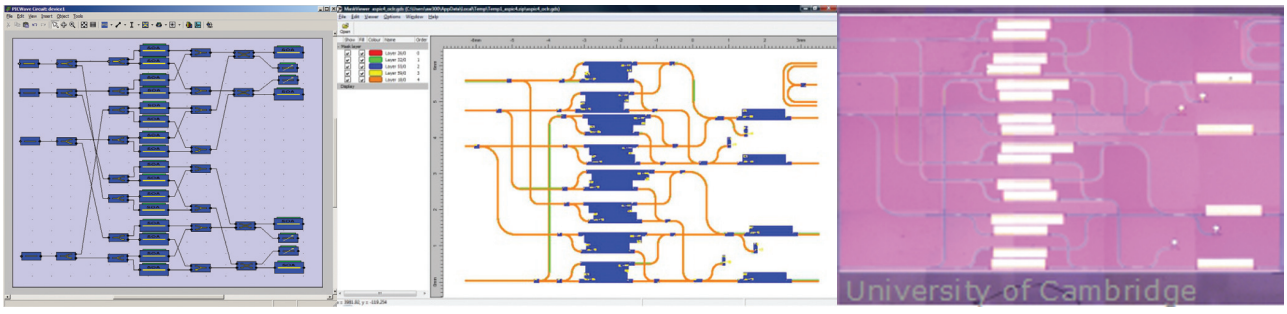

Figure 7: From circuit design with PICWave to layout with OptoDesigner to manufactured chip, demonstrating the use of a PDK-driven top-down design methodology based on PDAFlow standards.

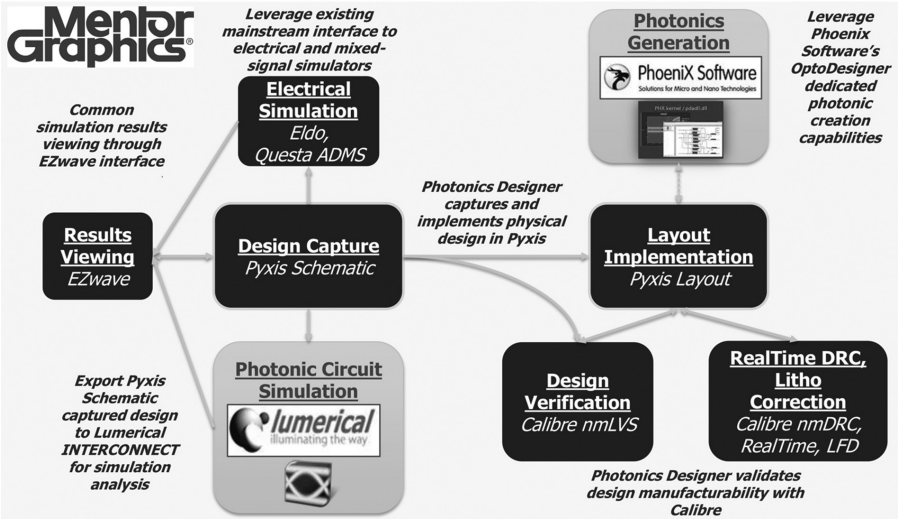

Figure 8: PDK-driven photonics electronics unified design flow.

achieve industry adoption of collaborative technology and services that deliver higher levels of silicon design integration, enabling improved design capabilities. The SP-TAB will help enable photonic-electronic design flows. The project has originated from the European Plat4M project, which is addressing electronics and photonics co-design, integration, and interoperability challenges. First results are the addition of special photonics extensions to the most widely applied design database in electronics: OpenAccess [18] and the creation of a standard to store S-matrices for photonic circuit simulations: OpenMatrices [18].

Members of SP-TAB are Aurrion, CEA-Leti, IMEC, IEF-PSUD, Luceda Photonics, Lumerical, Mentor Graphics, and PhoeniX Software.

In addition to these activities aiming at standardization, software vendors from EDA and PDA are collaborating to improve design flows for silicon and other photonics technologies, leveraging an electronics design framework by integrating photonics capabilities for simulations, layout generation, verification, and design rule checking (Figure 8) [19].

8 Conclusions

Photonics IC design tools are now around for almost 25 years and became instrumental in the growth of the photonic integration market over the last years. The design approach has evolved from a highly specialized skill focused on physical simulations related to continuous process development for each application to a circuit-driven approach for standardized and generic fabrication processes; hence, the development of several solutions and methods for photonic circuit simulations. To support these standardized fabrication processes, the use of non-shared internal libraries has been replaced by rich and powerful PDKs that are standardized through the PDAFlow Foundation. This gives fabless designers access to physical layer and circuit simulation, verification, and layout generation, all according to the technology settings and design rules of the applied technology and foundry. In the last 2 to 3 years, this resulted in more than 300 designs being made and fabricated on these PDKs and compliant tools within multiple foundries around the world. In addition, the developed standards are being used by a wide variety of both commercial as well as academic organizations to streamline their internal design process. EDA is struggling with standardization due to the large market fragmentation and fierce competition. Si2 started to play a role by creating a special group for (silicon) photonics, and in addition, the PDA community is supporting this, bringing standards and methods into the EDA domain.

References

[1] OWTNM, http://www.city.ac.uk/owtnm-2015.
[2] Photon Design, http://www.photond.com.
[3] M. T. Korthorst, P. Bakopoulos, R. Stoffer, E. Kehayas, G. Maxwell, H. Avramopoulos, Software tools for photonic integrated circuit design using "monolithic-on-hybrid integration", Micro and Nano Engineering 2008, Poster presentation.
[4] Lumerical Solutions, Vancouver, Canada, https://www.lumerical.com.
[5] PhoeniX Software, Enschede, the Netherlands, http://www.phoenixbv.com/optodesigner.

[6] VPI Photonics, Berlin, Germany, http://www.vpiphotonics.com.
[7] Optiwave, Ottawa, Canada, http://www.optiwave.com.
[8] Filarete, Milano, Italy, http://www.aspicdesign.com.
[9] Luceda Photonics, Gent, Belgium, http://www.lucedaphotonics.com.
[10] R. F. Klein Breteler, J. J. G. M. van der Tol, M. Felicetti, G. D. J. Sasbrink and M. K. Smit, Opt. Eng. 50, 07111 (2011).
[11] Mentor Graphics, Wilsonville, USA, http://www.mentor.com.
[12] Design Workshop Technologies, Ottawa, Canada, http://www.designw.com.
[13] M. Smit, X. Leijtens, H. Ambrosius, E. Bente, J. van der Tol, et al., Semicond. Sci. Technol. 29, 083001 (41pp), (2014).
[14] Technobis ipps, Alkmaar, the Netherlands, http://www.technobis.com/business-units/technobis-ipps/.
[15] Chiral Photonics, Pine Brook, USA, http://www.chiralphotonics.com.
[16] W. Bogaerts, M. Fiers and P. Dumon, IEEE J. Sel. Top. Quant. Electron. 20, (2014).
[17] PDAFlow Foundation, Enschede, the Netherlands, http://www.pdaflow.org.
[18] Si2, SP-TAB, OpenAccess, OpenMatrices, http://www.si2.org.
[19] A. Arriordaz, A. Bakker, R. Cao, C. Cone, J. Ferguson, et al., Proceedings IEEE Photonics Conference (IPC) 2014, 63–64 (2014).

Twan Korthorst
PhoeniX Software, The Netherlands,
www.phoenixbv.com,
twan.korthorst@phoenixbv.com

Twan Korthorst did his Master's degree in Electrical Engineering at the University of Twente during the final stage of which he co-founded the Microflown Team, developing the world's first particle velocity microphone. In 1996, he started as product engineer at Twente Microproducts (TMP). After the acquisition of TMP by Kymata, a Scottish-based communications solutions provider based on integrated optics, he occupied the post of Director Operations at Kymata Netherlands (later Alcatel Optronics Netherlands). Having been a manager of Operations at DEMCON Advanced Mechatronics, Twan joined PhoeniX Software from August 2007 he went on to become General Manager and later CEO.

Remco Stoffer
PhoeniX Software, The Netherlands,
www.phoenixbv.com

Remco Stoffer received his MSc degree in Applied Physics and his PhD degree from the University of Twente, The Netherlands, in 1997 and 2001, respectively. He has worked for Kymata Netherlands B.V. and Alcatel Optronics Netherlands B.V. and as a postdoc at the University of Twente for the NAIS and NanoNED projects. He is currently employed by PhoeniX Software, formerly as numerical engineer and currently as a product specialist. He has contributed to the OptoDesigner, FieldDesigner, FlowDesigner, and MaskEngineer software packages. Furthermore, he has contributed as a work package leader and senior scientist to more than 10 European or Dutch research projects and has 20 journal papers and more than 30 conference contributions to his name.

Arjen Bakker
PhoeniX Software, The Netherlands,
www.phoenixbv.com

Arjen Bakker graduated in Electrical Engineering from the University of Twente in the Netherlands in the area of design and realization of high-efficiency fiber chip coupling. As co-founder of BBV in 1991, he started his working career as the chief architect of the software, which translated the design requirements needed by the BBV design group into software solutions of which Prometheus, Selene, and OlympIOs are the most well known. In 1992, he introduced the concept of parametric design, which, nowadays, is the standard design approach in integrated photonics mask layout and modeling. After selling and successfully integrating BBV into Kymata Netherlands (later Alcatel Optronics Netherlands), Arjen co-founded PhoeniX Software in 2003 together with Jan Bos, where he has continued to occupy the post of the CTO overseeing the technology roadmaps, research projects, and the overall architecture of the software products.