PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Rapid virtual prototyping of complex photonic integrated circuits using layout-aware schematic-driven design methodology

S. Mingaleev, A. Richter, E. Sokolov, S. Savitzki, A. Polatynski, et al.

S. Mingaleev, A. Richter, E. Sokolov, S. Savitzki, A. Polatynski, J. Farina, I. Koltchanov, "Rapid virtual prototyping of complex photonic integrated circuits using layout-aware schematic-driven design methodology," Proc. SPIE 10107, Smart Photonic and Optoelectronic Integrated Circuits XIX, 1010708 (20 February 2017); doi: 10.1117/12.2252001

SPIE.

Event: SPIE OPTO, 2017, San Francisco, California, United States

Rapid virtual prototyping of complex photonic integrated circuits using layout-aware schematic-driven design methodology

S. Mingaleev[c], A. Richter[1a], E. Sokolov[c], S. Savitski[c], A. Polatynski[a], J. Farina[b], I. Koltchanov[a]
[a]VPIphotonics GmbH, Carnotstr. 6, 10587 Berlin, Germany;
[b]VPIphotonics Inc., 1 Edgewater Drive, Norwood, MA 02062, USA;
[c]VPI Development Center, Filimonova str. 15, 220037 Minsk, Belarus

ABSTRACT

We present our versatile simulation framework for the schematic-driven and layout-aware design of photonic integrated circuits (PICs) realizing a fast and user-friendly design flow for large-scale PICs comprising passive and active building blocks (BBs). We show how the seamless interaction of circuit simulation with photonic layout design tools allows to specify and utilize directly physical locations and orientations of BBs of standardized process design kits (PDKs). We demonstrate how to combine graphical schematic capture and automated waveguide routing, and discuss by means of typical design applications how an optimized design flow can speed-up the virtual prototyping of complex PICs and optoelectronic applications.

Keywords: integrated photonics, photonic circuits, schematic-driven design, layout-aware design, photonic design automation, modeling, simulation, prototyping

1. INTRODUCTION

Despite impressive advances in integrated Si and InP photonics technologies, costs associated with the design and fabrication of photonic integrated circuits (PICs) are still several orders of magnitude higher than those for their microelectronic counterparts, which limits rapid application of PICs in many areas. The need to reduce these costs has been a major driving force for the integrated photonics development during the past few years. Similar to electronics, the main approach to cost reduction is based on introducing standardized process design kits (PDKs) and a generic foundry model [1-2], and integrating it into modern electronic design automation (EDA) and photonic design automation (PDA) tools. After interesting initial attempts to extend modern EDA tools to the needs of Si photonics [3], more recently the PIC design community came to the conclusion that dedicated PDA tools, closely integrated with EDA tools, should be developed to address the specific PIC design needs [4-7].

The fundamental difference between the design processes for PICs and electronic integrated circuits is caused by the fact that optical waveguides in PICs play a twofold role. In some cases, they just route optical signals from one BB of the circuit to another, and thus, are mere connectors playing the same role as electrical wires in electronic circuits. Often, waveguide properties (length, width, shape) may be ignored in such cases. However, waveguides determine quite often the interference conditions between signals propagating along different optical paths, and thus should be considered as functional BBs of the designed circuit. Waveguide properties are of crucial importance in this case, and should be known and taken into account already at the beginning of the circuit design.

The problem is that usually these two roles of optical waveguides cannot be clearly separated. As the designed PIC grows, adding new couplers of reflective photonic components may turn any connecting waveguide into a functional waveguide. And in contrast, functional waveguides serve in most cases also as connectors of other components, so their lengths and shapes are restricted not only by the desired functions, but also by the geometrical requirements of an adequate layout connectivity. Because of this, the design process for PICs cannot be so easily separated into two stages (principal scheme design followed by layout design) as it is possible in the case of electronic circuit design. Both, the PIC principal scheme and its layout should be designed simultaneously.

As we show in Section 2, this makes the standard schematic-driven design workflow very inefficient even for Si photonics (which allows very sharp waveguide bends and thus facilitates usage of waveguides with simple Manhattan-

[1] andre.richter@vpiphotonics.com; phone +49 30 3980580; fax +49 30 39805858; vpiphotonic.com

Smart Photonic and Optoelectronic Integrated Circuits XIX, edited by Louay A. Eldada, El-Hang Lee, Sailing He, Proc. of SPIE Vol. 10107, 1010708 · © 2017 SPIE
CCC code: 0277-786X/17/$18 · doi: 10.1117/12.2252001

type routing geometries). In Section 3, we briefly discuss an alternative layout-driven design approach and outline its current limitations, mostly with respect to enabling an automated design optimization and related tasks. In Section 4 we present a novel layout-aware schematic-driven design methodology and show how it greatly simplifies the PIC design process, allowing to avoid annoying back and forth switching between circuit simulator and layout design tool. We motivate that the support of automated sweeps and optimization of layout parameters are key enabling technologies of this methodology. We implemented this design methodology by combining VPIcomponentMaker Photonic Circuits [8] with photonic layout design tools such as OptoDesigner by PhoeniX Software [9] and IPKISS by Luceda Photonics [10]. Further, we describe our support of layout-aware multi-level hierarchical designs and pay particular attention to the support of advanced parameter scripting as highly required extension of standard parameter expressions. Designing large-scale PICs, such as optical interconnect networks, becomes very inefficient when performed manually by drag-and-drop BBs using the graphical user interface (GUI). As we discuss in Section 5, automating this task in a convenient way requires development of a layout-aware macro scripting language which can access and combine capabilities of both, a circuit simulator and layout design tool. Finally, in Section 6 we summarize our recent achievements and discuss directions of future work.

2. LIMITATIONS OF STANDARD SCHEMATIC-DRIVEN DESIGN APPROACH

The standard schematic-driven approach to PICs design is based on using a set of PDK building blocks whose layout is fully determined by the BB parameters such as length for a straight waveguide, bend angle and radius for a waveguide arc, length and height of the bounding box for an S-shaped waveguide bend (S-bend), and so on. In this case, the circuit layout is fully determined by the connectivity between optical and electrical ports of neighboring BBs (except for the absolute location and orientation of the whole circuit on the die). This allows to immediately perform circuit simulations of the designed PIC, since the compact simulation model of each BB is known at any design step. Designers only need to preserve correct layout connectivity when they change BB parameters – for PICs with low degree of complexity, this seems to be a simple problem.

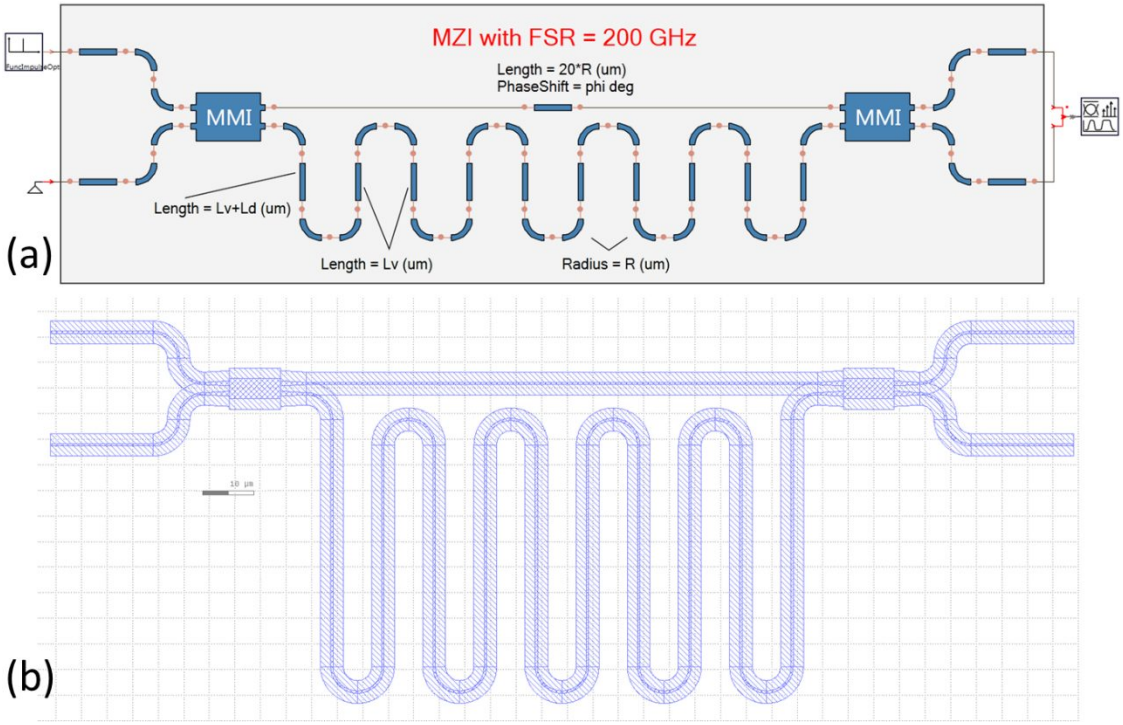

Figure 1. Example of a silicon photonics unbalanced Mach-Zehnder interferometer designed using a standard schematic-driven approach: (a) circuit setup in VPIcomponentMaker Photonic Circuits with VPItoolkit PDK DemoFab (IPKISS) extension; (b) GDSII mask of the corresponding circuit layout exported via IPKISS and visualized with KLayout.

As an example, let us consider the design of an unbalanced Mach-Zehnder interferometer (MZI) as shown in Fig. 1. This example, being one of the simplest and most common PICs, provides also a nice illustration of the twofold nature of optical waveguides in PICs. The waveguides connected to the outer ports of the 2x2 multi-mode interference (MMI) devices serve as connecting waveguides only – their lengths are not very important, and may be set to more or less realistic values and adjusted to satisfy layout connectivity at the last PIC design phase. However, the waveguides which compose the upper and bottom waveguide arms interconnecting the two MMIs are functional BBs – the whole functionality of the circuit is based on the interference of optical signals, propagating along these two arms. Adjusting the lengths of these waveguides represents the main design task.

For simplicity, let us assume that all waveguides in this circuit have the same cross-section and width, support only a single optical mode with the frequency-dependent effective mode index $n_{eff}(f)$, and that the bend radius of all bent waveguides is sufficiently large, so that the dependence of effective mode index on the bend radius may be ignored.

Then, the MZI properties are determined by the difference between phases accumulated by optical signals during their propagation along two waveguide arms, $\Delta\phi(f) = (2\pi f/c)\, n_{eff}(f)\, \Delta L$. Here, c is the speed of light in vacuum and ΔL is the difference between lengths of the bottom and upper arms. For frequencies at which $\Delta\phi(f)$ is equal to $2m\pi$ with m an integer, the MZI works as a cross-switch, while for frequencies at which $\Delta\phi(f)$ is equal to $(2m+1)\pi$, it works as a bar-switch (accounting for $\pi/2$ phase shift between output signals of MMIs). The frequencies with identical routing behavior are separated by the free spectral range (FSR) $\Delta f_{FSR} = c/n_{gr}(f)\Delta L$, where $n_{gr}(f) = n_{eff}(f) + f\,(dn_{eff}/df)$ is the group mode index at frequency f.

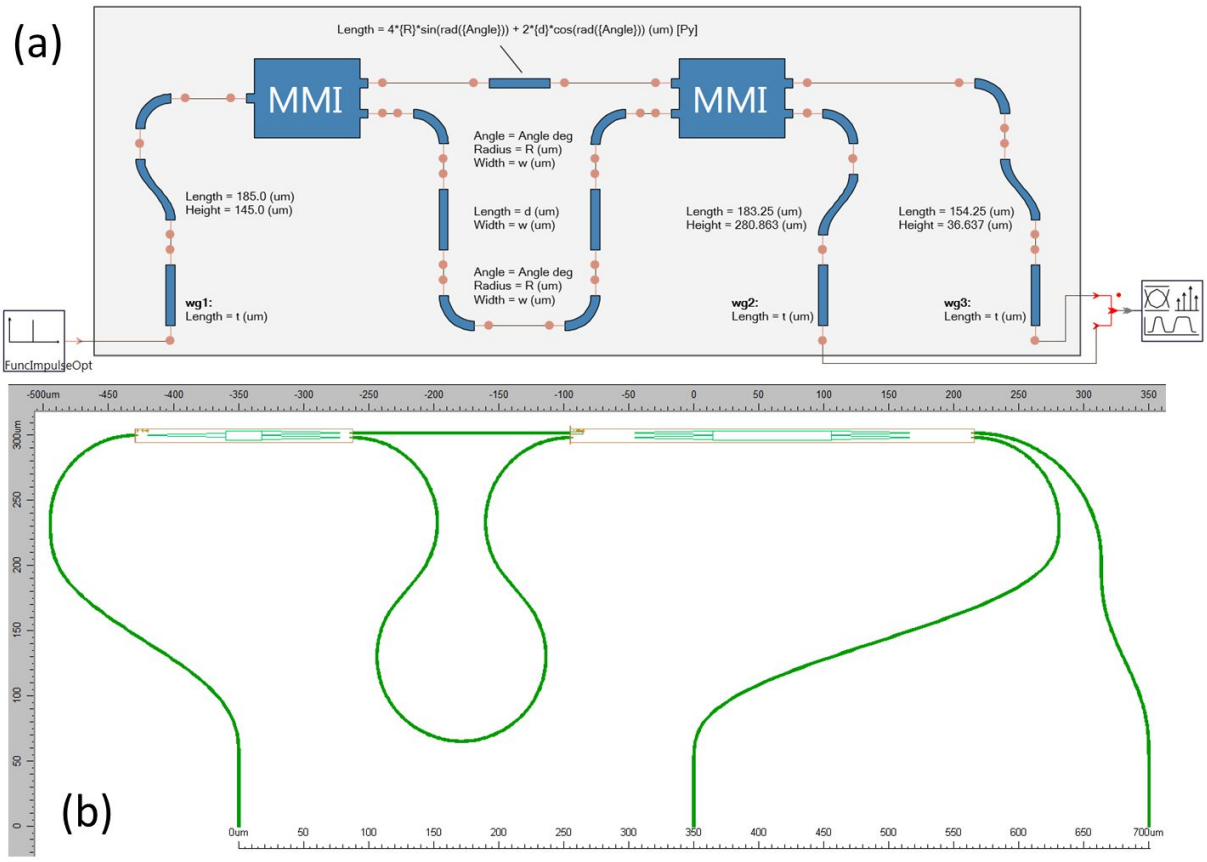

Figure 2. Example of InP photonics unbalanced Mach-Zehnder interferometer designed using a standard schematic-driven approach: (a) circuit setup in VPIcomponentMaker Photonic Circuits with VPItoolkit PDK DemoFab (OptoDesigner) extension; (b) GDSII mask of the corresponding circuit layout exported via OptoDesigner.

To design a MZI with certain Δf_{FSR} around a given reference frequency, f_0, we should use waveguide arms with length difference $\Delta L_0 = c/n_{gr}(f_0)\Delta f_{FSR}$. This is a functional requirement on the used optical waveguides. In addition, there exist also a geometrical restriction: the waveguide arms should nicely connect both MMIs – that is, their lengths in horizontal direction should be identical.

This specific problem has a simple solution when we employ silicon photonics technology. In this case, one can safely use waveguide bends with very small bend radius (say, 5 microns) and, therefore, it is natural to introduce the restriction that all the optical waveguides should be either horizontal and vertical straight waveguides, or 90-degrees waveguide bends (that is, employ the so-called Manhattan waveguide routing). This greatly simplifies the complexity of the geometrical problems which should be solved during the circuit design. Indeed, in the case of the MZI shown in Fig. 1, it is easy to derive that the geometrical restriction is satisfied if we use the length $L_s = 4NR$ for the horizontal waveguide in the upper arm, while the functional requirement is satisfied if we use lengths $L_v = (\Delta L_0 - 2L_d)/(2N) - (\pi - 2)R$ for all internal vertical waveguides in the bottom arm. Here, we assume that N is the number of waveguide loops used in the bottom arm (equal to 5 in Fig. 1), R is the bend radius, and L_d is the length increment added to the left and right vertical straight waveguides in the bottom arm to avoid intersections between waveguides of upper and bottom arms. To set the MZI resonance frequency to the desired value f_0, one should also adjust (by thermal heating, for instance) the phase shift ψ in the upper arm waveguide: $\psi = 2\pi \cdot frac\{(f_0/c)n_{eff}(f_0)\Delta L_0\}$.

Importantly, the design task above can be fully and easily solved, resulting only in simple arithmetic expressions for waveguide lengths, which are supported in any EDA and PDA tools. The only complication is that the PIC designer should know the effective and group mode indices for the employed waveguides – that is, such indices should always be provided by foundries as a standard part of their PDKs, preferably in the form of simple polynomial dependencies on waveguide width and light frequency.

However, the real tasks of PIC designers are usually much more complex. As an example, let us reconsider the same MZI design task, but add two small complications: (i) we assume that the MZI will be fabricated using a photonics technology with a lower index contrast (InP, for instance), and (ii) we take into account chip packaging requirements.

The first complication means that we should apply waveguide bends with significantly larger bend radii (hundreds of microns instead of few microns) – this forces us to use only a single large loop in the bottom MZI arm, as illustrated in Fig. 2. To minimize the device size, we cannot apply Manhattan waveguide routing with only 90-degree bends anymore, but should introduce the bend angle θ as a user-controlled parameter of the waveguide arcs in the bottom arm. This makes the geometrical problem of the layout connectivity between two MMIs more complex. Assuming that the distances between the neighboring ports in the 1x2 MMI and 2x2 MMI are identical, we can still find its analytical solution: the length of the upper waveguide arm should be equal to $L_s = 4R\sin\theta + 2d\cos\theta$, where R and d are, correspondingly, the bend radii of waveguide arcs in the bottom arm and lengths of straight waveguides placed between these arcs. The functional MZI requirement now leads to solving the transcendental equation $\Delta L_0 = 4R(\theta - \sin\theta) + 2d(1 - \cos\theta) = c/n_{gr}(f_0)\Delta f_{FSR}$ with respect to the required bend angle θ. Solving such equations goes already beyond the standard support of arithmetic parameter expressions, and requires adding support of advanced parameter scripting capabilities, outlined in more details in Ref. [5].

In real life, the distances between neighboring ports in the 1x2 and 2x2 MMIs are not identical. To solve the problem, we need not only to resolve a much more complex geometrical problem, but also to know relative port locations of the MMIs on the layout. Attempts to properly route outer waveguides connecting the MZI with custom packaging (for instance, in the assumption that they should end up along one line and be separated by 350 microns, as shown in Fig. 2) result in complex geometrical problems as well requiring knowledge of MMIs port locations on the layout. At the same time, such waveguides may have sharp bends, introducing non-negligible optical losses which should be accounted in circuit simulations.

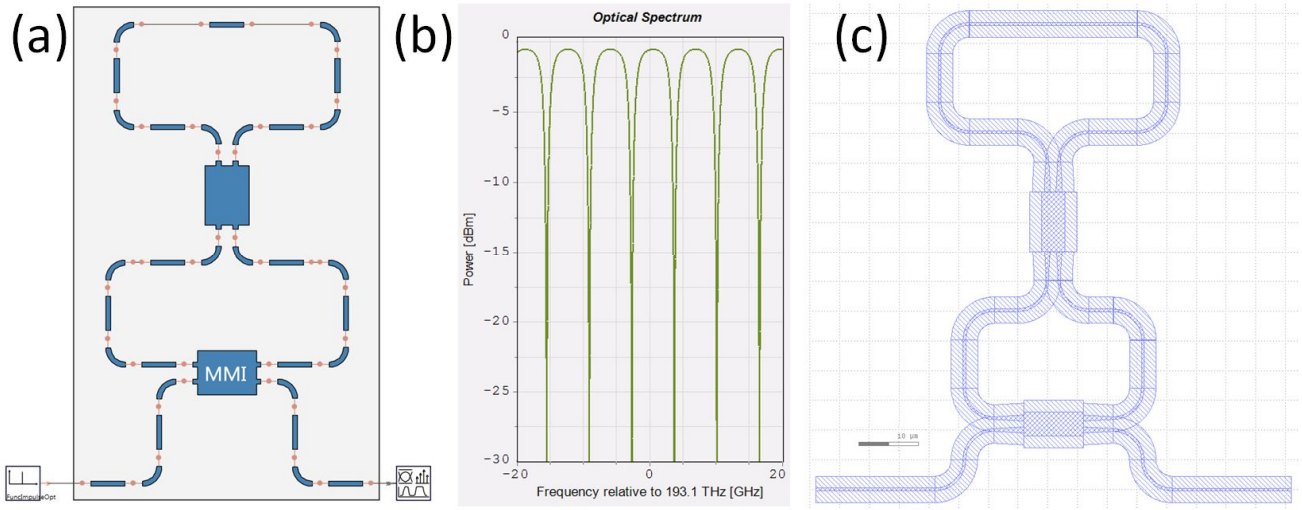

Figure 3. Example of a silicon photonics ring resonator with an internal Sagnac loop, designed using a standard schematic-driven approach: (a) circuit setup in VPIcomponentMaker Photonic Circuits with VPItoolkit PDK DemoFab (IPKISS) extension; (b) simulated device transfer function; (c) GDSII mask of the corresponding circuit layout exported via IPKISS and visualized with KLayout.

Importantly, the problems described in this example may appear also for silicon photonics chips with a simple Manhattan waveguide routing. As an illustration, let us consider an MMI-based ring resonator with an internal Sagnac loop, as shown in Fig. 3. Although resolving the geometric layout connectivity problem is sufficiently simple here, it again requires knowledge of relative MMI port locations on the layout. Furthermore, resolving the functional problem of having user-defined FSR and resonance frequencies cannot be formulated in terms of analytical nonlinear equations any more: its solution requires knowledge of group delays and phase shifts accumulated inside the MMIs, as well as their reflectivity properties, which are not handled by simple analytical expressions, even in the first approximation.

In practice, attempts to design more or less complex PICs employing the standard schematic-driven design approach results in very time-consuming back and forth iterations between the circuit simulator and layout tool. Moreover, such design tasks such as device performance and yield optimization (being a standard for EDA tools) cannot be automated and thus become merely unmanageable.

3. LAYOUT-DRIVEN DESIGN APPROACH

Currently, the problems outlined above can be solved by employing a layout-driven design approach. For example, dedicated photonic layout design tools, such as OptoDesigner [9] and IPKISS [10], provide advanced scripting capabilities with support of an easy netlist-based PIC design process and immediate access to relative and absolute locations of any BB ports inside the circuit netlist. Moreover, they provide advanced routing capabilities, including support of “elastic waveguide connectors”, which allow designers to easily connect any two given BB ports without manual solving complex geometrical problems. Furthermore, they provide basic simulation capabilities, in particular access to the effective and group mode indices of the used optical waveguides, thus allowing to easily design such PICs as illustrated in Figs. 1-2, and even proceed with the design example in Fig. 3.

However, the simulation capabilities of layout design tools remain limited (as we discuss in [5], accurate and efficient simulation of modern large-scale PICs with a mix of passive and active BBs requires development of complex specialized circuit simulation engines). Because of this, the layout-driven design approach should include the simulation of the designed PIC with a dedicated photonic circuit simulator as a separate stage. But simulations of complex non-passive circuits require embedding of the designed PIC into a complex simulation framework: adding properly configured electrical and optical sources, post-processing of output signals with specialized (and very often customized) instrumentation tools, and visualization of obtained results.

As a result, designers need again to frequently switch between a layout design tool and circuit simulator, and experience serious difficulties trying to organize automated sweeps over layout parameters or perform automated design optimization.

4. NEW LAYOUT-AWARE SCHEMATIC-DRIVEN DESIGN APPROACH

As an efficient practical solution to all the problems outlined above, we suggest a new layout-aware schematic-driven design approach [7]. With this approach, both circuit simulation and layout design tasks are performed simultaneously, using the same schematic capture environment. For this, a circuit-level simulator is seamlessly integrated with a layout design tool, providing designers transparent access to the capabilities of both software packages, as illustrated in Fig. 4.

In particular, it becomes possible to directly specify in the circuit-level simulator physical locations and orientations of PDK BBs on the final layout (when this is required by DRC and packaging specifications or layout optimization). Also, the library of PDK BBs is extended with a set of smart elastic waveguide connectors (the same as provided by the underlying layout design tool), allowing to easily interconnect sub-circuits having fixed locations without the need to manually solve complex geometrical problems.

The main problem with implementing this approach is that lengths and shapes of elastic waveguide connectors are not instantly known to the circuit simulator, though, they need to be known to perform circuit simulations. To resolve this problem, the circuit simulator automatically and invisibly for users invokes the layout design tool to determine the actual physical lengths and shapes of all elastic connectors, constructs compact simulation models for them, and after that initiates the circuit simulations. This allows combining graphical schematic capture, layout design with automated waveguide routing, and circuit simulations, which are currently considered separately representing a major problem for PIC designers.

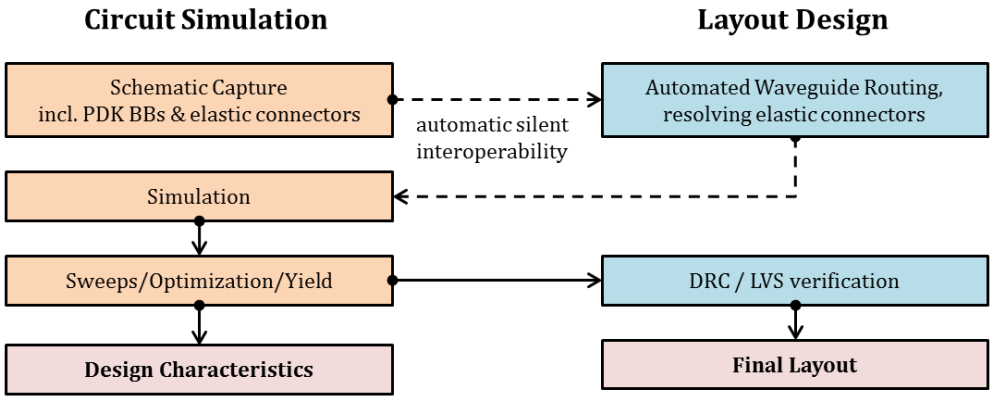

Figure 4. Layout-aware schematic-driven design methodology and its implementation with VPIcomponentMaker Photonic Circuits and OptoDesigner.

4.1 Example of Mach-Zehnder interferometer design

As an illustration of this design workflow, let us reconsider the design example from Fig. 2. Now we can replace all the problematic waveguides with elastic waveguide connectors, as shown in Fig. 5. The physical locations and orientations of all the outer waveguides can now be directly specified with absolute coordinates, allowing to easily fulfill our packaging requirements. Finally, the physical location and orientation of the MZI sub-circuit can now be specified using relative coordinates with respect to the location of the output port of the left bottom waveguide (denoted as "wg1@out0", according to the OptoDesigner notation [9]). All such port locations can be specified using advanced parameter scripts – so that, if required, locations of BBs can be specified after solving arbitrarily complex equations. That is, the circuit may now be designed in almost the same way as it would be designed with the layout-driven approach.

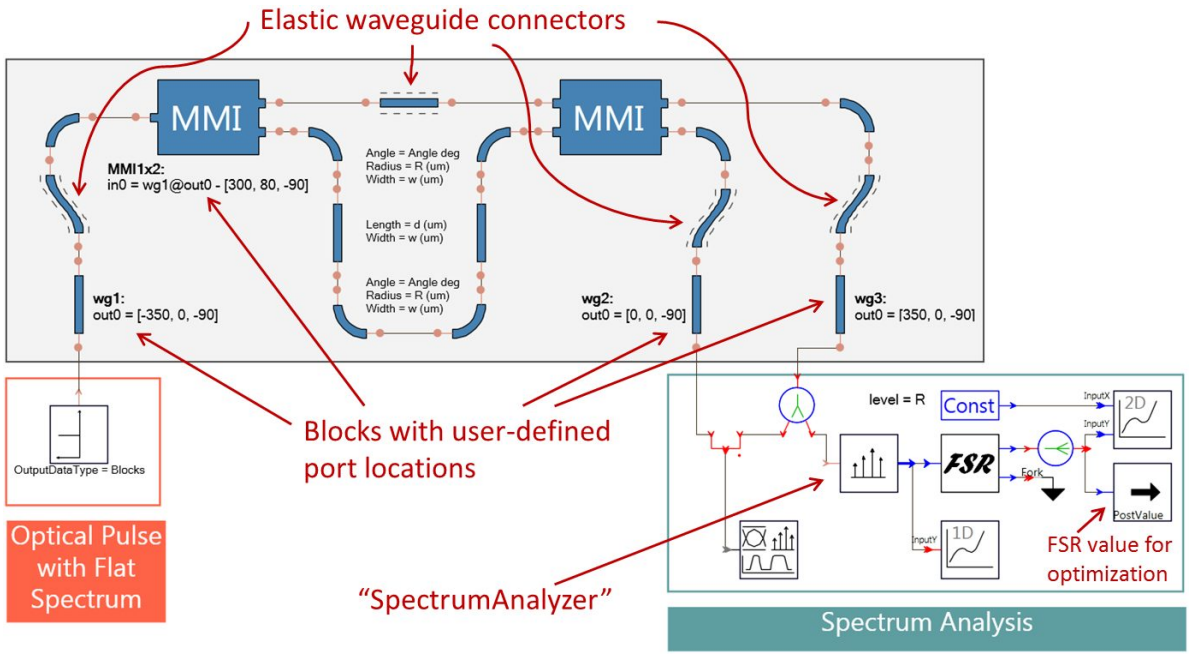

Figure 5. Example of the same MZI as shown in Fig. 2, but now designed using a layout-aware schematic-driven approach. Notice how the usage of elastic connectors and user-defined port locations solves all the main difficulties of the design in Fig. 2. Notice also the usage of advanced instrumentation for automated signal post-processing.

4.2 Support of automated parameter sweeps and optimization

At the same time, since this design is still performed in a circuit simulator environment, one can easily combine PDK BBs with any signal sources and instrumentation modules required for signal post-processing and organization of parameter sweeps and/or optimization. As an example, the "Spectrum Analysis" block in Fig. 5 allows to visualize both "BAR" (from waveguide "wg3") and "CROSS" (from waveguide "wg2") transfer function spectra (shown in top chart on Fig. 6), automatically extract the peak properties from "BAR" spectrum using the built-in instrumentation module "SpectrumAnalyzer" (shown in left bottom chart on Fig. 6), and finally estimate and plot the device FSR (shown in right bottom chart on Fig. 6).

Importantly, our implementation of the layout-aware schematic-driven design approach supports automated sweeps and optimization of layout parameters (such as the bend radius R of waveguide arcs inside the MZI in Fig. 5 – notice that changing its value affects the whole circuit layout). For this, the layout design tool is automatically invoked to determine the actual physical lengths and shapes of all elastic connectors at each simulation step.

As an example, the right bottom chart on Fig. 6 shows the dependence of MZI FSR on the bend radius R as a result of an automated parameter sweep, which can be easily organized by a PIC designer by merely specifying the name and value range of the swept parameter. In addition, Fig. 7 illustrates automated optimization of the same bend radius for getting the desired FSR of 200 GHz (in few seconds converging to the optimum value R = 61.64 microns).

Other frequently required simulation capabilities, such as multi-dimensional parameter sweeps and optimization, combination of sweeps with optimization, sensitivity and yield analysis, as well as yield optimization are also supported with this approach (although enabling easy user-friendly access to such more advanced capabilities is still under active development).

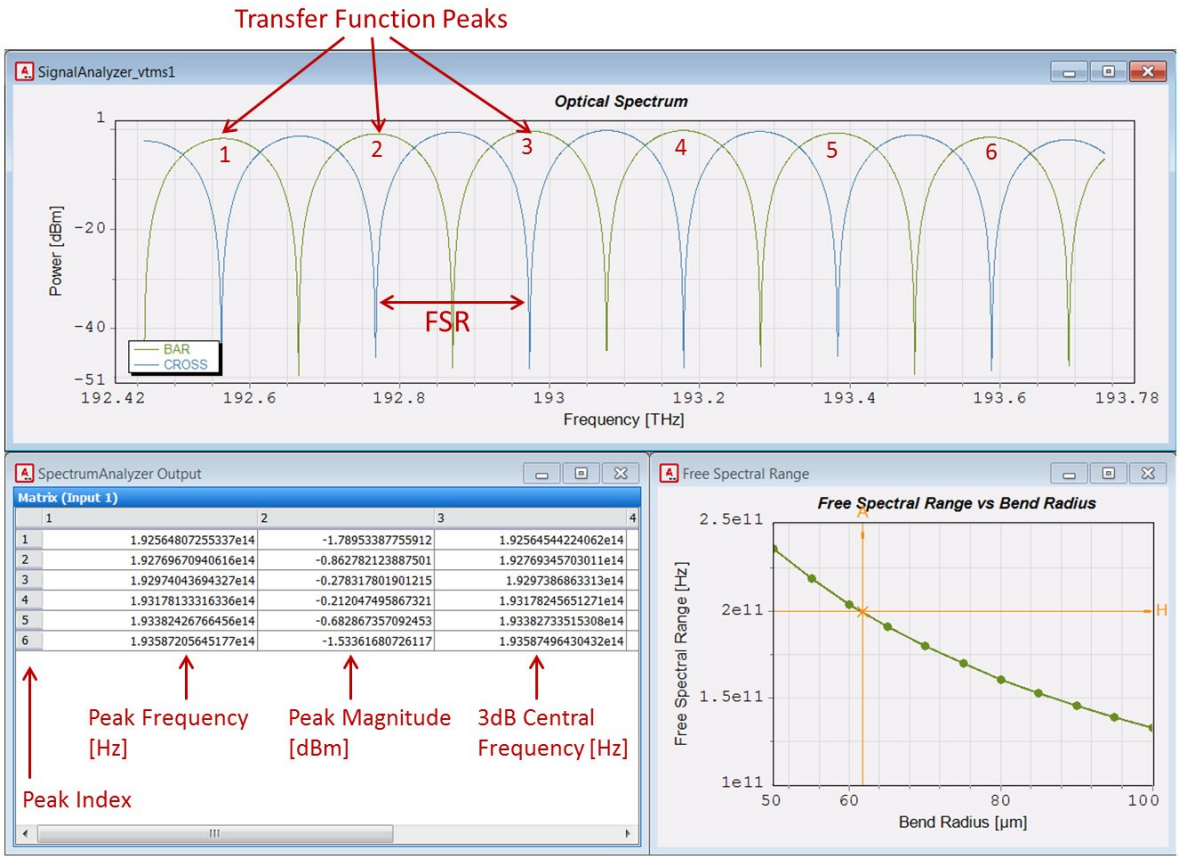

Figure 6. Results of automated sweep over the bend radius for MZI shown in Fig. 5. Here, the MZI transfer function (shown in top chart) is automatically analyzed by "SpectrumAnalyzer" module which extracts properties of all its peaks (such as peak frequency and magnitude, 3dB bandwidth, passband ripple, etc), shown on the bottom left chart. These peak properties are then used to calculate the average Free Spectral Range of the designed MZI, plotted as function of bend radius on the bottom right chart.

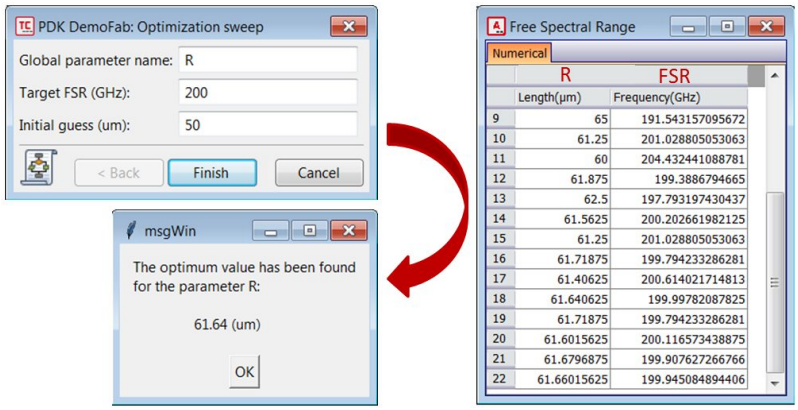

Figure 7. Results of automated optimization of the bend radius *R* for MZI shown in Fig. 5 with respect to getting the desired Free Spectral Range of 200 GHz.

4.3 Support of conversion between standard waveguides and elastic connectors

In practice, the PIC design usually starts with elaborating a circuit idea – at this stage, the circuit layout is ignored, and designers concentrate on elaborating a principal scheme which enables the required circuit functionality. This stage can be addressed by a standard schematic-driven design methodology, based on the usage of standard PDK waveguides.

At the second stage, the circuit layout is taken into account ensuring layout connectivity and accounting for packaging requirements. To simplify transition from the first to the second stage, we provide the possibility to easily replace all selected standard waveguides with their elastic counterparts, and thus introducing layout flexibility which is used to optimize the circuit design and easily satisfy all packaging requirements. This is the stage where the layout-aware schematic-driven approach is most intensively used.

However, once the design of the optimal circuit design is fully elaborated, it may be a good idea to "fix" its layout. For this, we provide the possibility to easily replace all (or all selected) elastic waveguide connectors with their standard counterparts, fully preserving the optimal circuit layout. This permits to eliminate elastic connectors from the circuit, and thus, significantly speed-up its simulations. Such a circuit may now be automatically converted into a compound building block and be used as part of a more complex circuit, as we discuss in the next subsection.

4.4 Support of multi-level hierarchical designs

Similar to electronics, design of large-scale PICs requires splitting the whole circuit into manageable functional sub-circuits, which are possibly further decomposed into yet smaller sub-circuits until the level of foundry-certified PDK BBs will be reached. This approach, commonly called "hierarchical design approach", is natively supported nowadays in modern circuit simulators and layout design tools.

Its support together with the layout-aware schematic-driven design approach is, however, non-trivial and deserves a special consideration. Our implementation of this functionality is illustrated in Fig. 8 for an example of a cascaded-MZI demultiplexer. Here, the input optical signal is split between two cascaded MZIs, both being instances of the same user-defined parameterized compound BB "CascadedMZI" with only different parameter settings. The "CascadedMZI" block contains, in its turn, three instances of another compound BB "MZI", made of only PDK BBs and elastic connectors.

Importantly, we assume that user-defined compound building blocks should support the same design workflow as built-in PDK BBs. That is, they should provide users with the possibility to specify physical locations and orientations of their ports on the final layout. In our example on Fig. 8, this mechanism is used to specify locations of both instances of the "CascadedMZI" block.

Another important functionality is the possibility to create custom compound BBs that are described by functional design parameters rather than structural layout parameters. In the case of the "MZI" block in Fig. 8, we employ design parameters such as "FreeSpectralRange" and "ResonanceFrequency" instead of specifying waveguide lengths and bend angles. As we already discussed in Section 2, this requires the application of parameter scripting, so that the required waveguide lengths and bend angles are calculated by solving a system of nonlinear equations derived from the condition to obtain a maximally compact device layout enabling the desired "FreeSpectralRange" and "ResonanceFrequency" (see [5] for details on parameter scripting support).

The hierarchy at circuit simulator level is maximally preserved when exporting the layout to the layout design tool, ranging from a fully isomorph circuit representation (one-to-one correspondence of compound building blocks in the case when only simple arithmetic expressions are used to define BB parameters) to one-to-many correspondence (in the case when advanced parameter scripting is used at the circuit simulator level, which is not supported by the layout tool).

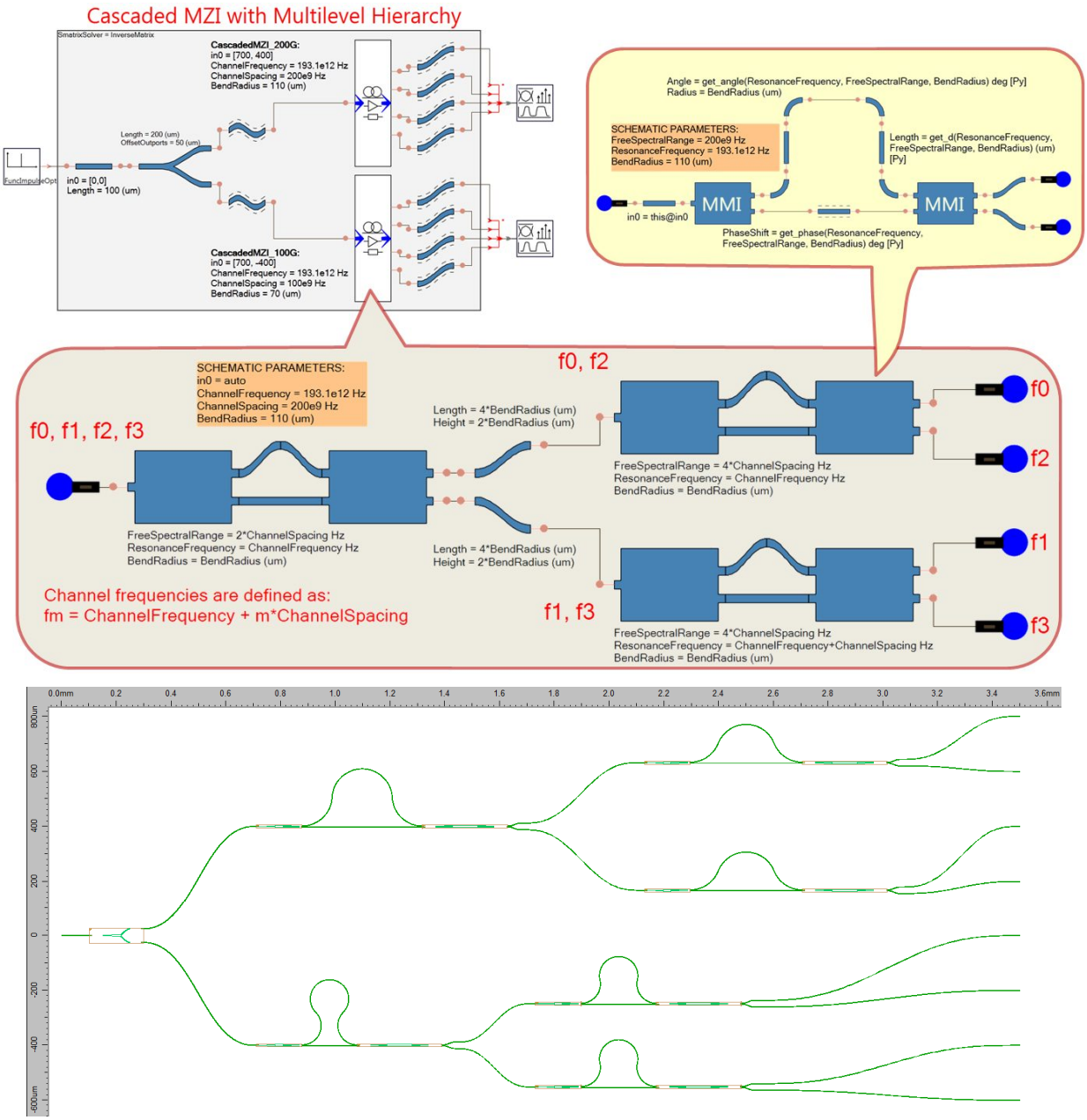

Figure 8. Example of cascaded MZI designed with the layout-aware schematic-driven design approach employing multi-level hierarchy. The layout (bottom) is exported through OptoDesigner.

4.5 Example of dynamic WDM channel equalizer design

To complete this section, let us illustrate application of the layout-aware schematic-driven approach to the design of a more complex InP photonics circuit: an Arrayed Waveguide Grating (AWG)-based dynamic WDM channel equalizer, as shown in Fig. 9.

The input signal (consisting of five WDM channels spaced by 200 GHz) is demultiplexed using 1x5 AWG, so that each channel is carried by its own waveguide. After passing through five fast dynamic phase shifters, these channels are multiplexed back by 5x1 AWG. Importantly, by using the 1x2 MMI at the beginning and 2x1 MMI at the end of the circuit, we effectively organize five independent Mach-Zehnder interferometers for each of the channels. Because of this,

the output amplitude of each channel can be independently varied over a broad range by adjusting the voltage applied to each of the phase shifters. Key design tasks for this circuit include achieving smooth layout connectivity (especially between the angled AWG waveguides and bottom phase shifters) and obtaining adequate FSR values (significantly exceeding the channel spacing) for each of the internal MZIs.

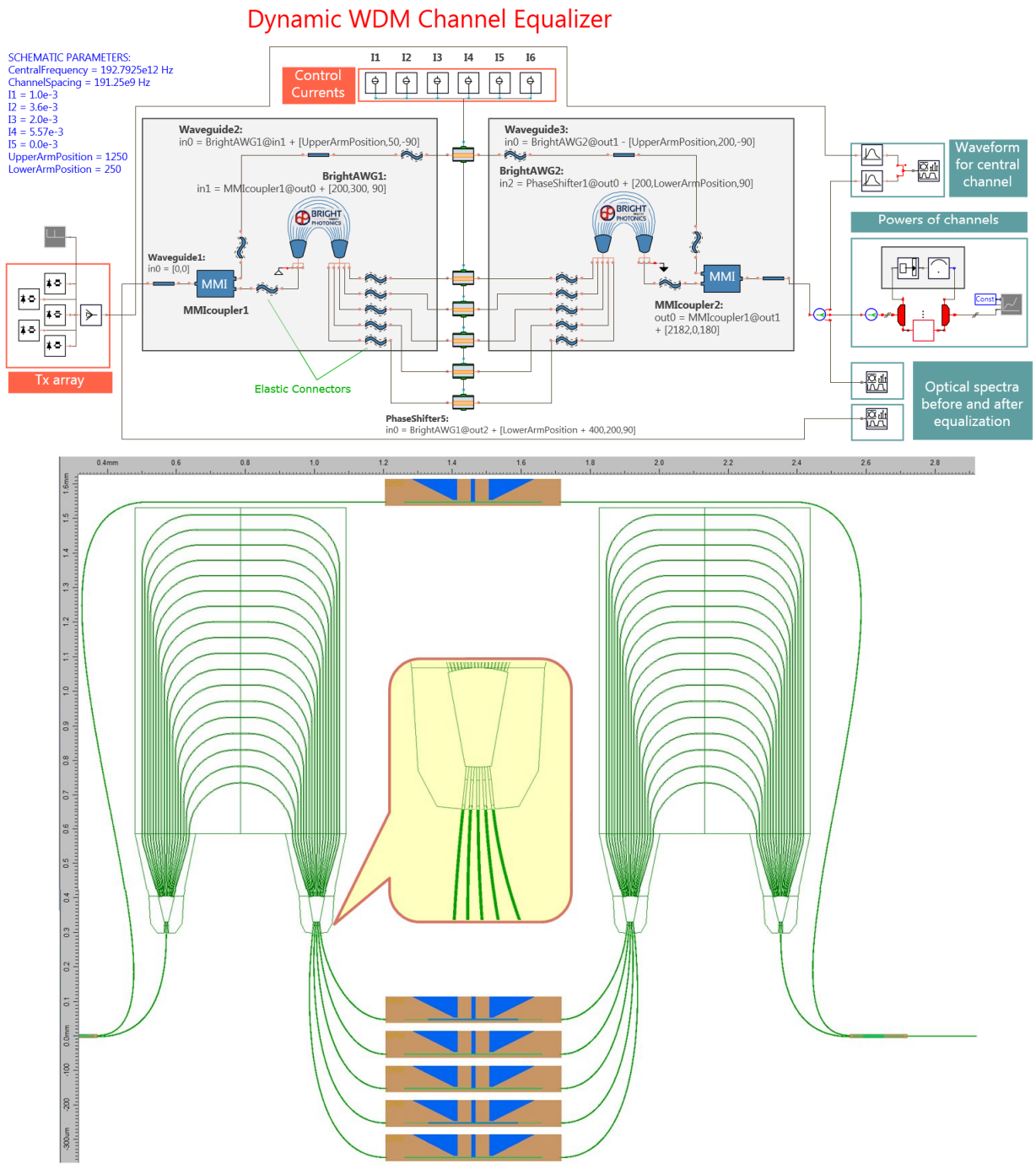

Figure 9. Example of designing an AWG-based dynamic channel equalizer for wavelength division multiplexing (WDM) applications based on layout-aware schematic-driven design approach. Top: circuit setup in VPIcomponentMaker Photonic Circuits with VPItoolkit PDK DemoFab (OptoDesigner); bottom: mask layout in OptoDesigner.

The first of these tasks is easily solved by using free-shape elastic connectors (based on "pxConnectorCurve" blocks in OptoDesigner), while the second task is addressed by automated sweeping the positions of the phase shifters until proper FSRs for all internal MZIs are obtained. Clearly, any additional packaging requirements can be easily satisfied as well with this approach by adding a few more elastic waveguide connectors.

5. TOWARDS LAYOUT-AWARE MACRO SCRIPTING

As we discussed in section 4.4, one way to handle the complexity of large-scale PIC designs is the usage of a hierarchical design approach. However, this approach alone is not always sufficient – especially in applications addressing the design of large-scale optical networks on chip (NoC), as is illustrated in Fig. 10.

The major problem in designing such networks is given by properly interconnecting all optical paths between the optical switches, which is complicated by resolving the waveguide crossings. An NxN switching network requires N^2 interconnections inside the circuit, which may be organized in different ways: minimizing the total number of waveguide crossings (thus minimizing total optical power losses), or trying to balance the number of crossings for each optical path (thus balancing optical power losses). Depending on the optical properties of the used components (waveguides, bends, and crossings), either one or another scheme may become most efficient. Therefore, optimization of such networks requires construction and then simulation of a large number of cumbersome circuits. Trying to construct such networks manually, by dragging, dropping, and connecting many hundreds of BBs, is an extremely inefficient and error-prone process.

The solution to this problem is provided by supporting macro scripting capabilities in the circuit simulator, which permit to place and connect any BBs on the schematic and adjust their parameters in a programmatic way instead of using drag-and-drop capabilities of a graphical user interface (GUI). VPIcomponentMaker Photonic Circuits supports macro scripting with two general-purpose scripting languages: Python and TCL, each extended by a library of special commands which allow performing almost any task that can be performed through the standard GUI. Usage of Python macro scripting is especially convenient since it provides designers with very powerful object-oriented programming capabilities and is closely integrated, in particular, with VPImodeDesigner [11] (which can be used to calculate on-the-fly any missing waveguide properties) and the SciPy ecosystem [12] (which provides an advanced set of optimization algorithms).

However, successful designing complex optical networks as shown in Fig. 10 with macro scripting requires extension of the macro scripting language with an additional set of "layout-aware" commands, which allow to determine physical locations and orientations of BB ports on layout, locations of waveguide crossing, and so on. Enabling such capabilities (which are standard for layout design tools) in circuit simulators requires even closer interfacing between the circuit and layout tools compared to the "standard" layout-aware schematic-driven design approach described in section 4. Although this is still work in progress, first results on this way show already that enabling layout-aware macro scripting inside circuit simulators is indeed a favored approach to handle complex PIC design problems efficiently.

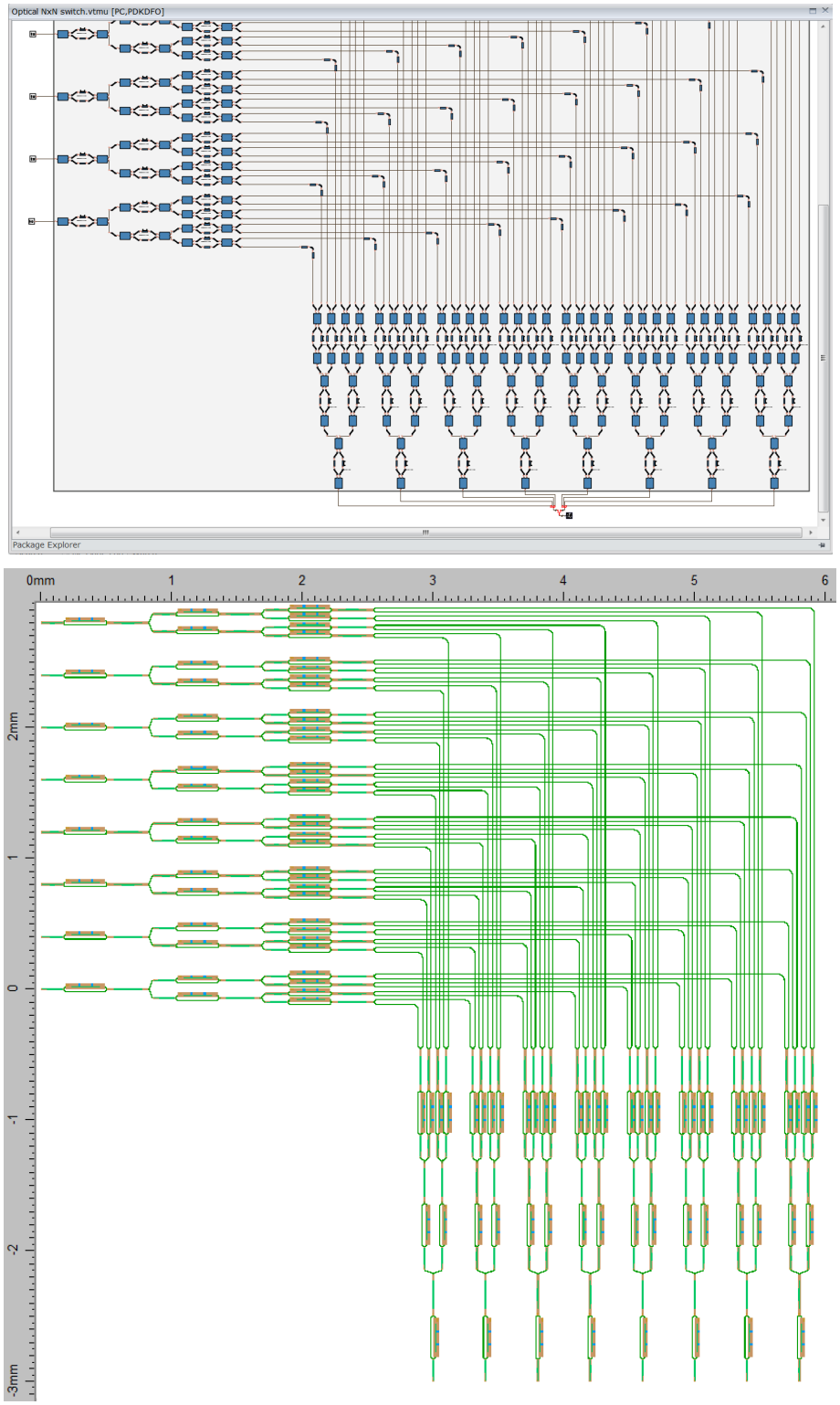

Figure 10. Example of a large-scale optical interconnect switching network, designed using layout-aware macro scripting with VPIcomponentMaker Photonic Circuits; its layout is exported through OptoDesigner.

6. CONCLUSIONS

We presented the main ideas and advantages of the layout-aware schematic-driven PIC design methodology, which enables a smooth photonic circuit design workflow starting with its logical idea and allowing to gradually elaborate (still staying with a circuit simulator) its layout at any desired level of detail, easily incorporating requirements of adequate layout connectivity, packaging, and functional specifications. This methodology is based on extending a circuit-level simulator with the capabilities of layout design tools, such as the capability to specify directly physical locations and orientations of PDK building blocks on the final layout and to connect sub-circuits having fixed locations by smart elastic optical connectors. Consequently, graphical schematic capture and automated waveguide routing can be combined, which otherwise need to be addressed separately representing a major problem for PIC designers.

Key enabler for this functionality is the seamless integration of circuit and layout tools: a circuit simulator automatically and invisibly for users invokes a layout design tool to determine the actual physical lengths and shapes of all elastic connectors, constructs compact simulation models for them, and after that initiates the circuit simulations. Importantly, the presented approach enables automated parameter sweeps and optimization even for parameters that affect the circuit layout – a task which cannot be performed by standard schematic-driven or layout-driven design approaches. Based on our experience, we predicate that, driven by the fundamental difference between electronic and photonic circuits, the future development of PDA tools will proceed further with much closer integration between circuit simulation and layout design tools than it was ever required for EDA tools.

REFERENCES

[1] Smit M. et al., "An introduction to InP-based generic integration technology," Semiconductor Science and Technology 29 (8), 083001 (2014).
[2] Melati D. et al., "Validation of the building-block-based approach for the design of photonic integrated circuits," J. Lightwave Technology 30 (23), 3610-3616 (2012).
[3] Mekis A. et al., "Computer-aided design for CMOS photonics," [Silicon Photonics for Telecommunications and Biomedicine], CRC Press (2011).
[4] Bogaerts W. et al., "Integrated design for integrated photonics: from the physical to the circuit level and back," Proc. SPIE 8781, 878102 (2013).
[5] Mingaleev S. et al., "Towards an automated design framework for large-scale photonic integrated circuits," Proc. SPIE 9516, 951602 (2015).
[6] Heins M. et al., "Design Flow Automation for Silicon Photonics: Challenges, Collaboration, and Standardization," [Silicon Photonics III], Springer, Berlin 99-156 (2016).
[7] Mingaleev S. et al., "Layout-Aware Schematic-Driven Design Methodology for Photonic Integrated Circuits," European Conference on Integrated Optics, p. 21 (2016).
[8] VPIcomponentMaker Photonic Circuits, http://vpiphotonics.com/Tools/PhotonicCircuits (06 January 2017)
[9] OptoDesigner, http://www.phoenixbv.com/optodesigner (06 January 2017)
[10] IPKISS, http://www.lucedaphotonics.com/en/product/ipkiss-eda (06 January 2017)
[11] VPImodeDesigner, http://vpiphotonics.com/Tools/ModeDesigner (06 January 2017)
[12] SciPy, See http://scipy.org (06 January 2017)