LUCEDA PHOTONICS

Designing Photonic ICs

Pieter Dumon

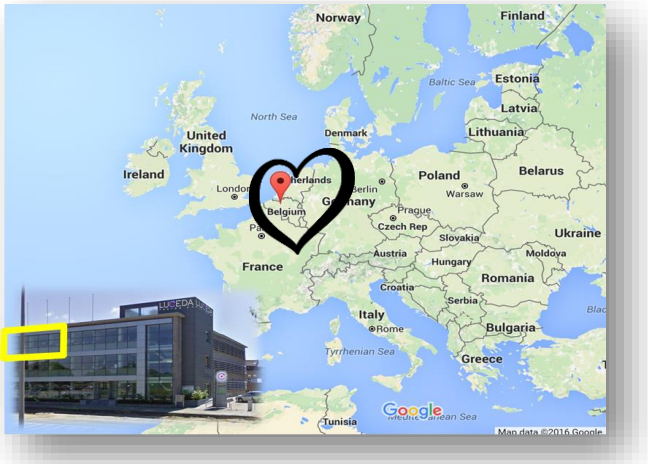
07/09/2022

EPIC Meeting on CMOS Compatible Integrated Photonics

Overview

- Intro to Luceda and our software
- What we believe in
- Where we’re going

Luceda Photonics



Photonic IC design software

Mission:

Help photonic IC designers enjoy the same **first-time** right experience as electronic IC designers

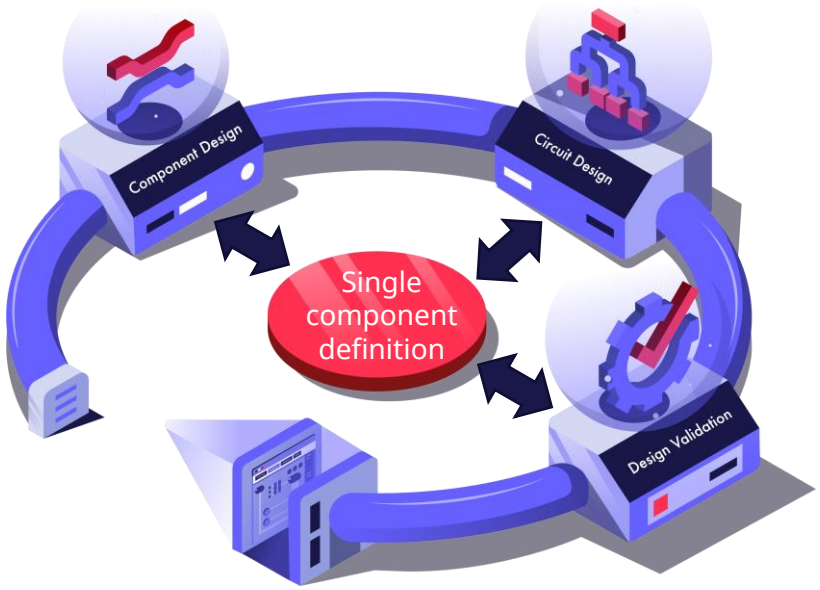
Facts & Figures

Founded in 2014 as spin-off from imec, Ghent University & VUB

HQ in Dendermonde, Belgium
China branch in Shanghai

Growing international team (>20 and hiring)

IPKISS Design Platform



- IC implementation: layout, placement, routing
- GDSII tape-out
- Virtual fabrication
- Device simulation (EM)
- Circuit simulation (behavioral/compact model)
- Netlist extraction and circuit verification
- Standard language: Python
- PDKs for wide range of photonic foundries

Additional modules

IPKISS AWG Designer – synthesis, implementation and simulation of AWG

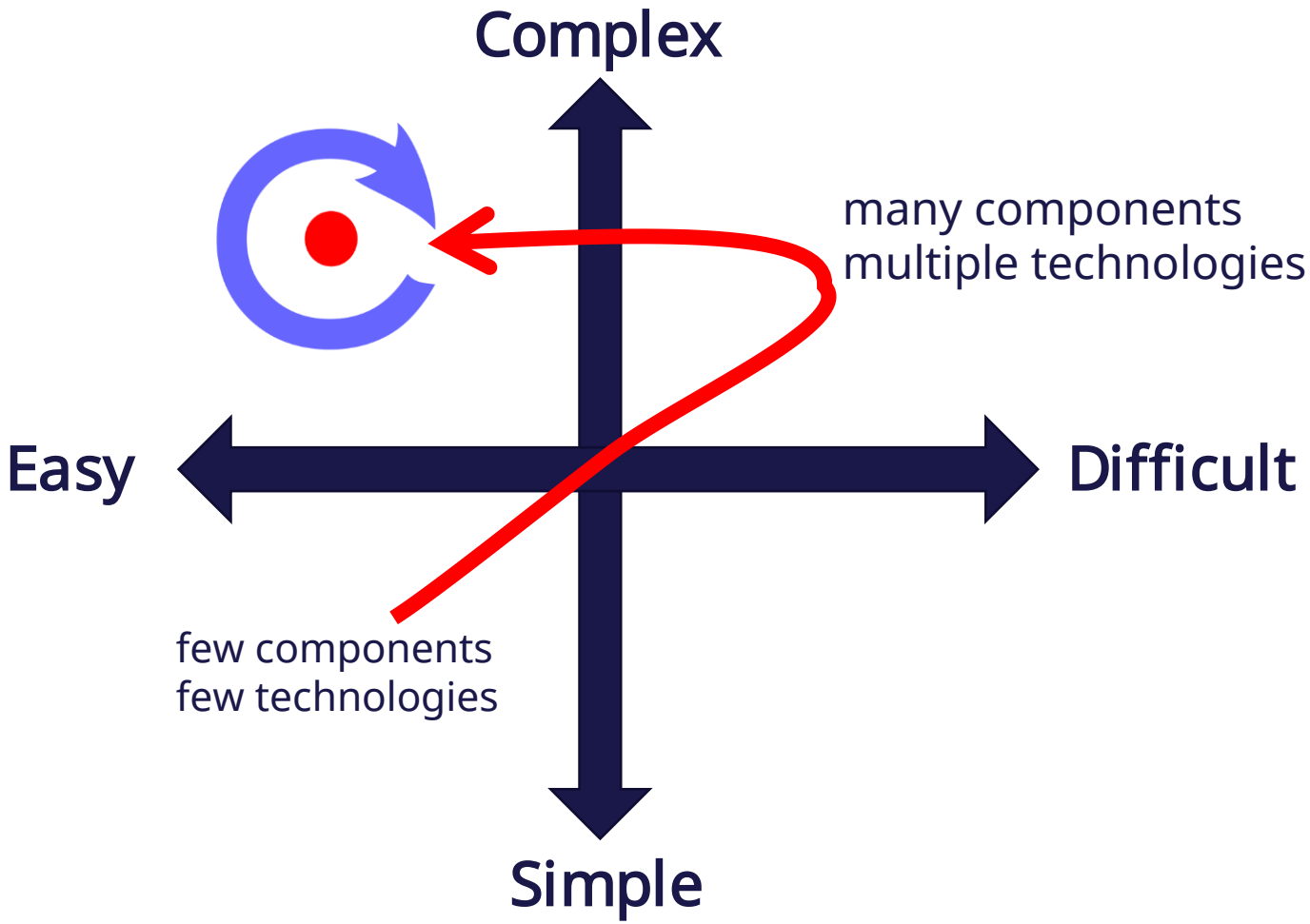
IPKISS IP Manager – regression testing and management of design IP for teams

IPKISS Link for Ansys Lumerical – automation of FDTD and MODE simulations

IPKISS Link for Dassault Systèmes Simulia – automation of EM simulations

IPKISS Link for Siemens EDA – link with L-Edit for graphical electrical IC design

Photonic IC Design



What we believe in

How to make complex photonic IC design easy?

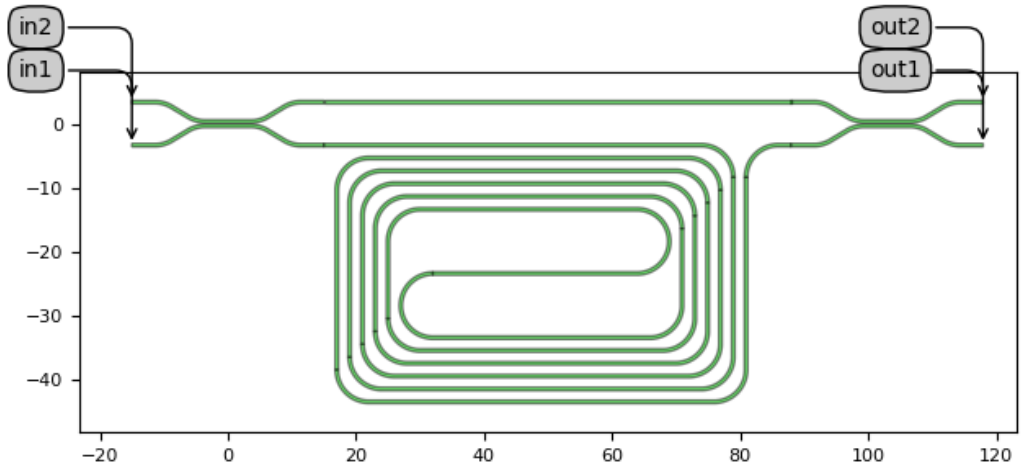
- Automation – Luceda **expertise** built-in
- Toolbox – **empowerment** of the designer, **flexible** and **human centric**

What we believe in

How to make complex photonic IC design easy?

- Use of specification-driven design and HDL – *correctness and automation*
- Direct coupling of specification, layout and simulation – *avoid mistakes*
- High degree of layout automation – *productivity* + *avoid mistakes*
- But low-level functions available – *do what needs to be done*
- Manage design IP – *persist and control design team knowledge*

Specification-driven placement and routing




```python
from picazzo3.wg.spirals import FixedLengthSpiralRounded

class MZI(i3.PCell):
    class Layout(i3.LayoutView):
        def _generate_instances(self, insts):

            spiral = FixedLengthSpiralRounded(name=self.name + "_SPIRAL",
                                              trace_template=wg_t, n_o_loops=3, total_length=1000)
            spiral_lo = spiral.Layout()

            insts += i3.place_and_route(
                insts={
                    'splitter': dc,
                    'combiner': dc,
                    'arm1': spiral,
                },
                specs=[
                    i3.Place('splitter', (0, 0)),
                    i3.Join([
                        ('splitter:out1', 'arm1:in'),
                        ('combiner:in1', 'arm1:out'),
                    ]),
                    i3.FlipV('arm1'),
                    i3.ConnectManhattan('splitter:out2', 'combiner:out2'),
                ]
            )
            return insts

        def _generate_ports(self, ports):
            return i3.expose_ports(self.instances,
                                   {'splitter:in1': 'in1',
                                    'splitter:in2': 'in2',
                                    'combiner:out1': 'out1',
                                    'combiner:out2': 'out2'})

mzi = MZI()
mzi_lo = mzi.Layout()
mzi_lo.visualize(annotate=True)
```

Built-in components

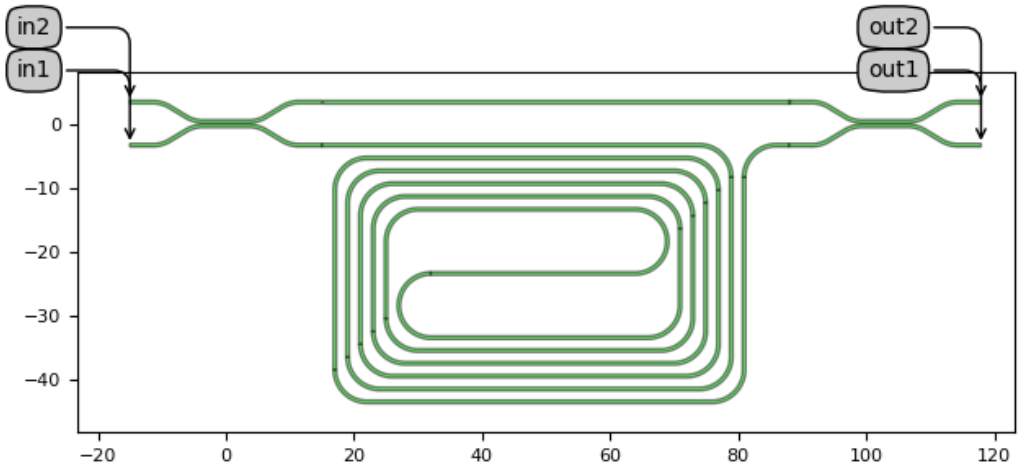
Luceda automates

Placement and routing specs

External connectivity spec

Valid and complete layout

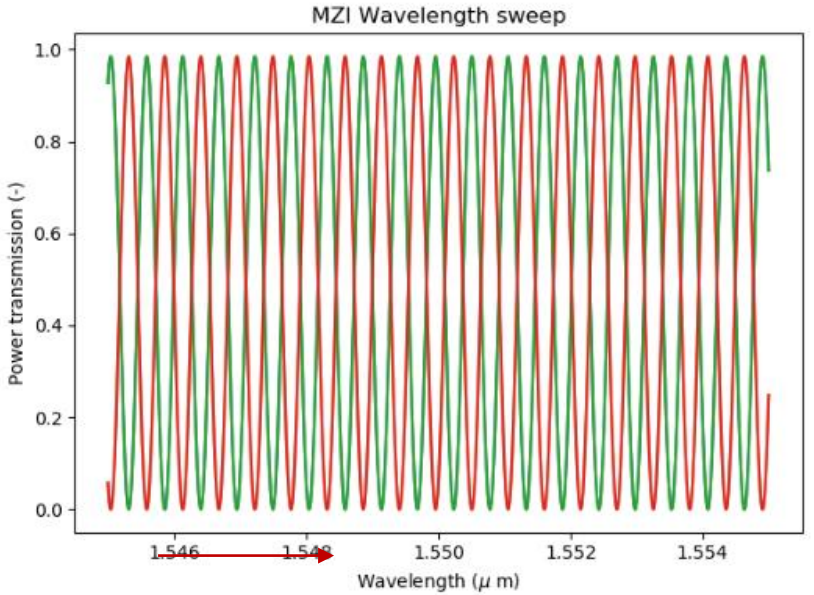
Specification-driven placement and routing



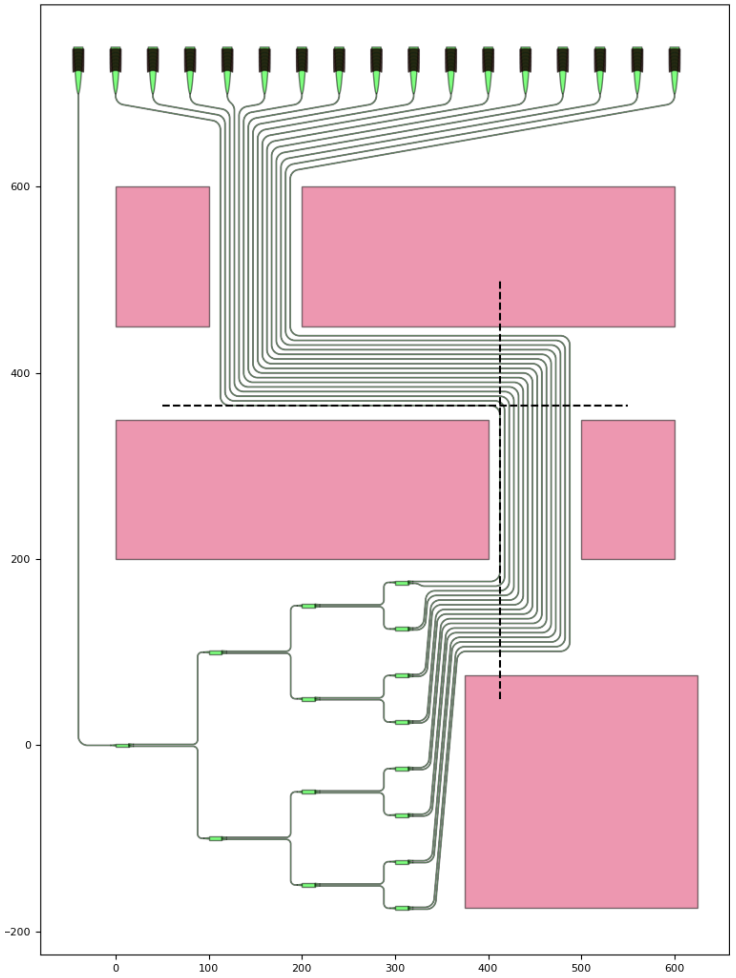

```python
import numpy as np
from pylab import plt

mzi_cm = mzi.CircuitModel()

wavelengths = np.linspace(1.545, 1.555, 1001)
S = mzi_cm.get_smatrix(wavelengths=wavelengths)

plt.plot(wavelengths, np.abs(S['out1', 'in1'])**2, label='MZI straight')
plt.plot(wavelengths, np.abs(S['out2', 'in1'])**2, label='MZI cross')
plt.title("MZI Wavelength sweep")
plt.xlabel("Wavelength ($\mu$ m)")
plt.ylabel("Power transmission (-)")
plt.show()
```

Model automatically built from subcomponents

Circuit model linked to specification and layout

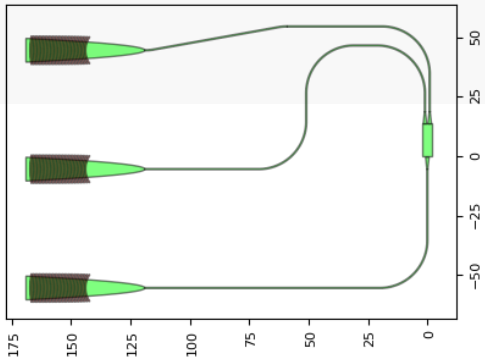


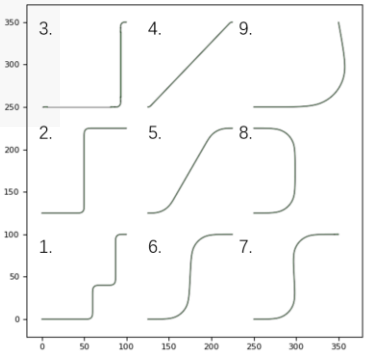
Routing



Individual routing

```
i3.ConnectBend("gr_out_{}:out".format(cnt), intermediate_port, bend_radius=self.bend_radius),
i3.ConnectManhattan(
    intermediate_port.flip_copy(), "DUT:out{}".format(n_gratings - cnt),
    bend_radius=self.bend_radius,
    min_straight=0.0, start_straight=0.0,
)
```



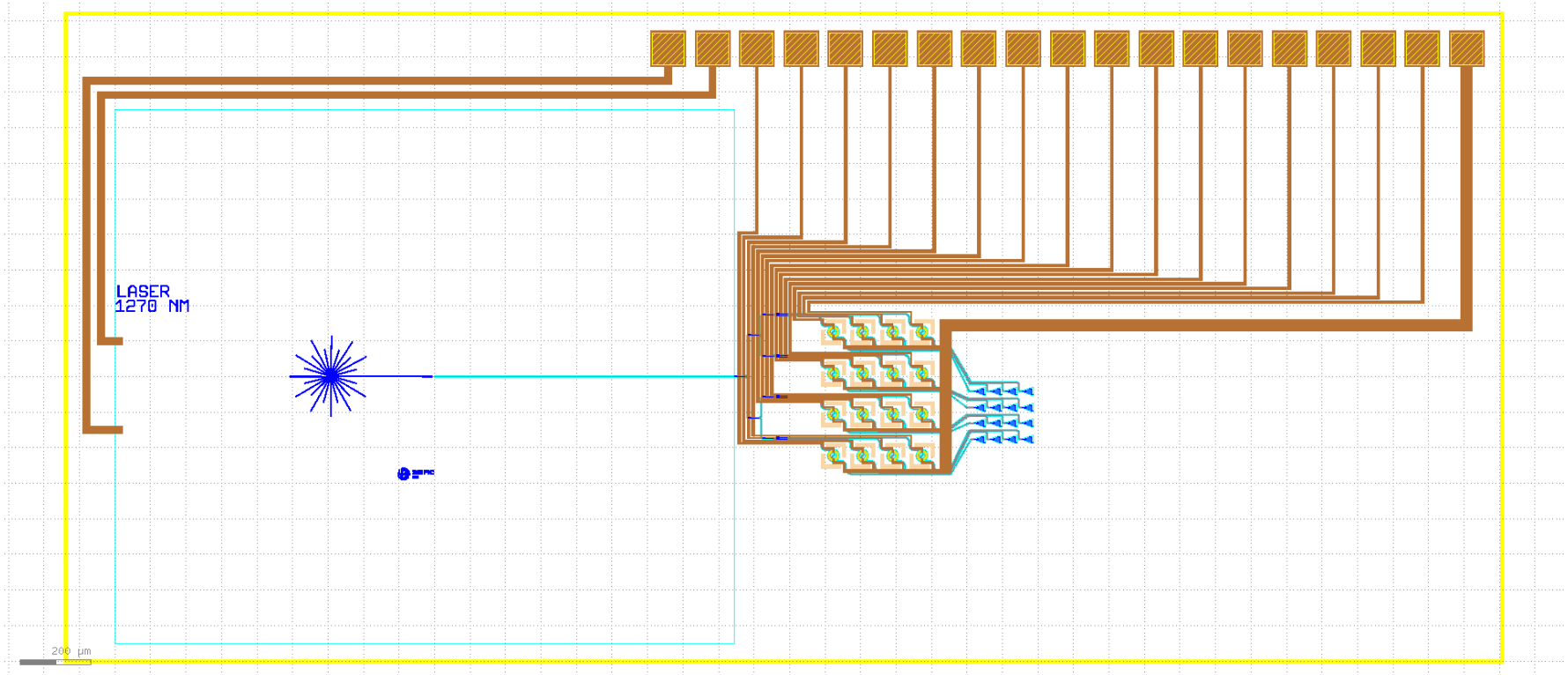


Bundle routing

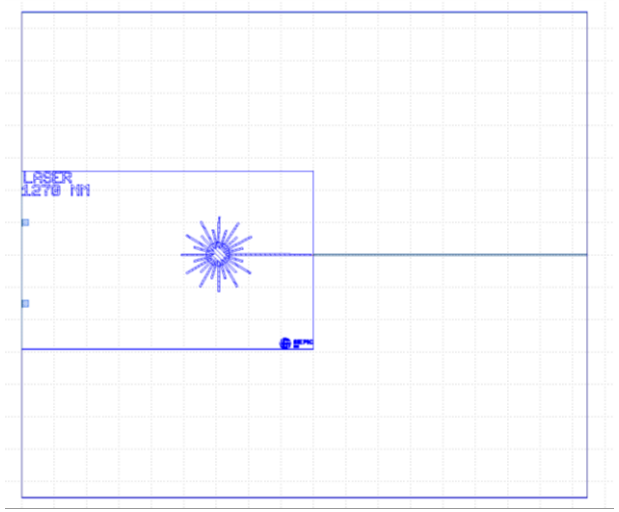
```
bundle = [
    i3.ConnectManhattanBundle(
        connections=[
            ('tree:out{}'.format(16 - n), 'gr_out_{}:out'.format(n)) for n in range(16)
        ],
        start_fanout=i3.SBendFanout(max_sbend_angle=85., reference='tree:out16'),
        end_fanout=i3.SBendFanout(max_sbend_angle=80, end_position=(150, 600)),
        pitch=pitch,
        bend_radius=bend_radius,
        control_points=[i3.V(412.5), i3.H(365)],
        control_point_reference='tree:out16',
    )
]
```

Hierarchical, parametric design

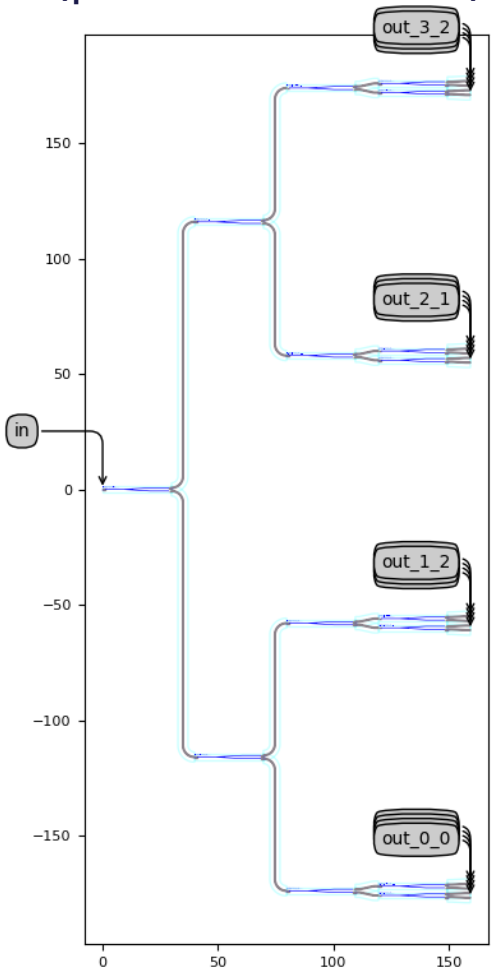
Example: Optical Phased Array with integrated laser (SiEPIC Shuksan PDK)



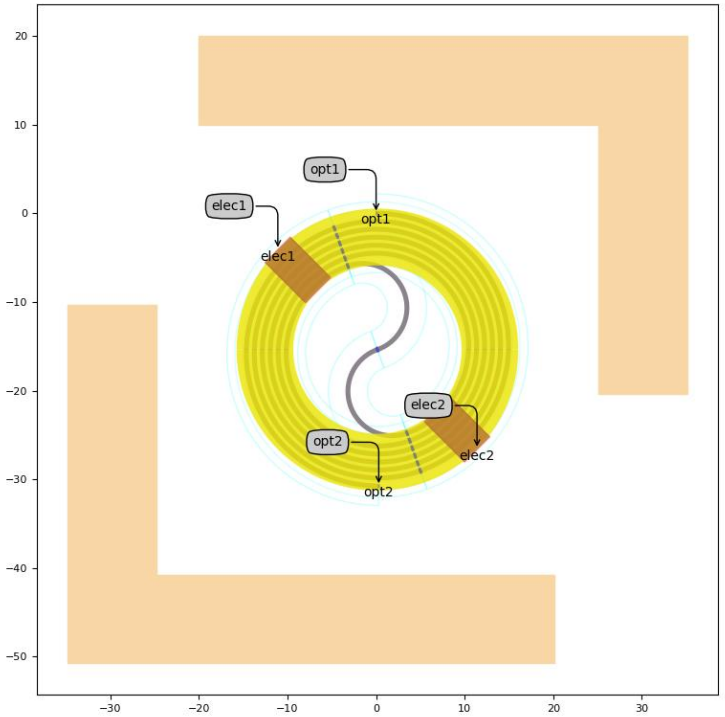
Laser (PDK)



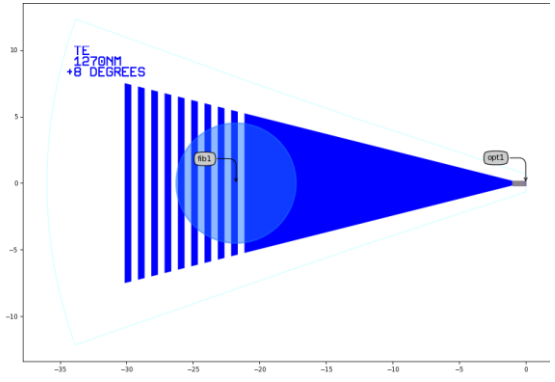
Splitter tree (parametric circuit)



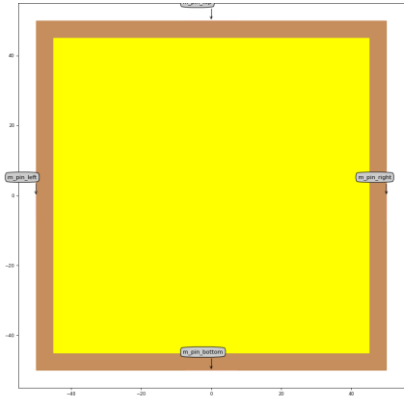
Spiral waveguide with heater (parametric circuit)



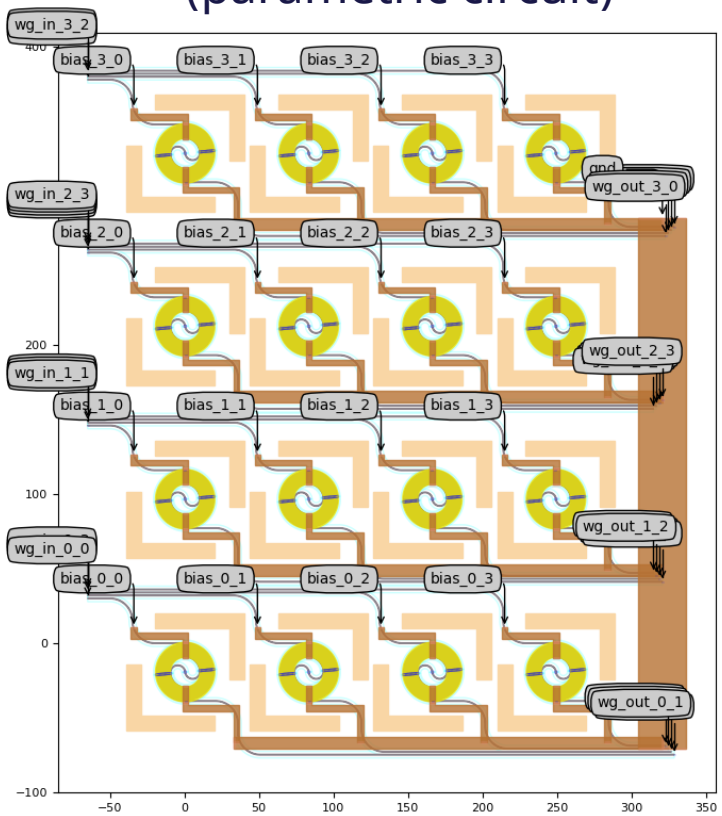
Grating coupler (PDK)



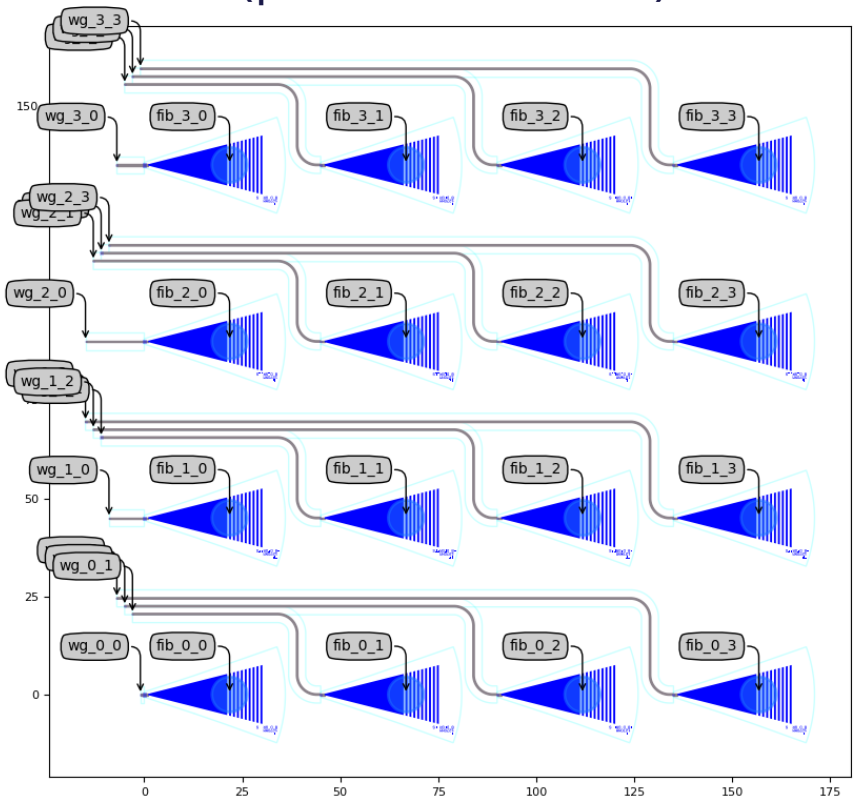
Bondpad (PDK)

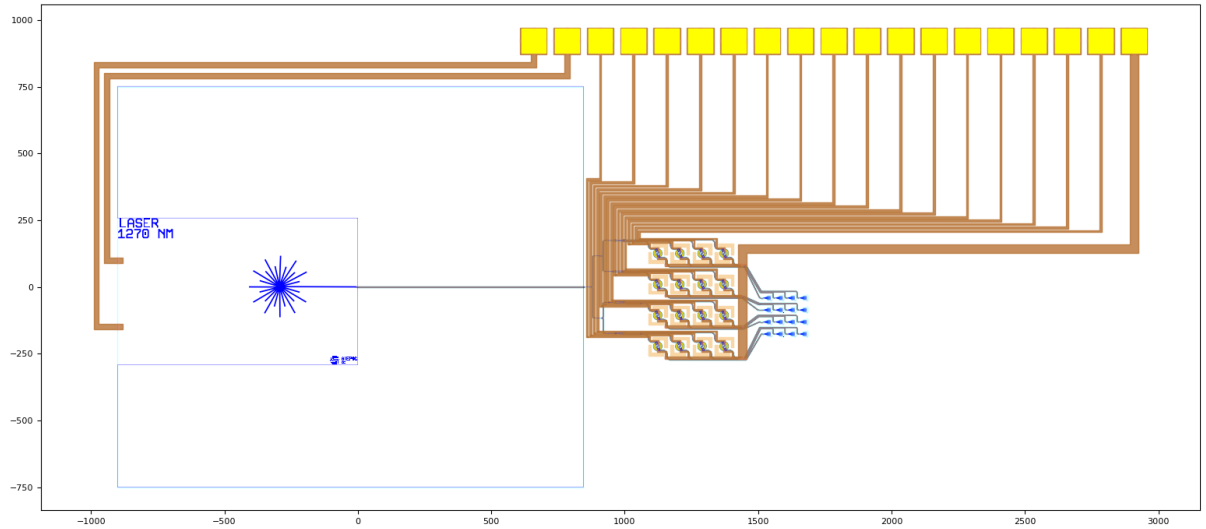


Matrix of phase shifters (parametric circuit)



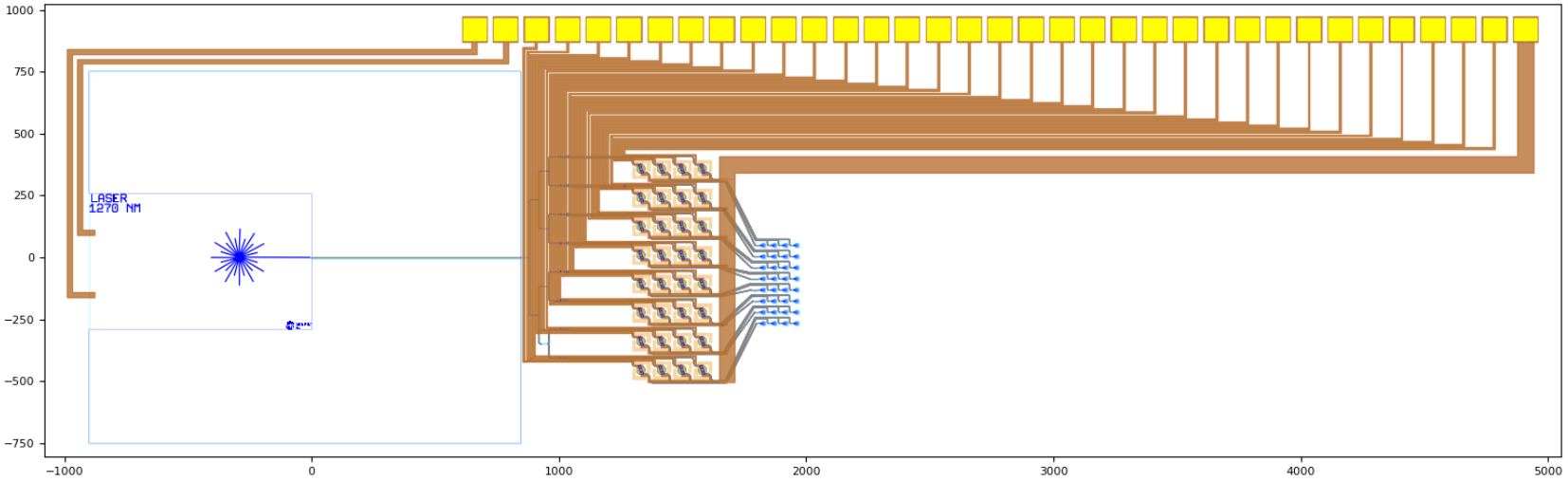
Matrix of grating couplers (parametric circuit)





Fully hierarchical – parametric !

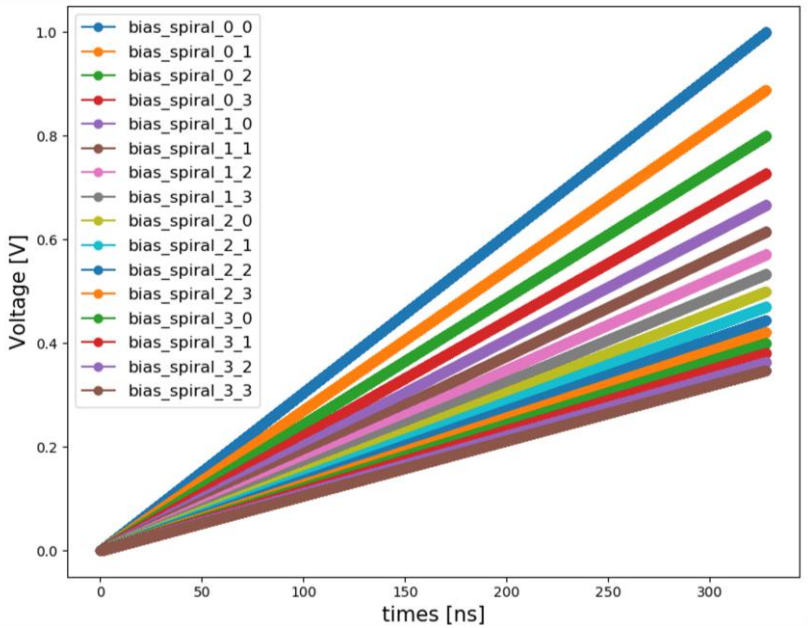
4x4 matrix



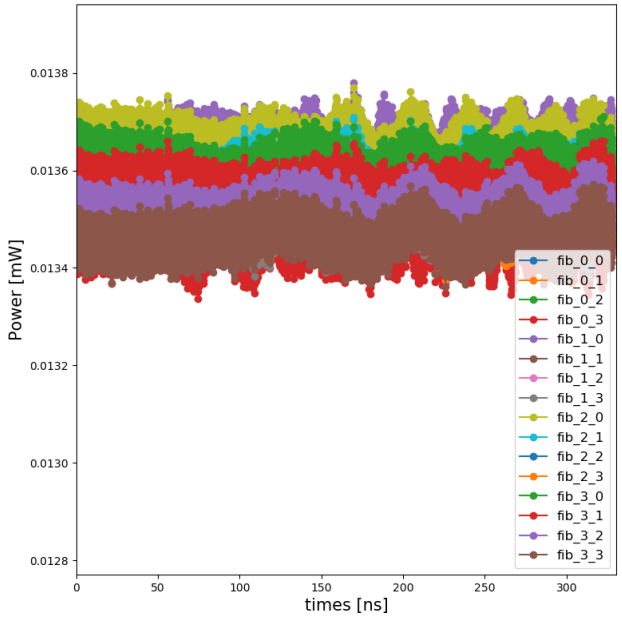
8x4 matrix

Circuit model generated automatically!

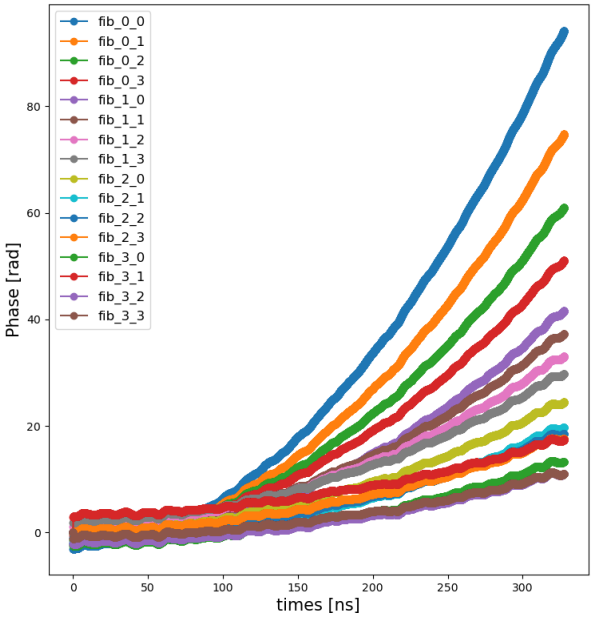
Time domain simulation



Bias voltage as function of time for each spiral



Optical power levels of emitted beams function of time

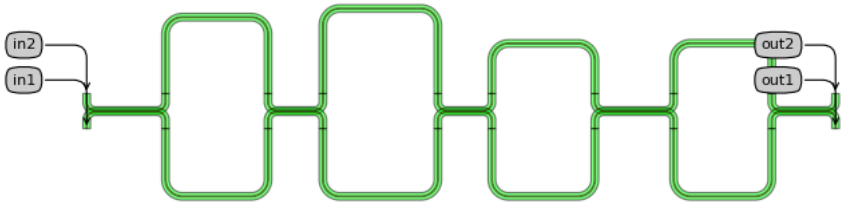


Phase of emitted beams Function of time

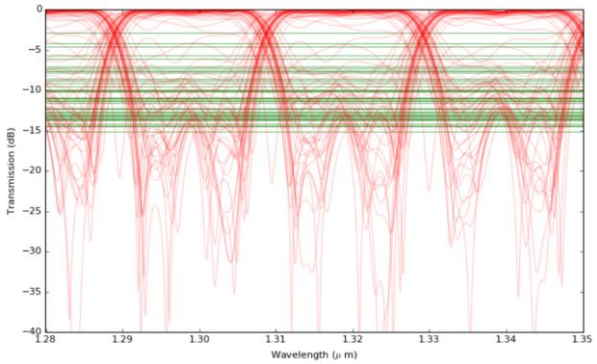
Where are we going?

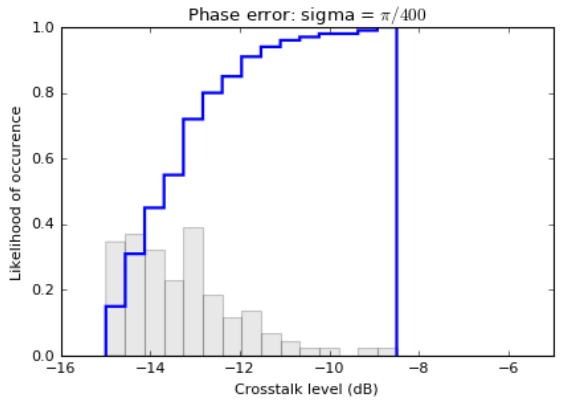
- Even more layout automation
- Simulation & modeling
- The chip within the process window: variability analysis
- Off the chip: assembly design kits

Variability and yield analysis example

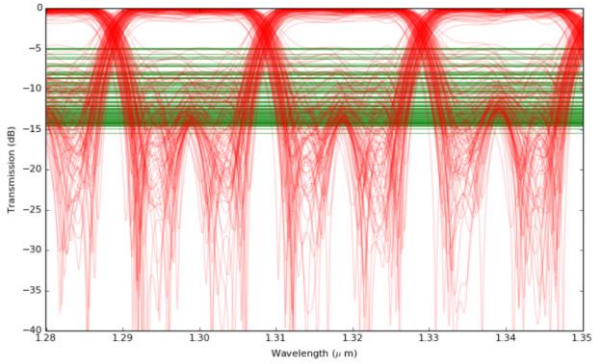


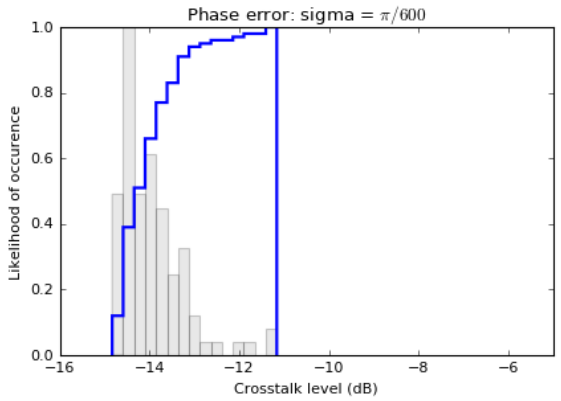
Bad phase control





Good phase control

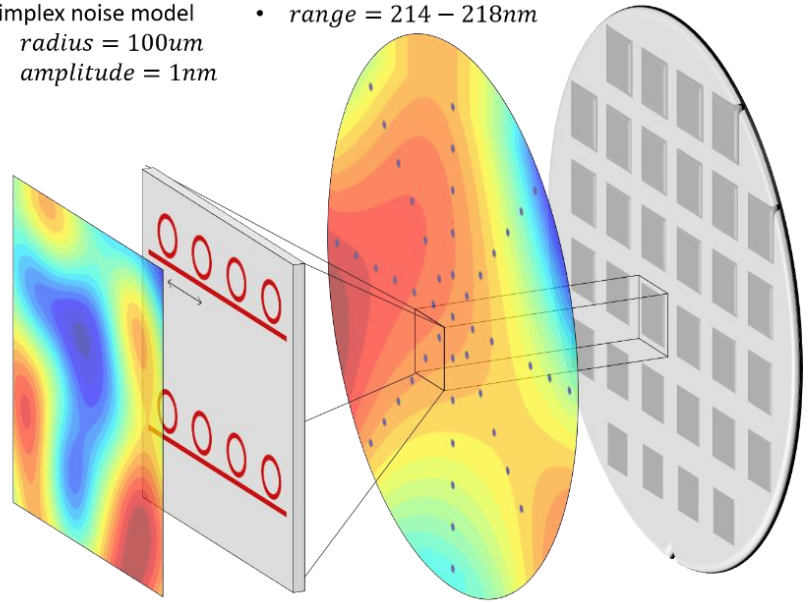




Cascaded MZI (lattice filter)

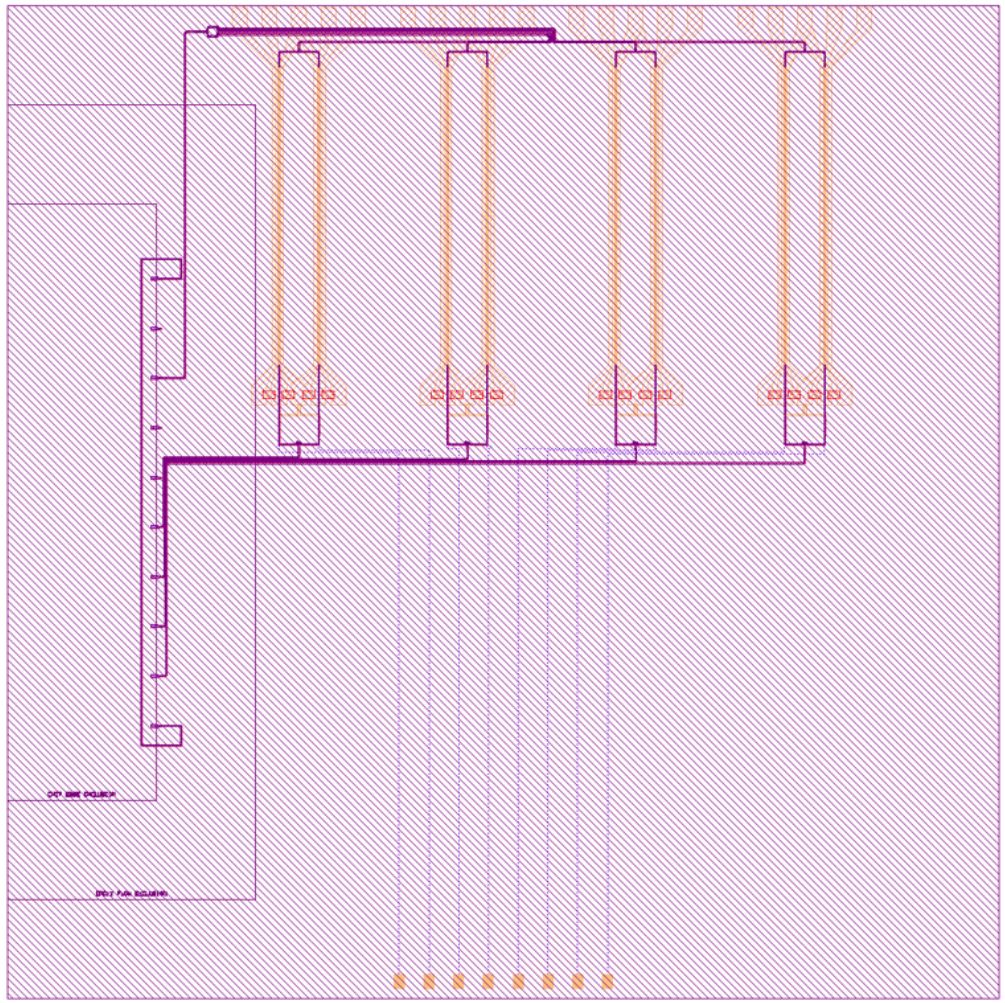
linewidth map
Simplex noise model
- $radius = 100um$
- $amplitude = 1nm$

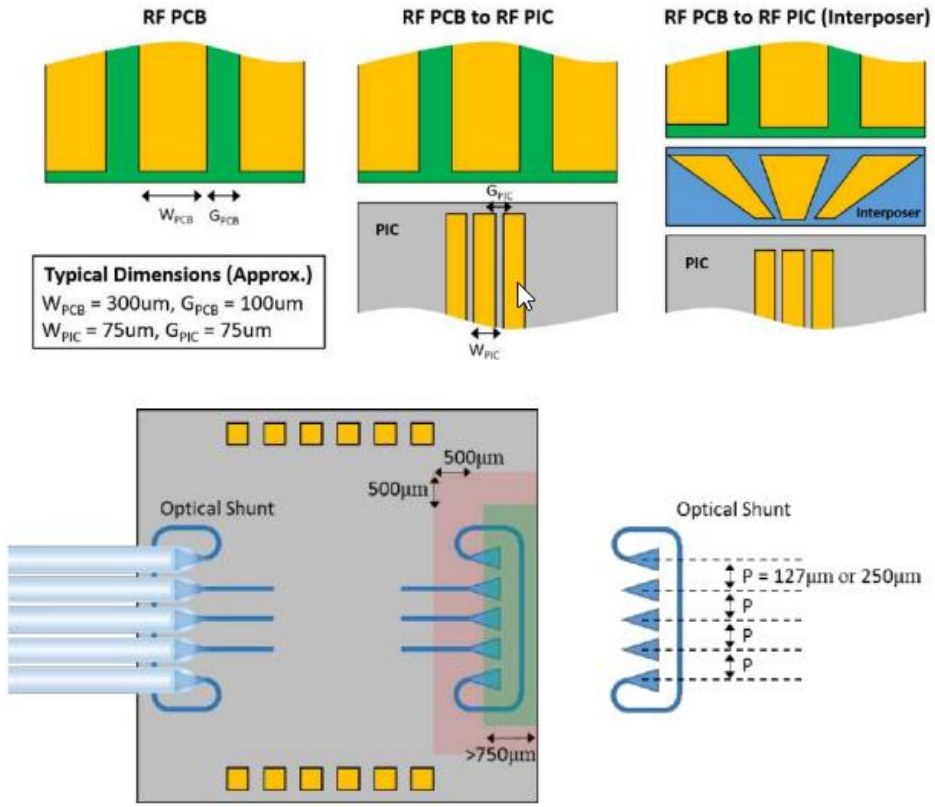
Thickness map (measured)
- $range = 214 - 218nm$



Take control of your photonics design flow 19

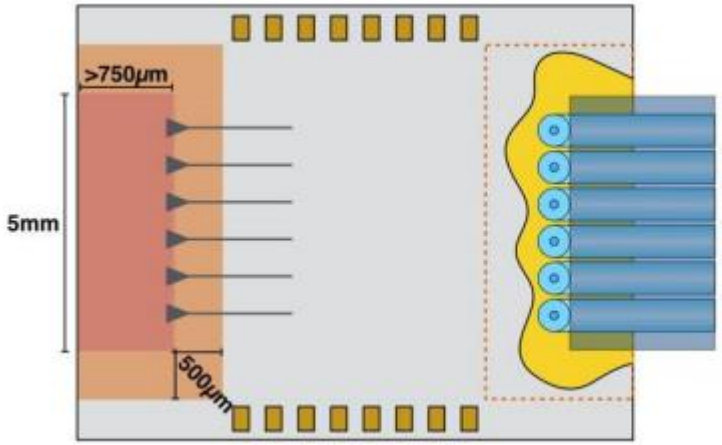
Example : Packaged 4-Lane Modulator

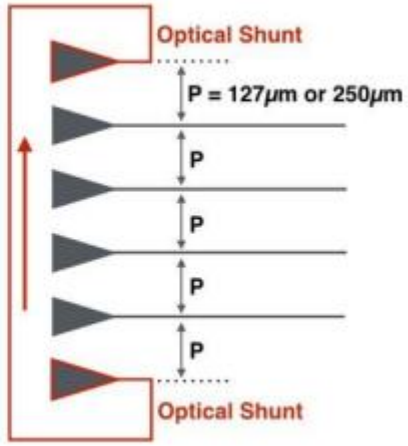




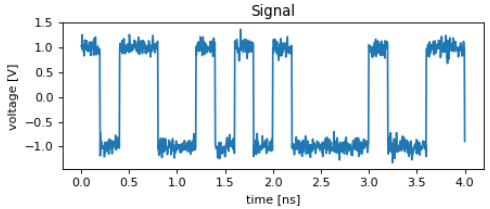
Respect the packing foundry rules

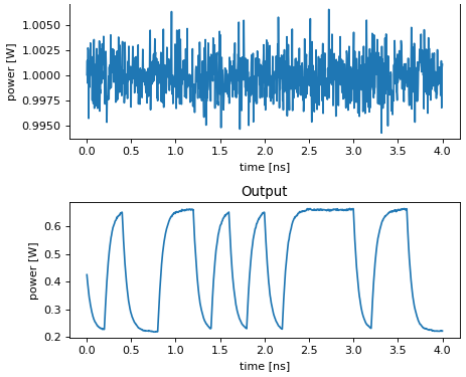
```
from si_fab import all as pdk
from tyndall_packaging.all import UniformPackagingArray

packaging_arrray = UniformPackagingArray(
    n_o_user_west_gratings=0,
    n_o_user_east_gratings=1,
    n_o_north_bondpads=10,
    n_o_south_bondpads=8,
    bondpad=pdk.BondPad(metal1_size=(50, 70), metal2_size=(50, 70)),
    grating=pdk.FC_TE_1300(),
)
packaging_array_lv = packaging_arrray.Layout()
packaging_array_lv.write_gdsii("sifab_chip_package_oband.gds")
```

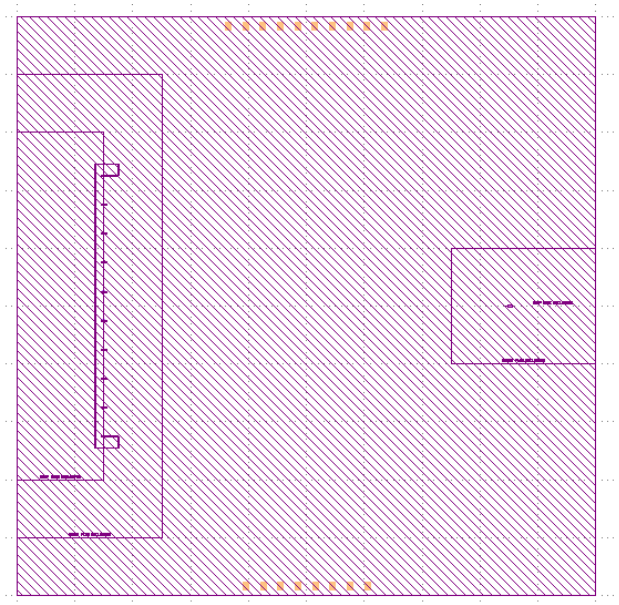




Simulate including interface models







Collaborate?

- Statistical models and PDK creation
- Circuit simulation for novel applications
- Heterogeneous integration, assembly and test

LUCEDA PHOTONICS