PHOTONICS RESEARCH GROUP

Introduction to Silicon Photonics Circuit Design

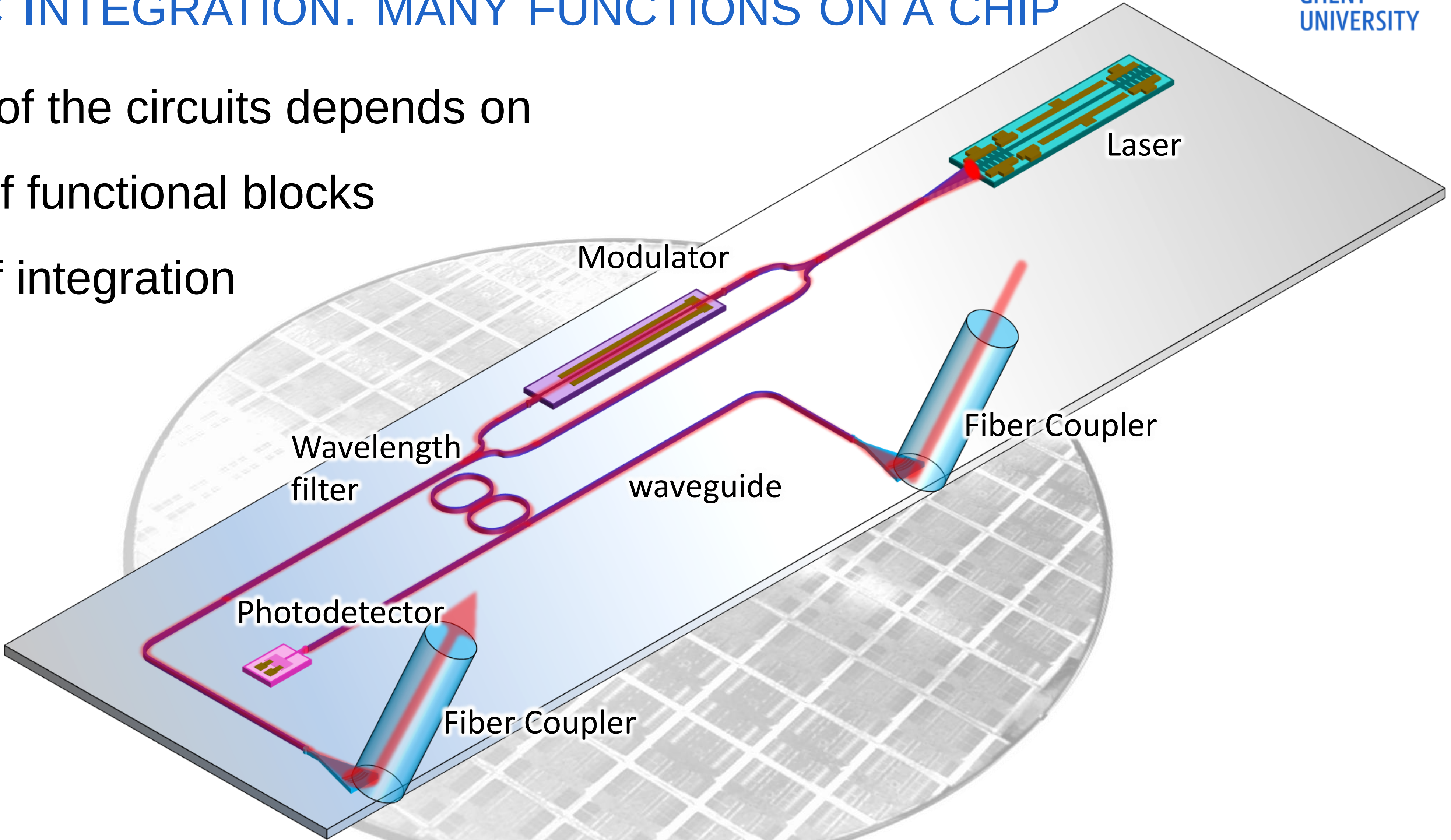
Wim Bogaerts

Short Course 454 - OFC 2021

PHOTONIC INTEGRATION: MANY FUNCTIONS ON A CHIP

Complexity of the circuits depends on

- number of functional blocks
- density of integration



Circuits connect elements together with waveguides

Manipulating light on Chips

Complexity

Overall Performance

Reliability

Ergonomy

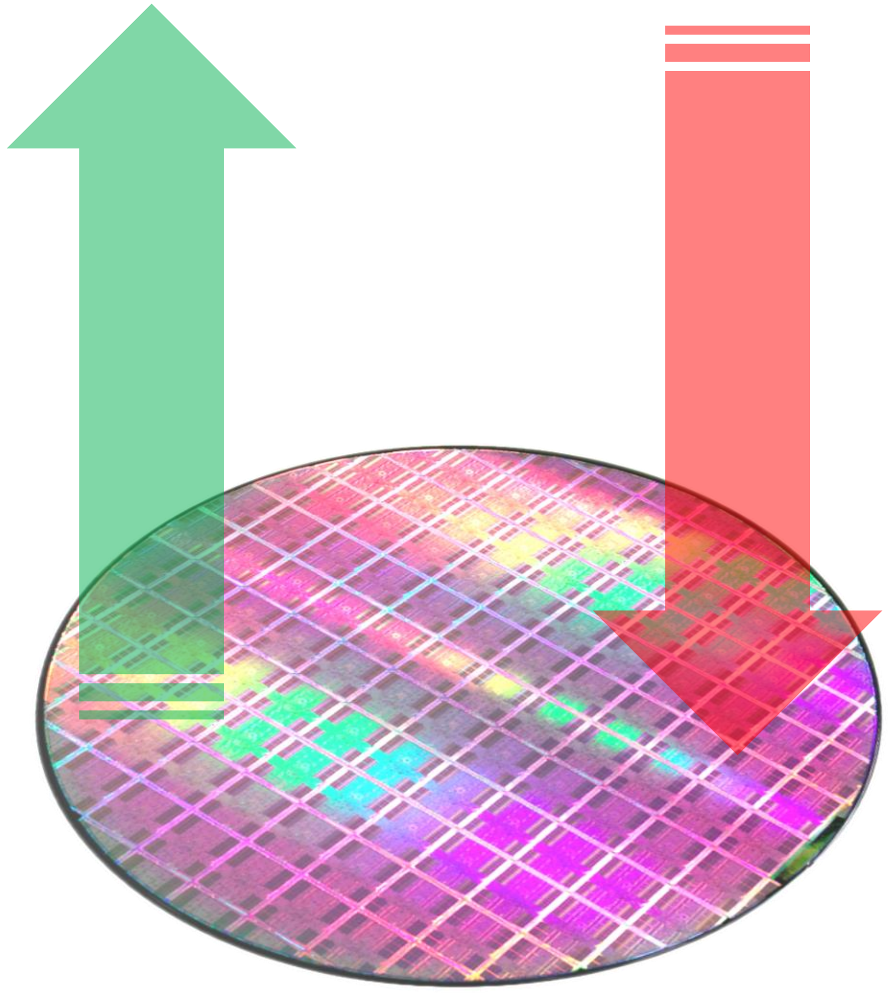
goes up

Power consumption

Ecological Footprint

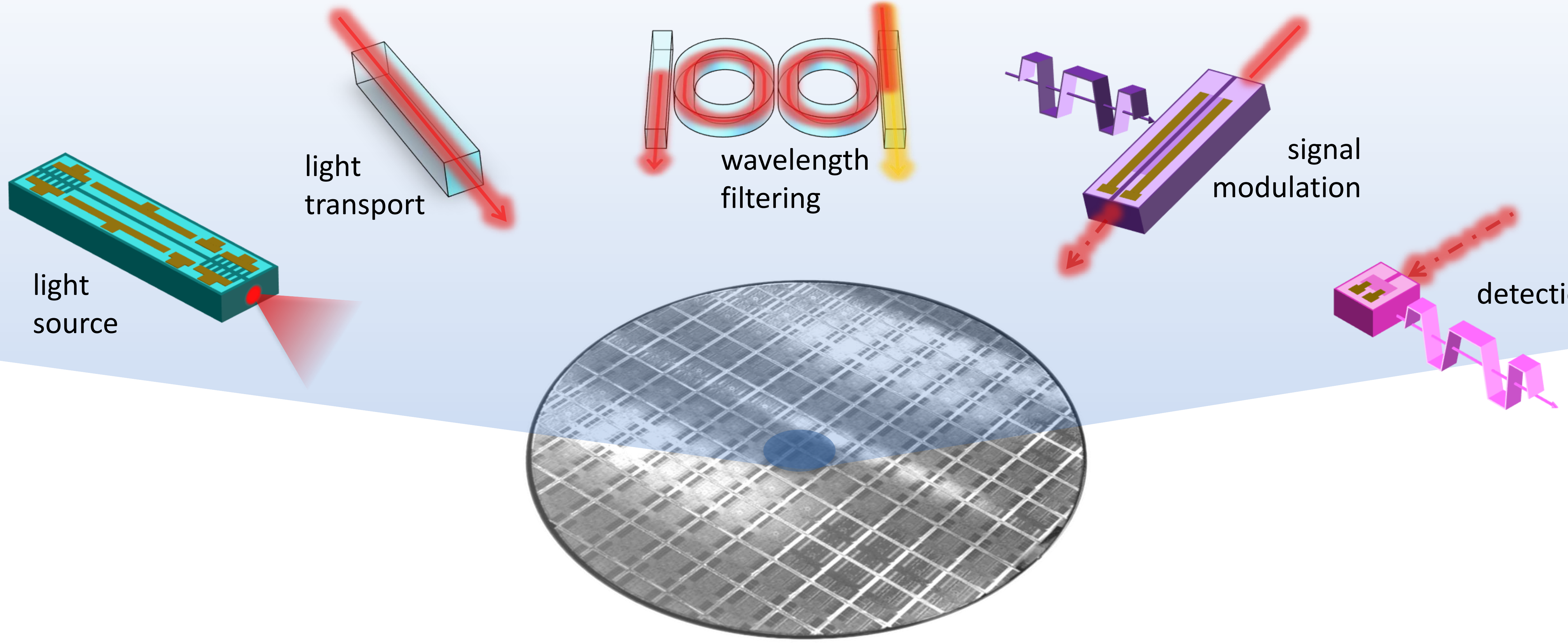
Cost

goes down



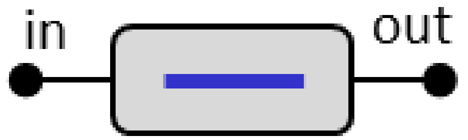
The benefits of scale

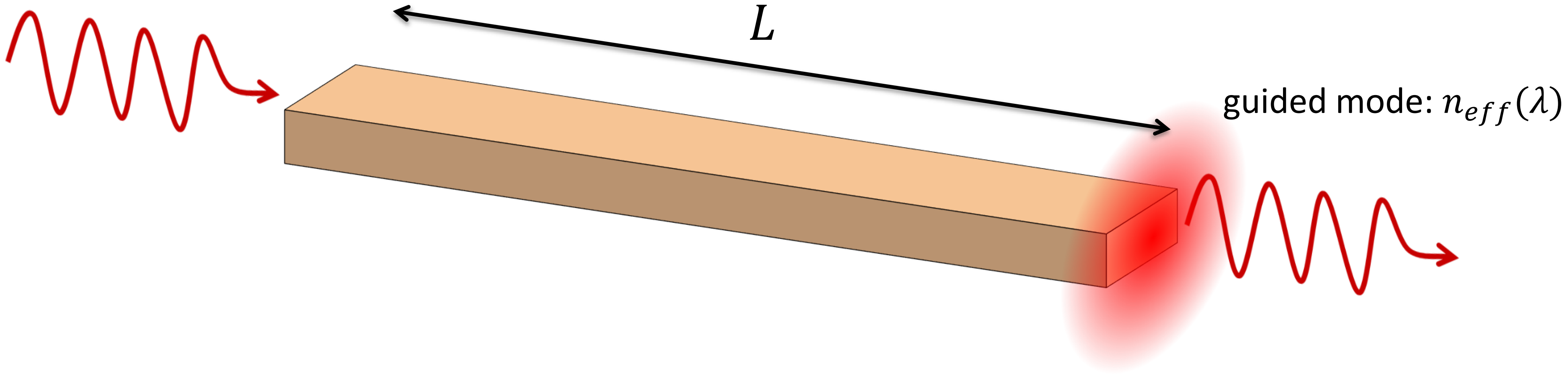
PHOTONIC INTEGRATION: MANY FUNCTIONS ON A CHIP



Circuits connect elements together with waveguides

WAVEGUIDES



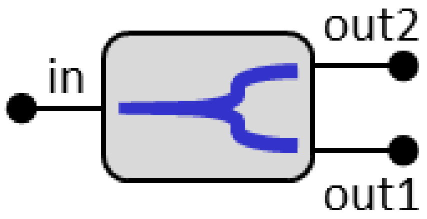


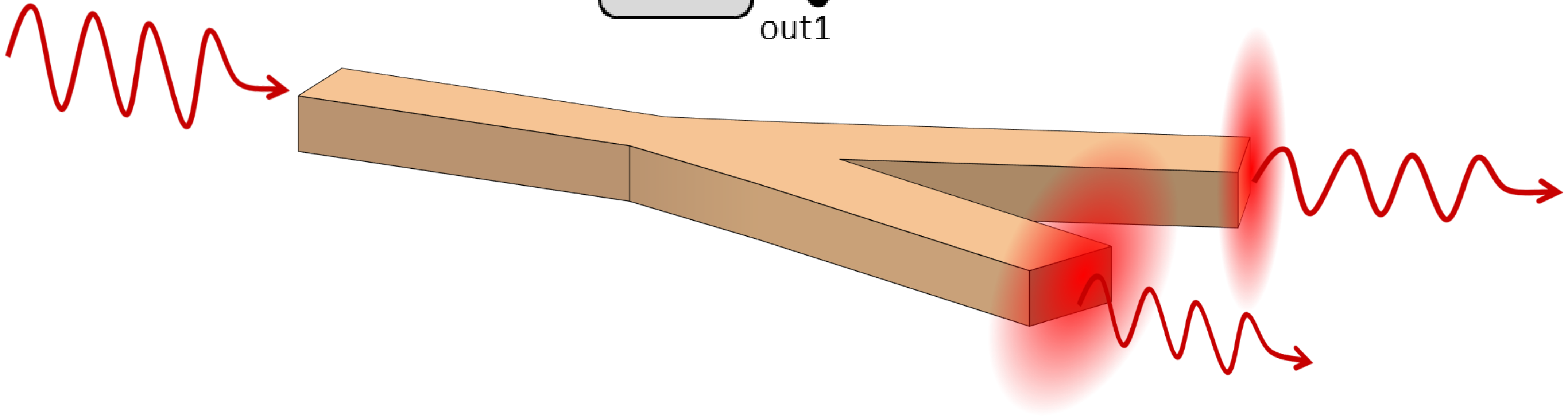
Propagate light from the input to the output

- wavefronts propagate with velocity $v_{ph}(\lambda) = \frac{c}{n_{eff}(\lambda)}$

  ($n_{eff}(\lambda)$= effective refractive index)
- Dispersion: $n_{eff}(\lambda)$ is wavelength dependent
- Group velocity: time delay of a wave packet: $v_g(\lambda) = \frac{c}{n_g(\lambda)}$

SPLITTERS



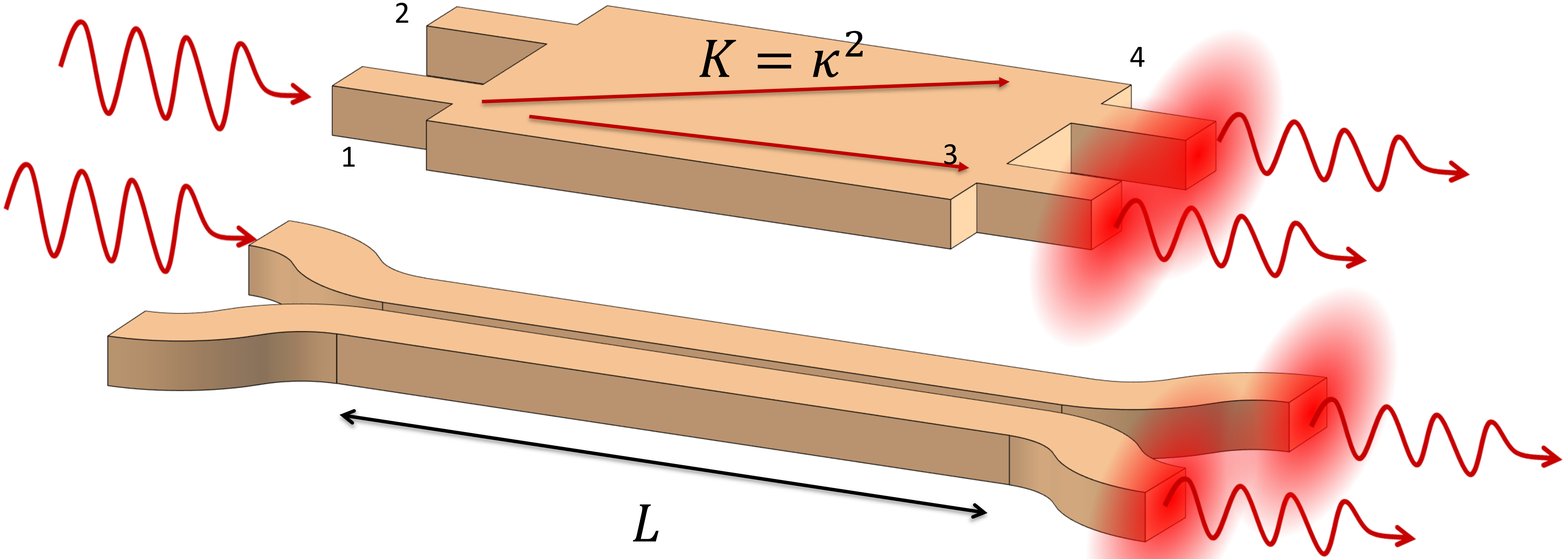


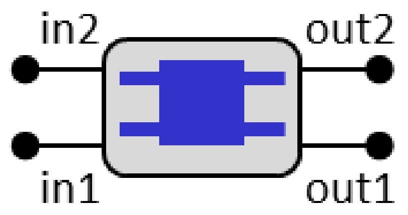
Splits light in two equal parts

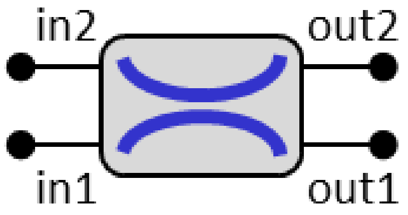
- one input
- two outputs
- symmetric

Reciprocal: Also has 3dB loss when used as a combiner.

2×2 COUPLERS



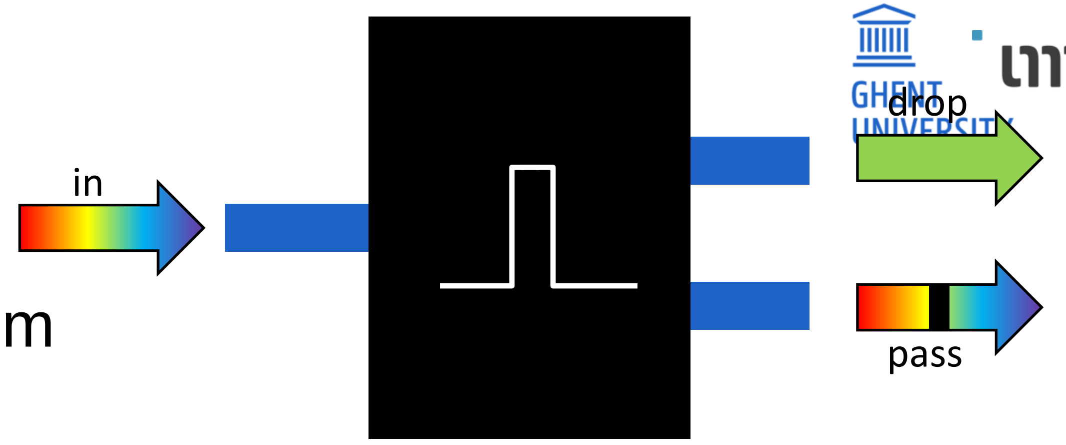




Wavelength Filtering

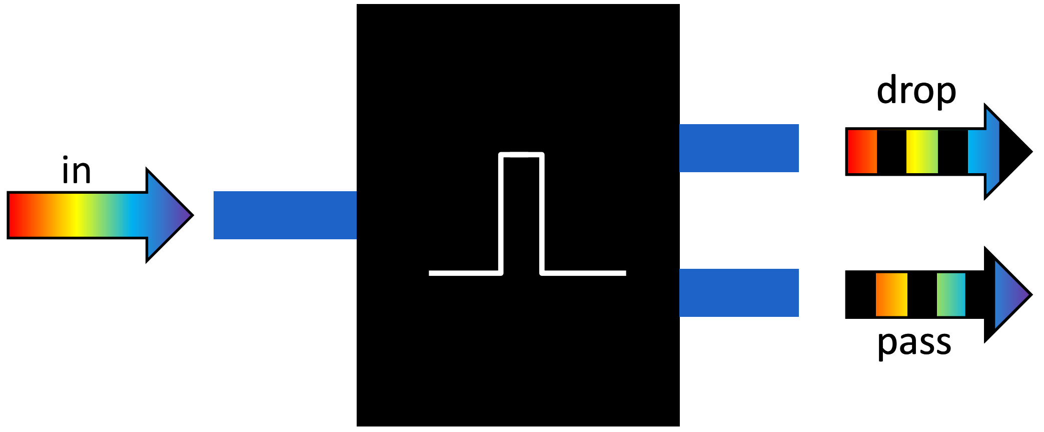
channel drop filter

- selects a passband from a wavelength range



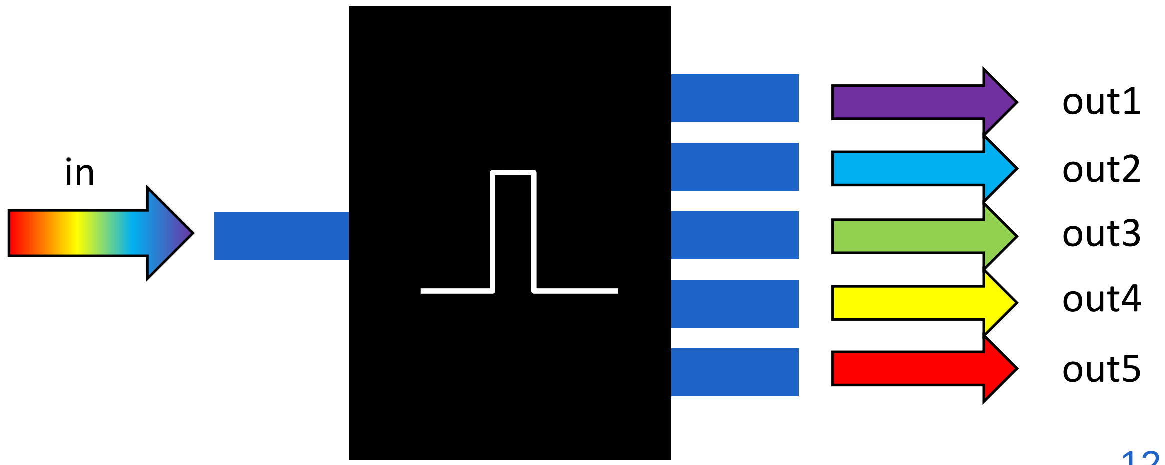
interleaver

- separates alternating wavelength bands

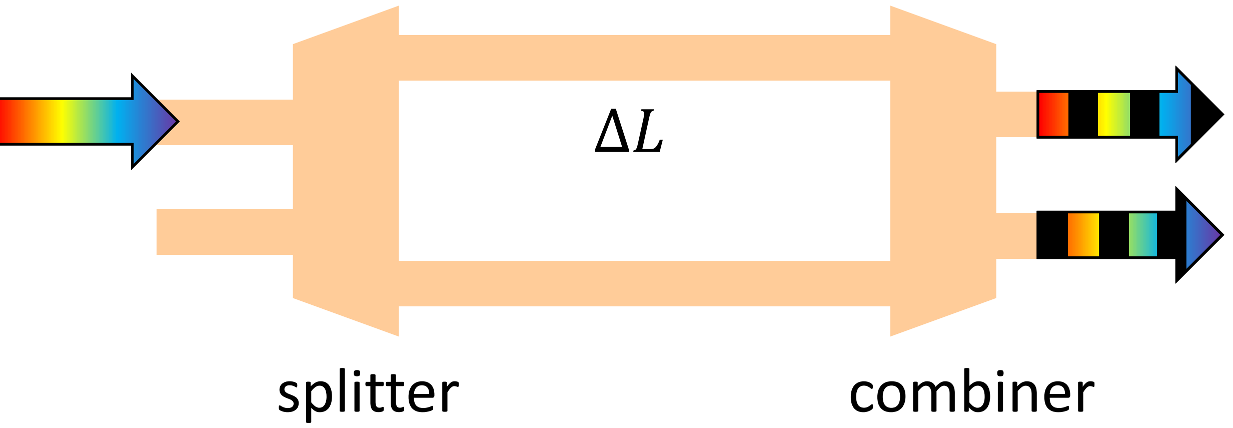


demultiplexer

- separates multiple wavelength channels



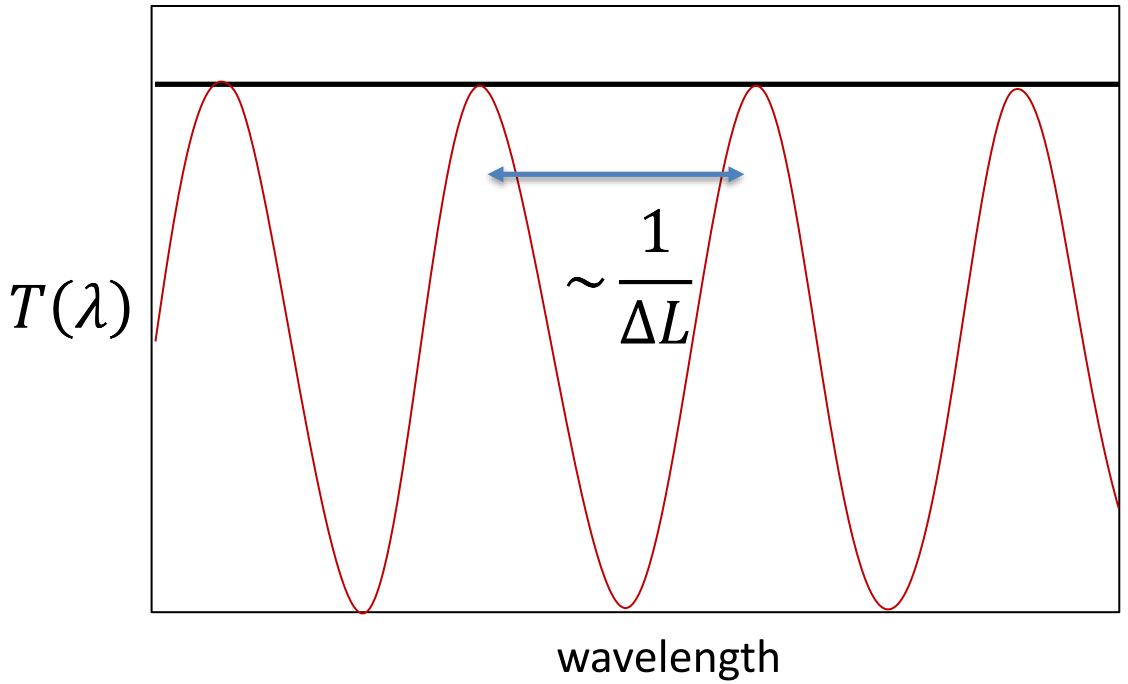
Wavelength Filtering



Mach-Zehnder filters

- two-arm interferometer
- fixed delay ΔL
- sinusoidal spectral response

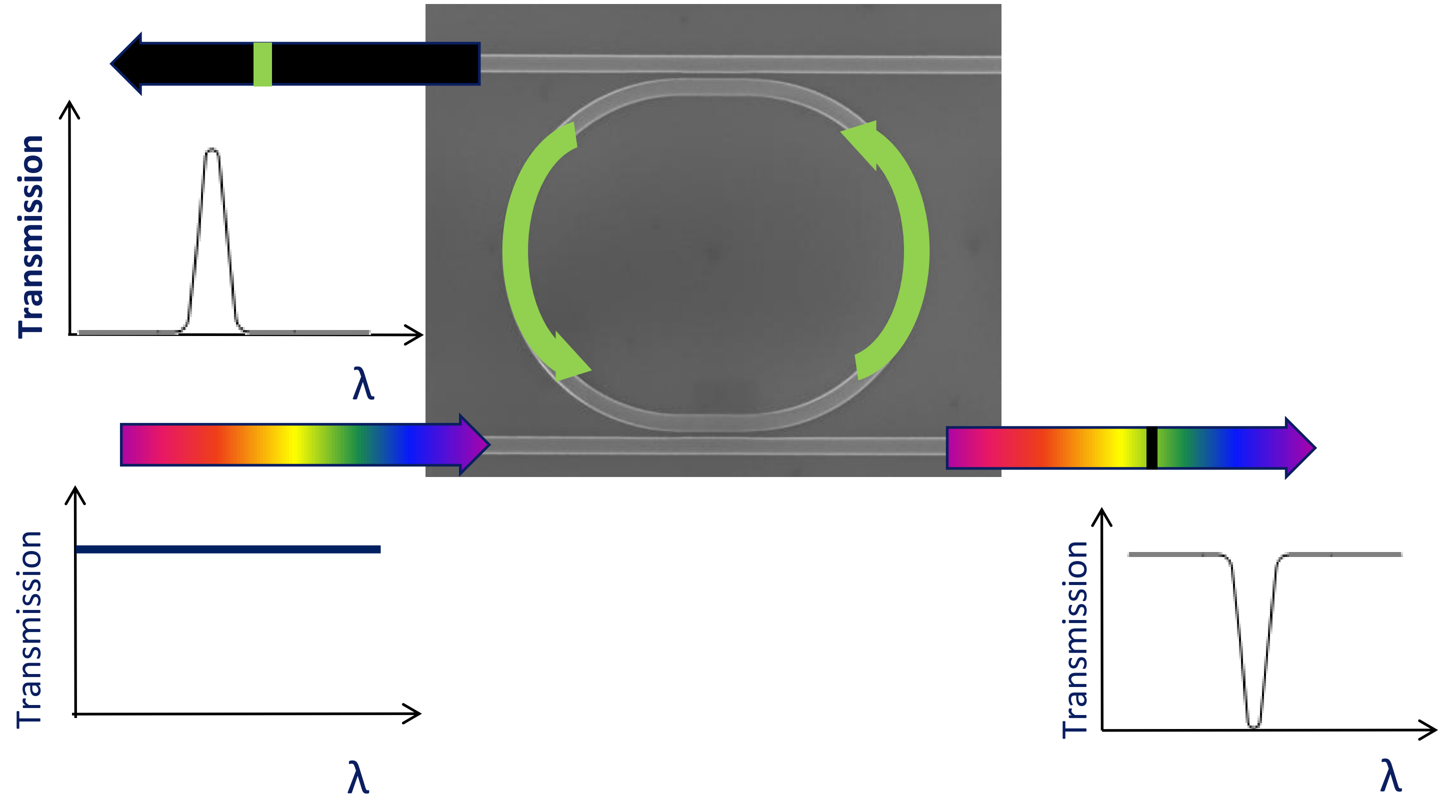
Can be cascaded for more complex filters



RING RESONATOR

Light resonates in ring cavity:

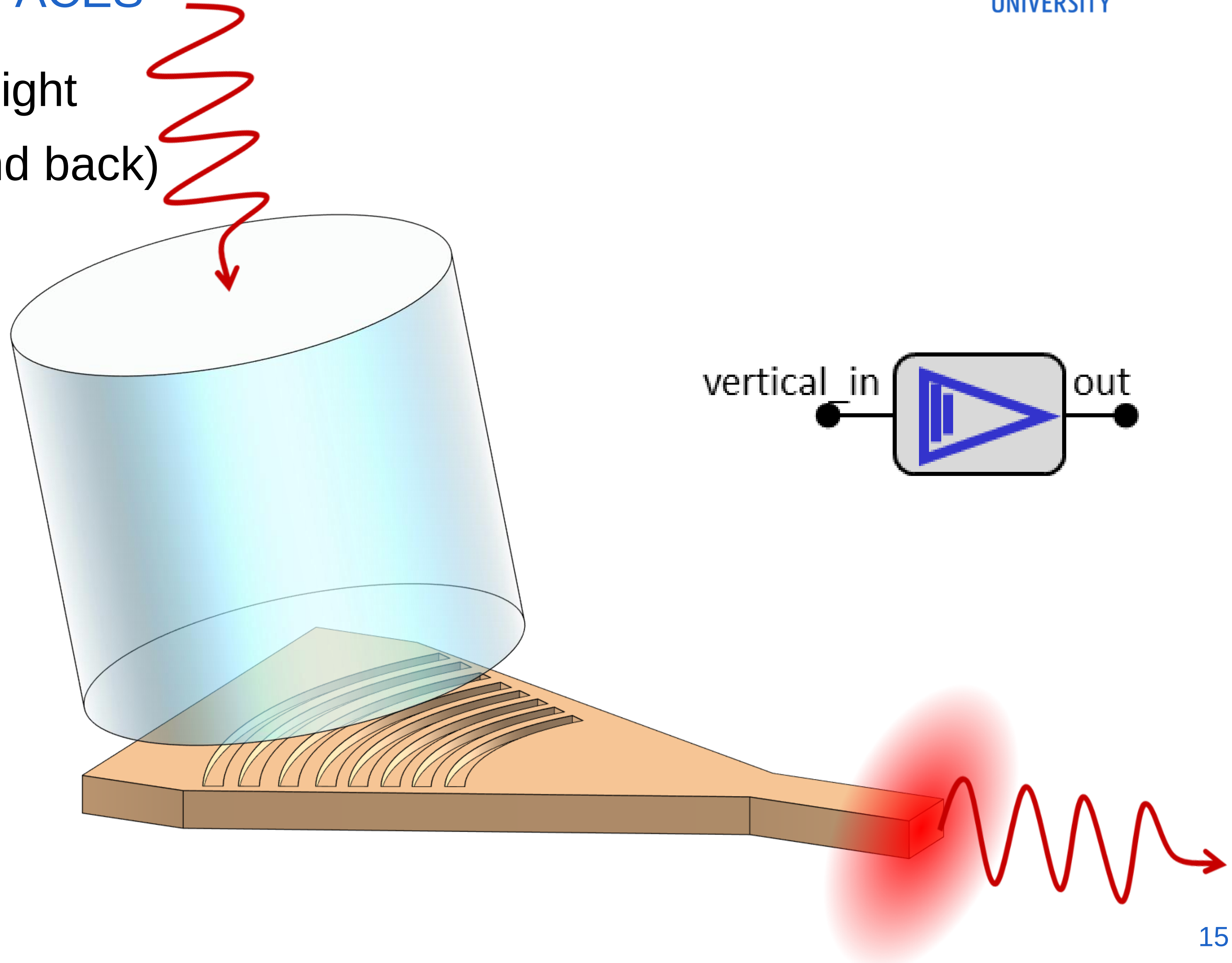
- $L_{optical} = L_{physical} . n_{eff} = m . \lambda$
- Quality factor Q ~ cavity losses (internal + coupling)

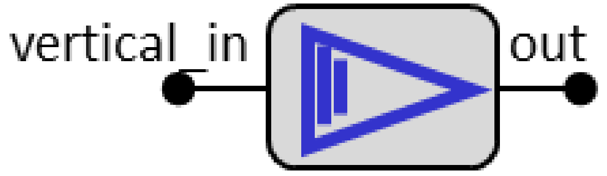


VERTICAL FIBER INTERFACES

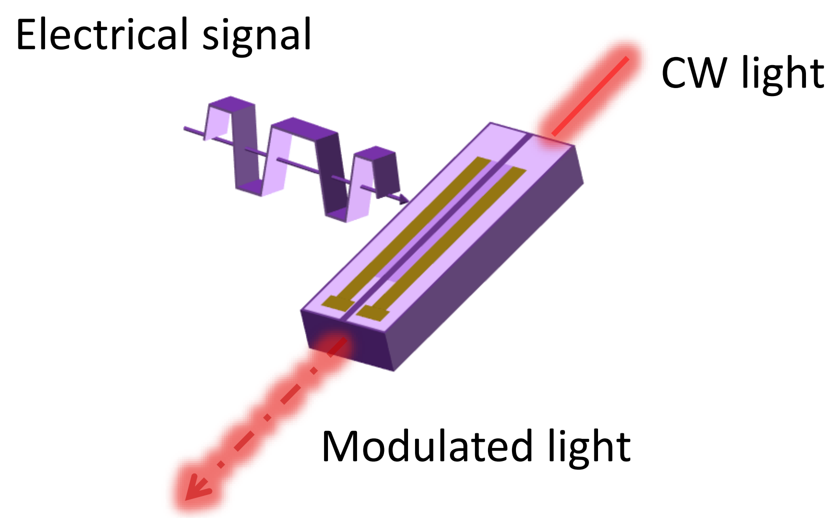
Diffraction grating couples light from fiber to waveguide (and back)

- wavelength dependent





ELECTRICAL MODULATION



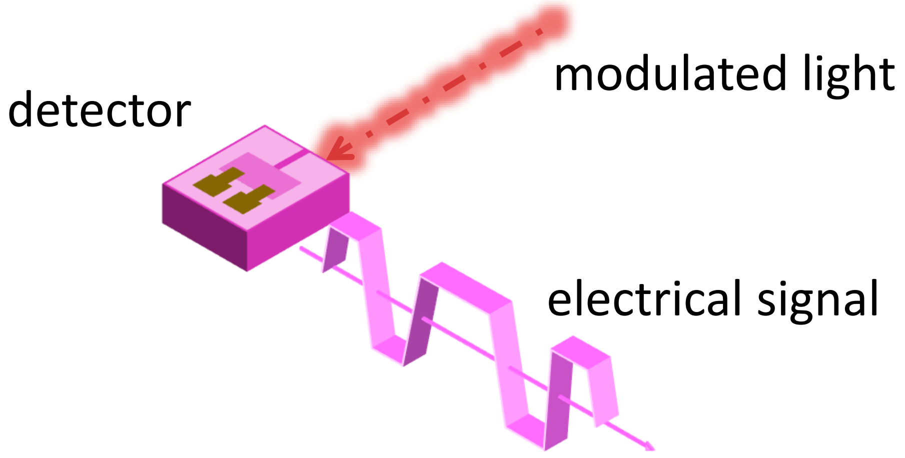
Electrical actuation: Switching and modulation

- Thermal
- Carrier injection/extraction
- Electro-optics

Different applications:

- Tuning: slow, analog
- Switching: slow, digital (<kHz), full amplitude
- Signal modulation: fast (GHz – 100GHz)
  - amplitude
  - phase

PHOTODETECTION



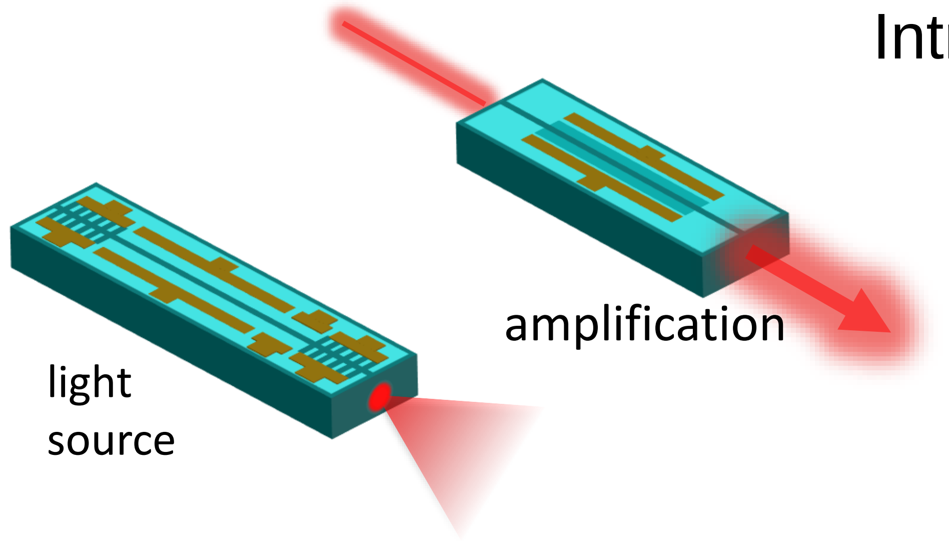
Mechanisms

- **photodiodes**: absorbed photon creates electron-hole pair.
  - p-i-n diode
  - metal-semiconductor-metal diode
- **photoconductors**: absorbed photon creates free carriers
- **photobolometers**: absorbed photon heats material, which then changes electrical resistivity

Examples

- III-V semiconductors (visible, telecom, MIR)
- Germanium (telecom)
- Silicon (visible, NIR)

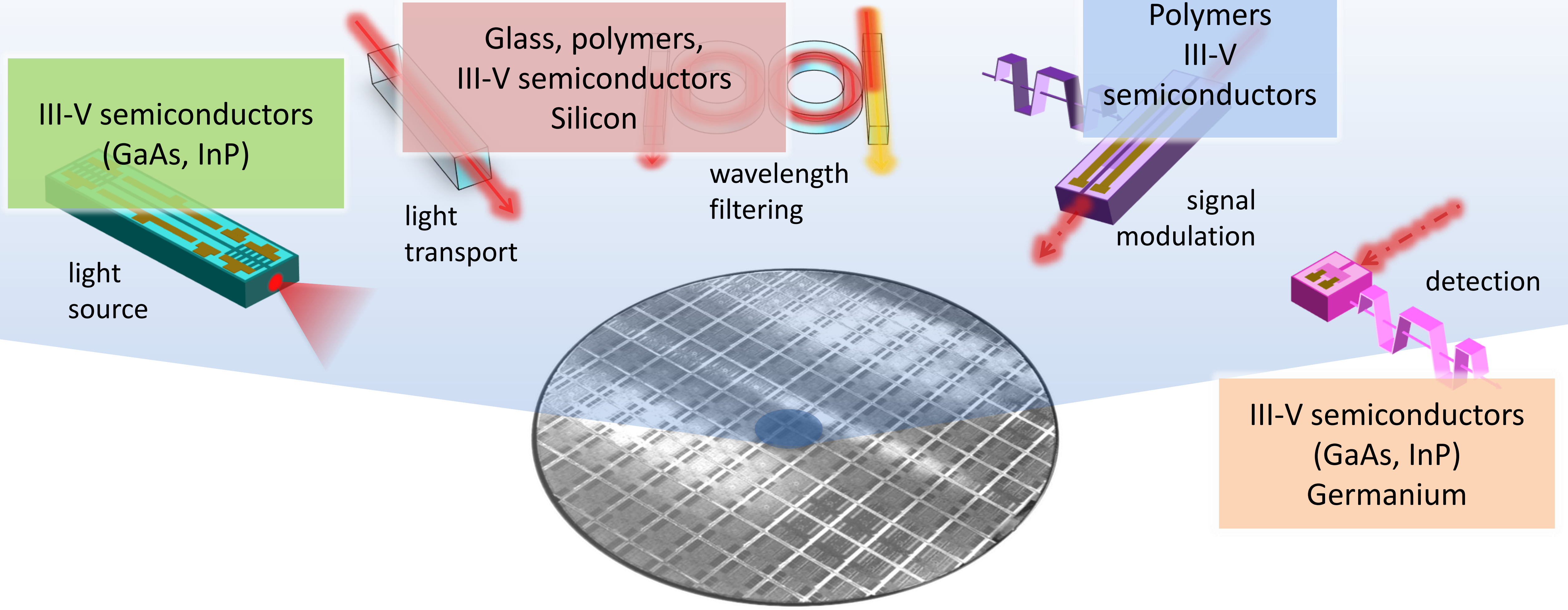
LASERS AND AMPLIFIERS



Introducing optical gain on a PIC

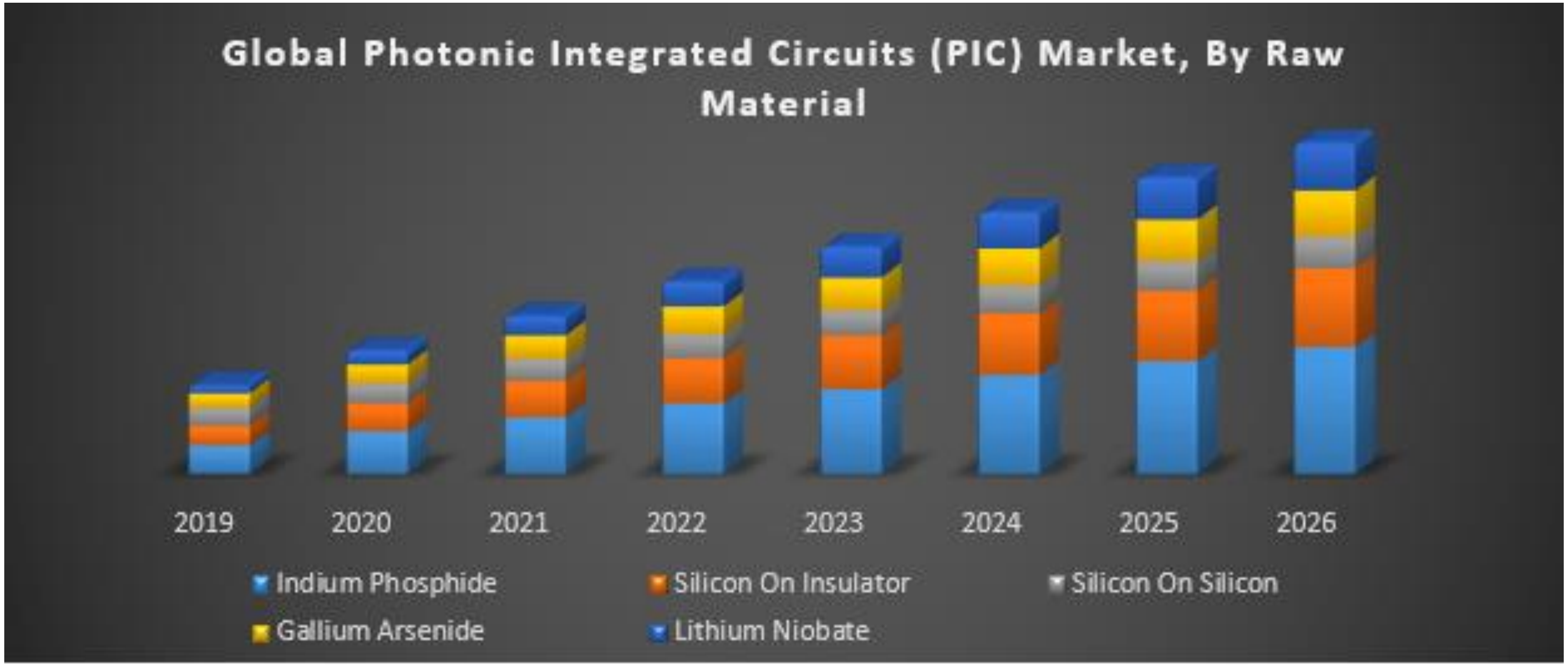
- semiconductors (III-V, Germanium) can be electrically pumped
- rare-earth (Erbium) can be incorporated in glass waveguides
- parametric gain (four wave mixing) requires nonlinear material

Photonic Integration: A mix of materials



GROWING PHOTONIC CHIP MARKET

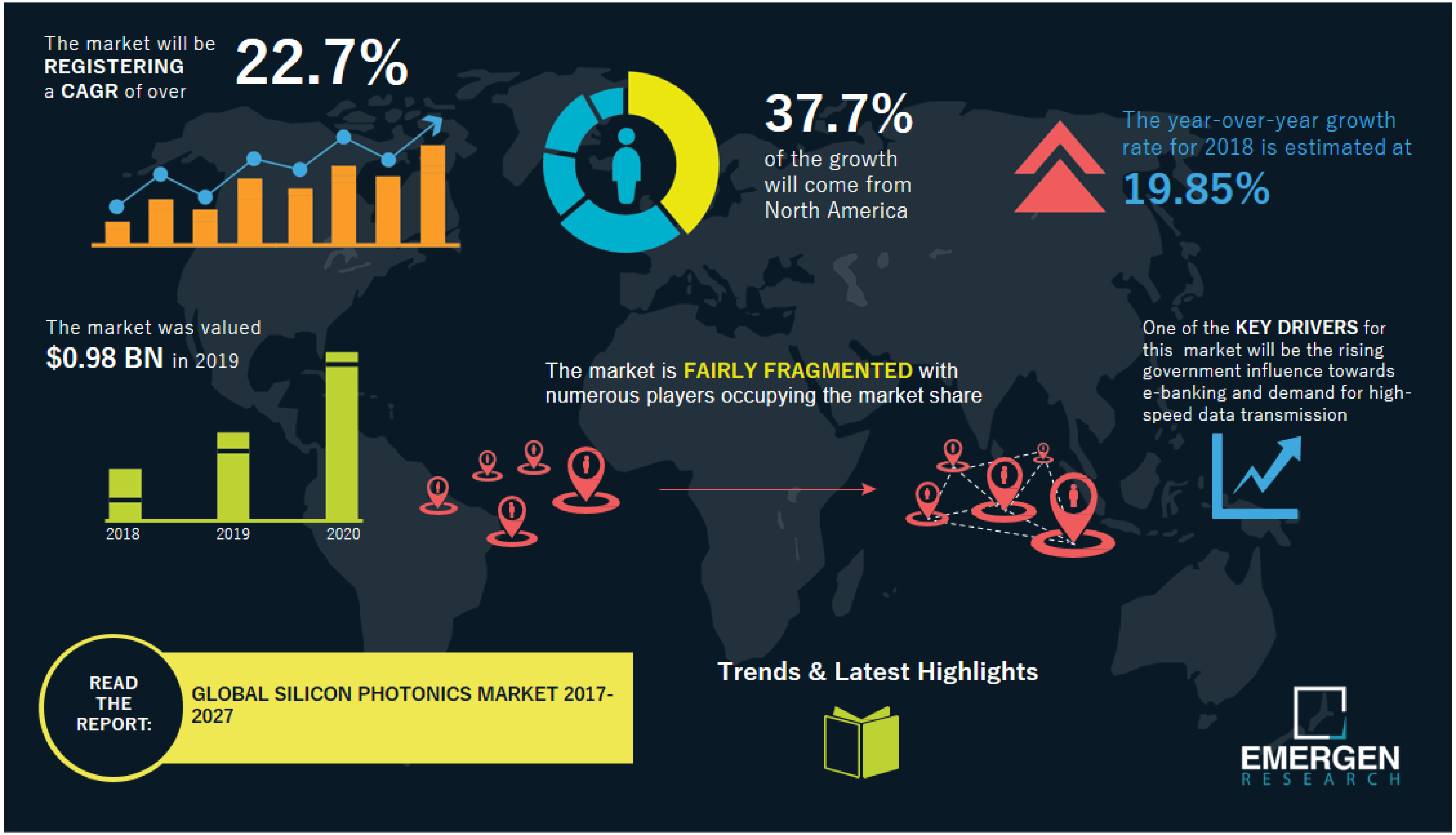
Different material systems



75% = semiconductor technology

source: maximizemarketresearch

WHAT IS SPECIAL ABOUT "SILICON PHOTONICS"?



The market will be **REGISTERING** a **CAGR** of over **22.7%**

37.7% of the growth will come from North America

The year-over-year growth rate for 2018 is estimated at **19.85%**

The market was valued **$0.98 BN** in 2019

The market is **FAIRLY FRAGMENTED** with numerous players occupying the market share

One of the **KEY DRIVERS** for this market will be the rising government influence towards e-banking and demand for high-speed data transmission

READ THE REPORT: **GLOBAL SILICON PHOTONICS MARKET 2017-2027**

Trends & Latest Highlights

source: emergen research

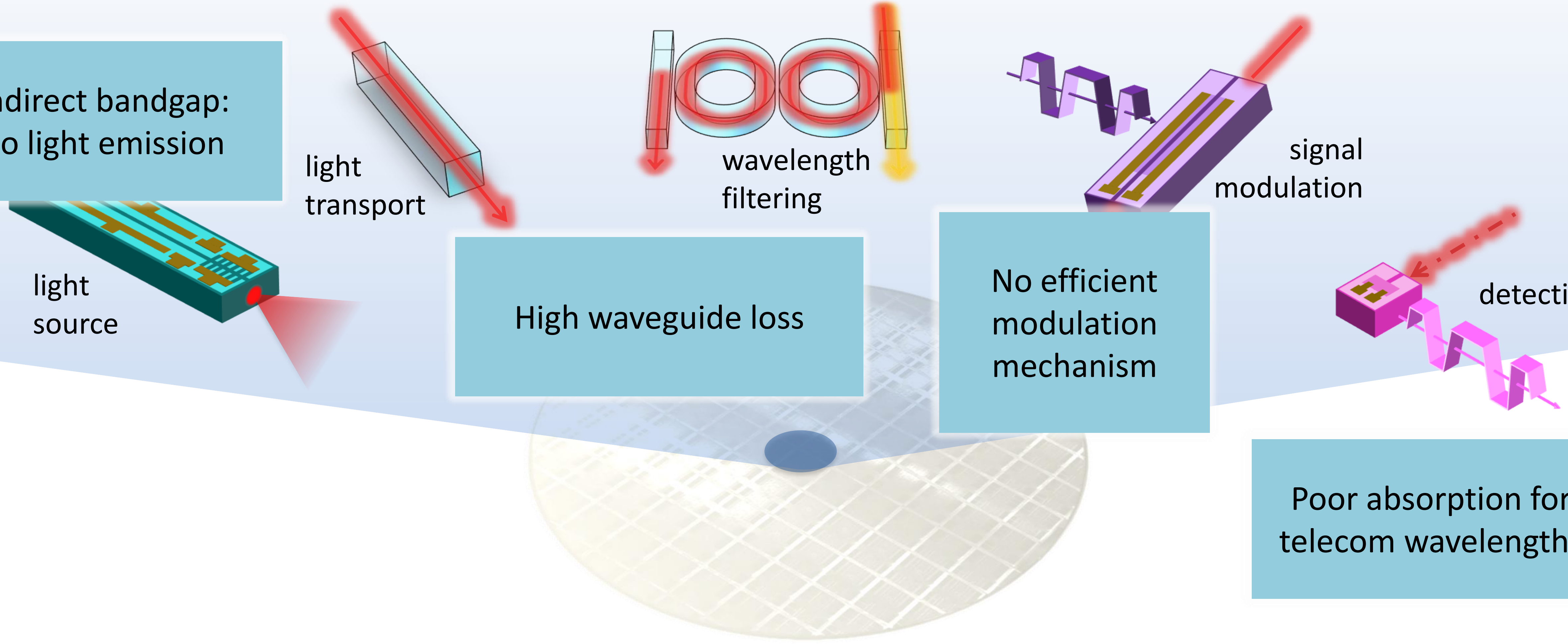
WHAT IS SILICON PHOTONICS?

GHENT UNIVERSITY imec

The implementation of high density photonic integrated circuits by means of CMOS process technology in a CMOS fab

Enabling complex optical functionality on a compact chip at low cost

SILICON IS NOT A GOOD PHOTONIC MATERIAL



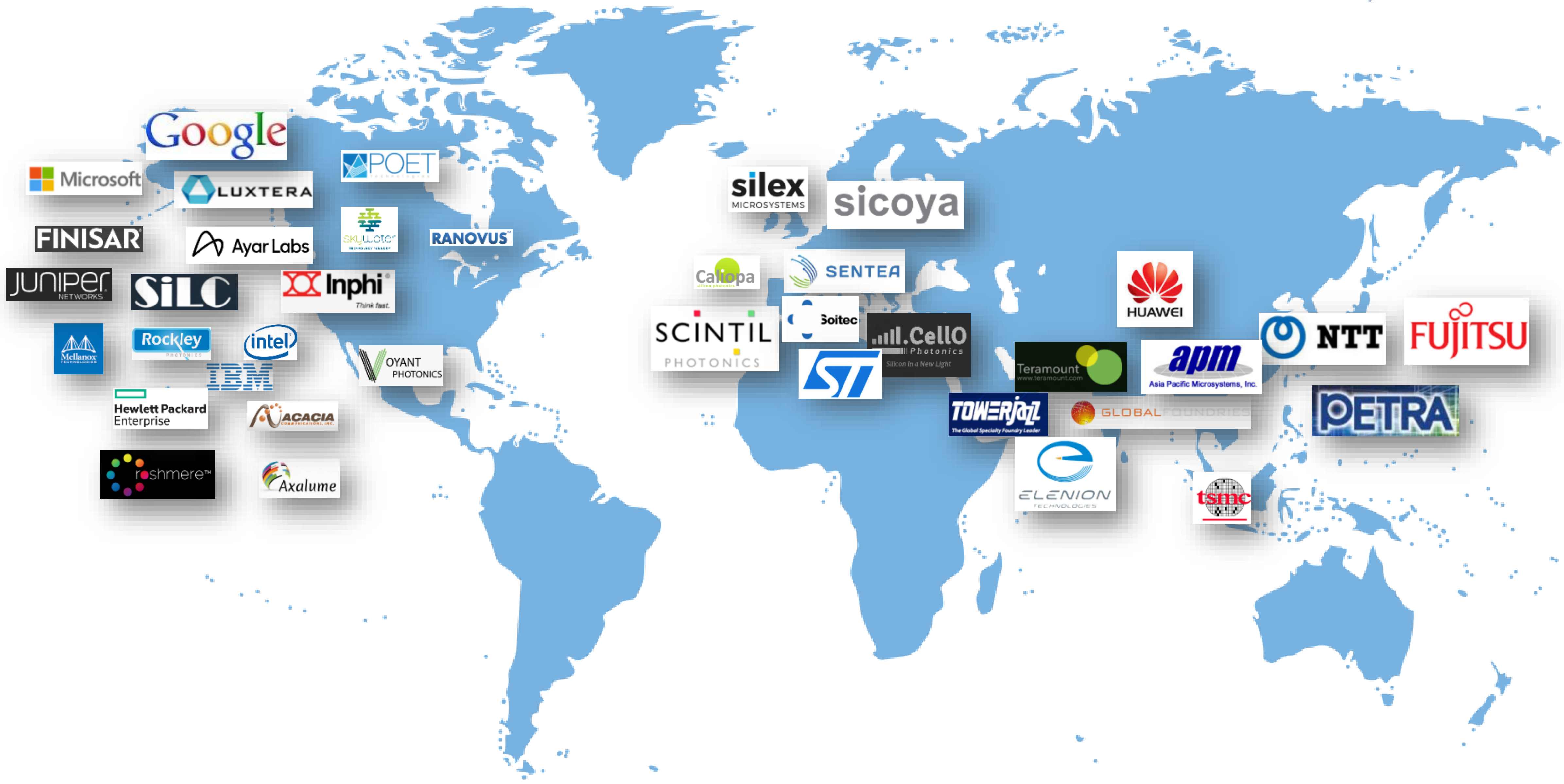
Indirect bandgap: no light emission

High waveguide loss

No efficient modulation mechanism

Poor absorption for telecom wavelengths

Silicon Photonics industrial landscape

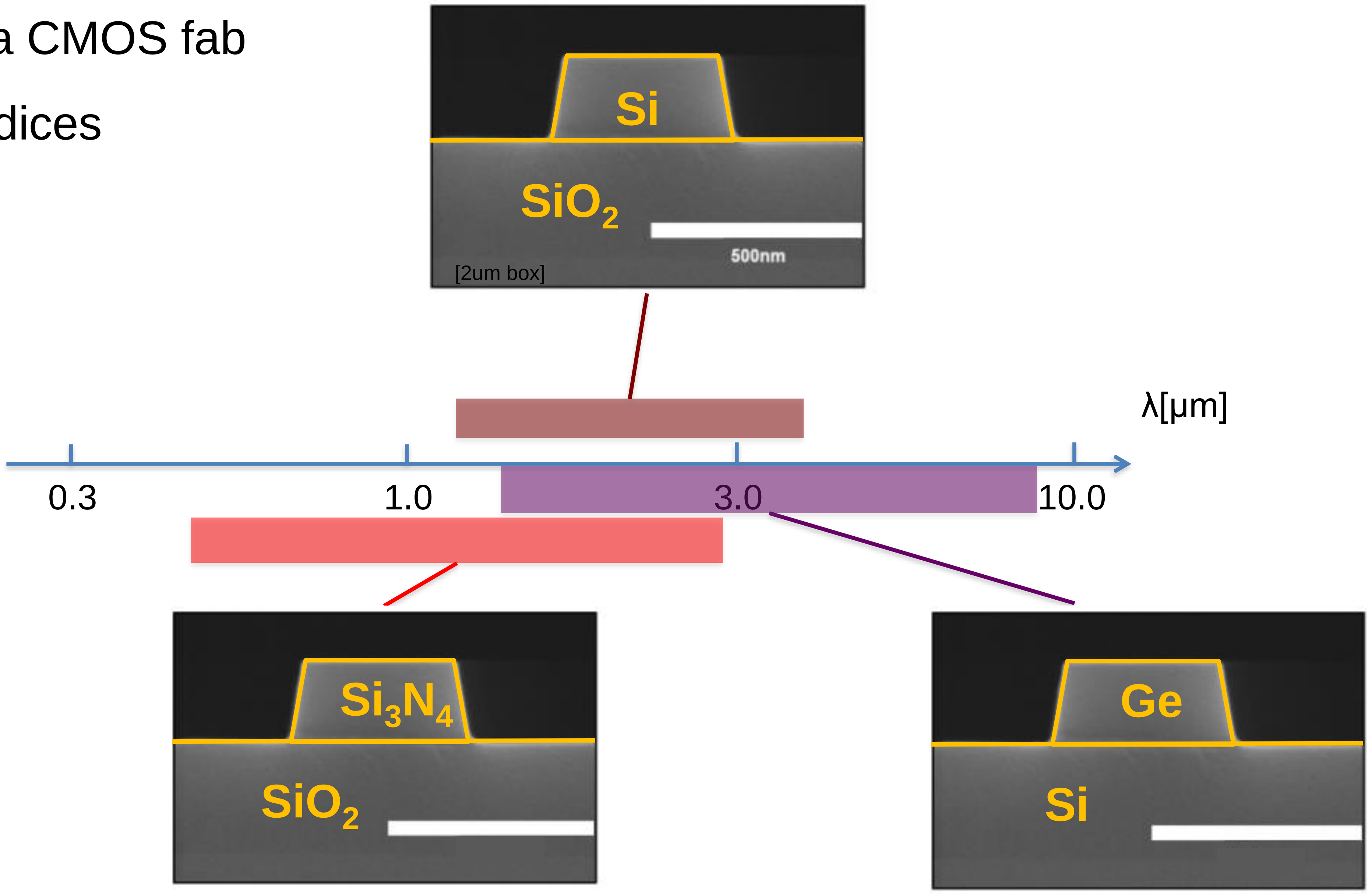


Source: Yole developpement

SILICON PHOTONICS: WAVELENGTHS AND MATERIALS

Compatible with a CMOS fab

High refractive indices



WHY SILICON PHOTONICS?

Large scale manufacturing

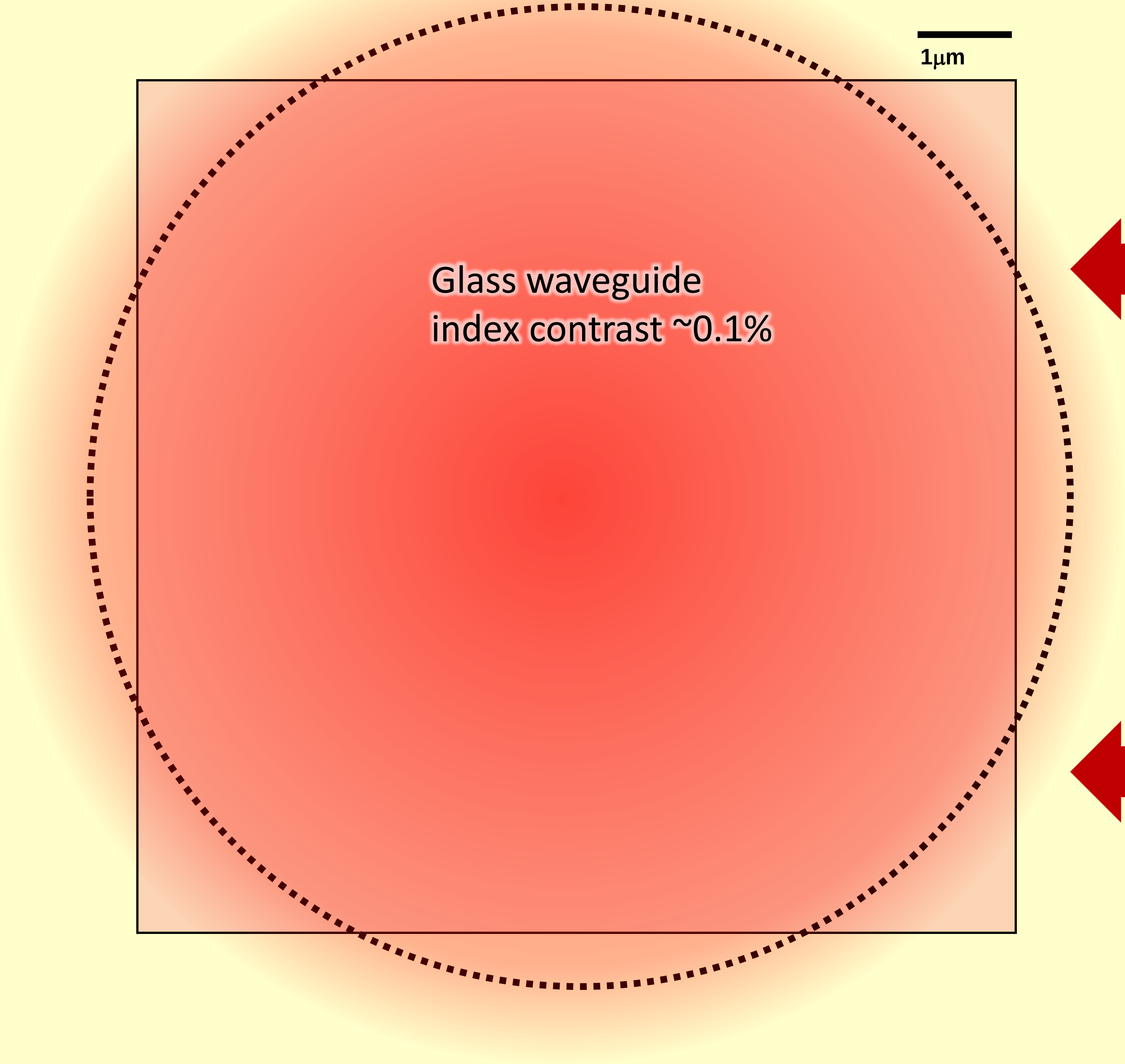
↑

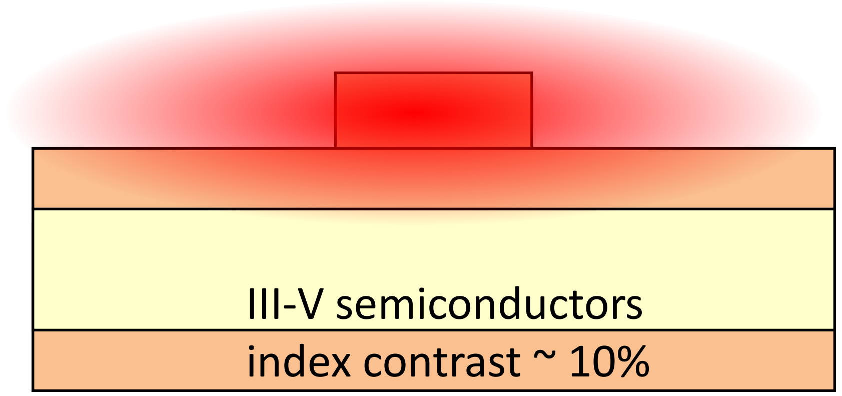
Scale

↓

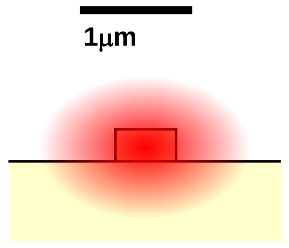
Submicron-scale waveguides

Scaling on-chip waveguides



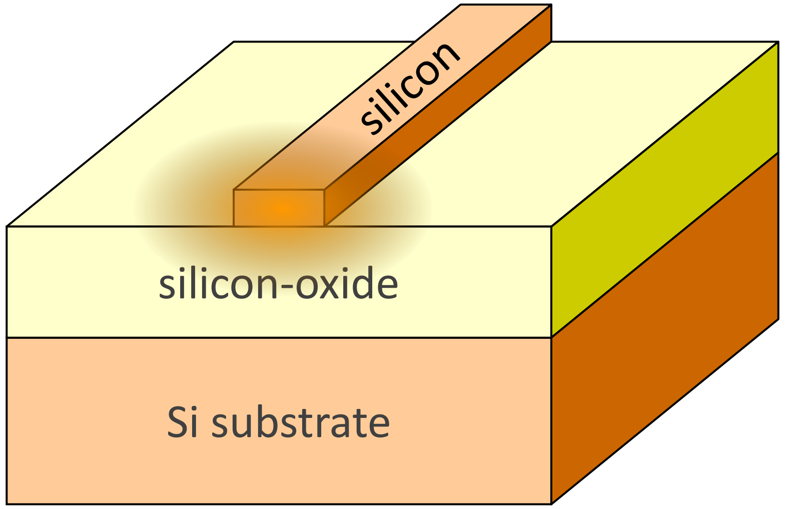


Higher index contrast
Smaller waveguides



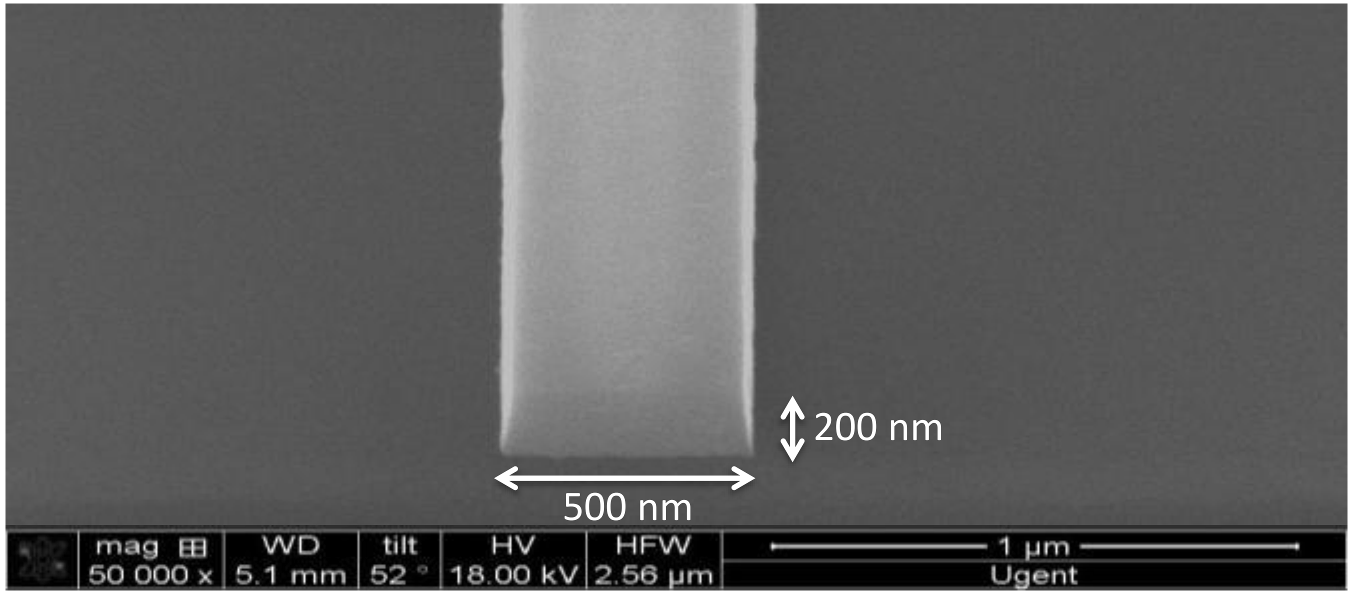
silicon wire:
index contrast ~ 200%

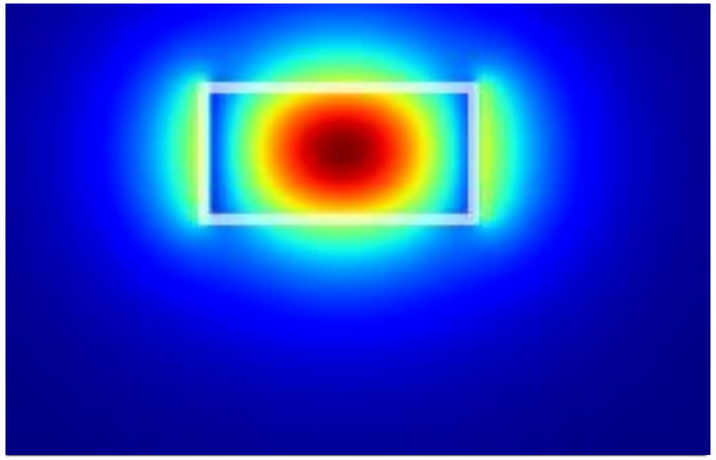
SILICON PHOTONIC WAVEGUIDES



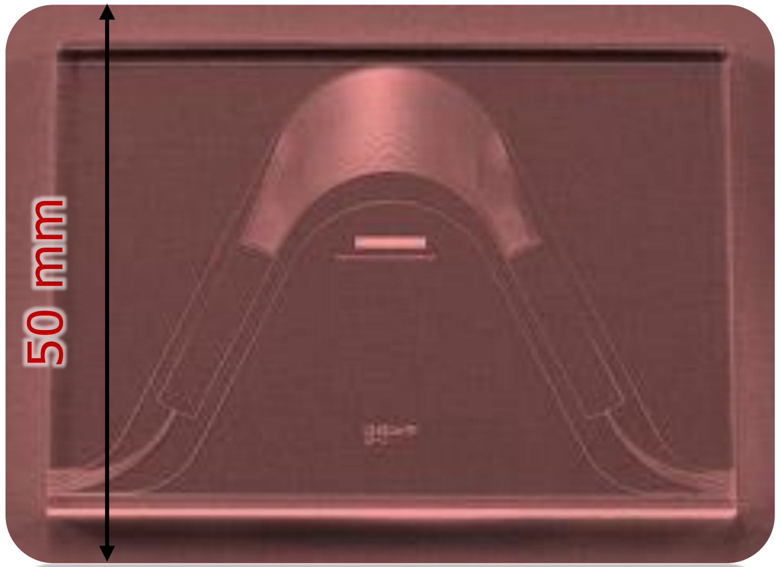
$$n_{core} = 3.45$$
$$n_{cladding} = 1.45$$

High intensity on sidewalls



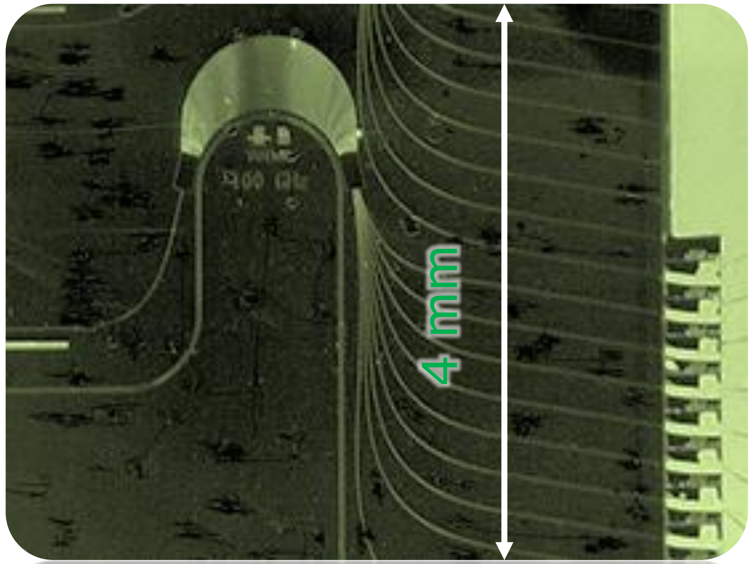


HIGHER CONTRAST, SMALLER CORES, TIGHTER BENDS



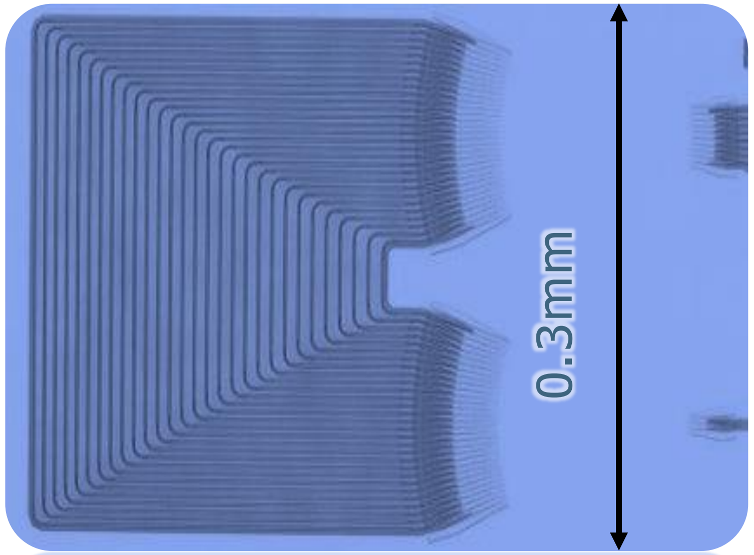
Silica on silicon

Contrast ~ 0.01 – 0.1
Mode diameter ~ 8µm
Bend radius ~ 5mm
Size ~ 10 cm^2



Indium Phosphide

Contrast ~ 0.2 – 0.5
Mode diameter ~ 2µm
Bend radius ~ 0.5mm
Size ~ 10mm^2



Silicon on insulator

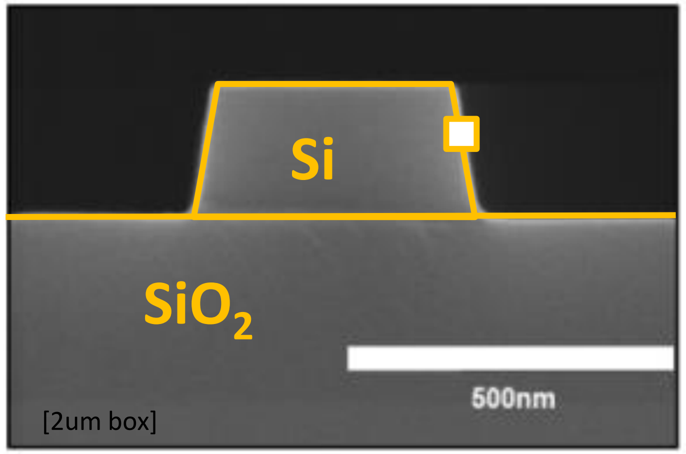
Contrast ~ 1.0 – 2.5
Mode diameter ~ 0.4µm
Bend radius ~ 5µm
Size ~ 0.1mm^2

10000 ×

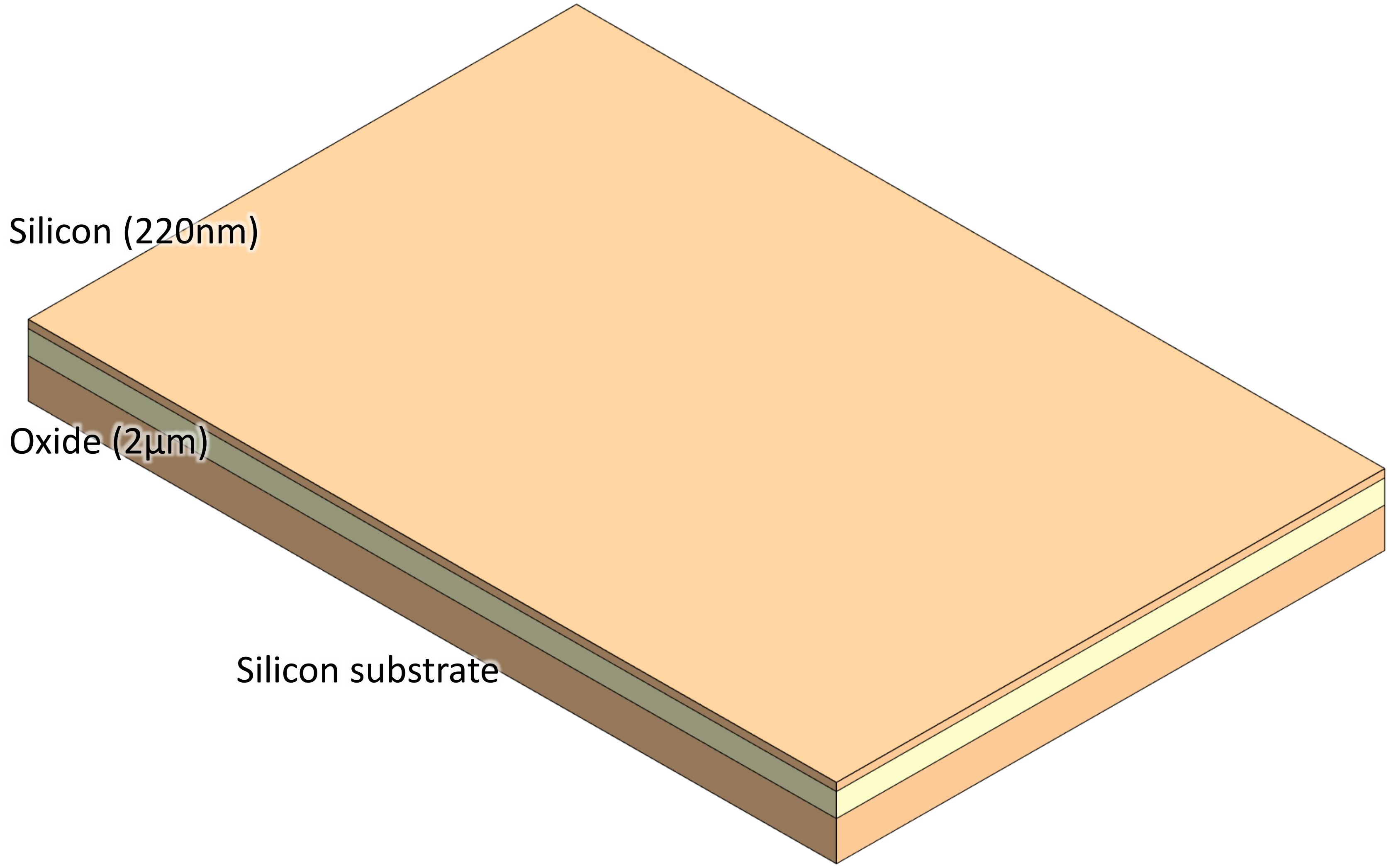
High index contrast: a blessing and a curse

Every nm^3 matters

CMOS technology is the only manufacturing technology with sufficient nm-process control to take advantage of the blessing without suffering from the curse

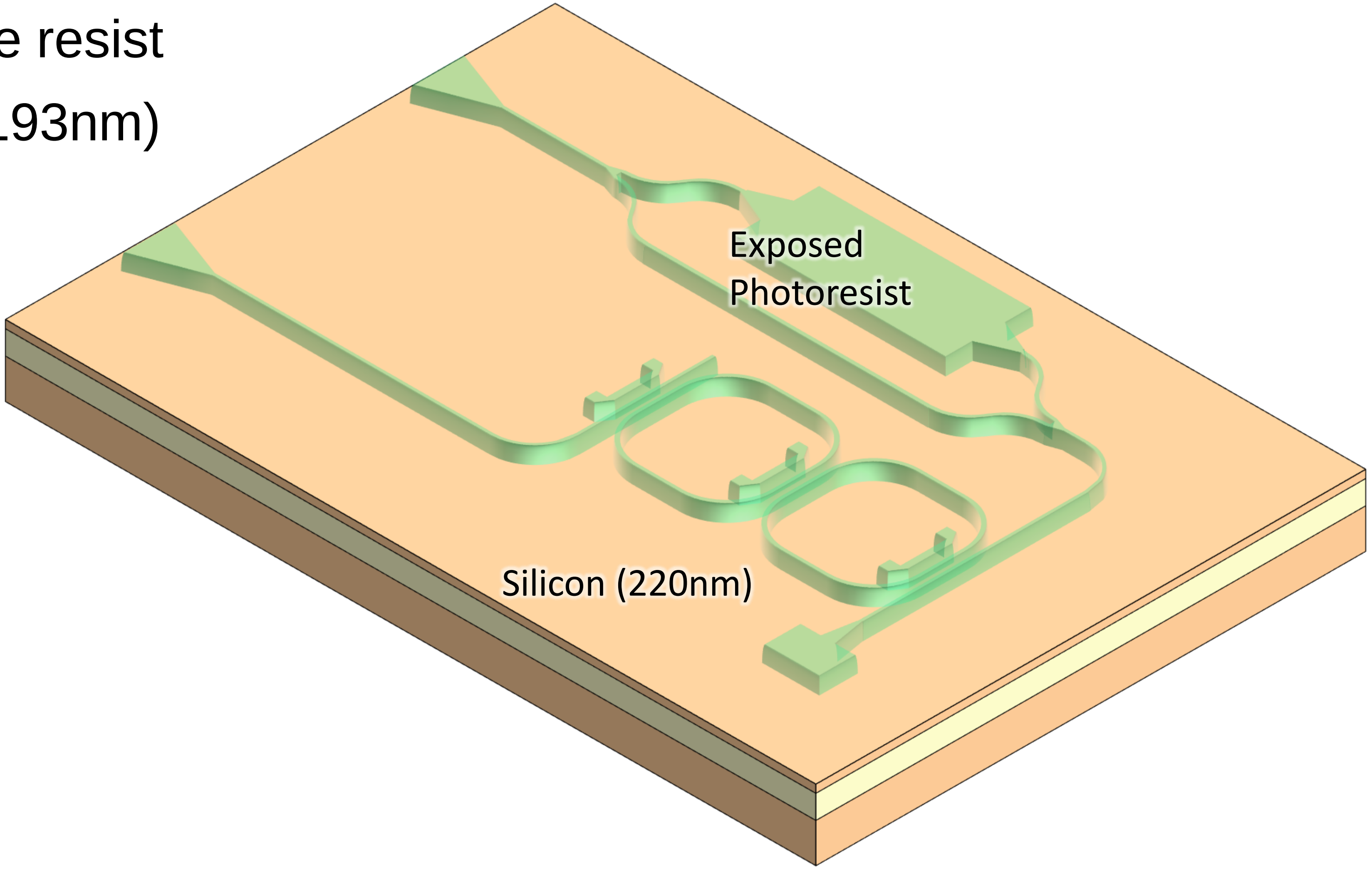


BARE SILICON-ON-INSULATOR WAFER



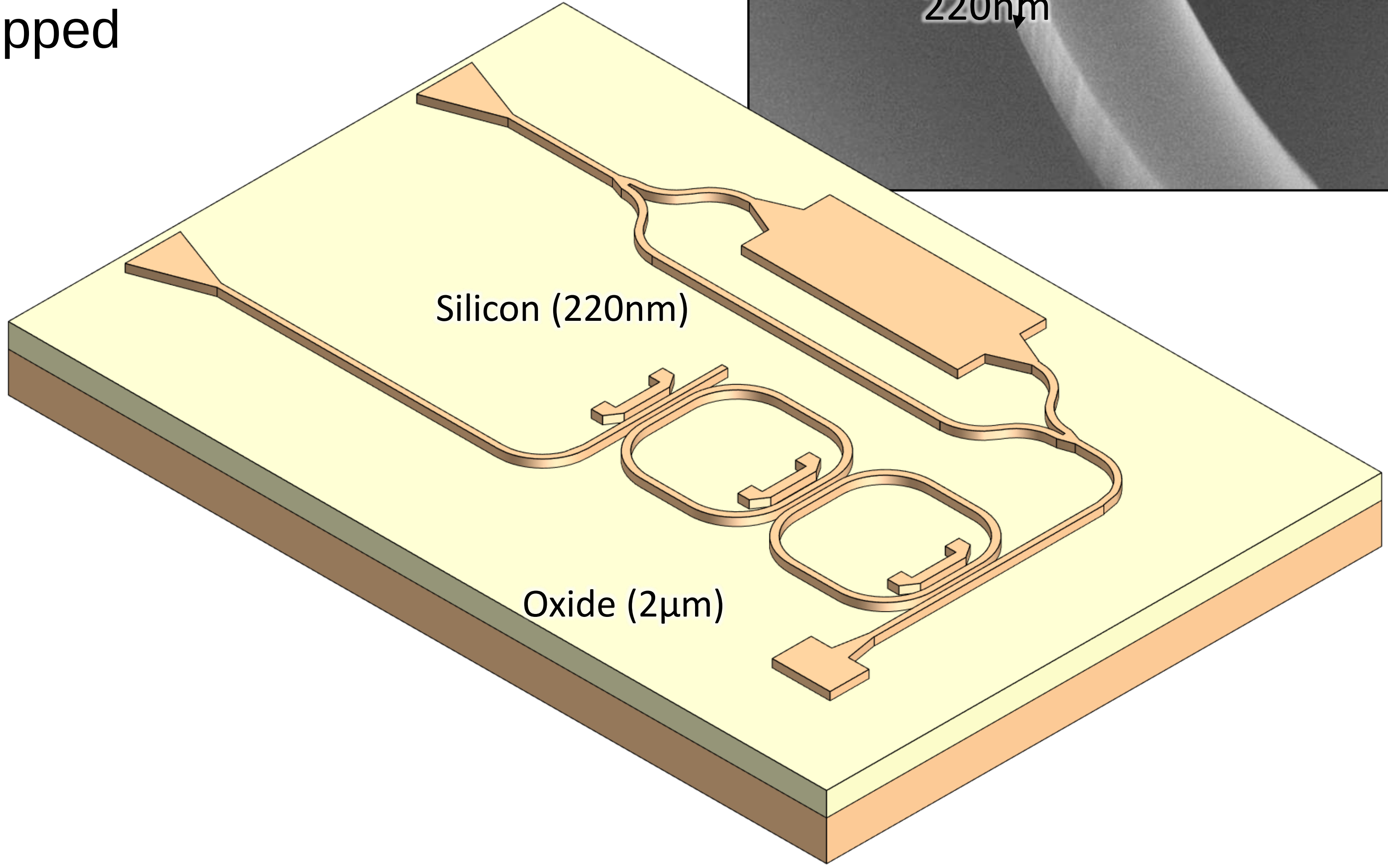
PHOTOLITHOGRAPHY

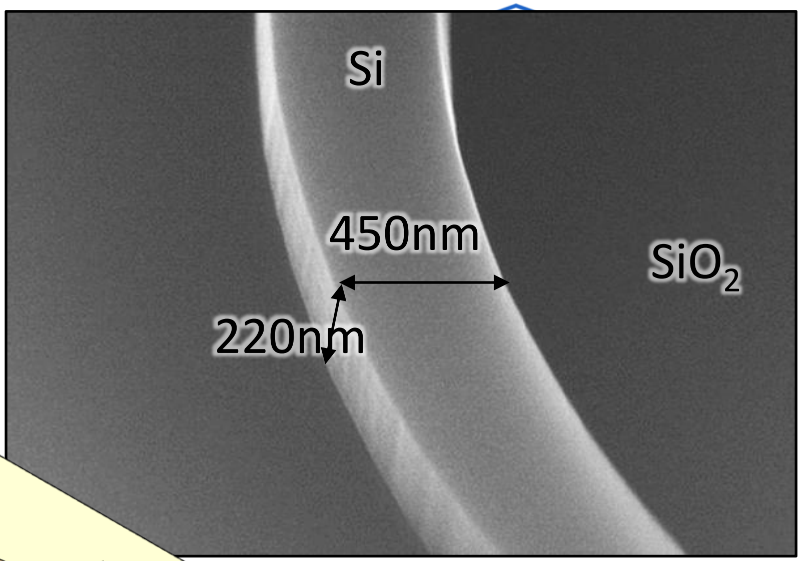
1. Spin-coat Photoresist + pre-bake
2. Mask is projected in the resist
   (UV light at 248nm or 193nm)
3. Post-Exposure bake
4. Resist is developed



SILICON ETCHING

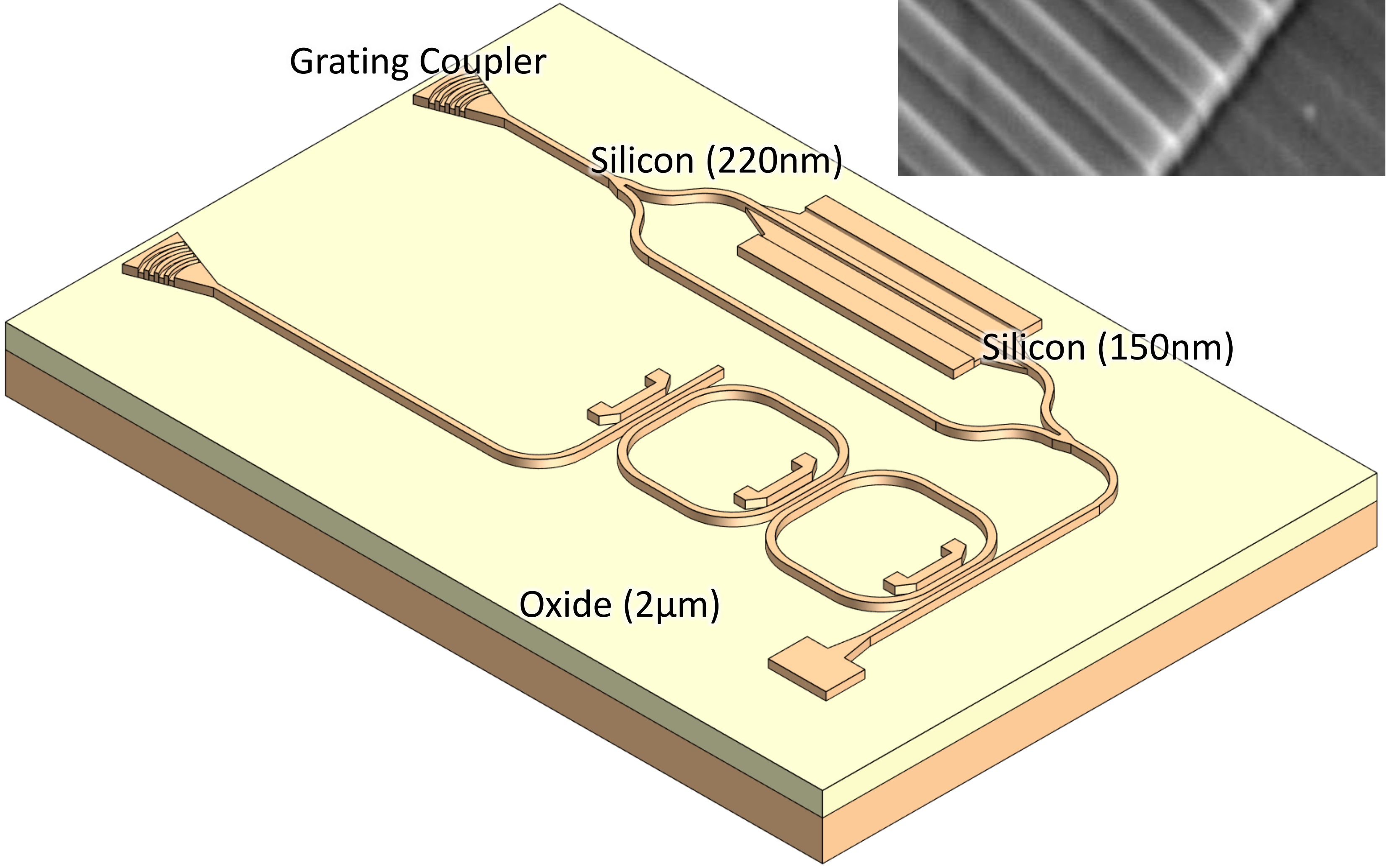
1. Plasma etches the exposed silicon
2. Remaining resist is stripped

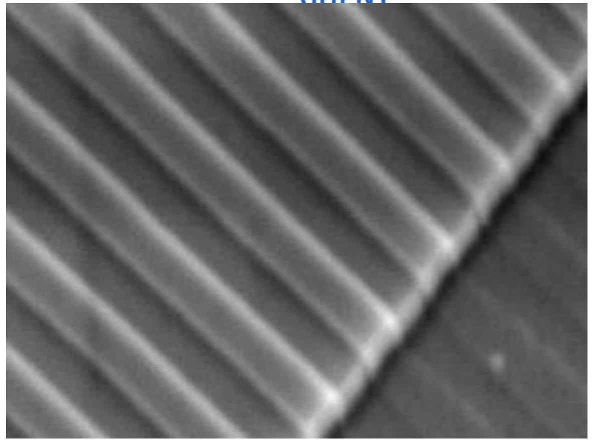




PARTIAL SILICON ETCHING

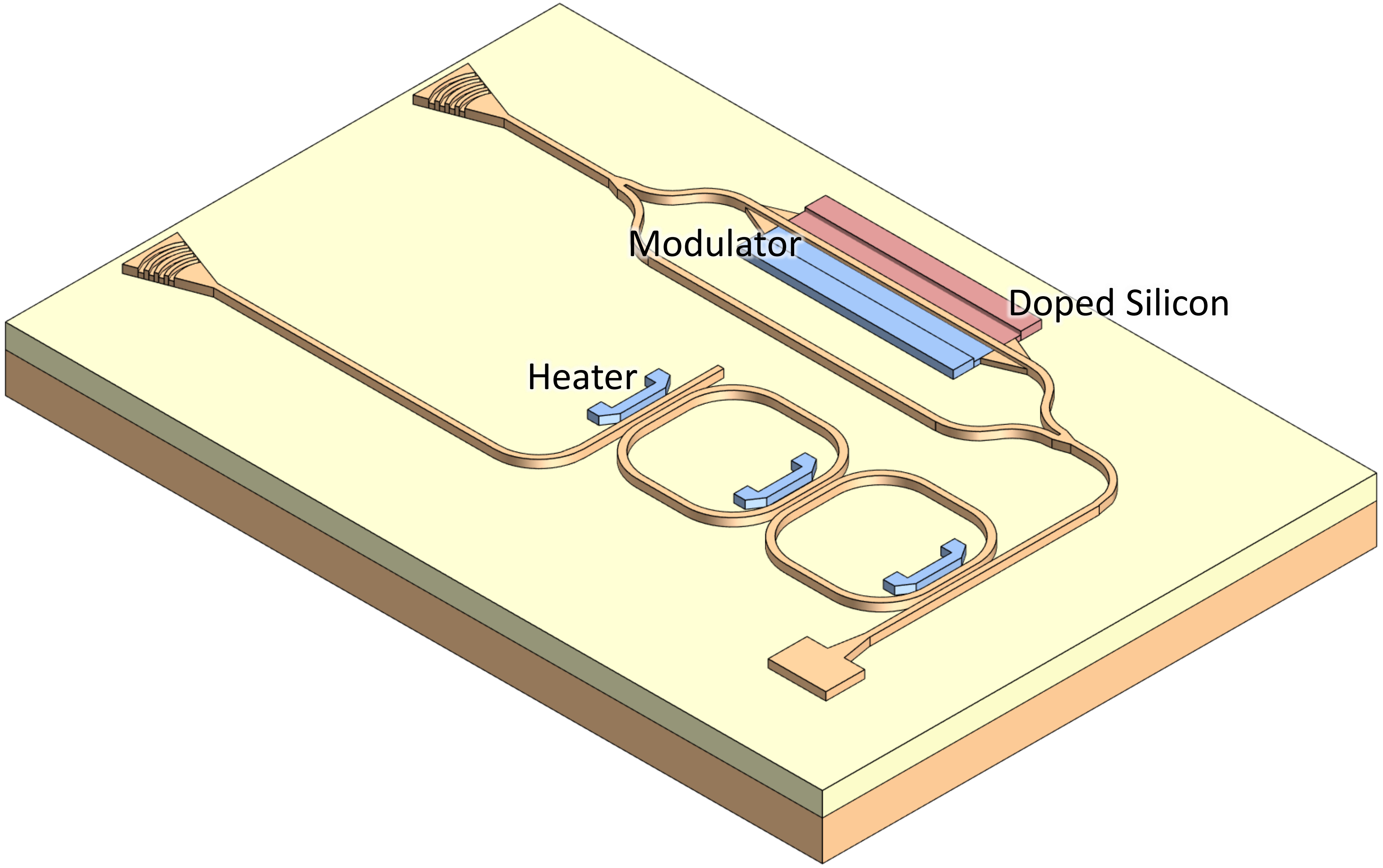
1. Lithography of second layer
2. Plasma etching
3. Resist Stripping





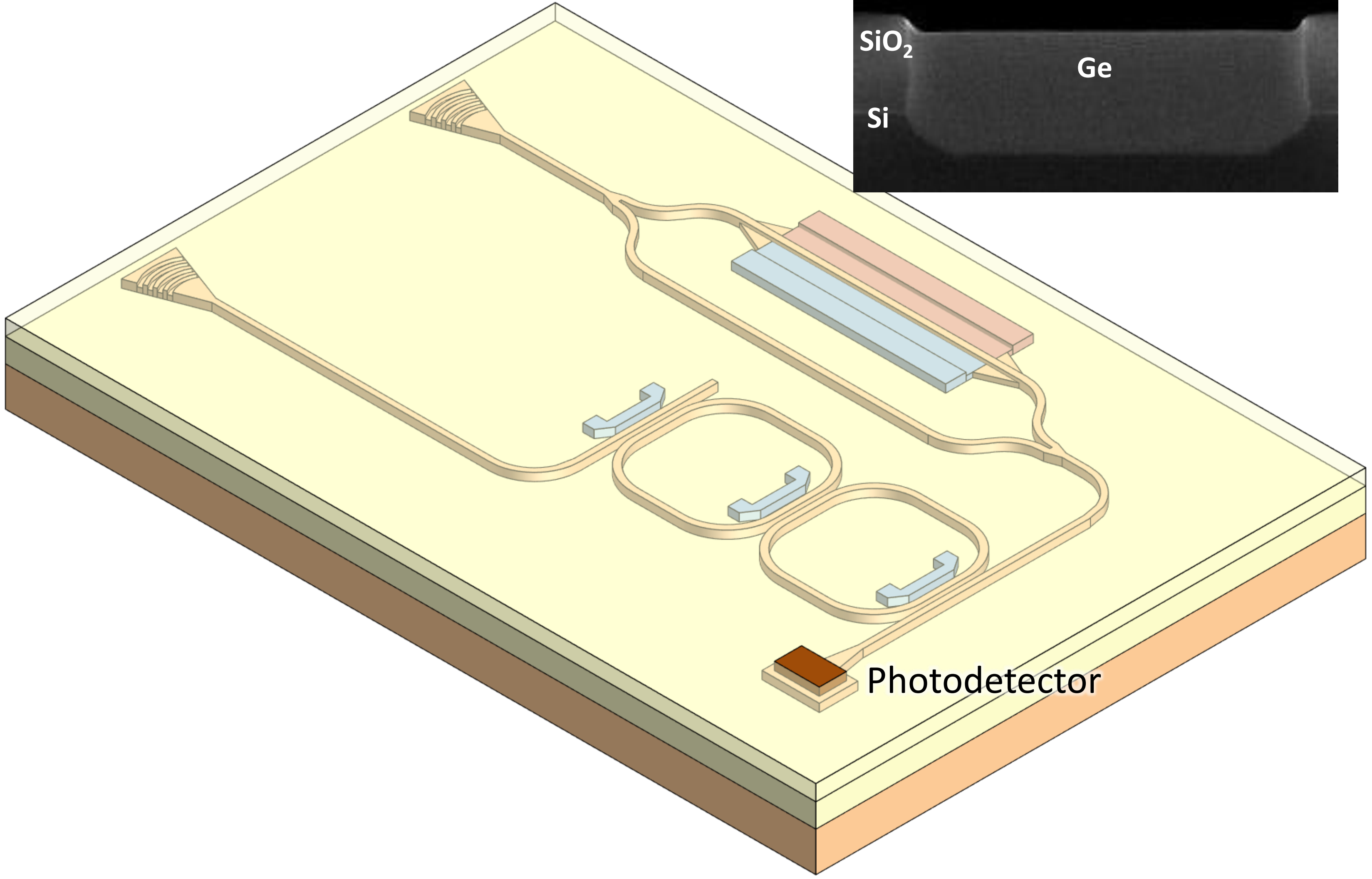
Doped regions for Modulators and Heaters

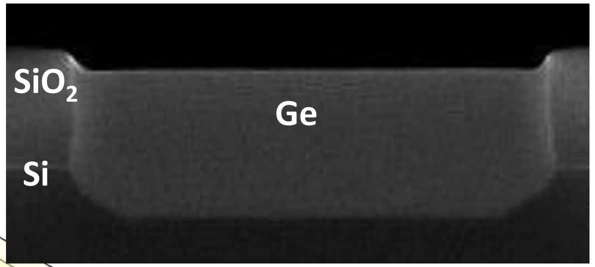
1. Lithography of windows
2. Ion implantation
3. Resist Stripping



GERMANIUM PHOTODETECTORS

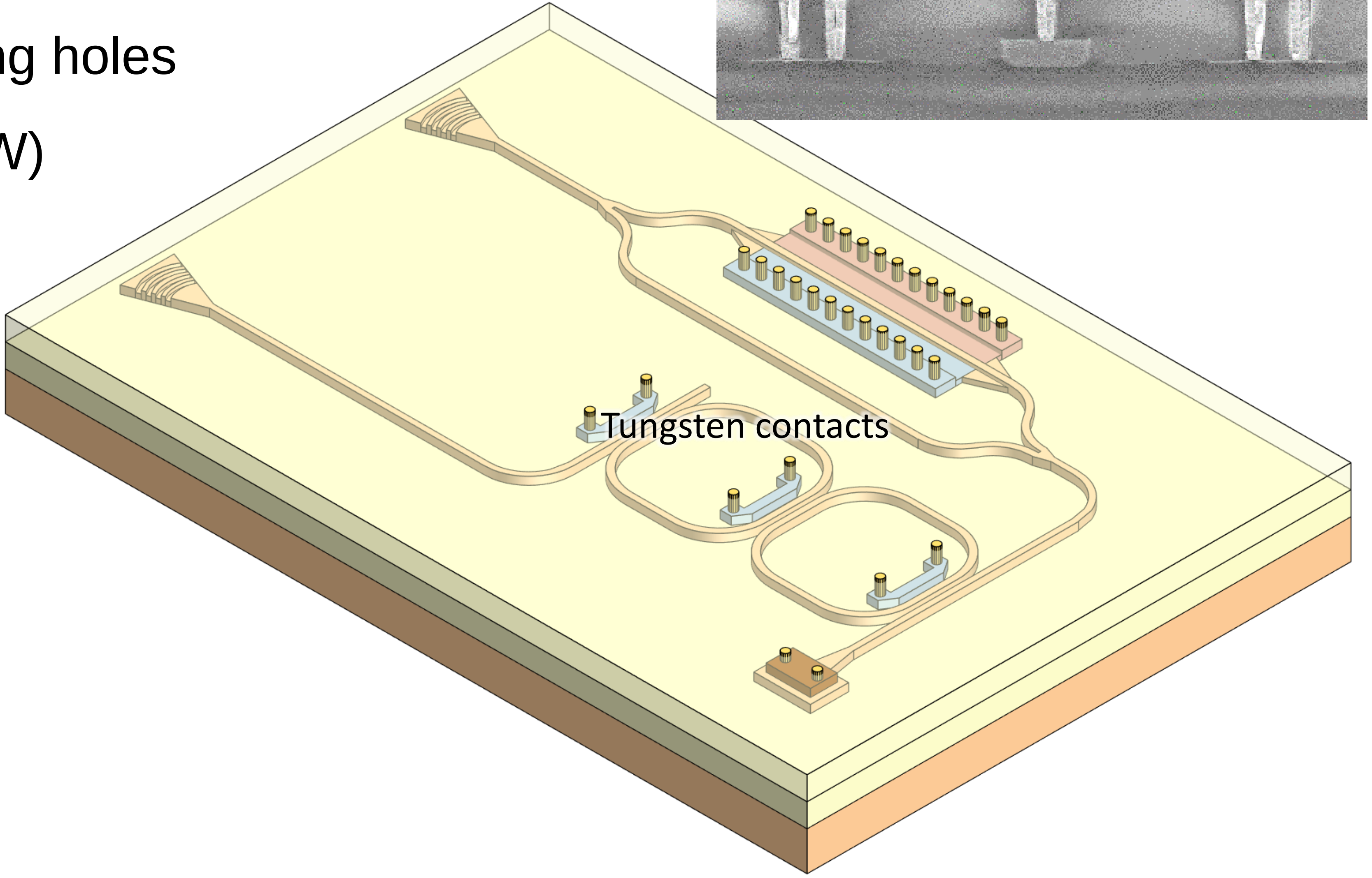
1. Oxide cladding
2. Planarization (CMP)
3. Opening of window
4. Epitaxial Growth of Ge
5. Planarization (CMP)





ELECTRICAL CONTACTS: DAMASCENE PROCESS

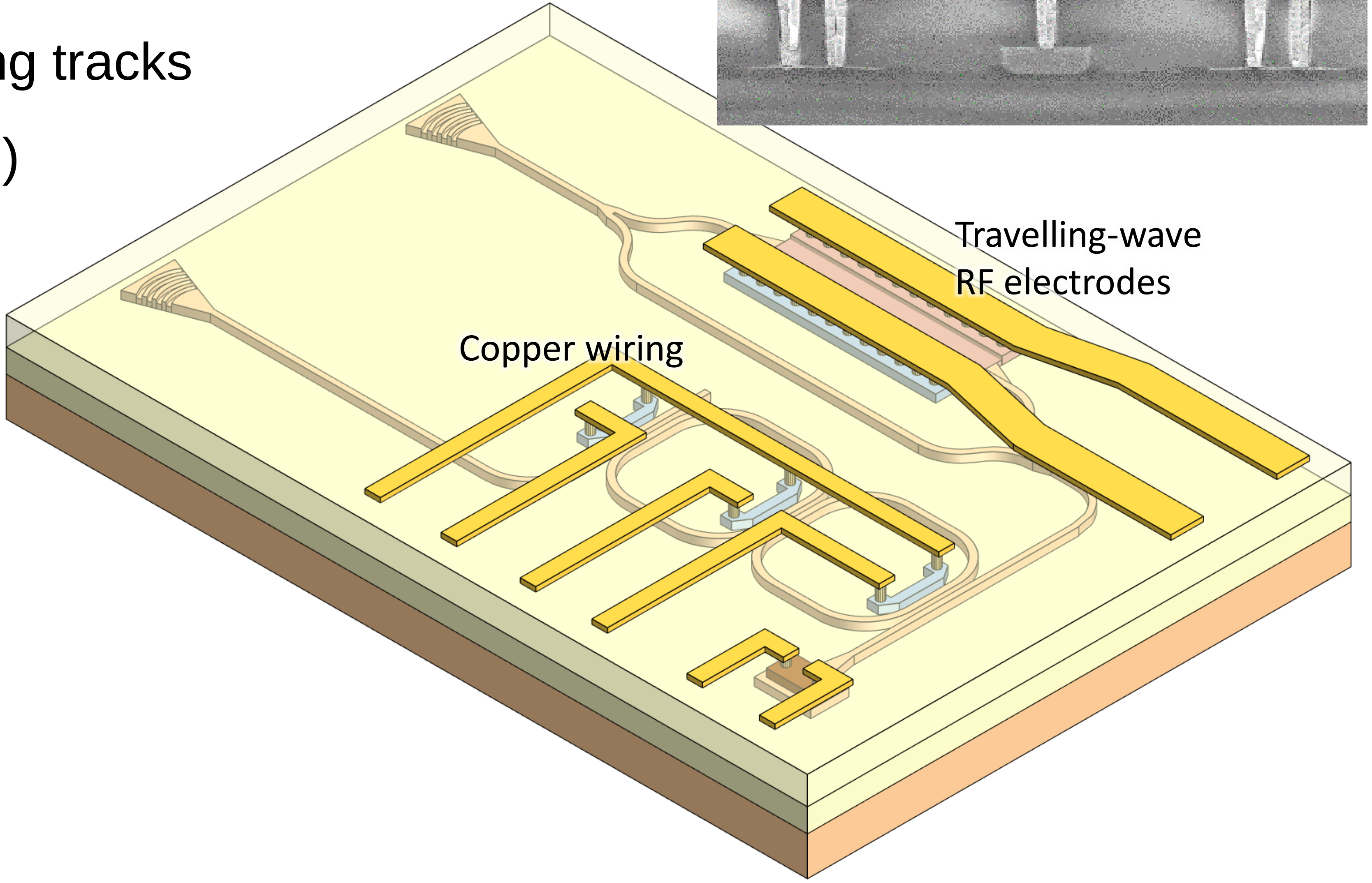
1. Depositing dielectric layers
2. Lithography and Etching holes
3. Filling with Tungsten (W)
4. Planarization (CMP)

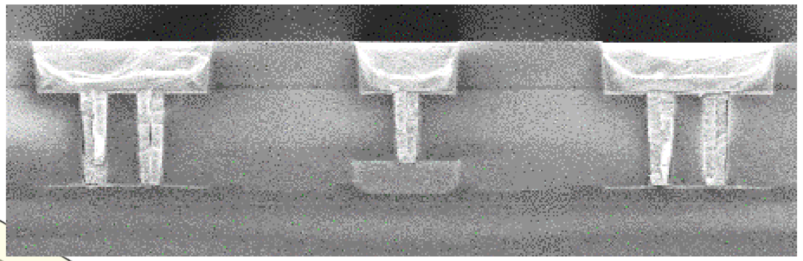


Metal Interconnects: Damascene process

1. Depositing dielectric layers
2. Lithography and Etching tracks
3. Filling with Copper (Cu)
4. Planarization (CMP)

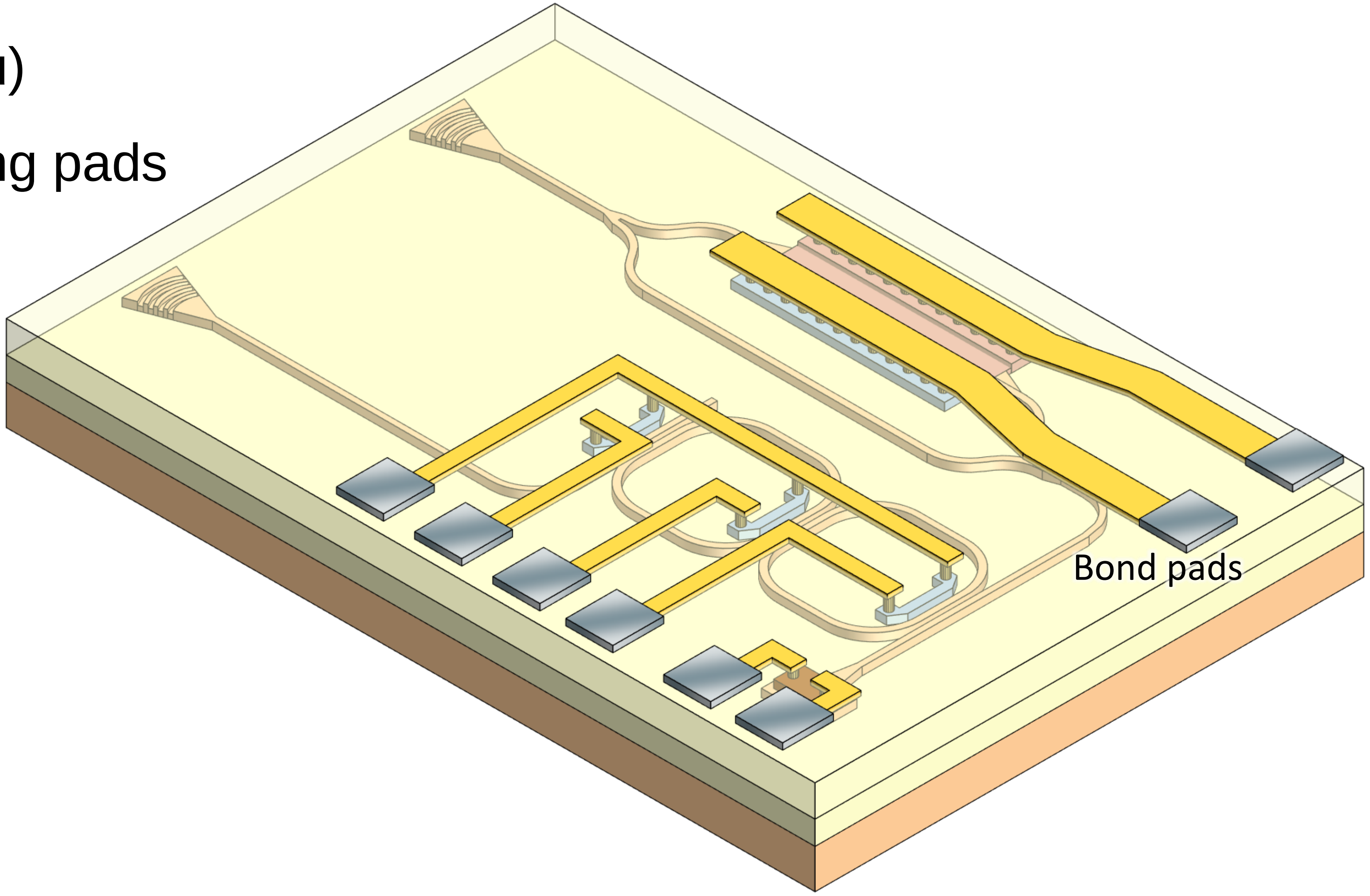
Repeat for more layers





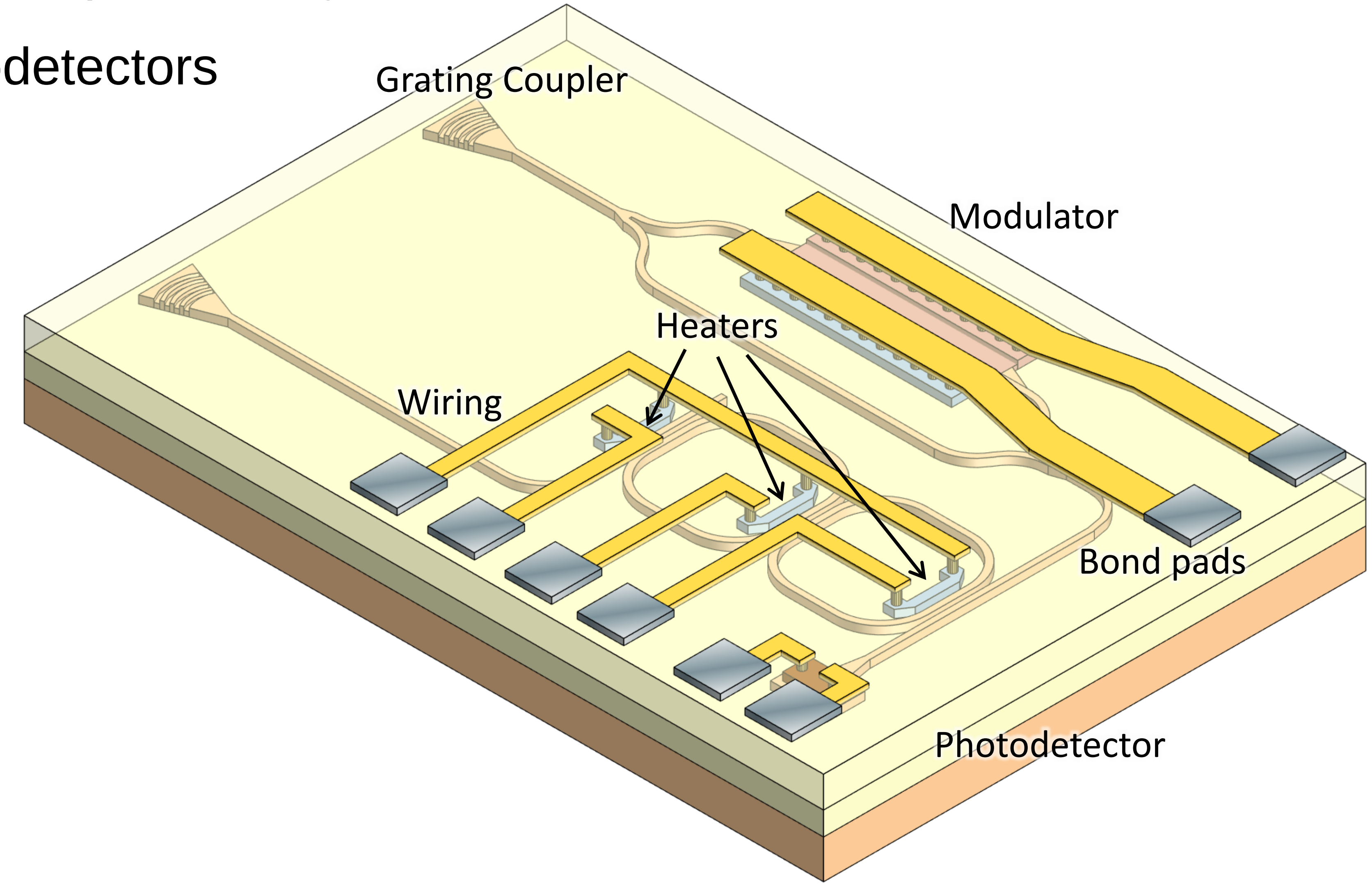
METAL BONDPADS

1. Deposit dielectric layers
2. Depositing Metal (AlCu)
3. Lithography and Etching pads



SILICON PHOTONICS CHIPS

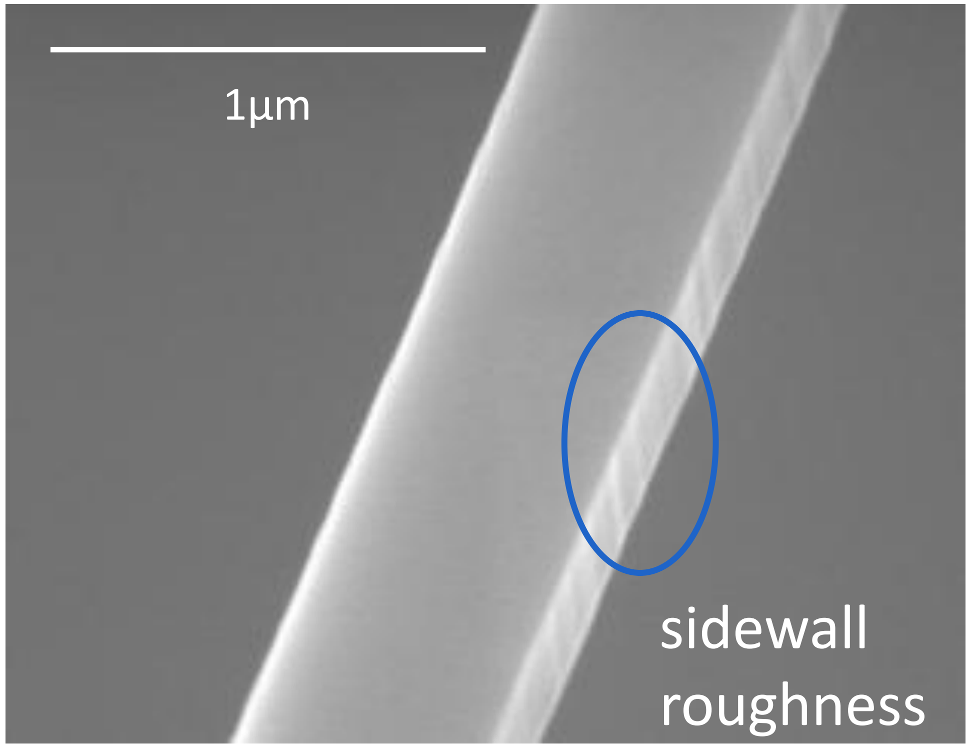
1. Passive circuits with multiple etch layers
2. Modulators and Photodetectors
3. Metal wiring

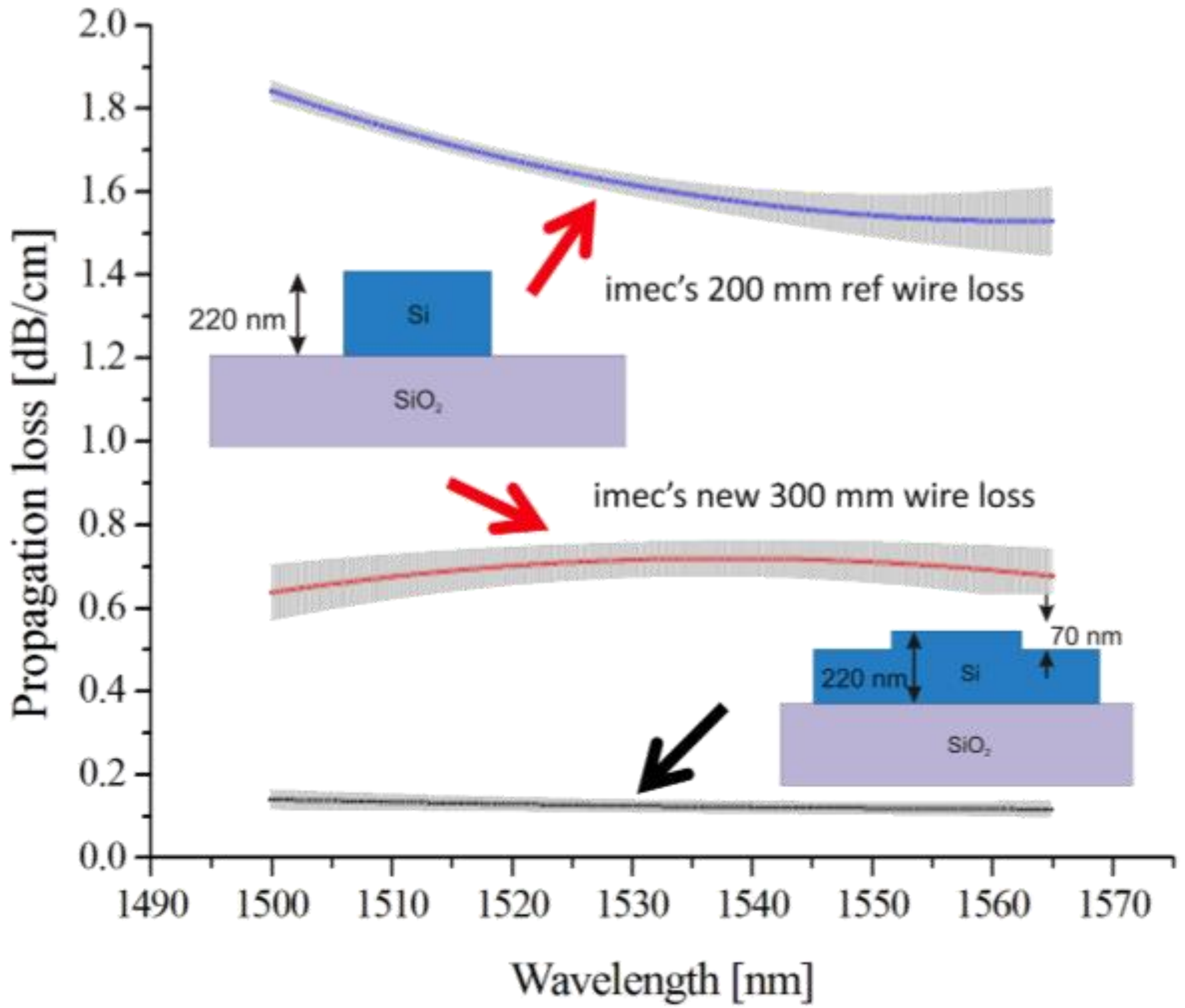


Waveguides

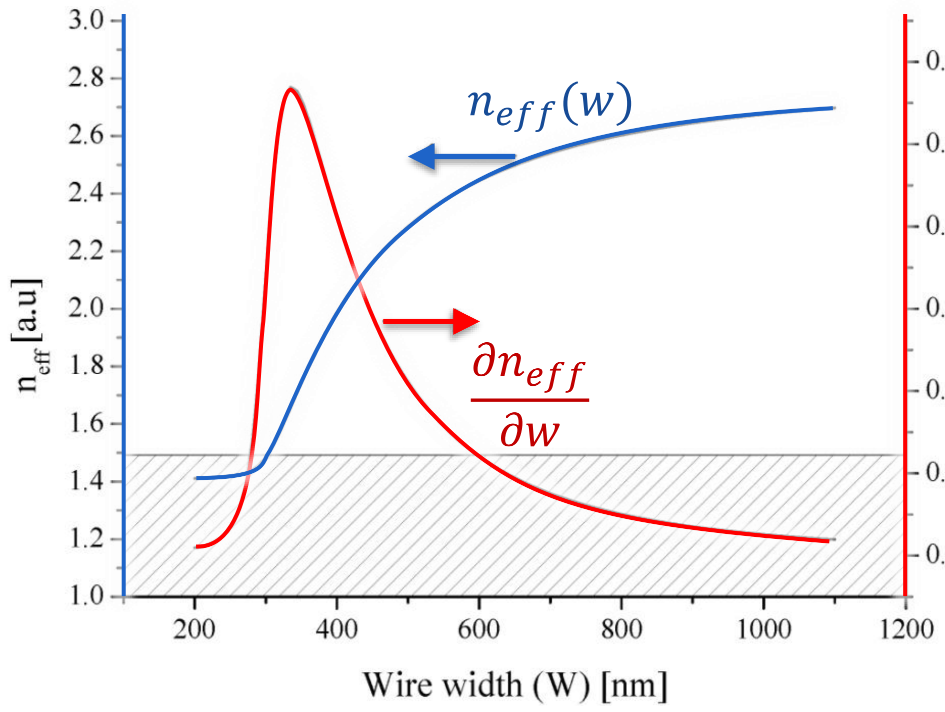
Waveguide losses dominated by scattering.

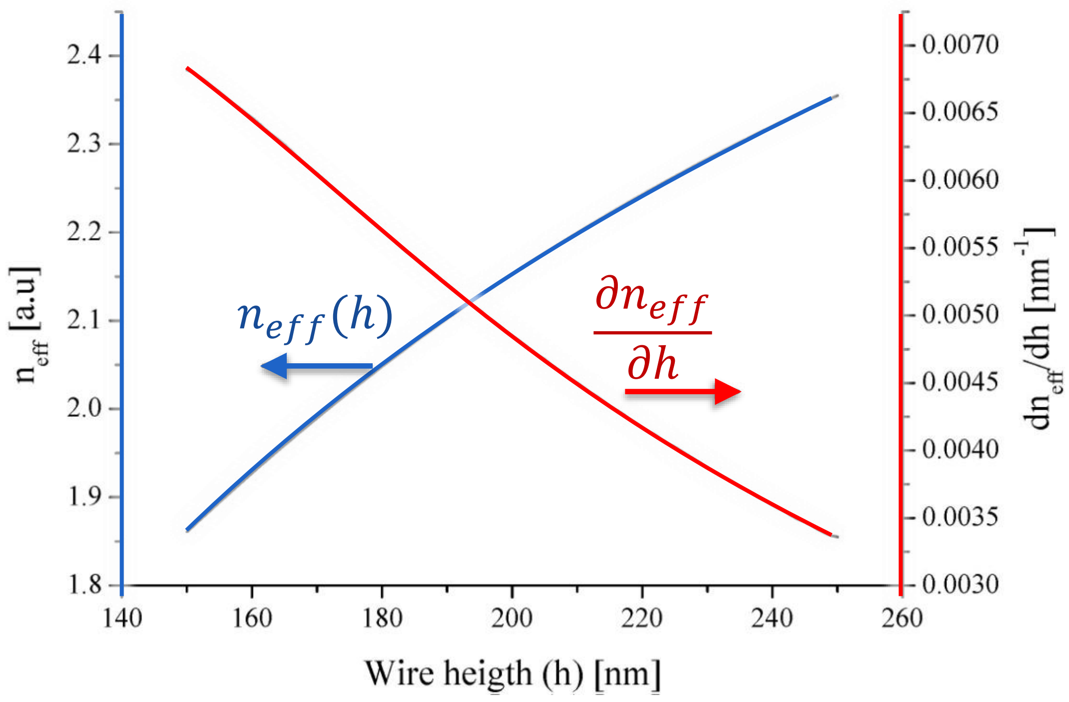
Use better litho + etch

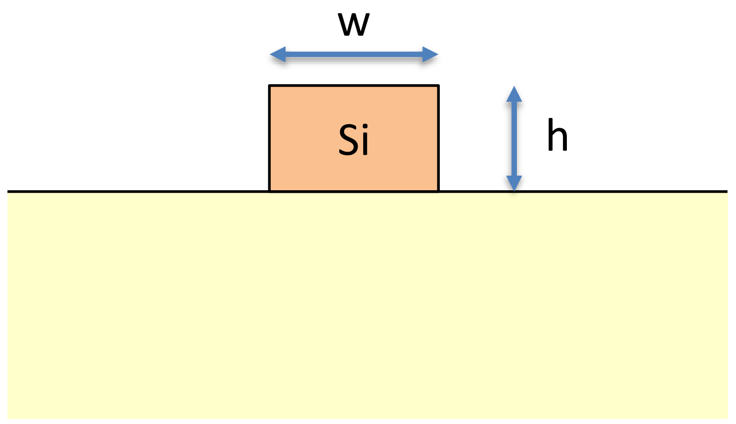




DIMENSIONAL DEPENDENCE OF A WAVEGUIDE



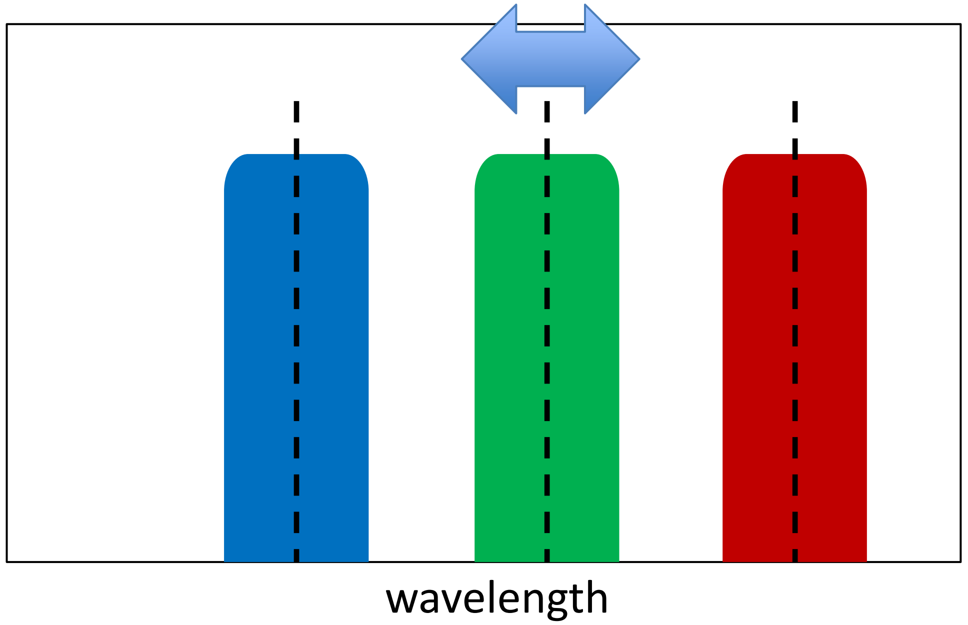




SENSITIVITY OF SILICON PHOTONICS WAVELENGTH FILTERS

Especially wavelength filters are sensitive:

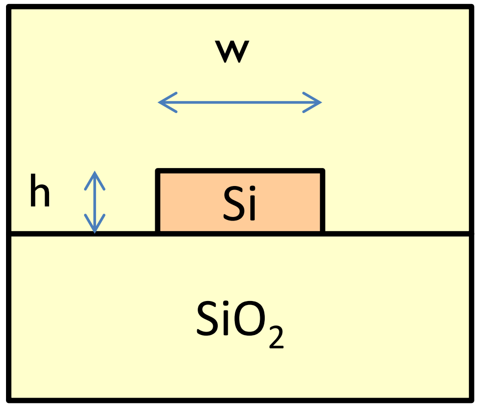
- geometry
- stress
- temperature



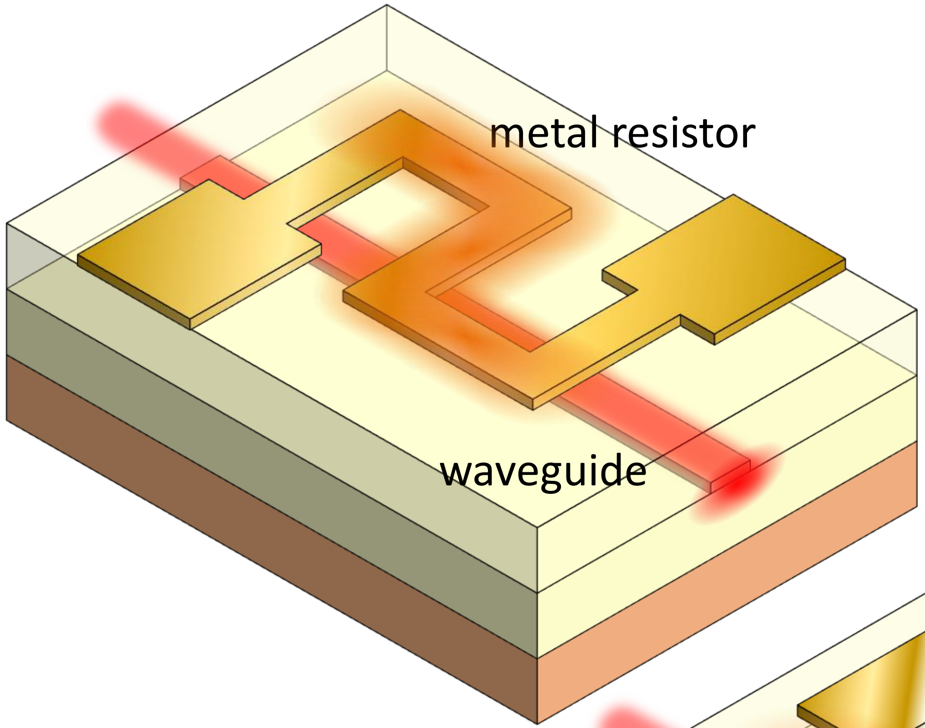
wire width $\frac{\partial\lambda}{\partial w} \approx 1\, {}^{nm}/_{nm}$

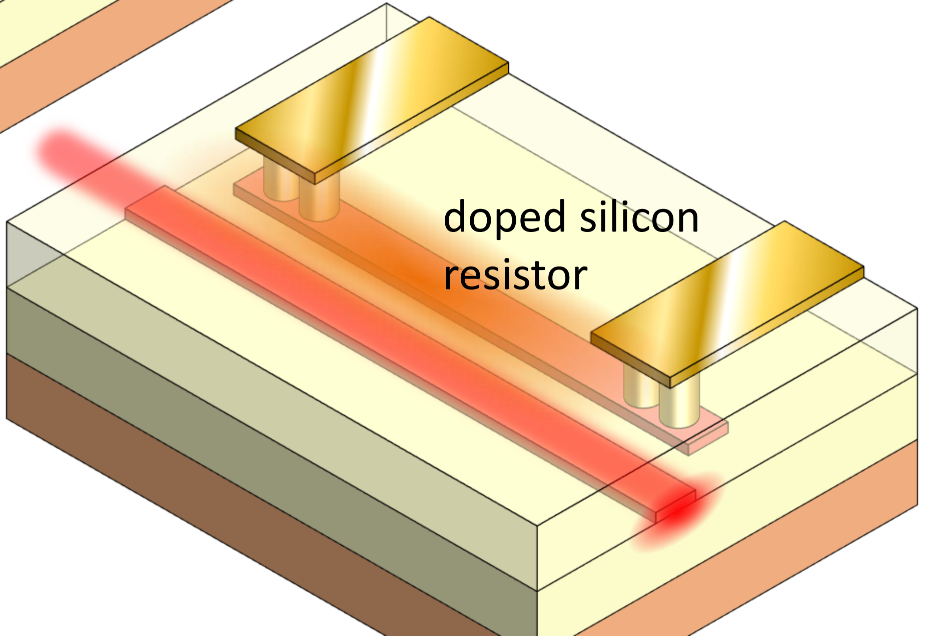
wire height $\frac{\partial\lambda}{\partial h} \approx 2\, {}^{nm}/_{nm}$

temperature $\frac{\partial\lambda}{\partial T} \approx 0.08\, {}^{nm}/_{K}$



The basic Optical Phase shifter: a Heater





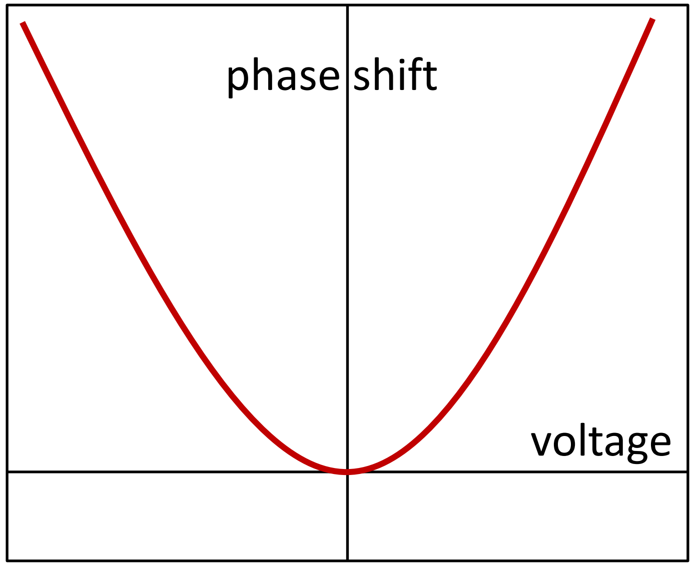
Waveguides are thermally sensitive:

$$\Delta\phi \sim \Delta n_{eff} \sim T \sim P_{elec} \sim V^2 \sim I^2$$

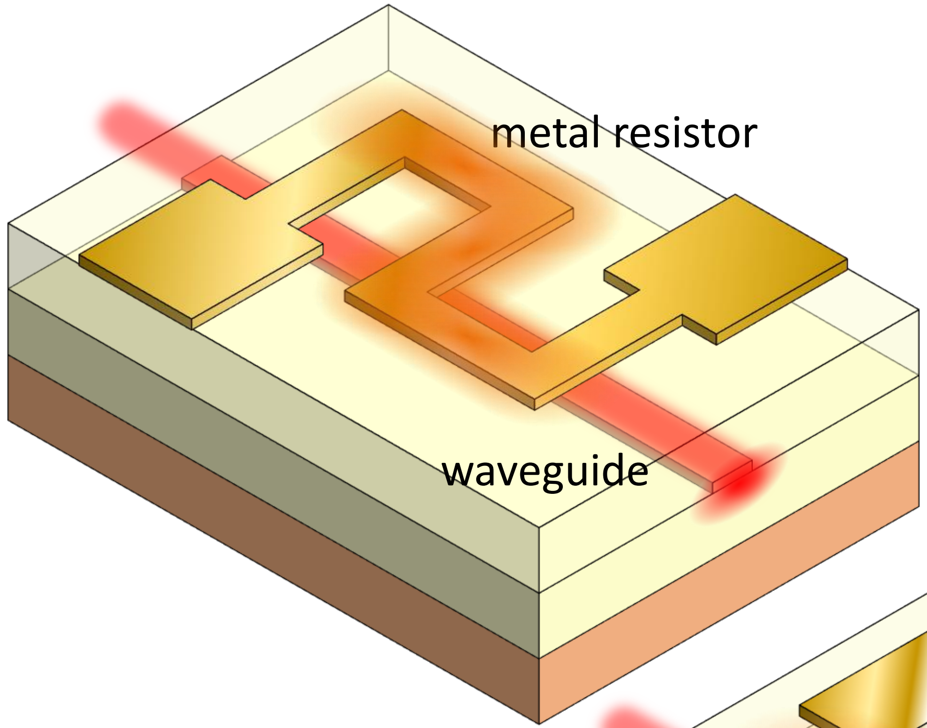
Integrate resistor close to the waveguide

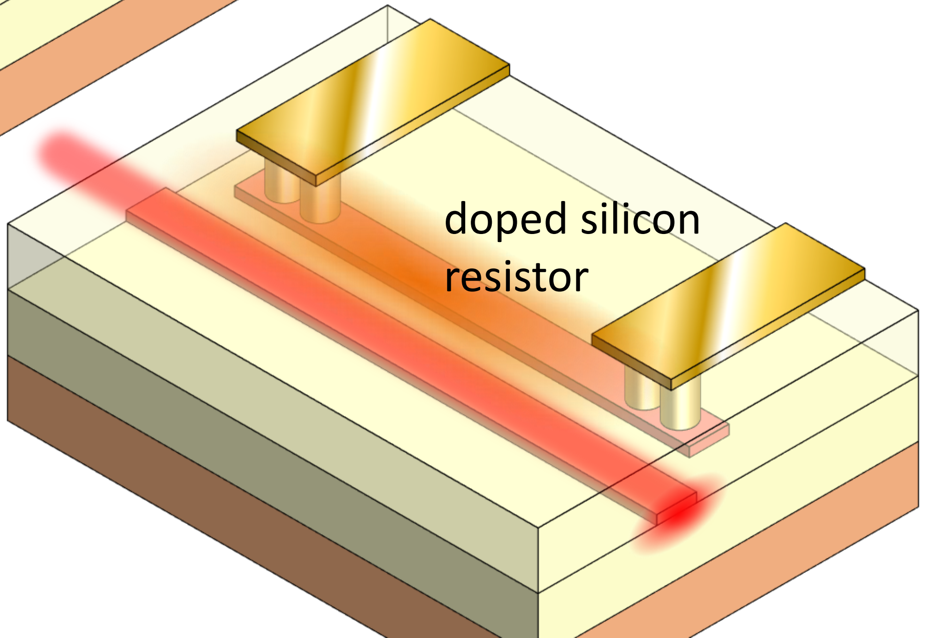
efficiency: $P_\pi \approx 5 - 30mW$

(for silicon waveguides)



THE BASIC OPTICAL PHASE SHIFTER: A HEATER





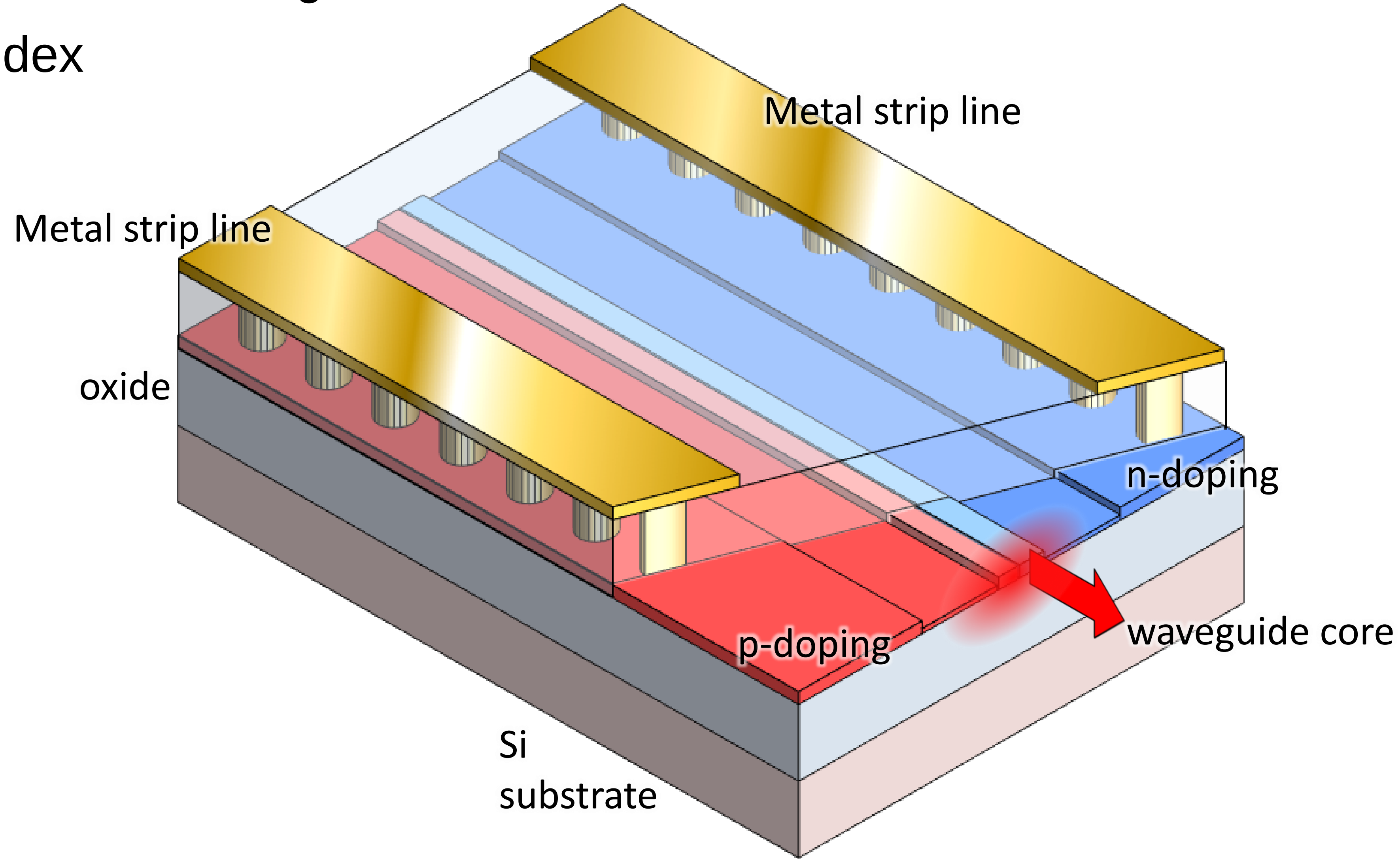
Performance determined by geometry

- not too close to waveguide (metal absorbs)
- volume to be heated (thermal mass)
- Thermal leakage paths

ELECTRICAL SIGNAL MODULATION

Add doped junction to silicon waveguide:

modulate refractive index

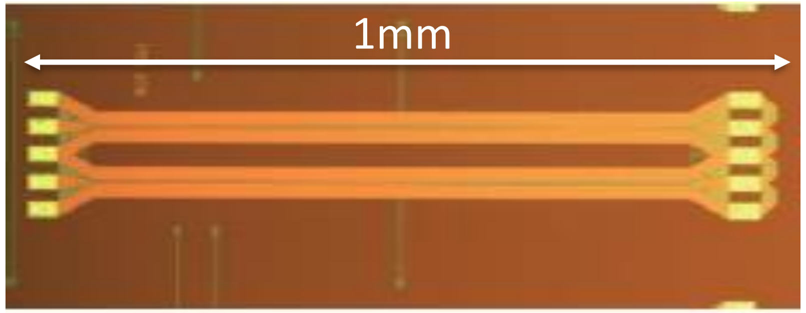


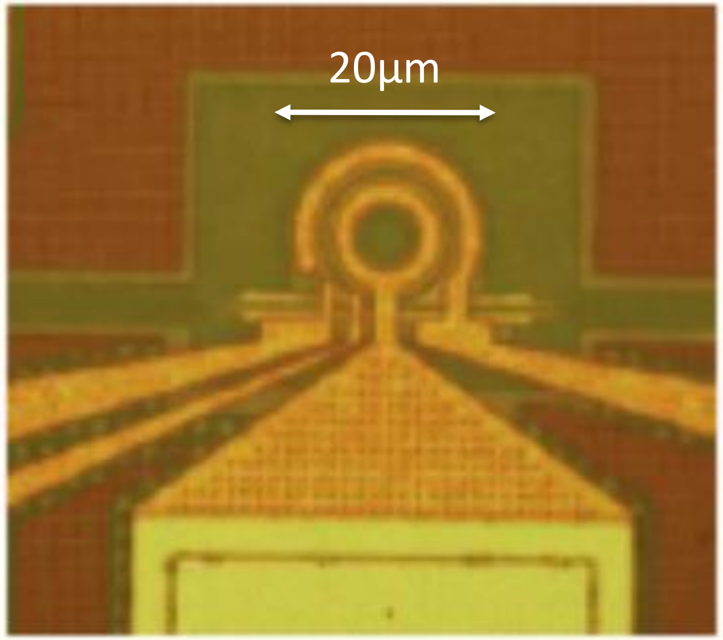
Electrical signal modulation

Add doped junction to silicon waveguide:

modulate refractive index

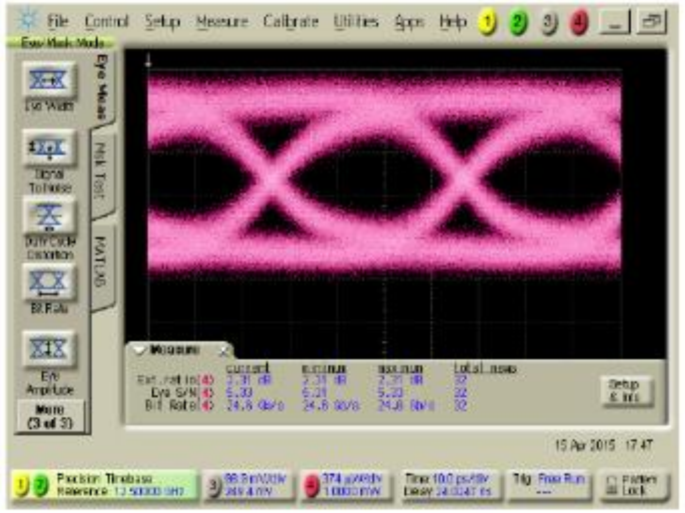
- travelling wave modulator
- ring resonator modulator





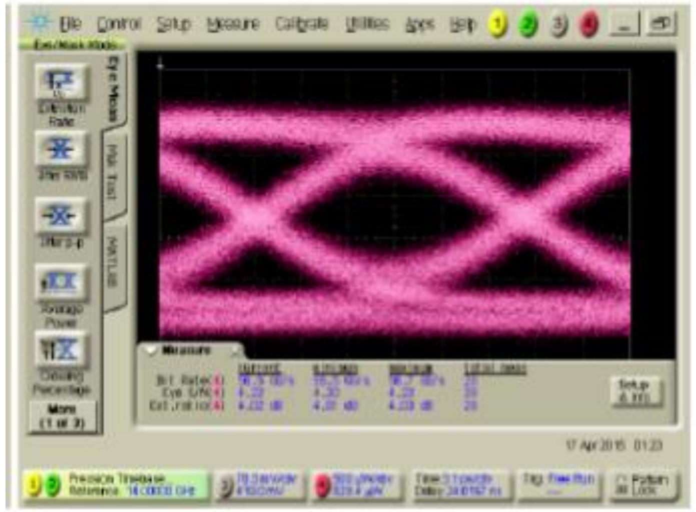
25Gb/s, 1Vpp

Vbias= -0.2V, ER = 2.3dB, Q = 5.3, Opt. Power=13dbm, 1560nm, PRBS=2e31-1

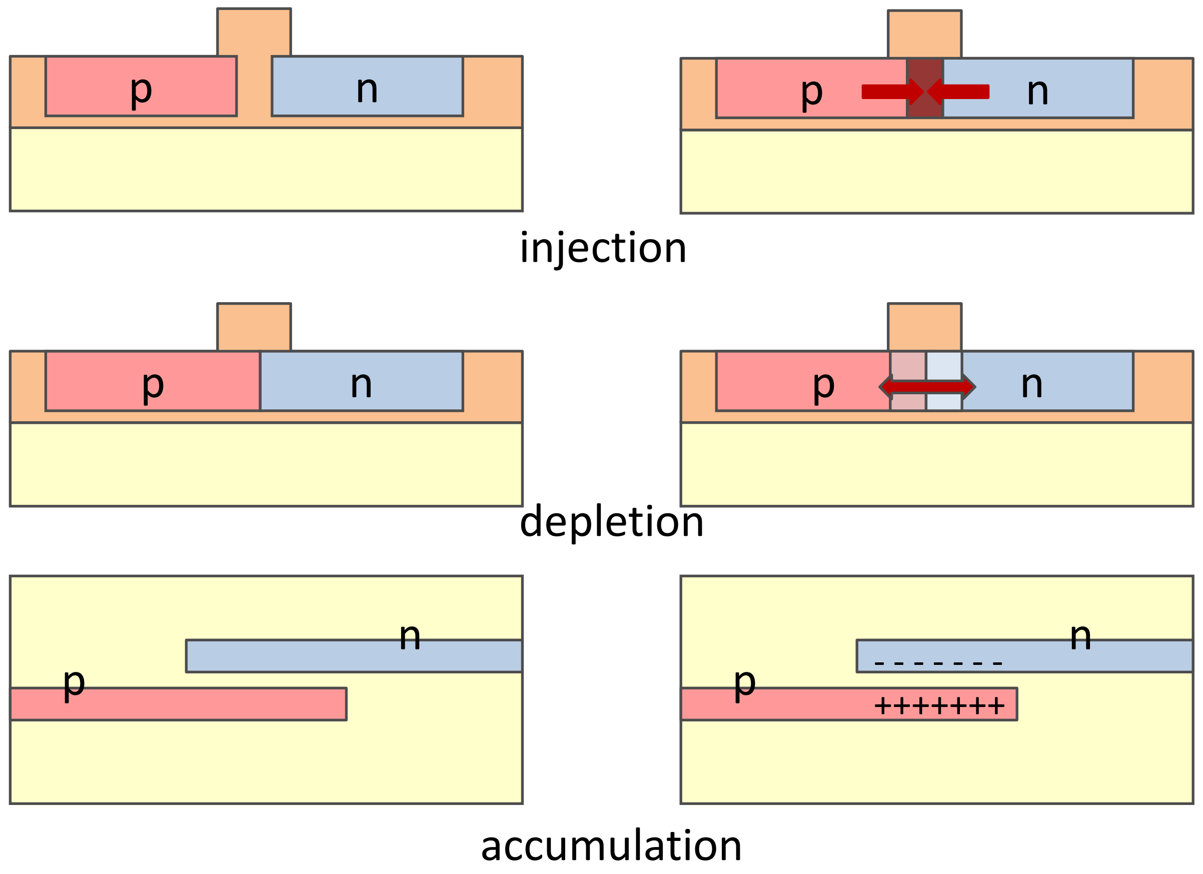


56Gb/s, 2.5Vpp

Vbias=-0.75V, ER=4dB, Q=4.2, PRBS=2e31-1



CARRIER-BASED MODULATION



Refractive-index of semiconductors depends on local carrier density

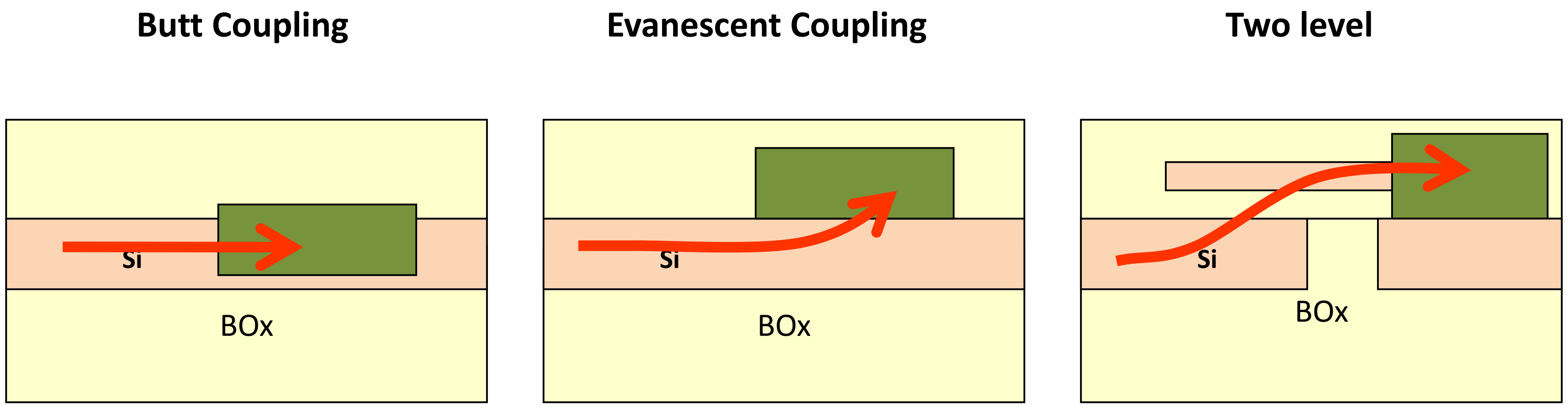
Modulate carrier density in waveguide

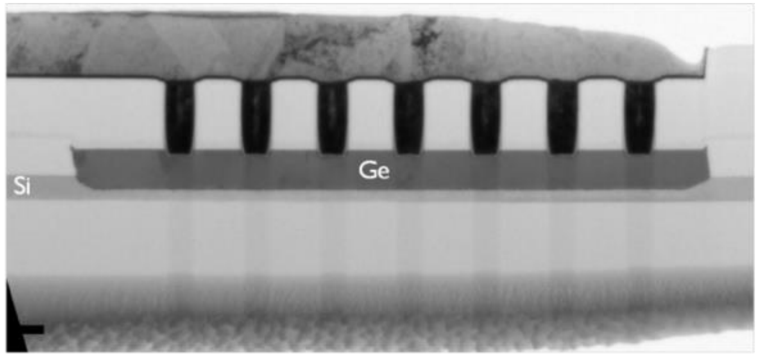
- phase modulation
- (spurious) amplitude modulation (free carrier absorption)

Modulation mechanisms

- carrier injection (in pin diode)
  speed limited by carrier recombination (~GHz)
- carrier depletion (in pn diode)
  speed limited by RC constant
- carrier accumulation (in capacitor)
  speed limited by RC constant

GE-DETECTORS COUPLING FROM SILICON WAVEGUIDES





Relevant parameters

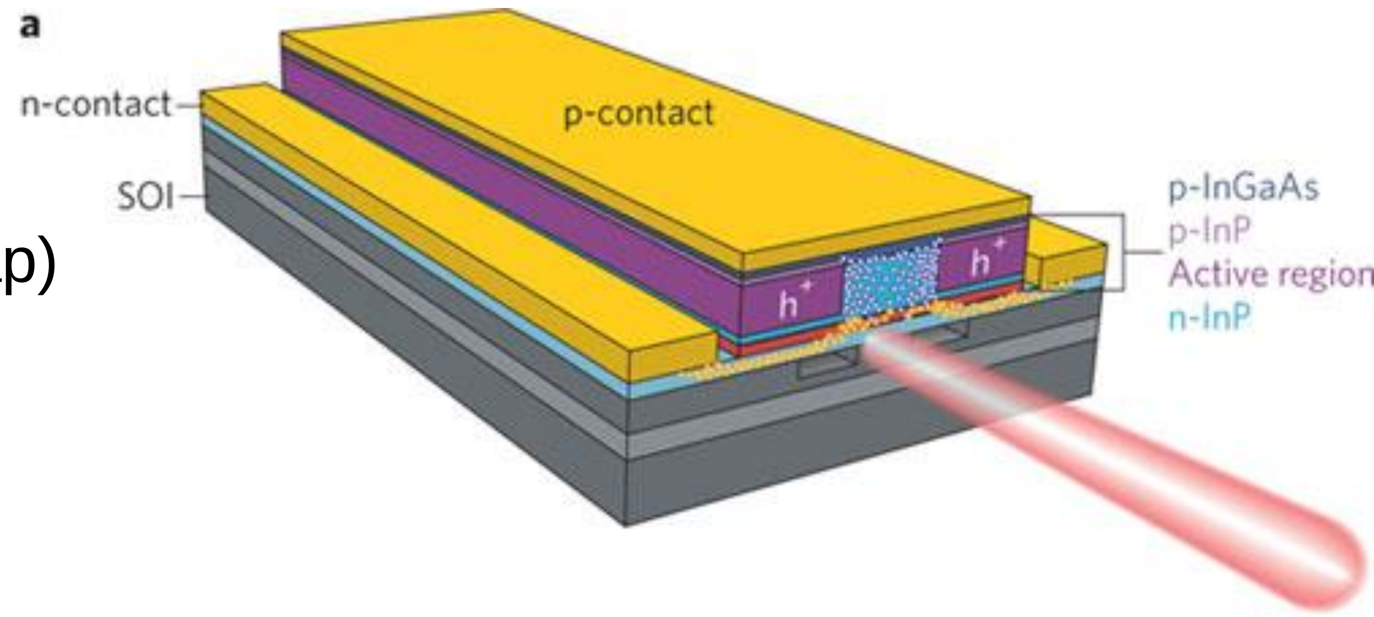
- Responsivity (A/W)
- Bandwidth (GHz)
- Dark current (nA)

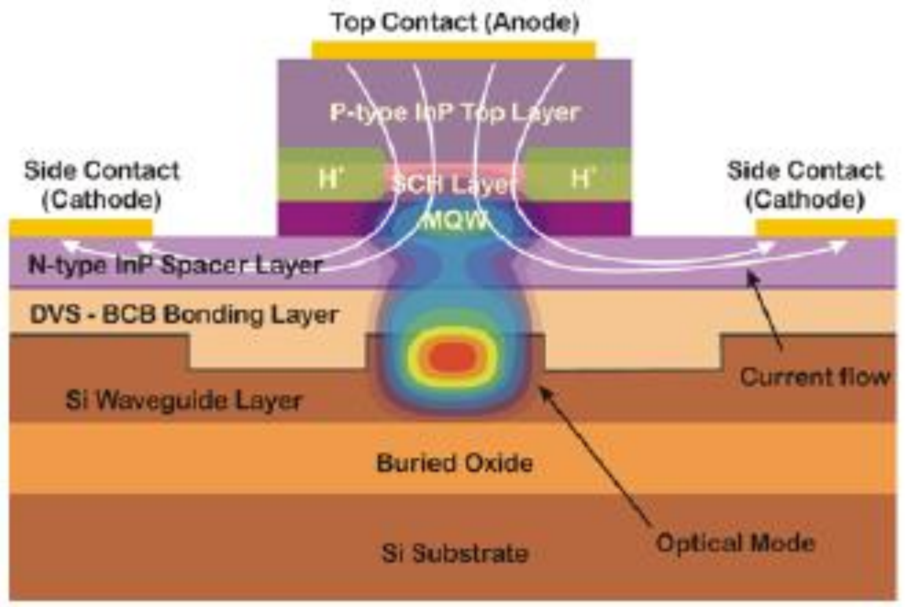
III-V LASERS ON SILICON

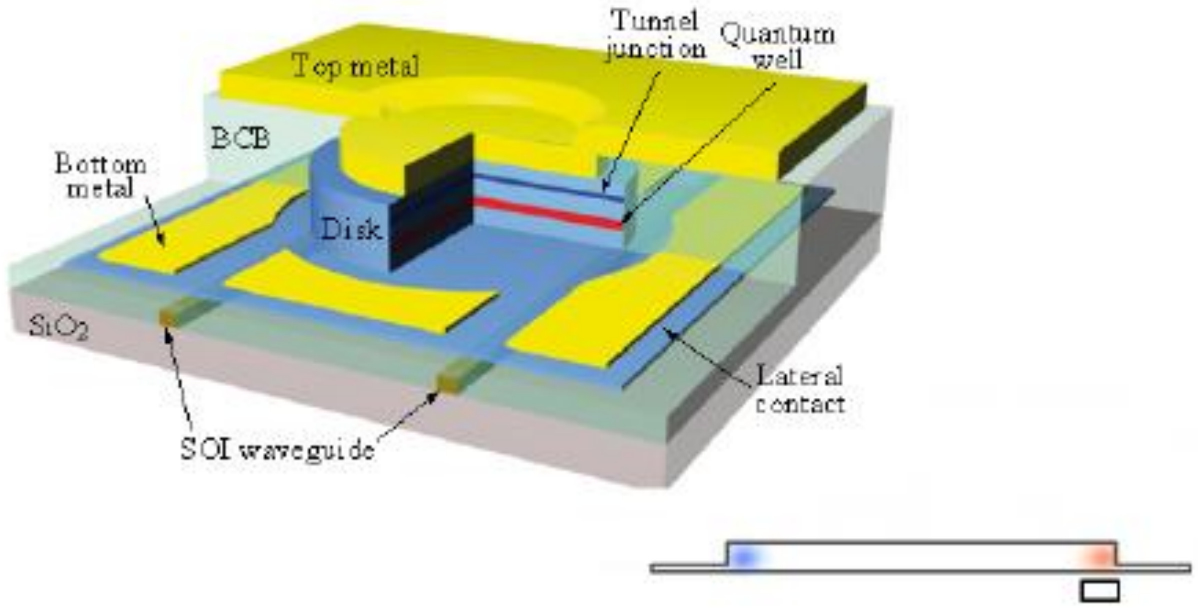
Silicon does not emit (indirect bandgap)

Bonding of III-V layer stack on silicon

Careful engineering of the transitions

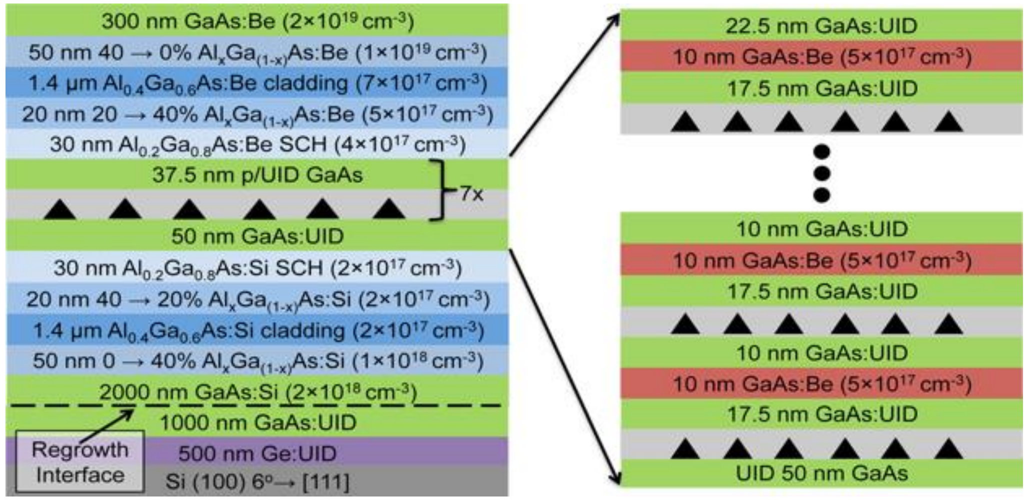


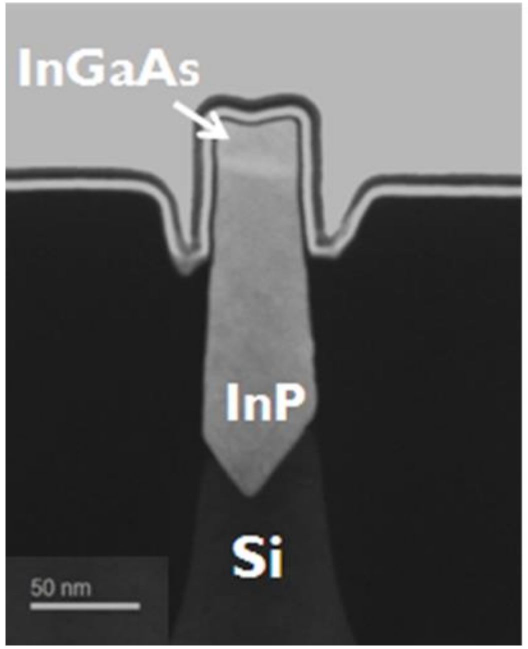




source: UCSB, UGent

III-V EPITAXY ON SILICON: DIFFICULT


300 nm GaAs:Be (2×10^{19} cm^{-3})
50 nm 40 → 0% Al$_x$Ga$_{(1-x)}$As:Be (1×10^{19} cm^{-3})
1.4 µm Al$_{0.4}$Ga$_{0.6}$As:Be cladding (7×10^{17} cm^{-3})
20 nm 20 → 40% Al$_x$Ga$_{(1-x)}$As:Be (5×10^{17} cm^{-3})
30 nm Al$_{0.2}$Ga$_{0.8}$As:Be SCH (4×10^{17} cm^{-3})
37.5 nm p/UID GaAs
7x
50 nm GaAs:UID
30 nm Al$_{0.2}$Ga$_{0.8}$As:Si SCH (2×10^{17} cm^{-3})
20 nm 40 → 20% Al$_x$Ga$_{(1-x)}$As:Si (2×10^{17} cm^{-3})
1.4 µm Al$_{0.4}$Ga$_{0.6}$As:Si cladding (2×10^{17} cm^{-3})
50 nm 0 → 40% Al$_x$Ga$_{(1-x)}$As:Si (1×10^{18} cm^{-3})
2000 nm GaAs:Si (2×10^{18} cm^{-3})
1000 nm GaAs:UID
Regrowth Interface
500 nm Ge:UID
Si (100) 6°→ [111]
22.5 nm GaAs:UID
10 nm GaAs:Be (5×10^{17} cm^{-3})
17.5 nm GaAs:UID
10 nm GaAs:UID
10 nm GaAs:Be (5×10^{17} cm^{-3})
17.5 nm GaAs:UID
10 nm GaAs:UID
10 nm GaAs:Be (5×10^{17} cm^{-3})
17.5 nm GaAs:UID
UID 50 nm GaAs



Challengin

- lattice constant mismatch
- polar vs. apolar material

Solutions:

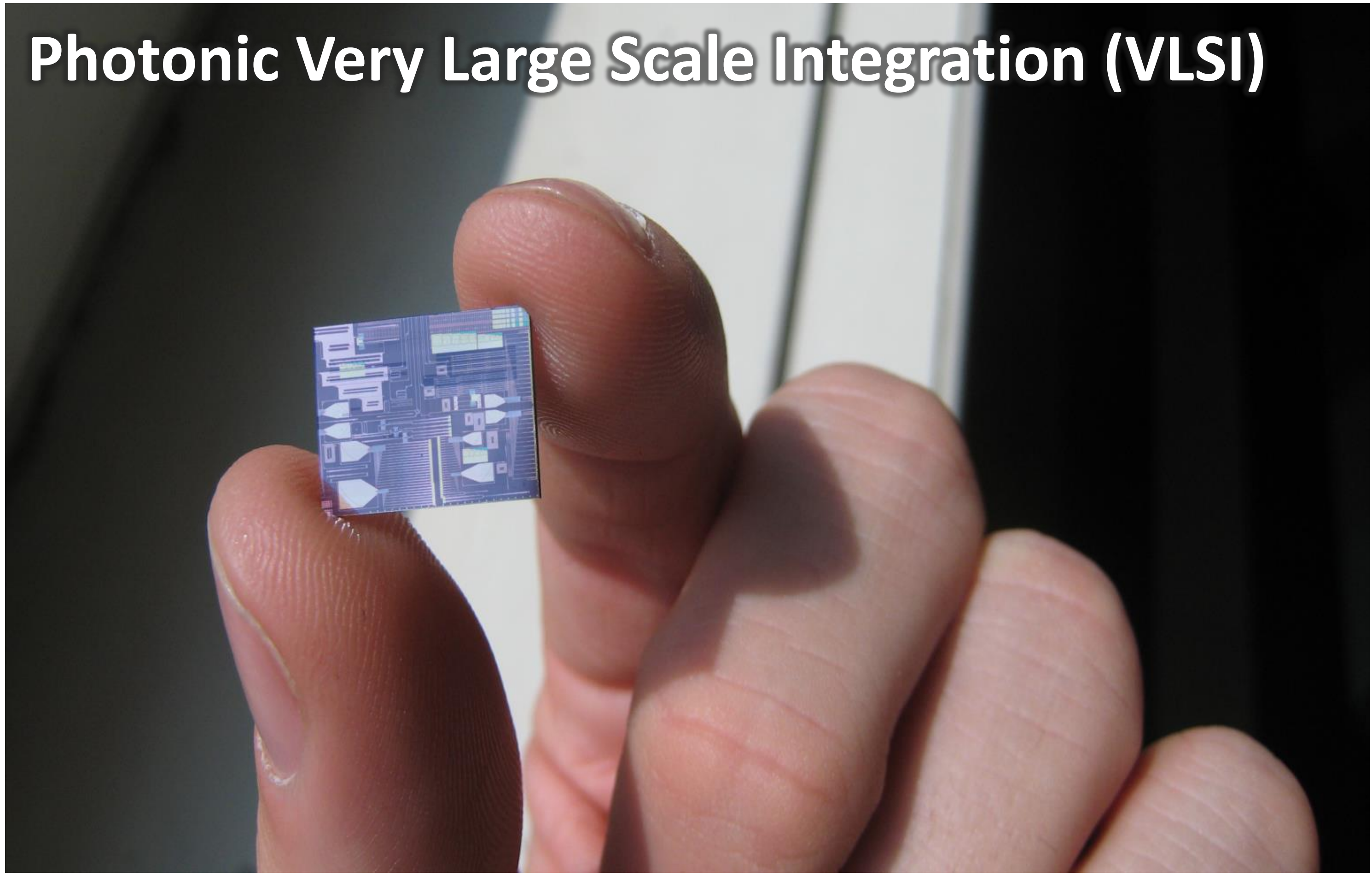
- Thick buffer layers
- high aspect ratio growth
- quantum dots

First lasing demonstrated (optically pumped)

GHENT UNIVERSITY imec

SMALL BUILDING BLOCKS → LARGE CIRCUITS

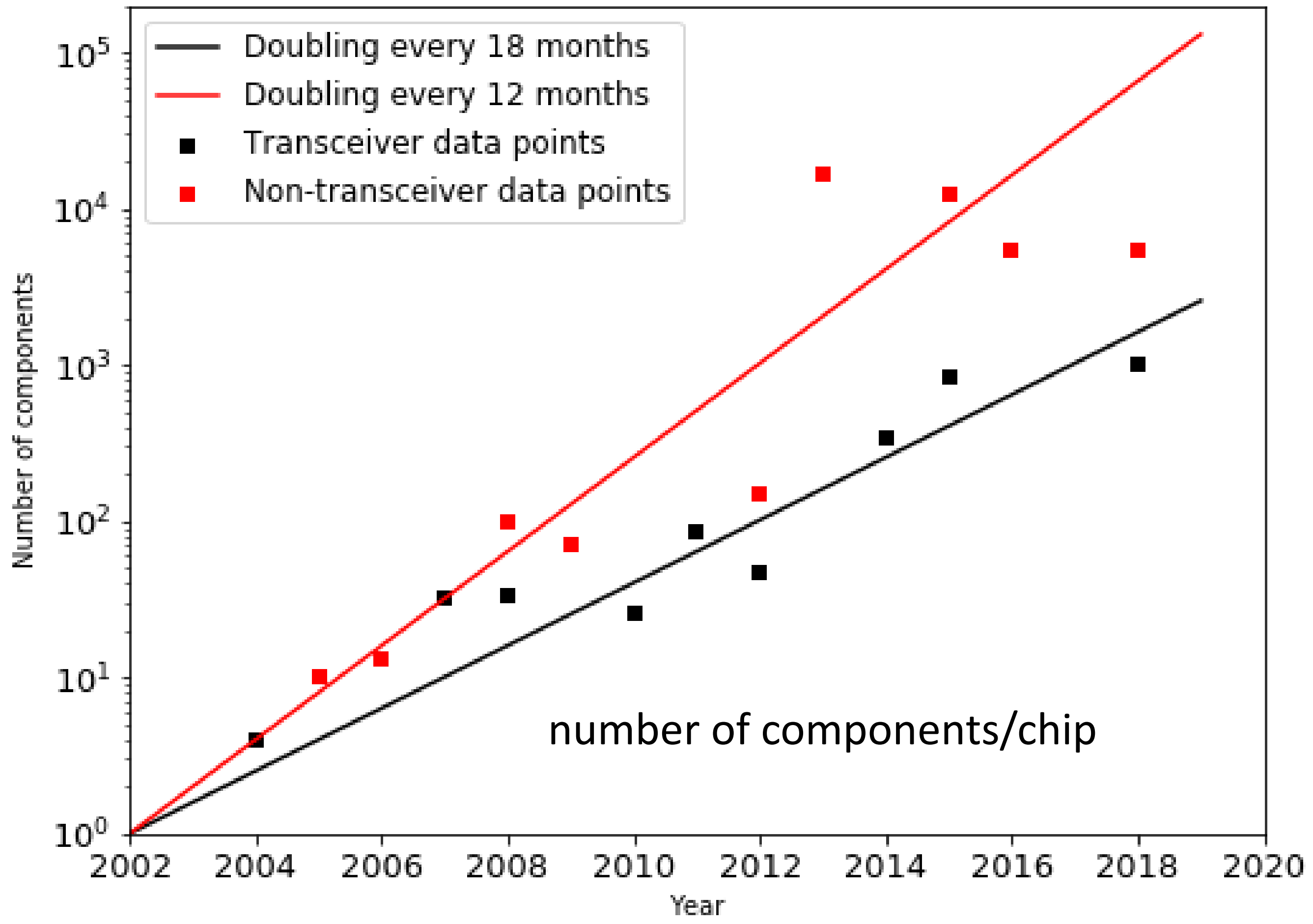
$$\left.\begin{array}{l}\text{µm-scale building blocks}\\ \\ \text{cm-scale chips}\end{array}\right\} \Rightarrow \text{thousands – millions components}$$



SILICON PHOTONIC CIRCUIT SCALING

Rapidly growing integration

- O(1000) components on a chip
- photonics + electronic drivers
- different applications (mostly comms)
- Relatively small chip volumes (compared to electronics)



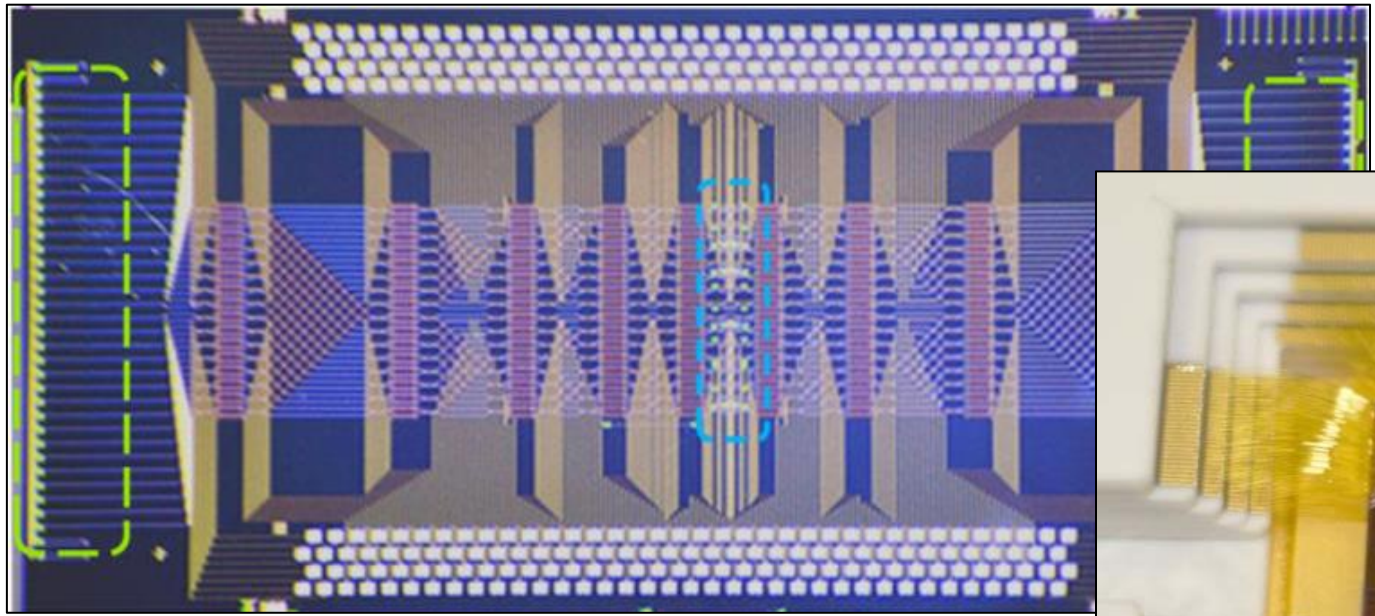
All photonic circuits are ASICs

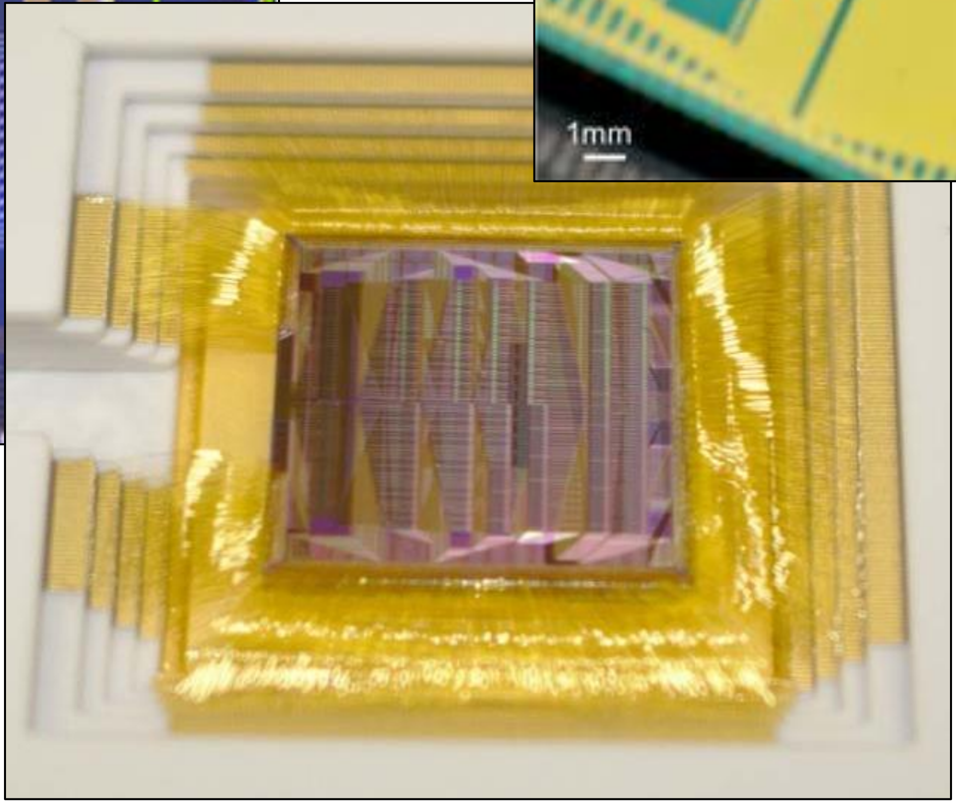
Khanna et al. 2016

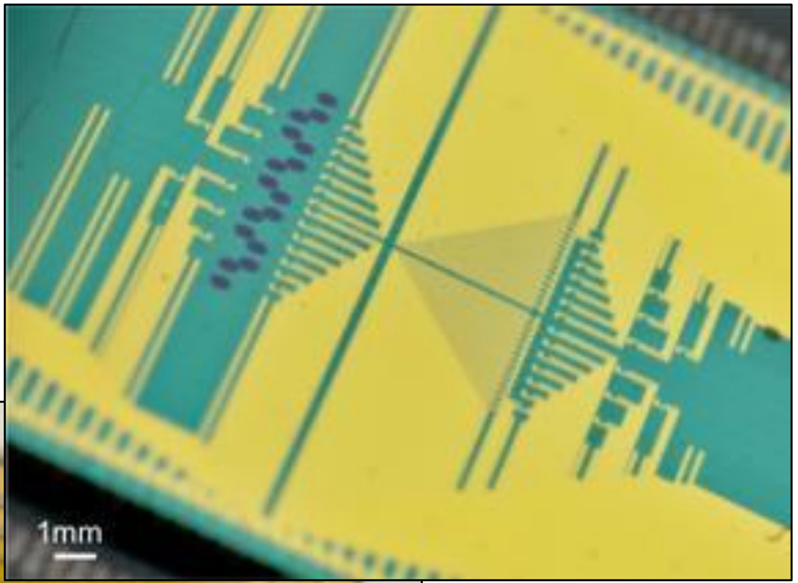
PHOTONIC LARGE-SCALE INTEGRATION IS HERE

That does not mean it is easy…

Larger circuits → lower fabrication yield?

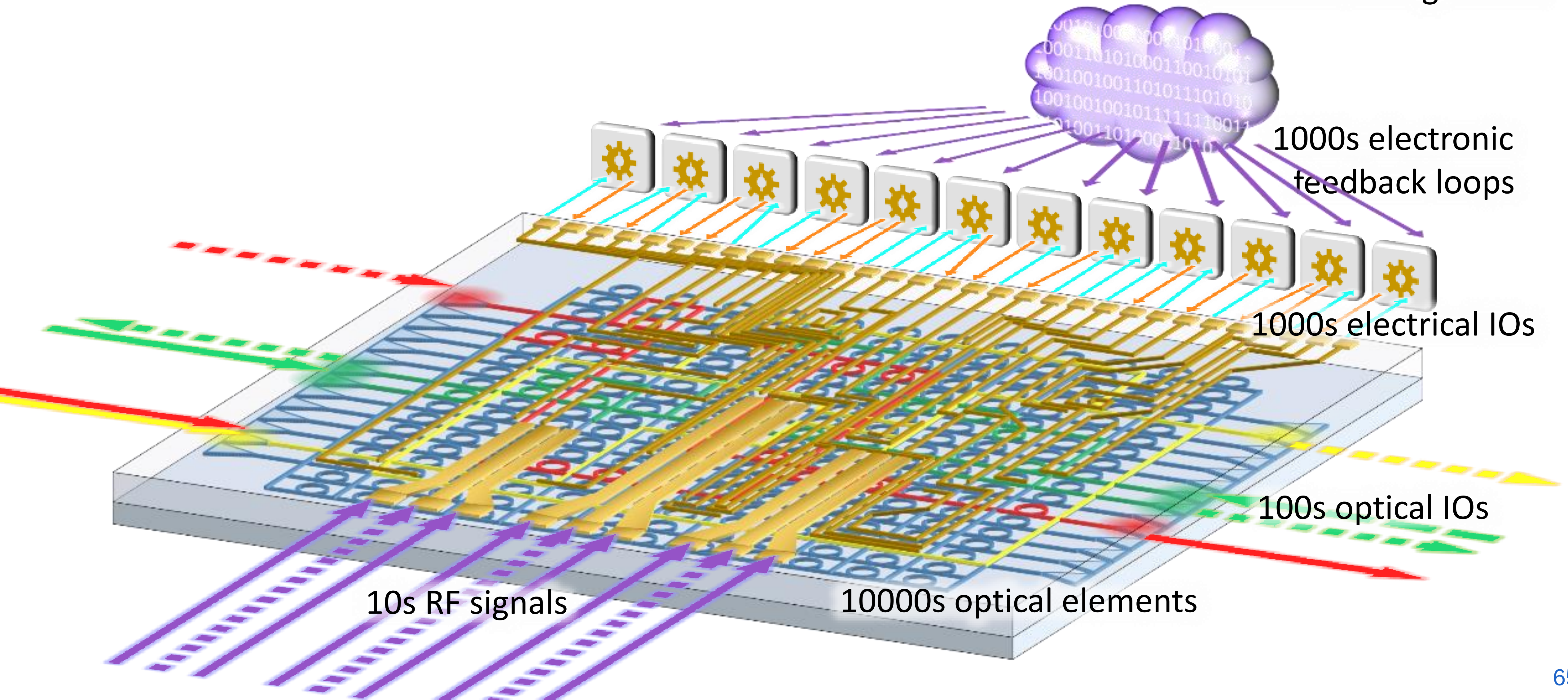




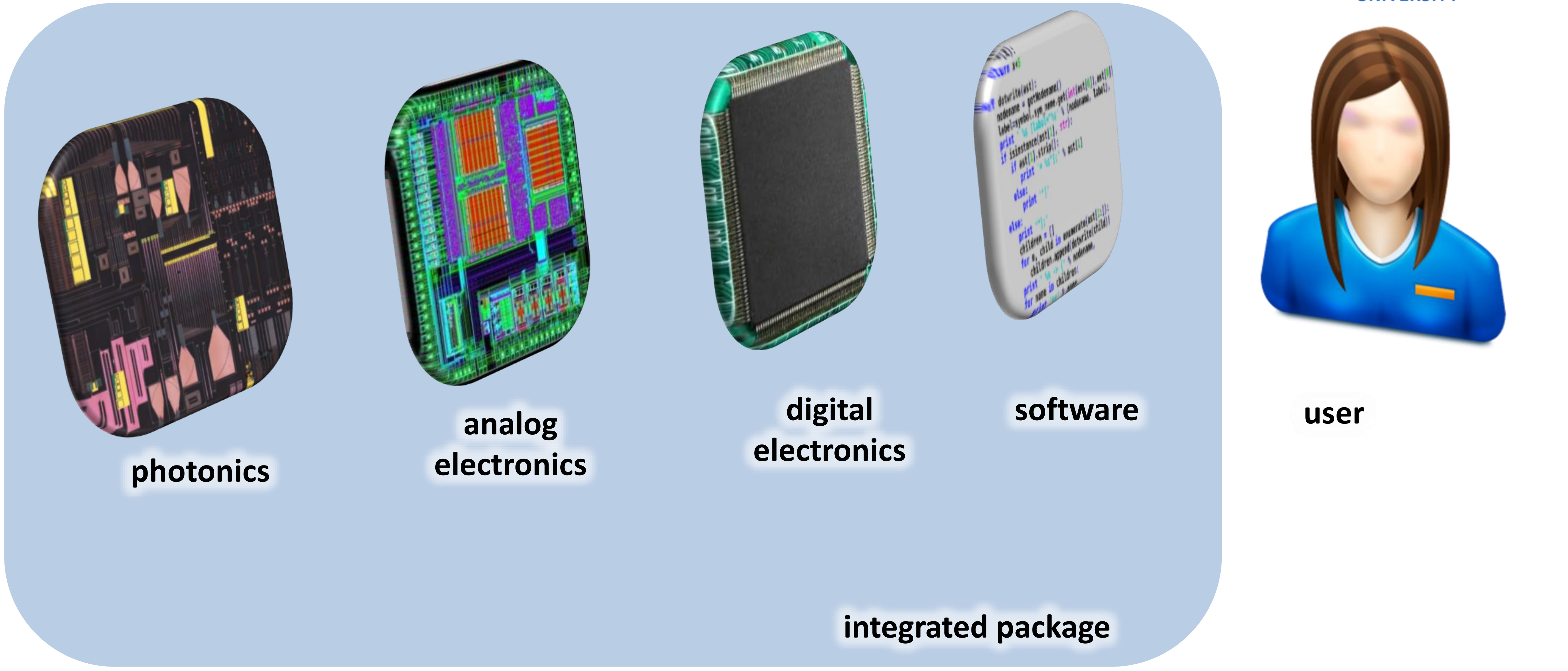


MORE THAN JUST PHOTONS

Silicon photonics goes beyond the optical chip

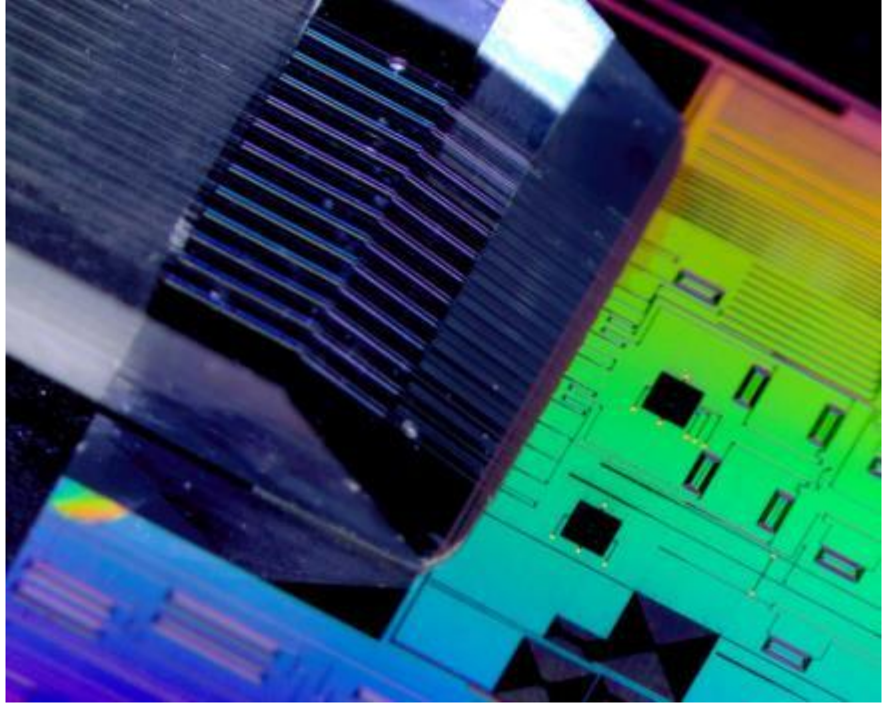


The Photonic Chip is Just a part of the system

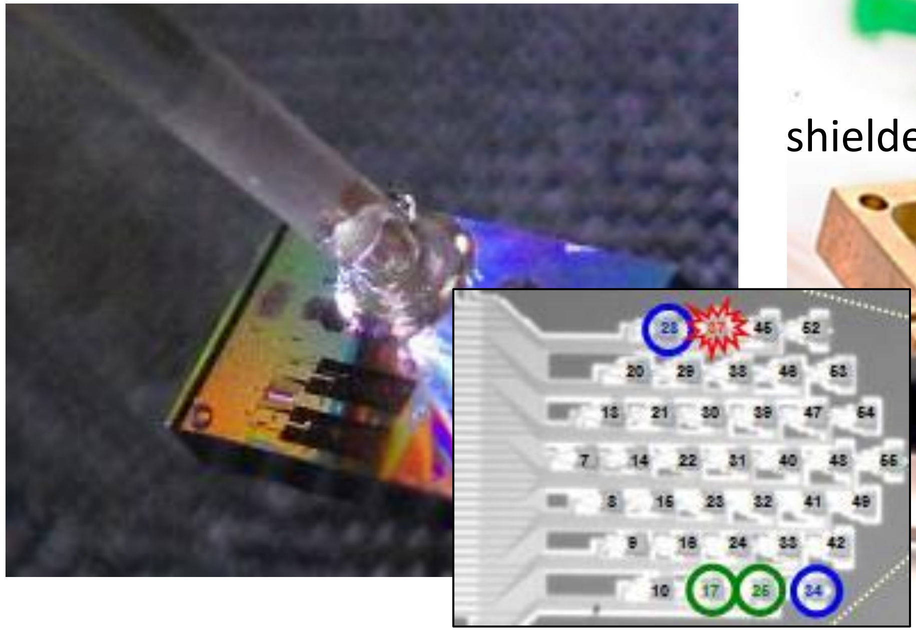


PACKAGING TECHNOLOGY

- Combining photonics and electronics
- Fiber interfaces
- RF connections
- Thermal and mechanical

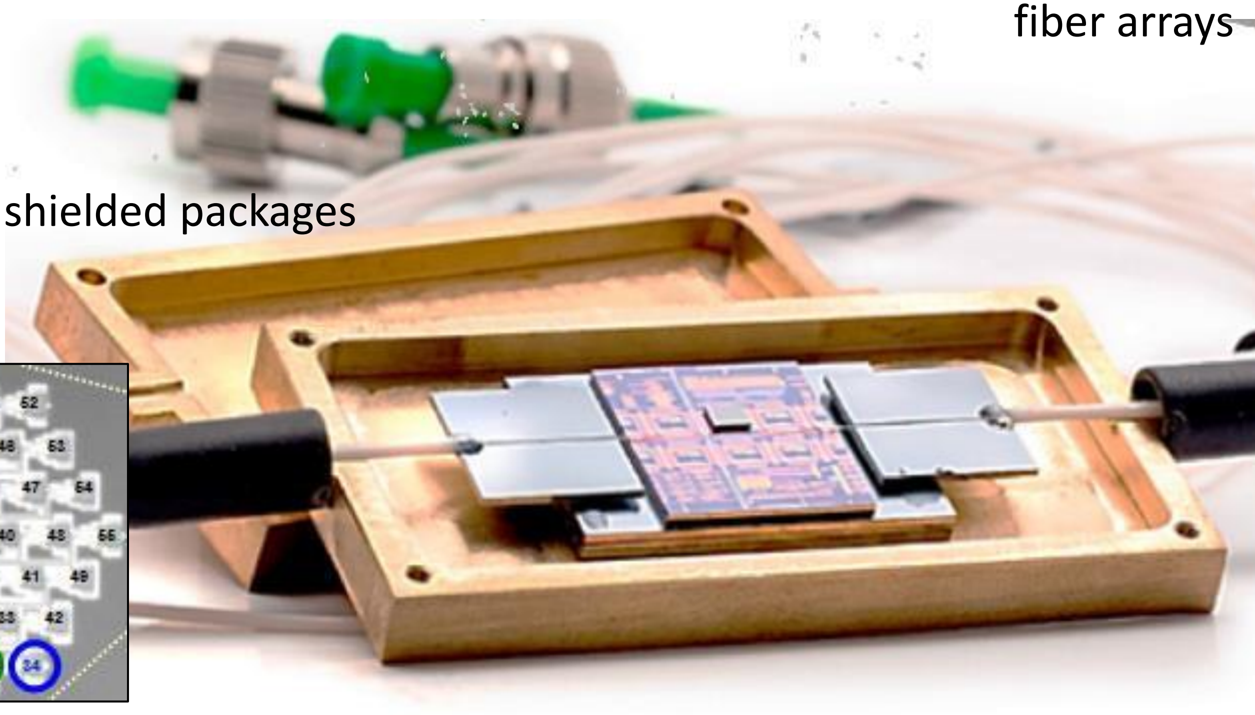


multi-core fibers



shielded packages

fiber arrays



FABLESS SILICON PHOTONICS

Many fabless Silicon Photonics companies have emerged

- from direct collaboration with fabs (Luxtera, ...)
- starting from MPW (Caliopa, Genalyte, Acacia)

Established players are also partnering

- e.g. Finisar with ST
- Many keep their fab a secret

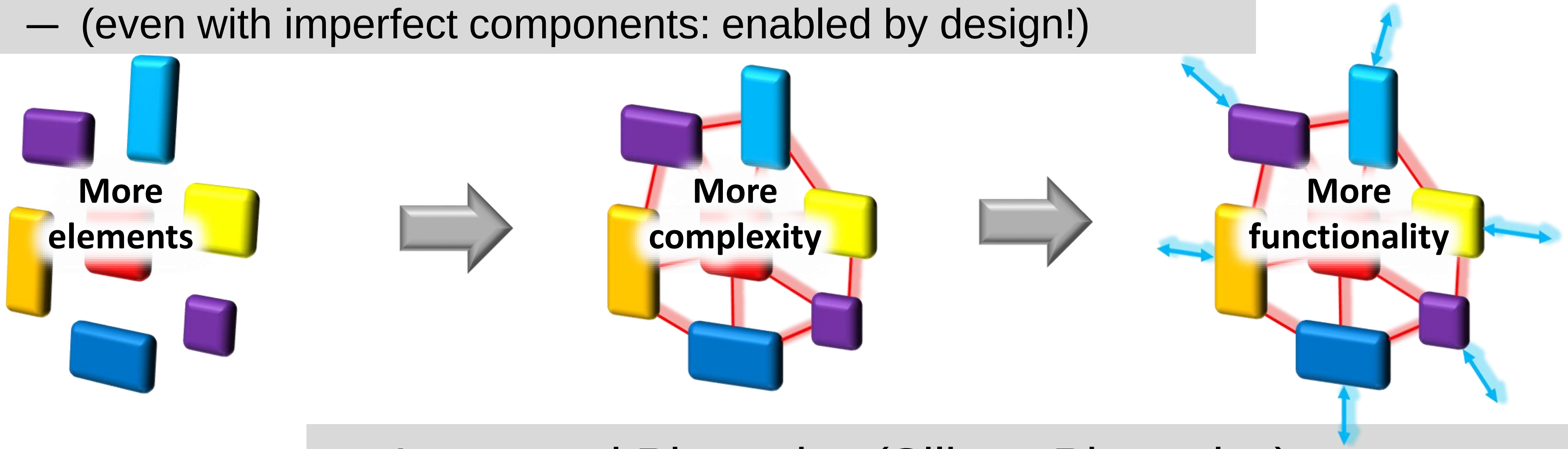
How to enter as a new (fabless) startup?

COMPLEXITY AS AN ENABLER

Integrated Electronics

- billions of digital gates: unprecedented logic performance
- millions of analog transistors: unprecedented control

— (even with imperfect components: enabled by design!)



Integrated Photonics (Silicon Photonics)

- **technological potential** of 10000+ photonic elements on a chip
- not even scratched the surface of what this could do

GHENT UNIVERSITY imec

PHOTONIC CIRCUIT DESIGN

ENABLING COMPLEXITY IN PHOTONICS

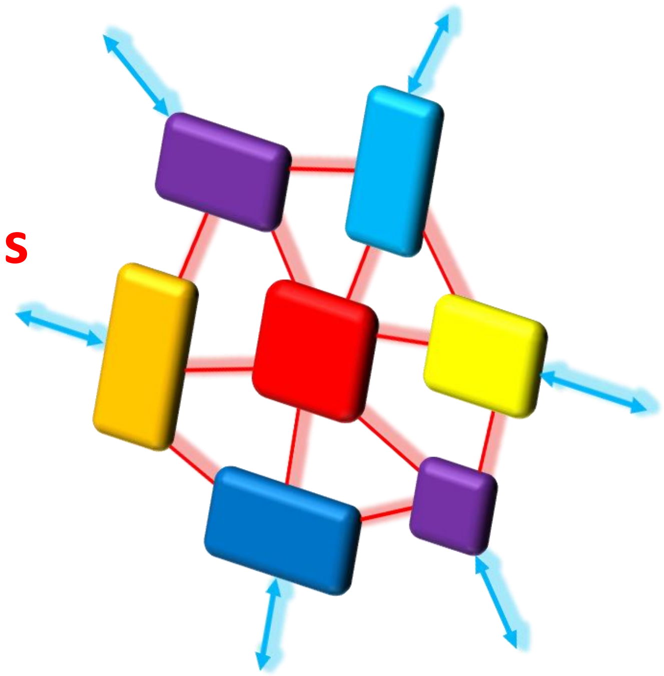
Industrial PIC technology platforms (Si, InP, …)

- demonstrations of sensors, spectrometers, …
- commercial products

But: fairly simple circuits ~ 1970s ICs

More complexity is enabled by design methods

- Design capture: translating ideas to circuits
- Circuit simulation (electrical+photonic)
- Variability analysis on circuits
- Yield prediction and improvement



COMPLEX CIRCUITS ≠ COMPLICATED BUILDING BLOCKS

You can do a lot with a few building blocks

Electronics: Transistors, Resistors, Diodes, …

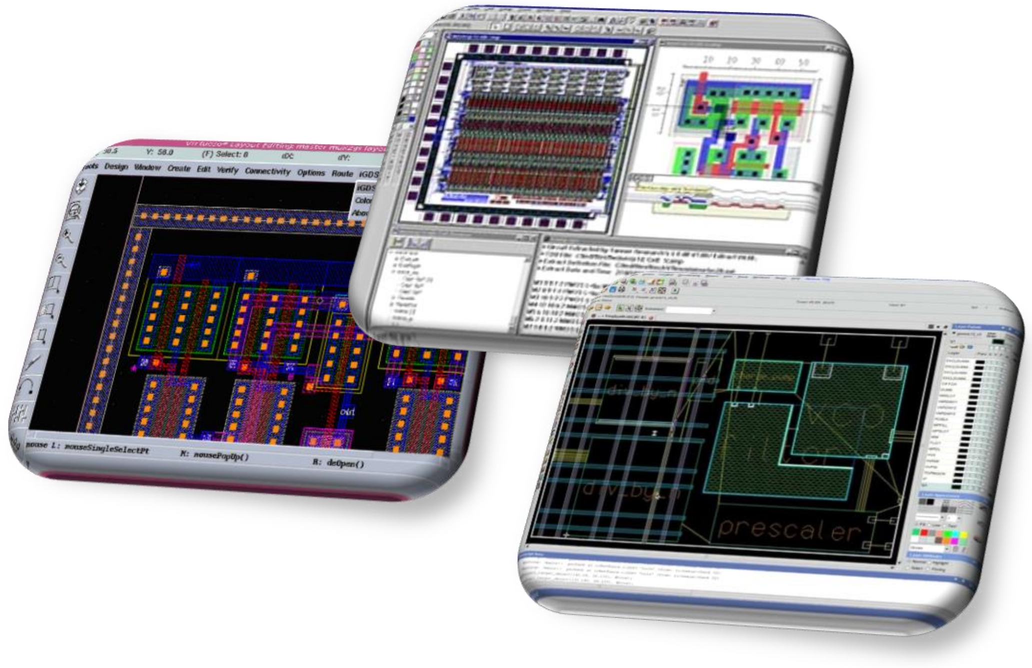
Photonics: Waveguides, Directional couplers, …

Complexity emerges from connectivity

But you need to support complexity

- Accurate models
- Variability
- Parasitics

DESIGNING PHOTONIC INTEGRATED CIRCUITS



Can we learn from electronic ICs?

- Millions of analog transistors
- Billions of digital transistors
- Power, timing and yield
- **First time right designs**

- Very mature Electronic Design Automation (EDA) tools!
- A well established design flow

Can we repurpose this for photonics?

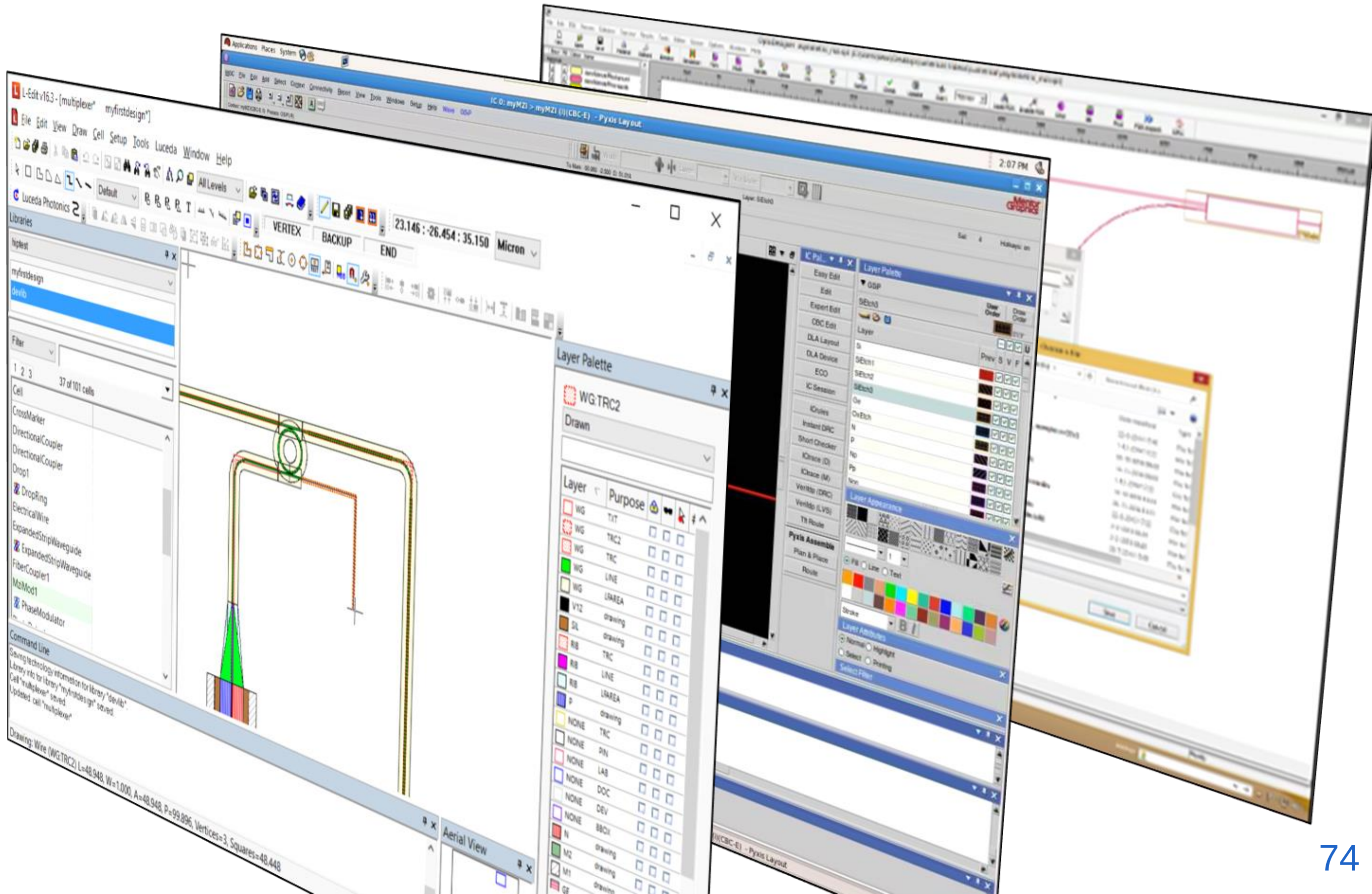
Design Environments are Emerging

Combinations of Photonics Design and EDA

Physical simulation combined with circuit design

Physical and functional verification

First PDKs with basic models



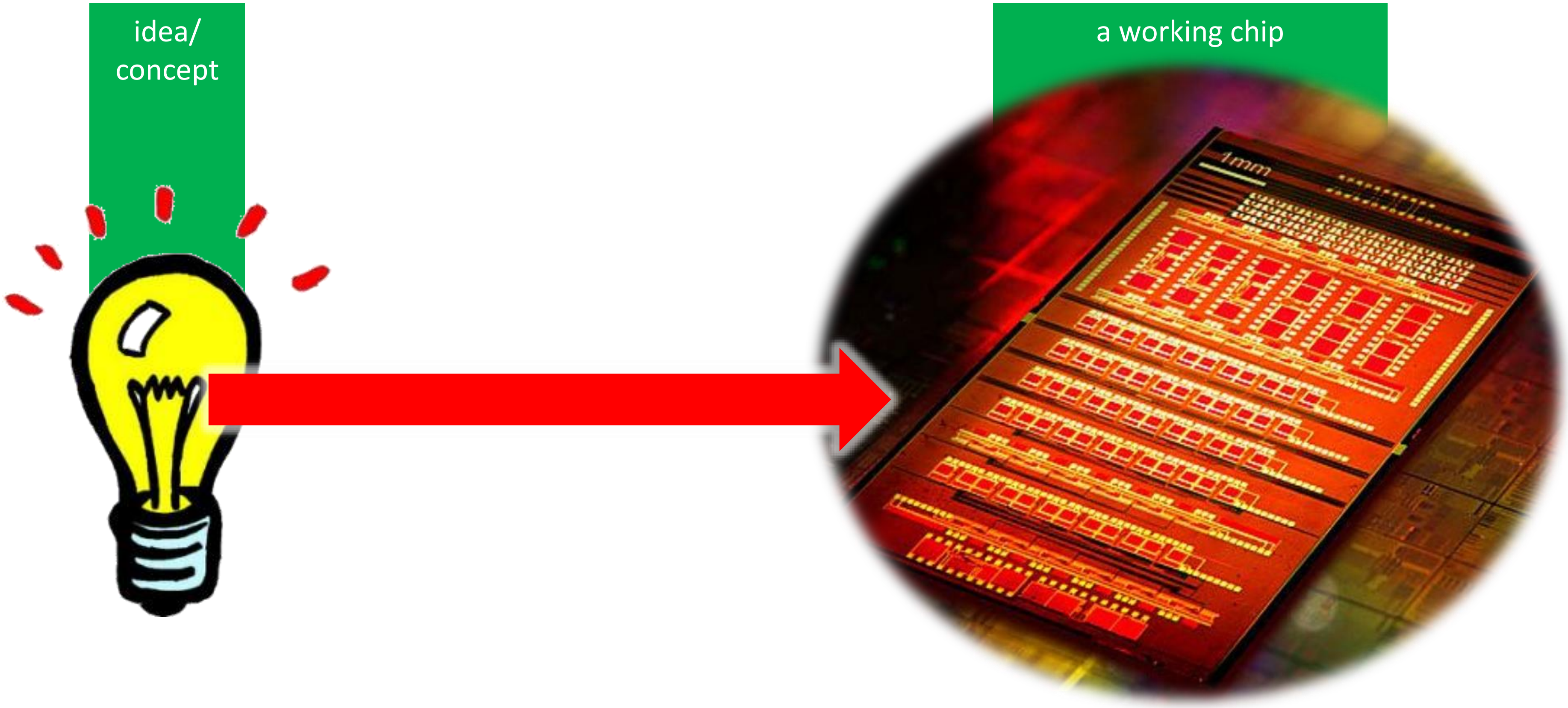
WHAT IS A DESIGN FLOW?

“Design is the creation of a plan or convention for the construction of an object or a system”

Design Flow

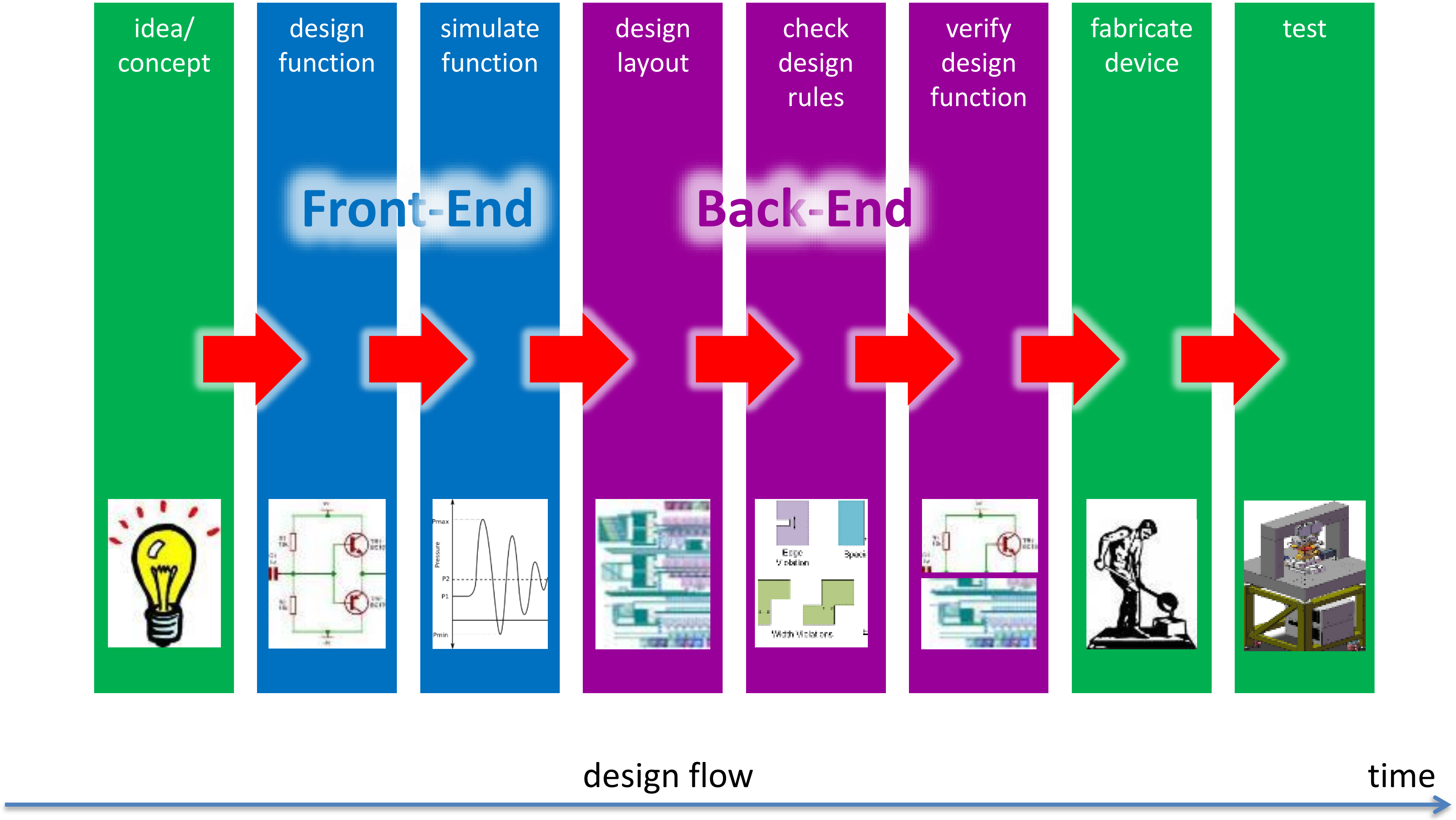
“a repeatable pattern of activity, usually involving multiple tasks with a specific set of outcomes”

WHAT IS THE PURPOSE OF A DESIGN FLOW?



to translate an idea into a **WORKING** chip.

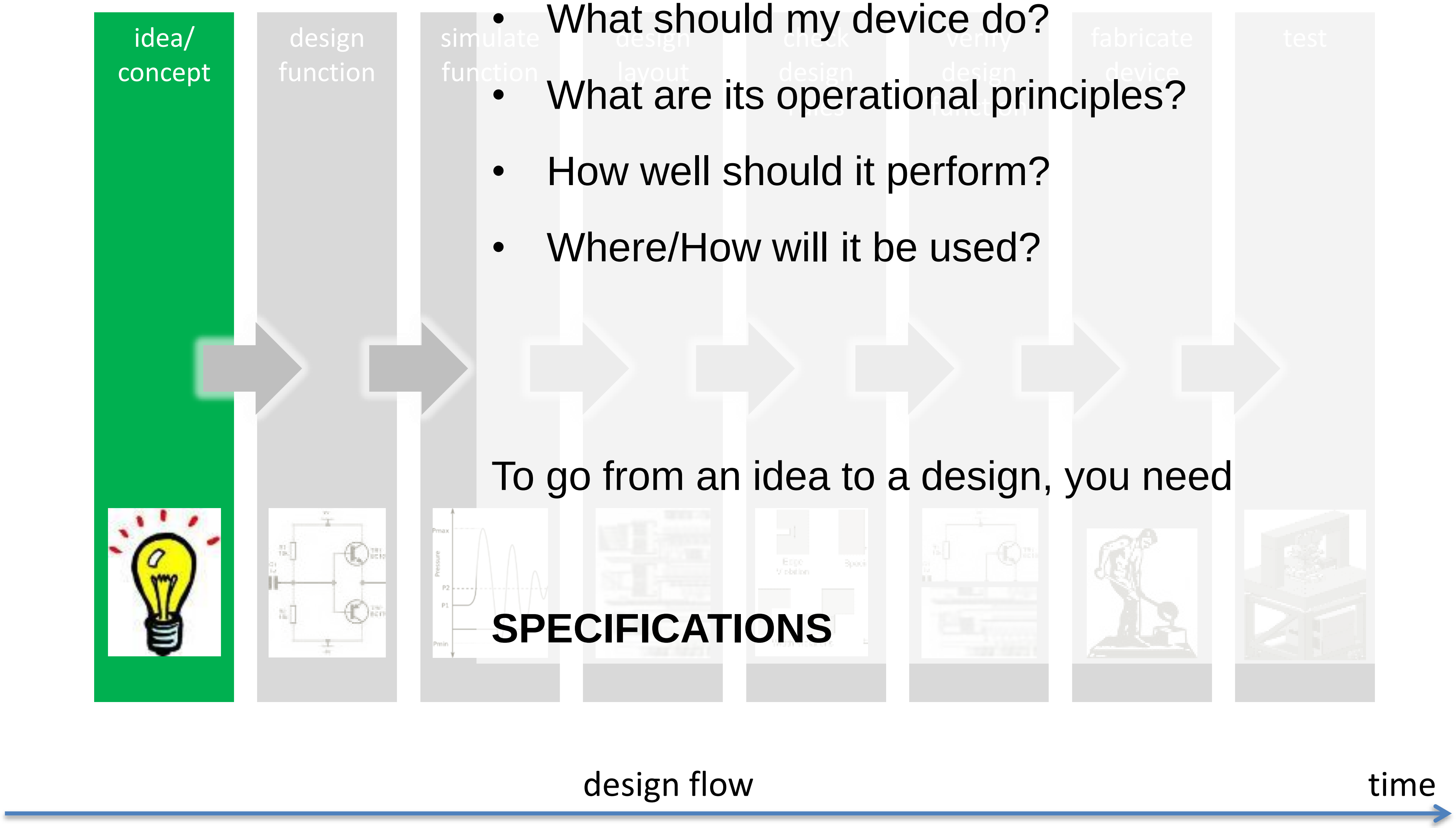
A TYPICAL DESIGN CYCLE



A GREAT IDEA?

Questions to be asked

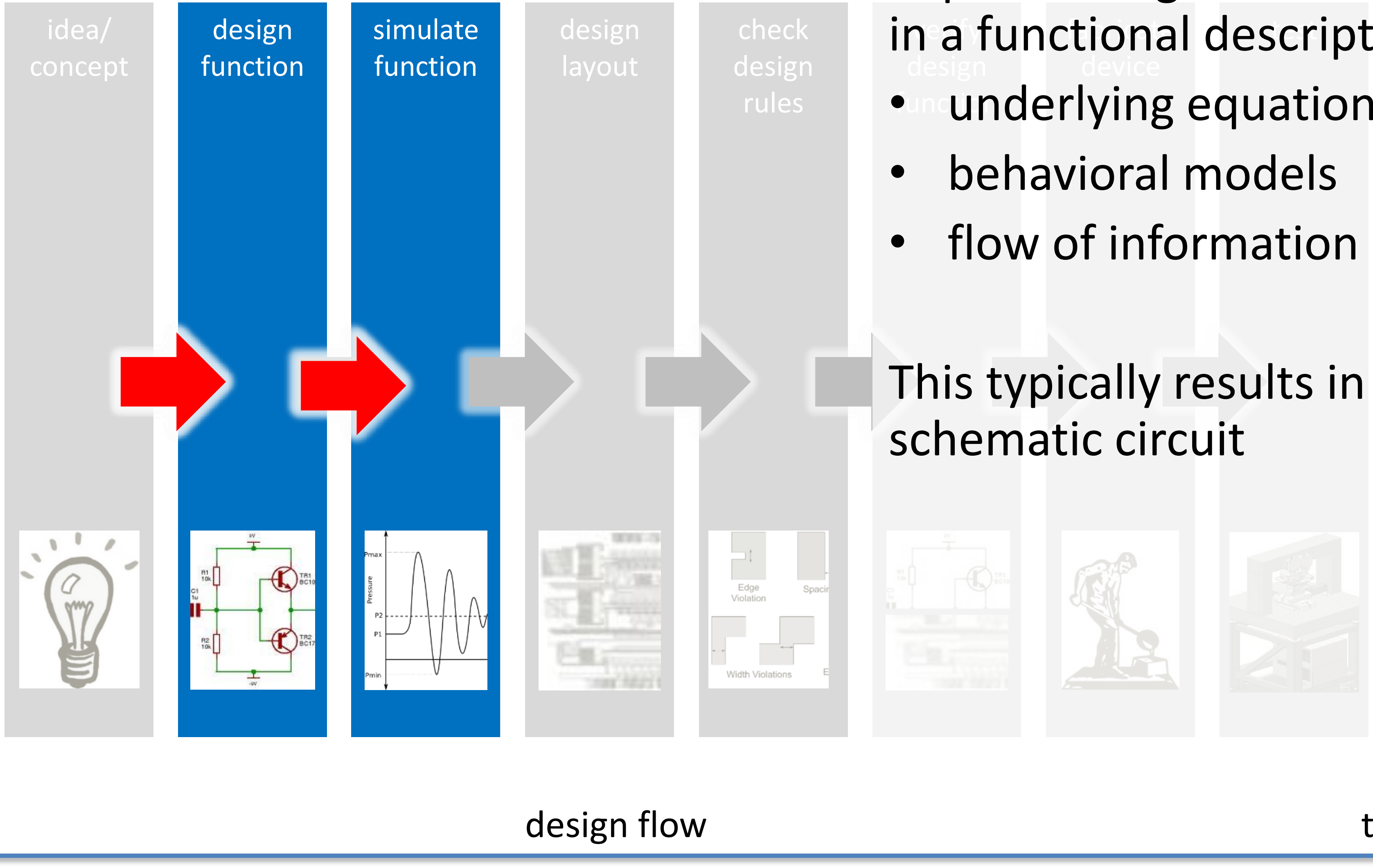
- What should my device do?
- What are its operational principles?
- How well should it perform?
- Where/How will it be used?



To go from an idea to a design, you need

SPECIFICATIONS

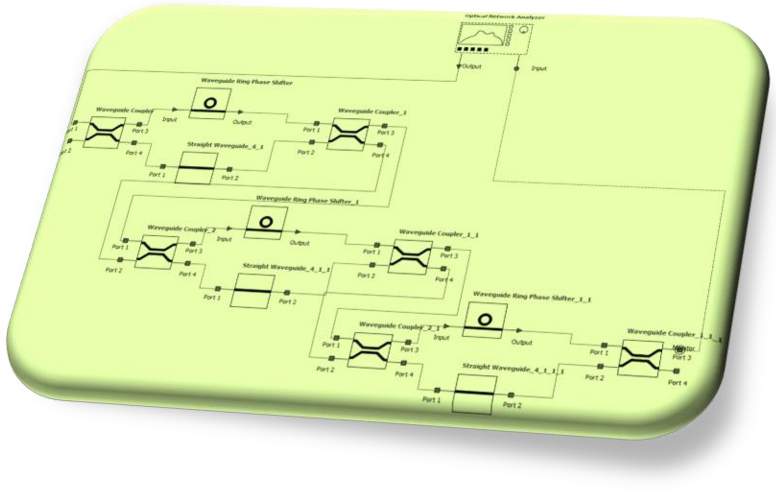
DESIGN CAPTURE AND SIMULATION



Capture design intent in a functional description

- underlying equations
- behavioral models
- flow of information

This typically results in a schematic circuit

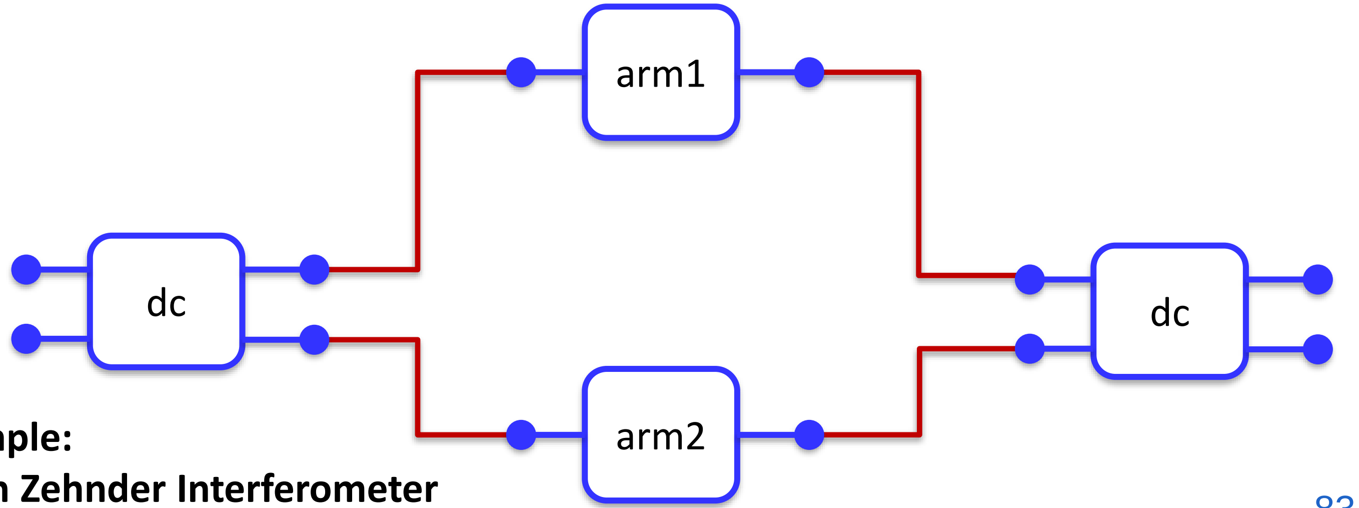


Design capture

Select/construct functional blocks

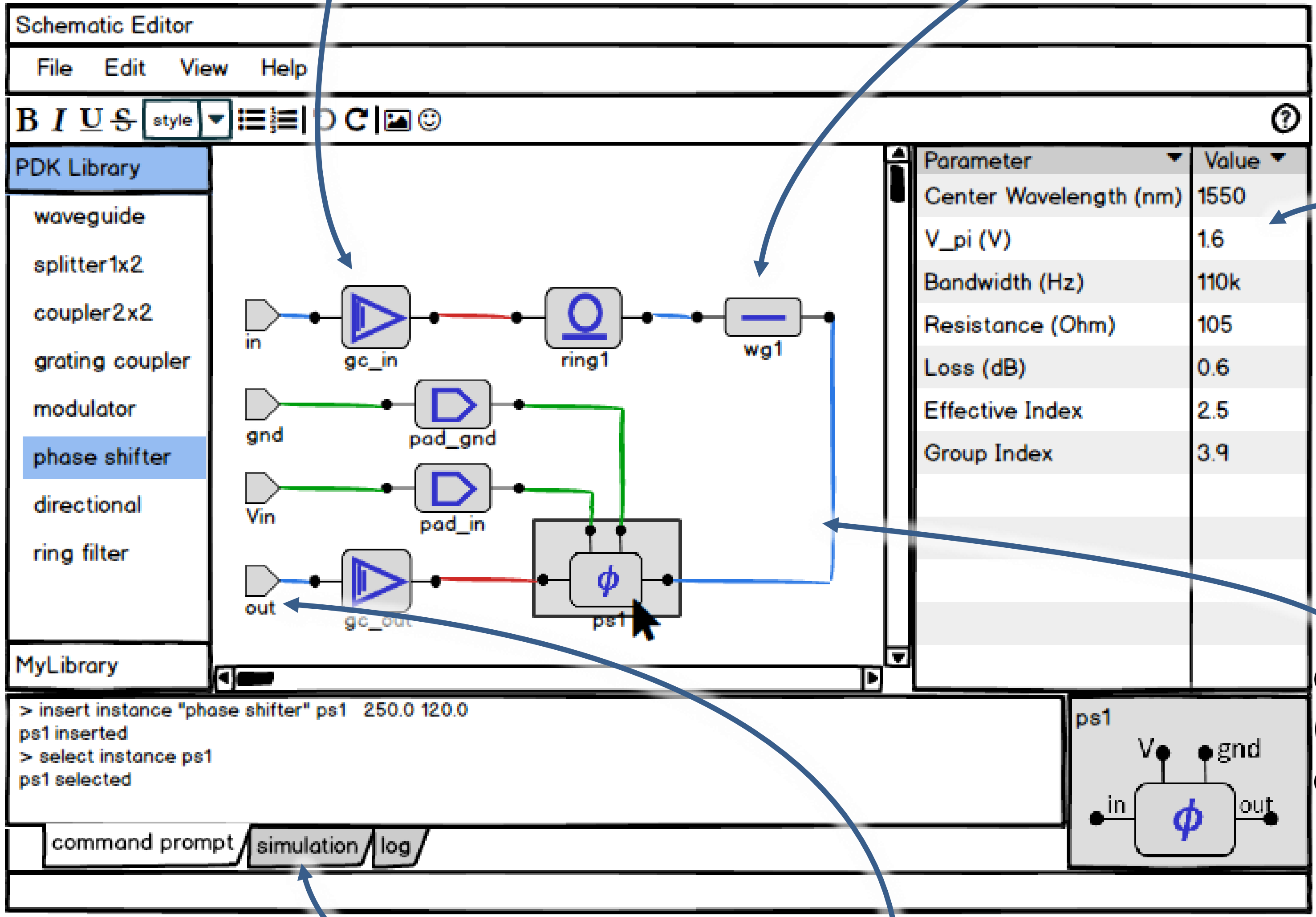
Connect them together

- **Netlist**:
  list of connections (“Nets”) and which components the nets are attached to.
- **Schematic**:
  graphical representation of a netlist, with placements



Example:
Mach Zehnder Interferometer

Schematic Editor



Parameter	Value
Center Wavelength (nm)	1550
V_pi (V)	1.6
Bandwidth (Hz)	110k
Resistance (Ohm)	105
Loss (dB)	0.6
Effective Index	2.5
Group Index	3.9

```
> insert instance "phase shifter" ps1   250.0 120.0
ps1 inserted
> select instance ps1
ps1 selected
```

drag and dropping components and drawing connections

make waveguides explicit if needed

component libraries

parametrization

scriptability

different connections (waveguides, direct optical, electrical)

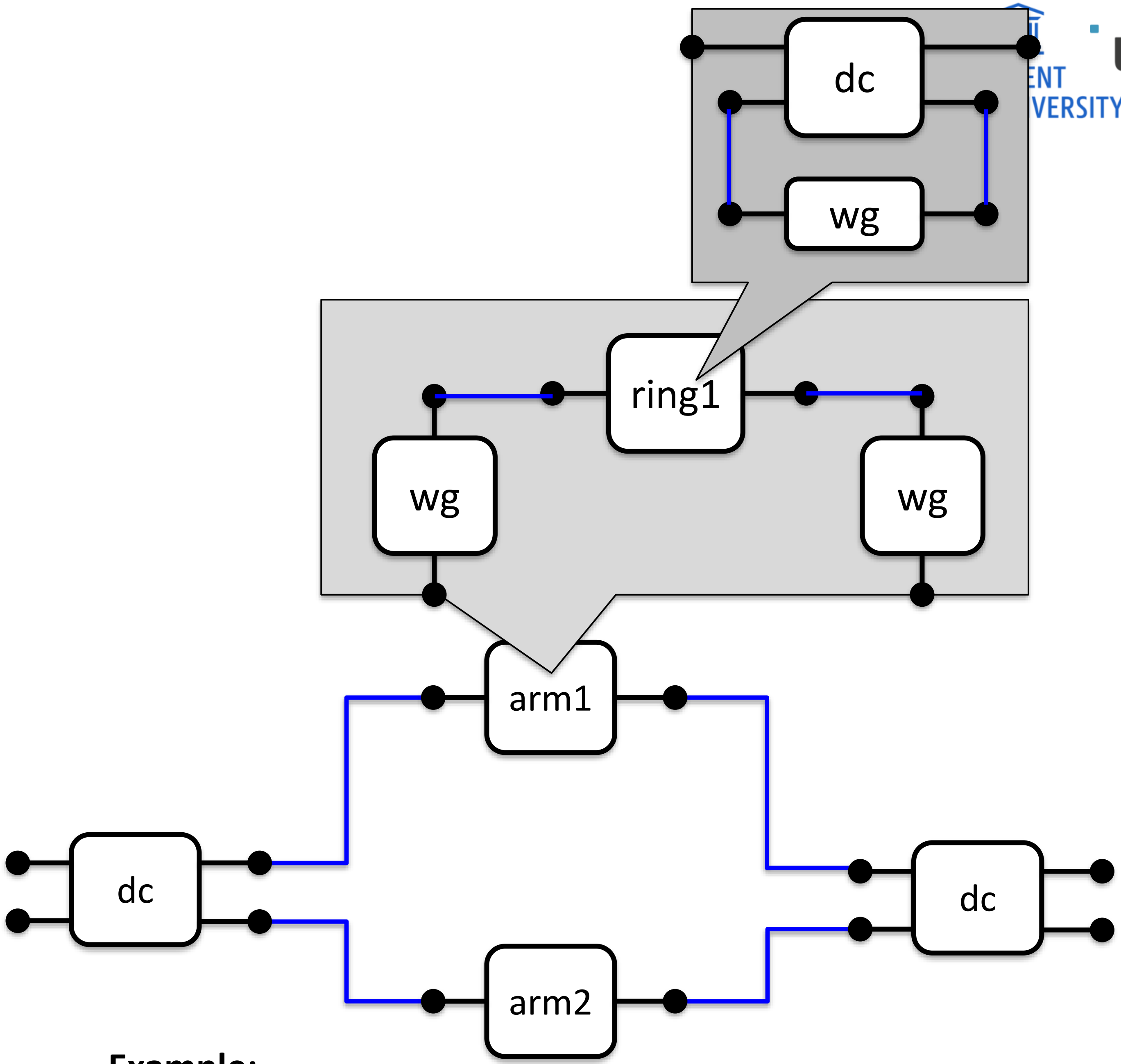
interface to circuit simulation

specify I/O ports

Hierarchy

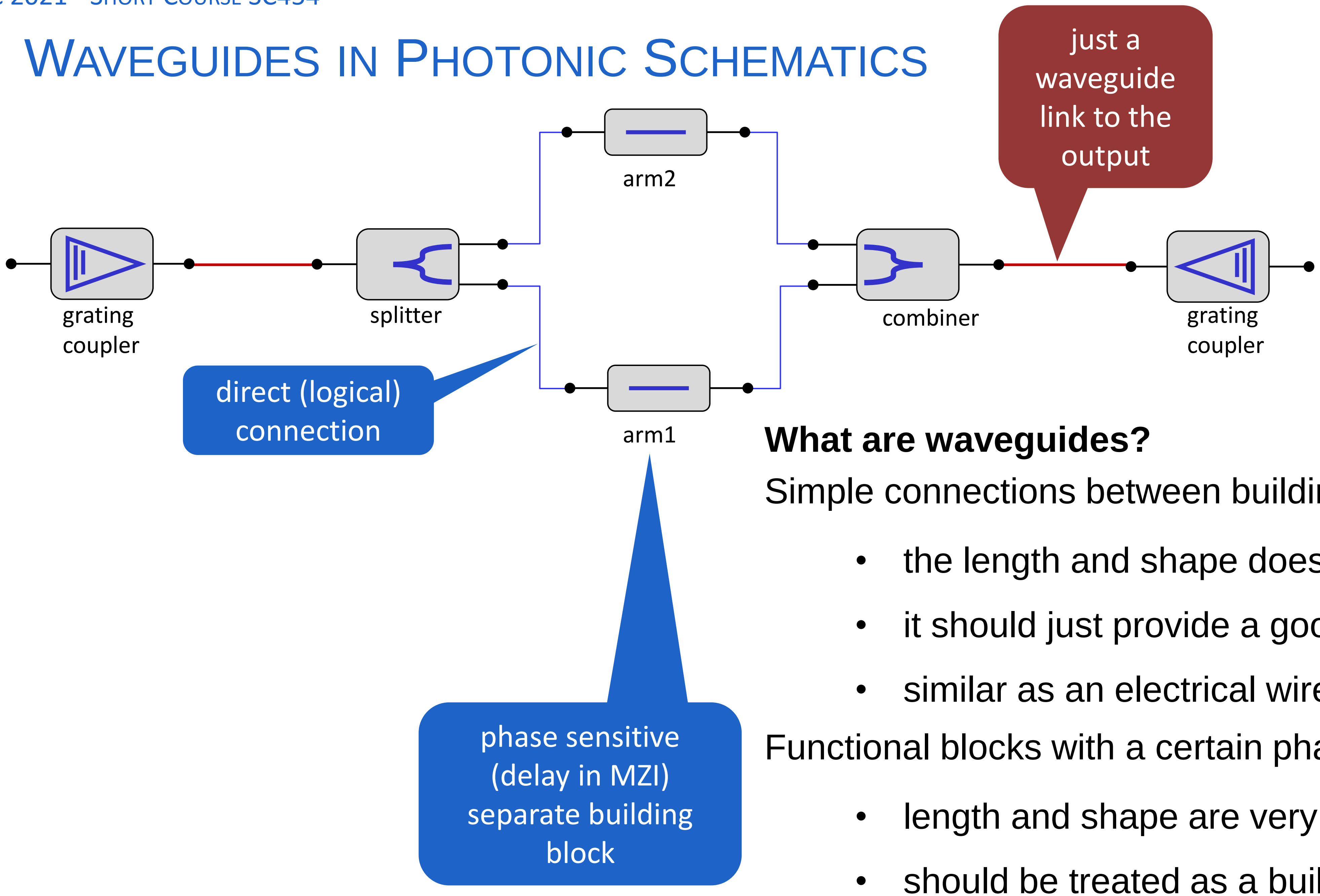
Netlists are hierarchical

- Hierarchical cells:
  contain another netlist
- Atomic cells:
  contain a circuit model



Example:
Ring-Loaded Mach Zehnder Interferometer

WAVEGUIDES IN PHOTONIC SCHEMATICS


just a waveguide link to the output
arm2
grating coupler
splitter
combiner
grating coupler
direct (logical) connection
arm1
phase sensitive (delay in MZI) separate building block

What are waveguides?

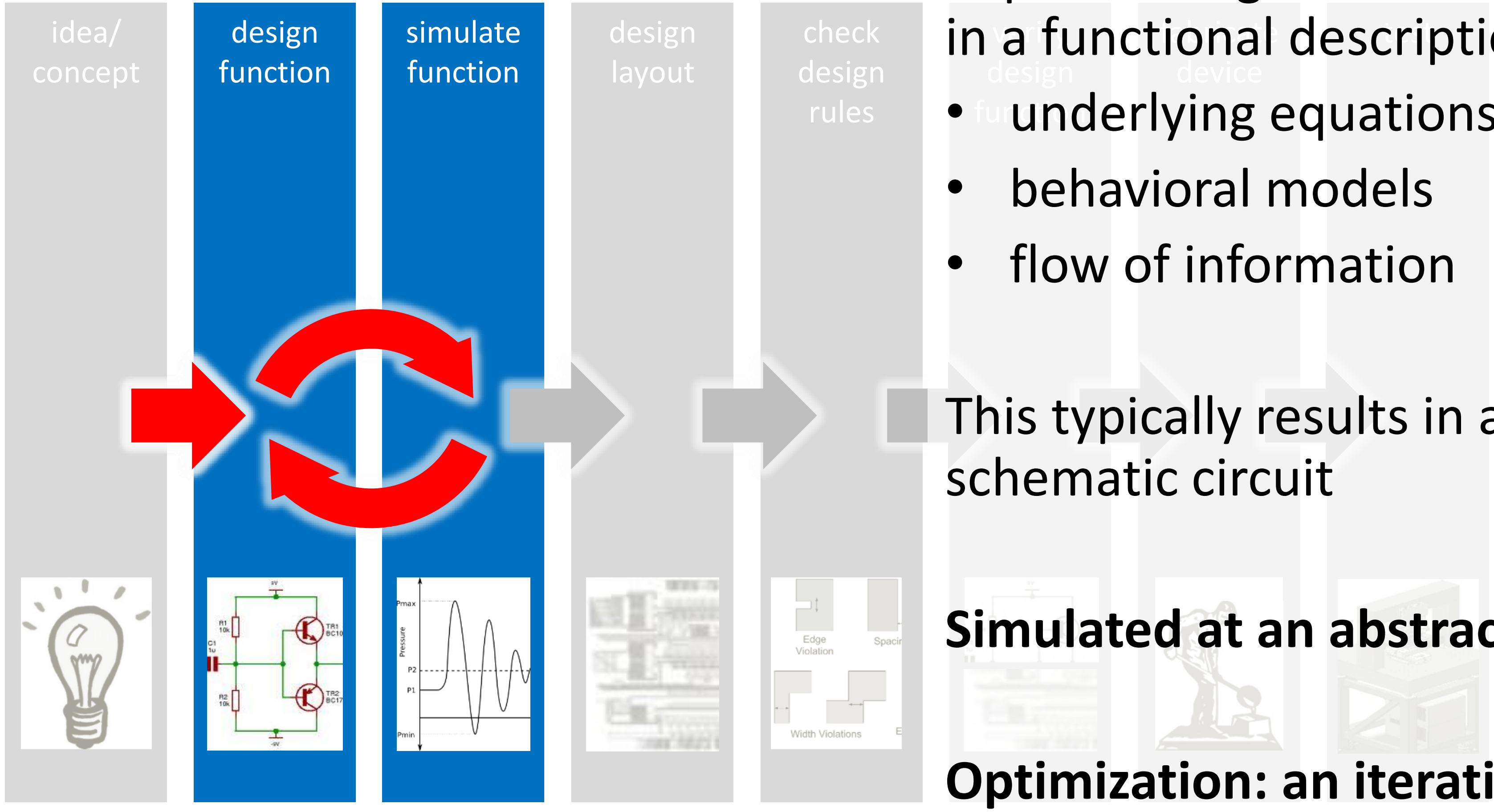
Simple connections between building blocks

- the length and shape does not really matter
- it should just provide a good connection
- similar as an electrical wire

Functional blocks with a certain phase/time delay

- length and shape are very important
- should be treated as a building block

Design Capture and Simulation



Capture design intent in a functional description

- underlying equations
- behavioral models
- flow of information

This typically results in a schematic circuit

Simulated at an abstract level

Optimization: an iterative process

MODELS FOR CIRCUIT SIMULATION

Should allow simulation in a larger circuit

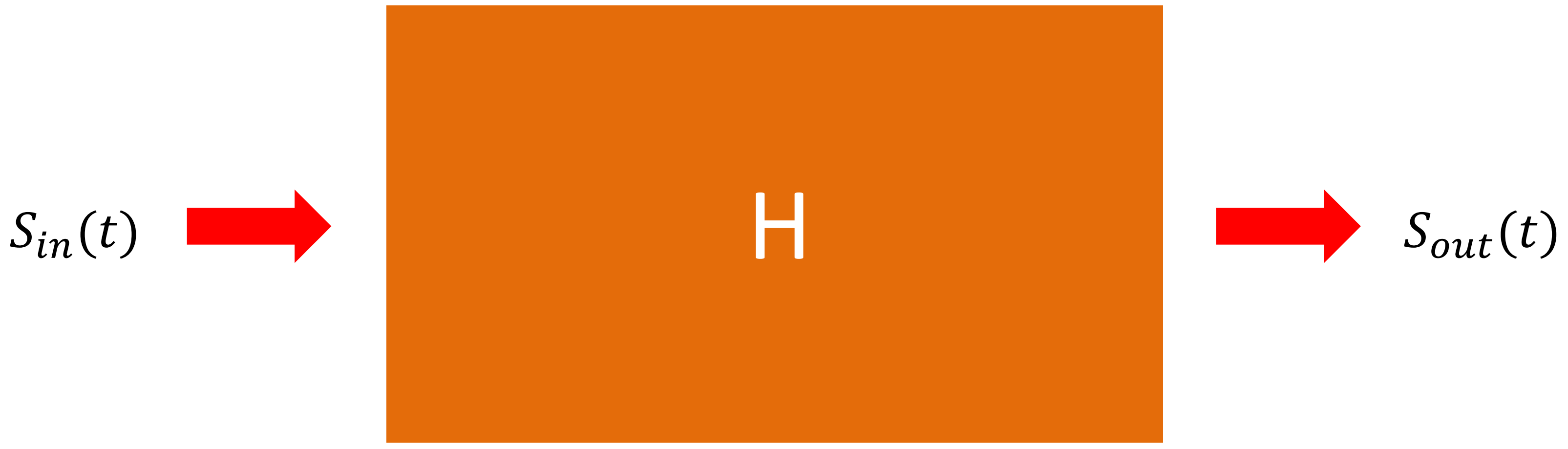
- based on equations
- based on measurement data
- based on EM simulations

Photonics: Nothing really standardized

- No standardized simulation method
- No standard model description
- No standard signals

A GOOD CIRCUIT MODEL

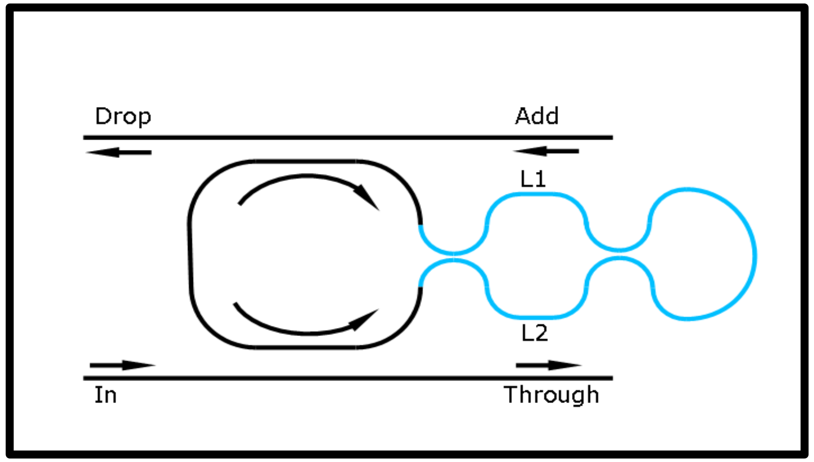
- Maps input signals correctly to output signals
- In frequency domain and time domain
- Is efficient (for circuit simulations)
- Has meaningful parameters
- Can be extracted from measurements



Black-box vs. White-box model

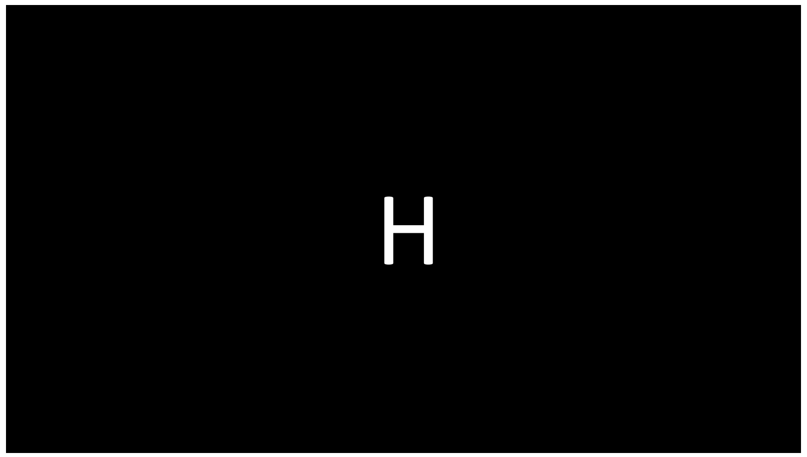
White-box:

- knows the circuit
- captures the physics

$S_{in}(t)$ →  → $S_{out}(t)$

Black-box:

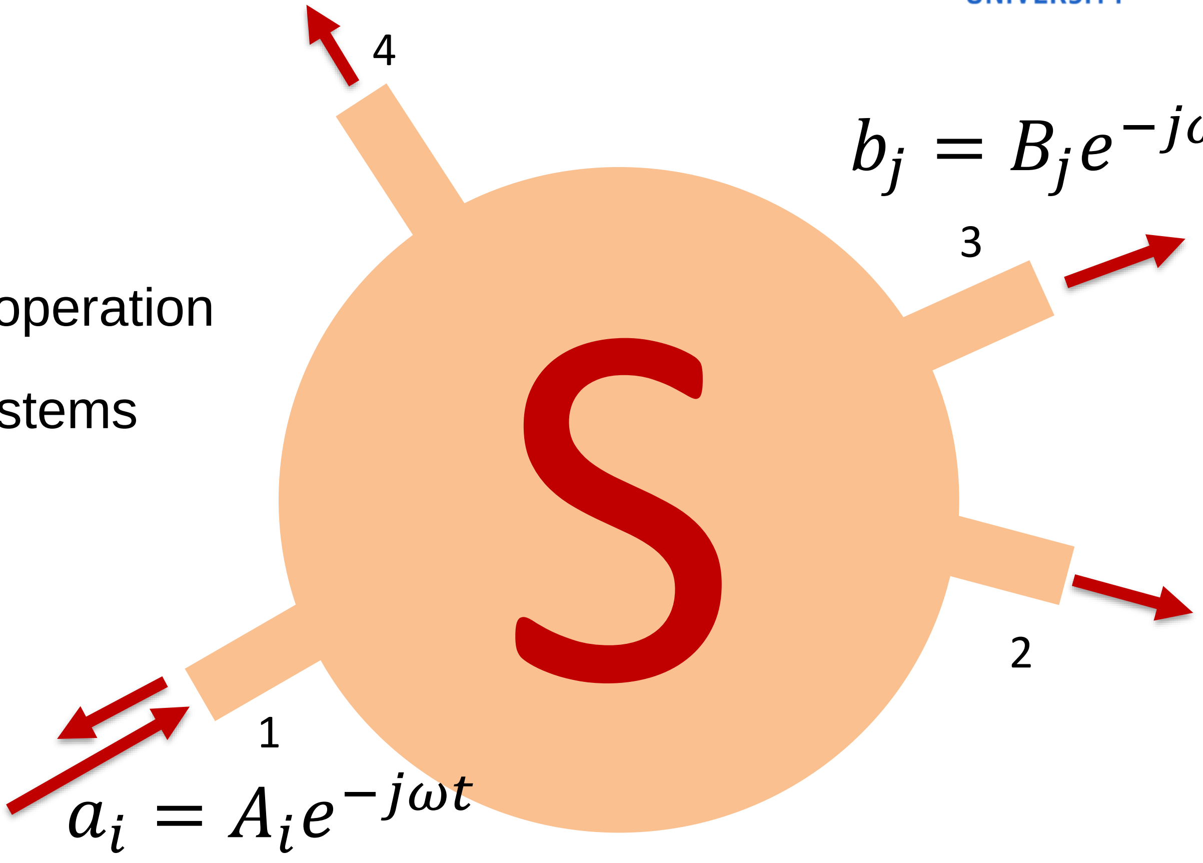
- internals unknown
- mathematical 'fit'

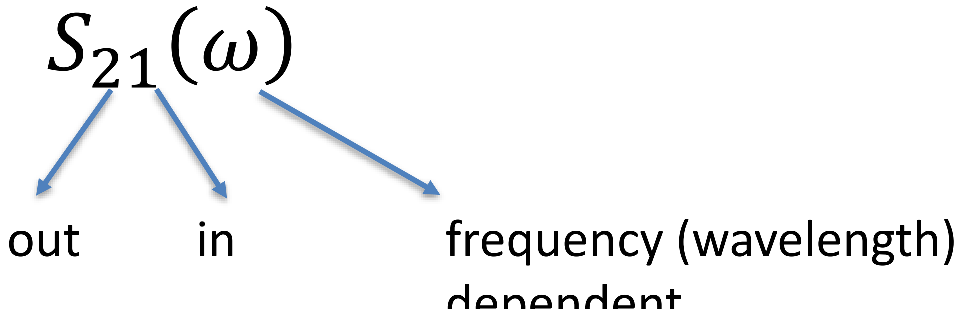
$S_{in}(t)$ →  → $S_{out}(t)$

OPTICAL CIRCUIT SIMULATION

Generalized scattering of an incoming wave

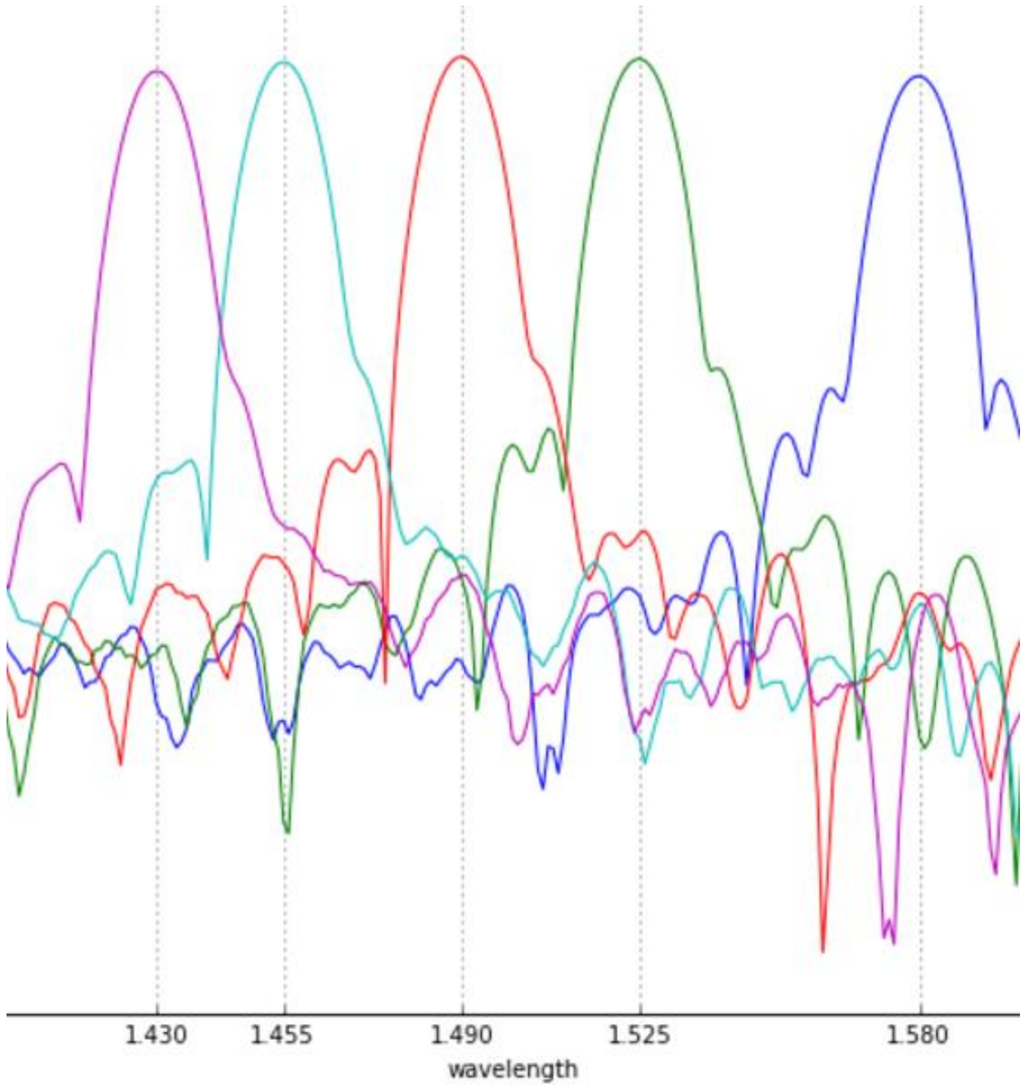
- Calculates one wavelength at a time
- gets response between all ports in one operation
- Can only model linear, time-invariant systems





FREQUENCY DOMAIN SIMULATIONS

GHENT UNIVERSITY imec



Frequency domain simulations are very useful for calculating

- Insertion losses
- Backreflections
- Dispersion (wavelength dependence)
- Wavelength filter response

and can also be extended to model

- Slowly varying effects
- Certain optical nonlinearities

WHAT IS A PORT OF A WAVEGUIDE COMPONENT?

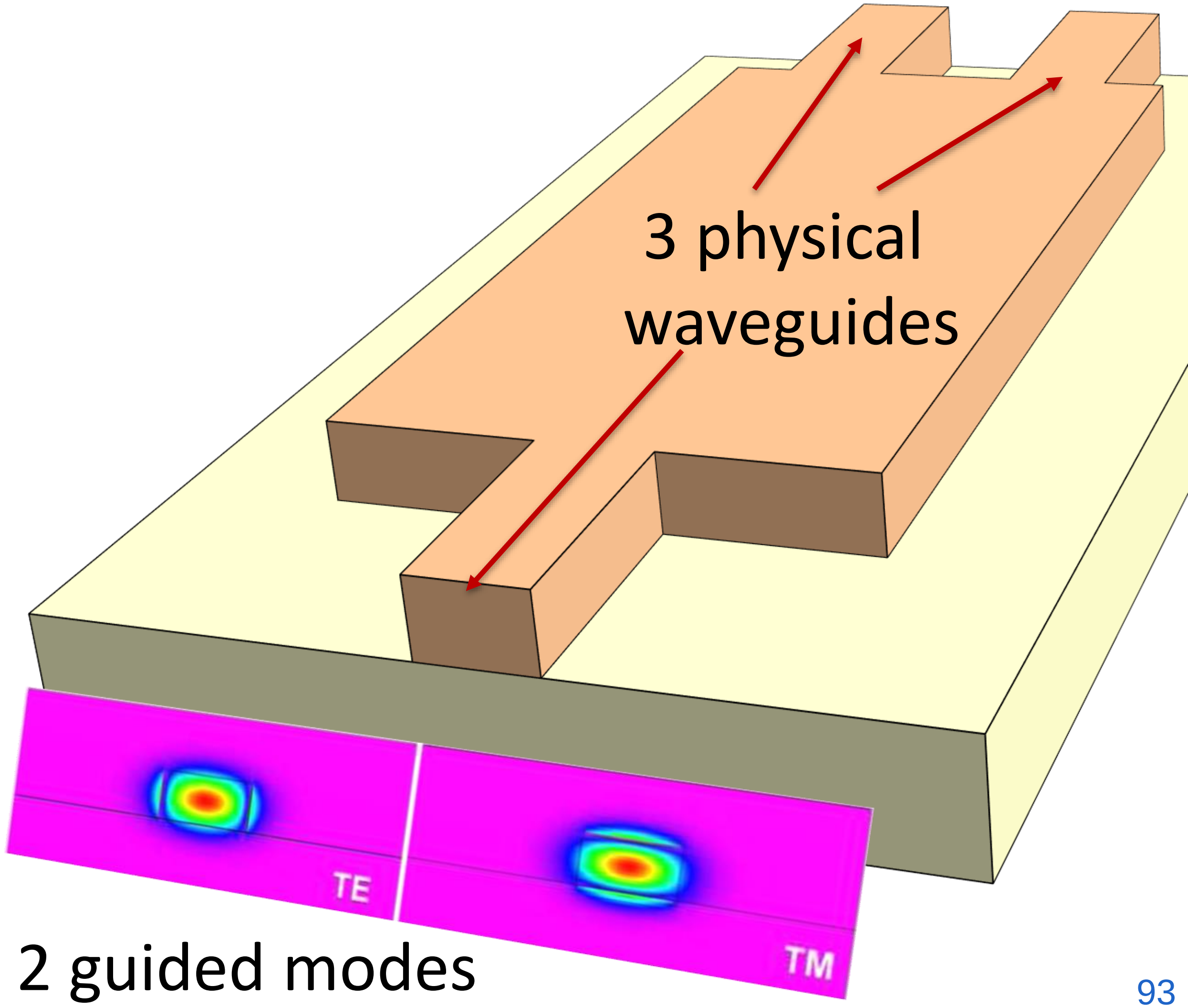
Orthogonal states

- Physically separated waveguides
- Each mode in the waveguide

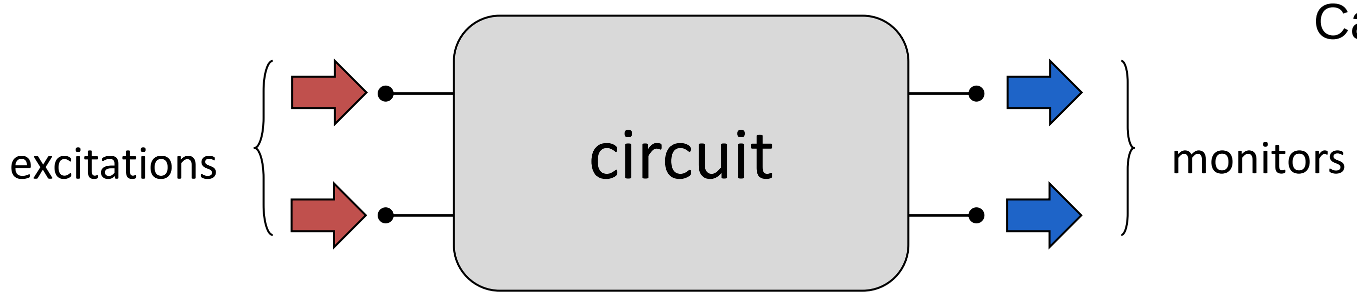
Example: 6 “ports” → 6×6 S-matrix

In practice:

Only use the relevant modes (rest is “loss”)



Time Domain Optical Circuit Simulation



Calculate time response of a circuit

- to a stimulus
  (or combination of excitations)
- at certain output monitors
- using discrete time steps

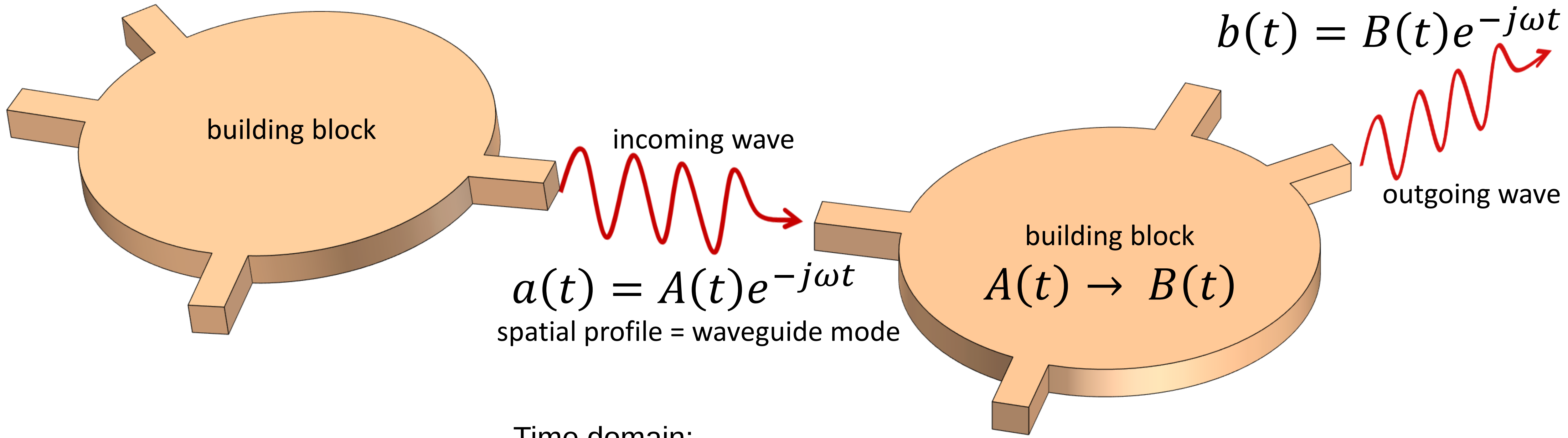
Pro:

- Faster than electromagnetic simulations
- Supports large circuits

Con:

- Slower than frequency domain
- Only response to specific stimulus

LIGHT PROPAGATES THROUGH CIRCUITS


building block
incoming wave
$a(t) = A(t)e^{-j\omega t}$
spatial profile = waveguide mode
building block
$A(t) \rightarrow B(t)$
$b(t) = B(t)e^{-j\omega t}$
outgoing wave

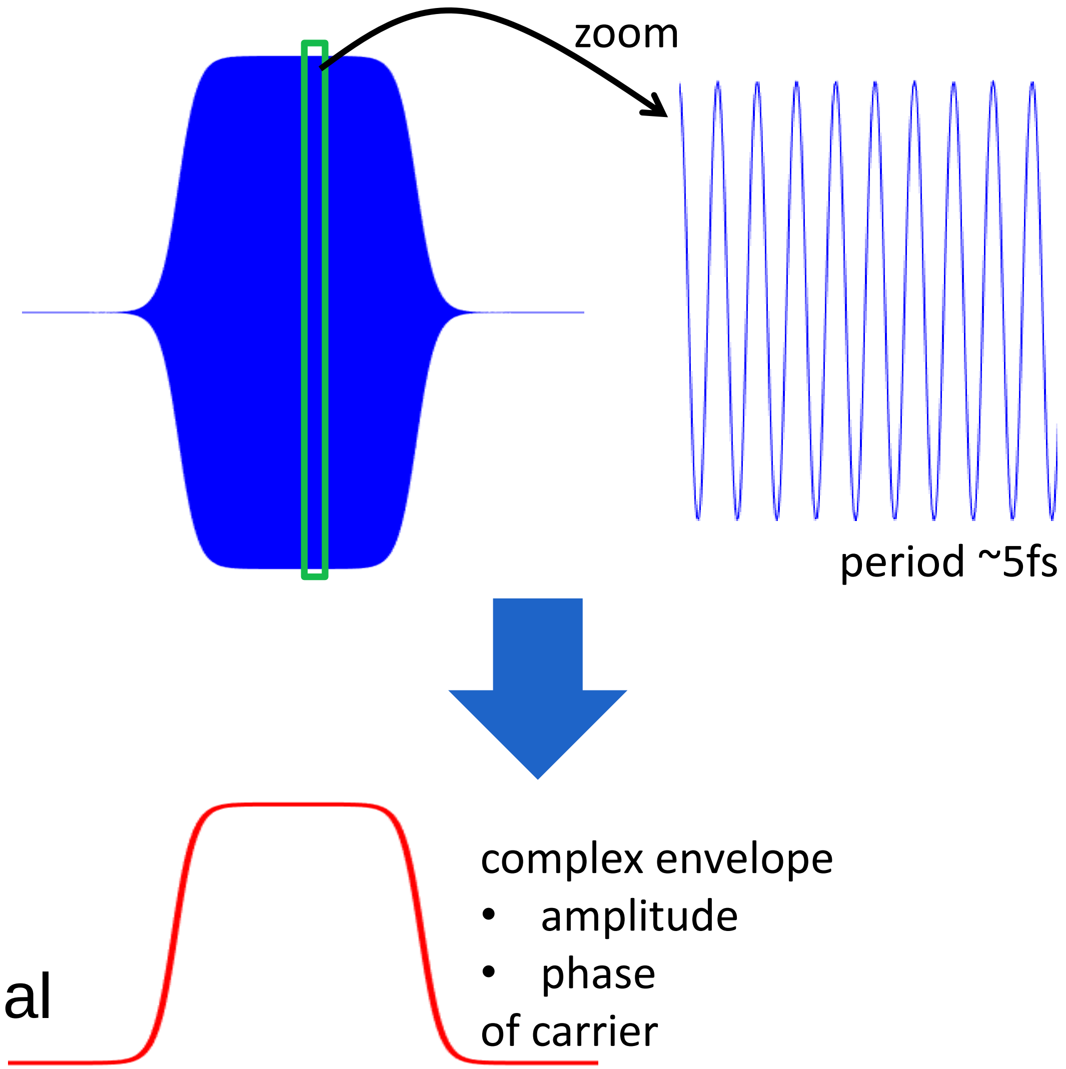
Time domain:

- time-varying signals propagating between nodes
- Linear, nonlinear and electro-optic systems
- Basically any equation can describe a node
- Still fast, but slower than frequency-domain
- Every excitation needs a new simulation

Optical Vs. Electrical Circuit Simulation

optical = electrical … at very high frequency

- ultra-small time steps (fs)
- ultra-long simulations (10^{12} time steps)
- high-bandwidth signals (200THz)

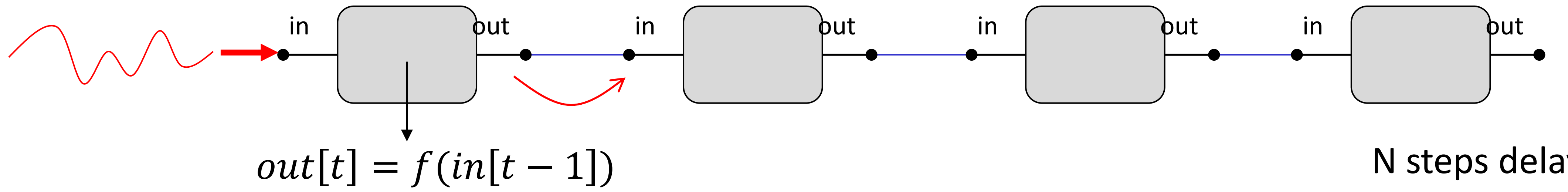


impractical!

Solution: analytic signal

= complex amplitude on carrier

TIME STEPS



need sufficient accuracy:

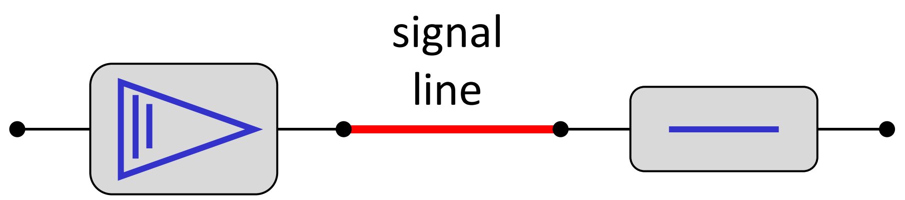
- circuit elements have a delay of at least 1 time step
- integration of differential equations get more accurate with smaller time steps
- Smaller steps = longer simulation

OPTICAL SIGNALS

$2 \times 2 \times N \times M$

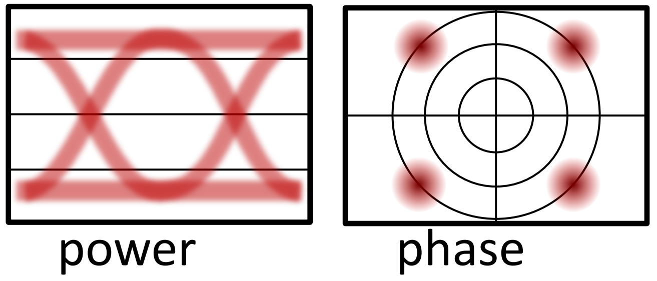
not all simulators support all combinations

An optical link carries an an optical signal…

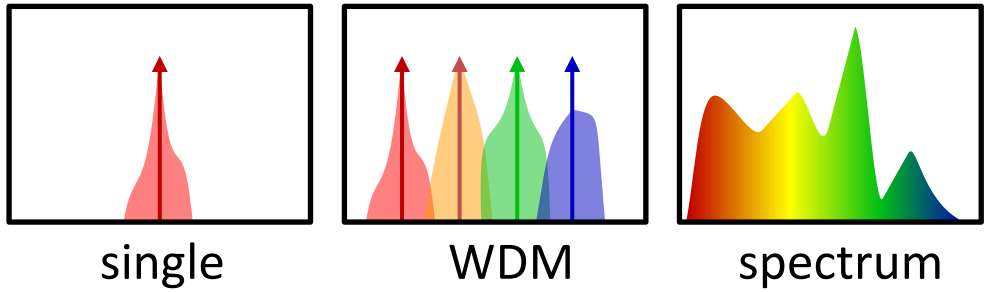


two directions

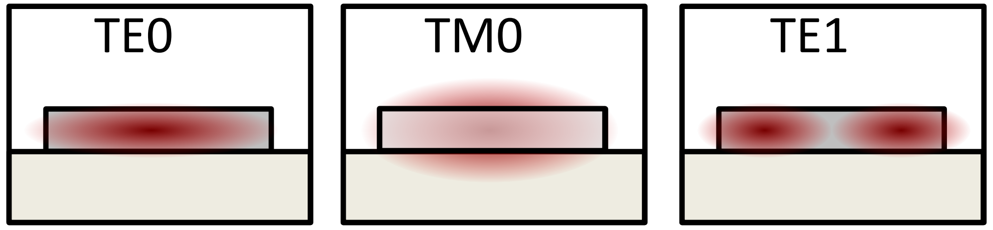
complex number



wavelength:
N channels



mode/polarization:
M modes



OPTICAL SIGNALS: EXAMPLE

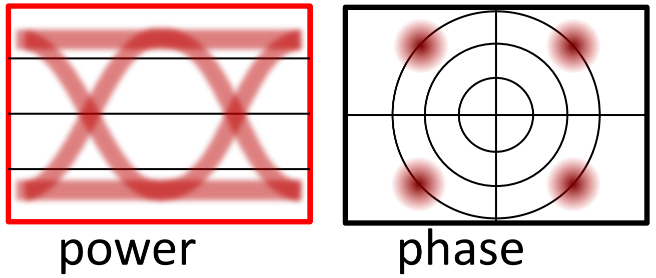
Example: Single-λ link

- One direction
- One wavelength
- On-off-keying: power
- One mode: TE

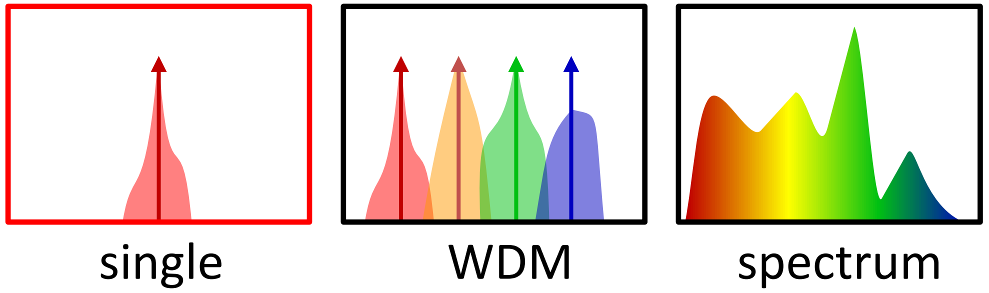
two directions

$1 \times 1 \times 1 \times 1$

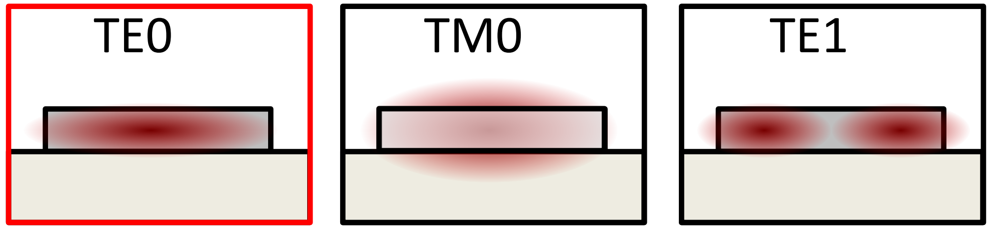
complex number



wavelength:
N channels



mode/polarization:
M modes



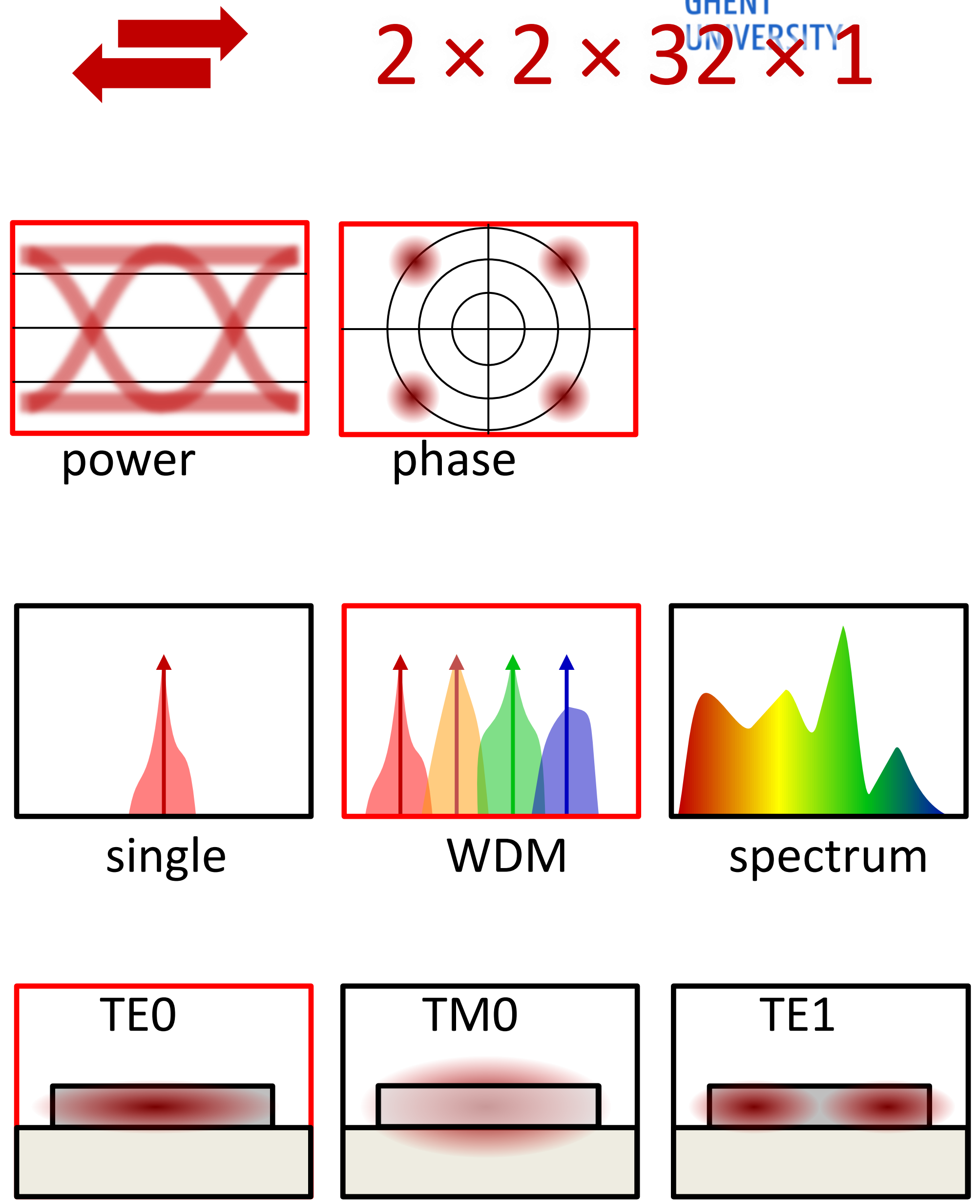
Optical Signals: Example

Example: WDM bidirectional link

- two directions
- QPSK modulation: phase
- 32 wavelength channels
- one mode

two directions

$2 \times 2 \times 32 \times 1$



OPTICAL SIGNALS: EXAMPLE

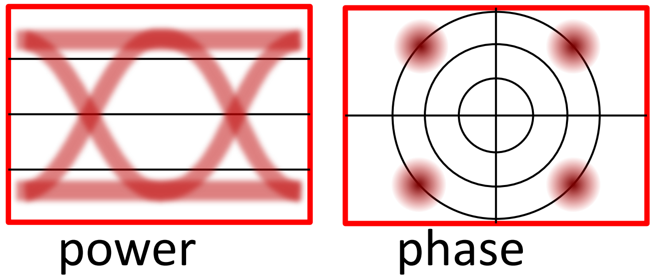
Example: DWDM multimode link

- two directions
- QAM64 modulation: phase
- 512 wavelength channels
- 4 modes

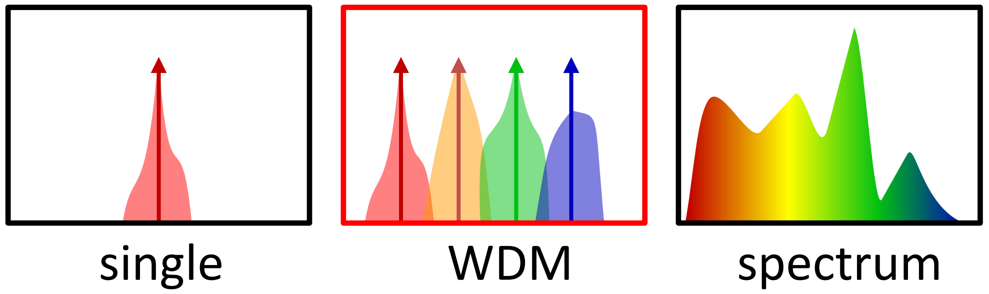
two directions

$2 \times 2 \times 512 \times 4$

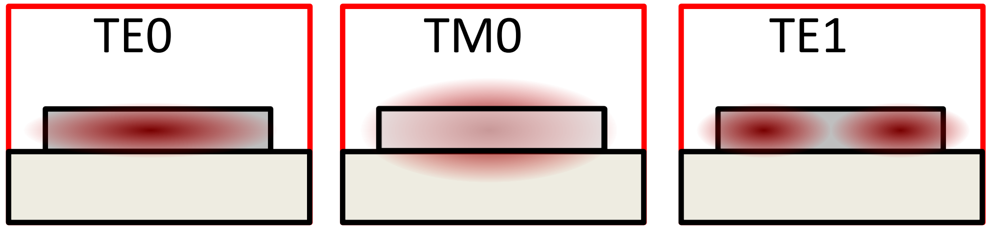
complex number



wavelength:
N channels



mode/polarization:
M modes



Optical Signals: Example

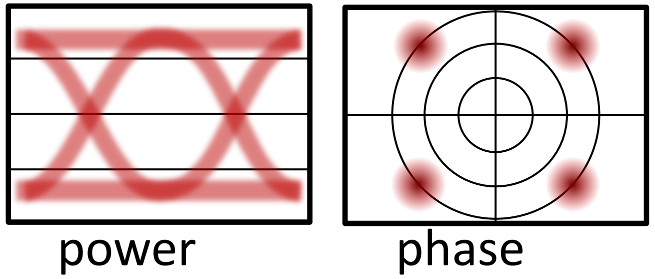
Example: Spectrometer

- two directions (parasitic reflections)
- wavelength filtering: phase
- continous spectrum: 100nm @ 5pm
- one mode

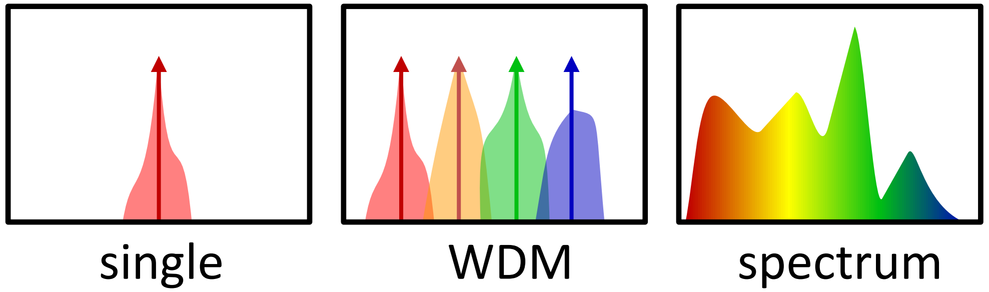
two directions

$2 \times 2 \times 20000 \times 1$

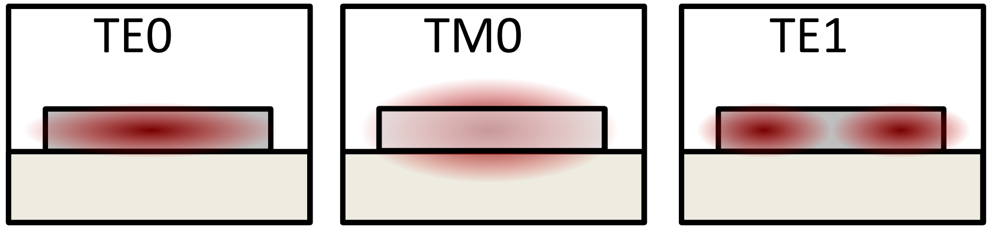
complex number



wavelength:
N channels



mode/polarization:
M modes



Simulating linear circuits

Photonics does not fit easily in Spice

Effort-flow systems

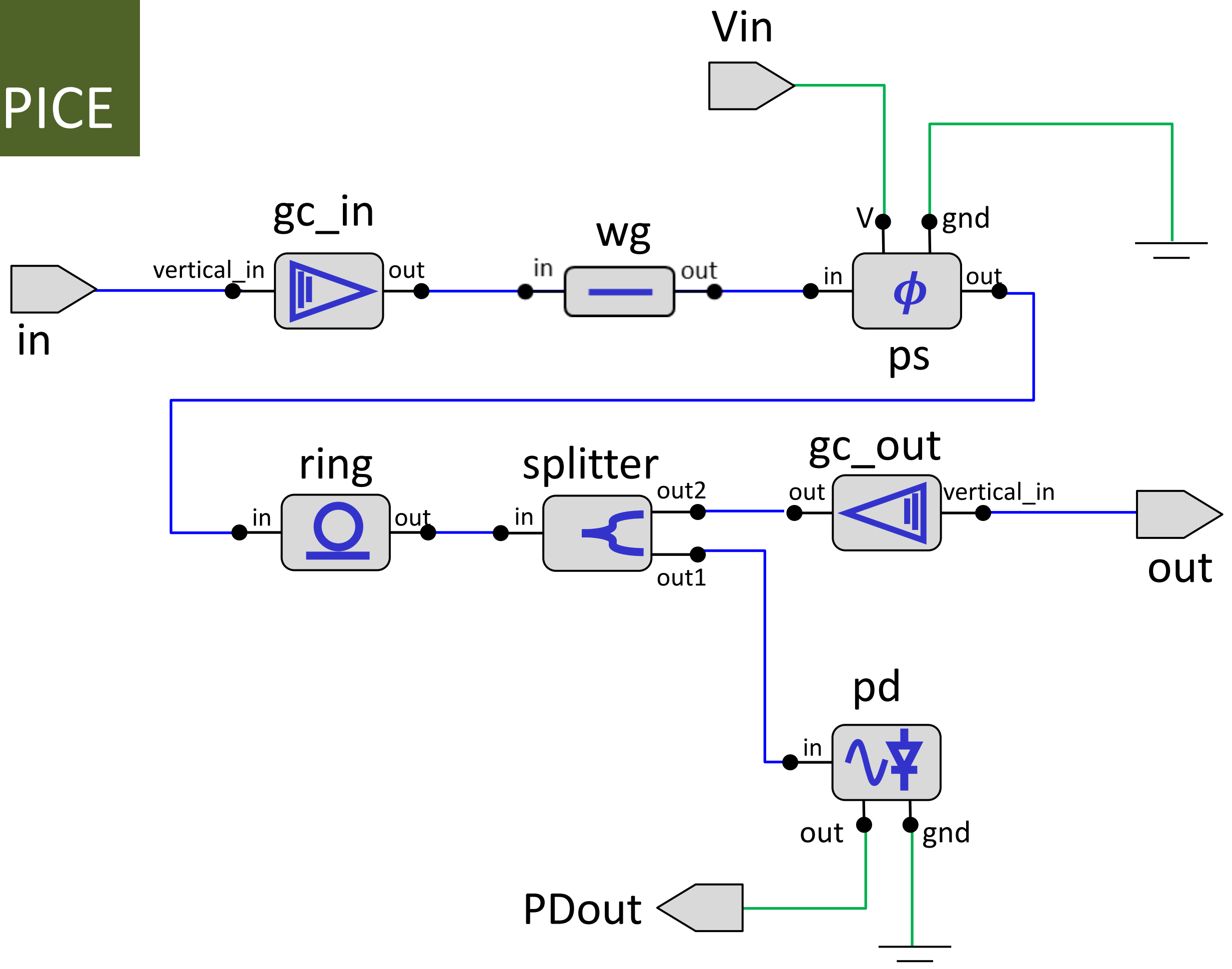
Electrical	Voltage	Current
Fluidic	Pressure	Flow
Thermal	Temperature	Heat Flow*
Mechanical	Force	Motion
Photonic?	E-field	H-field

Not the best formalism for photonics
(more like an RF wave)

Photonics and Electronics use different formalisms

electrical:
- effort-flow / SPICE

How to co-simulate?

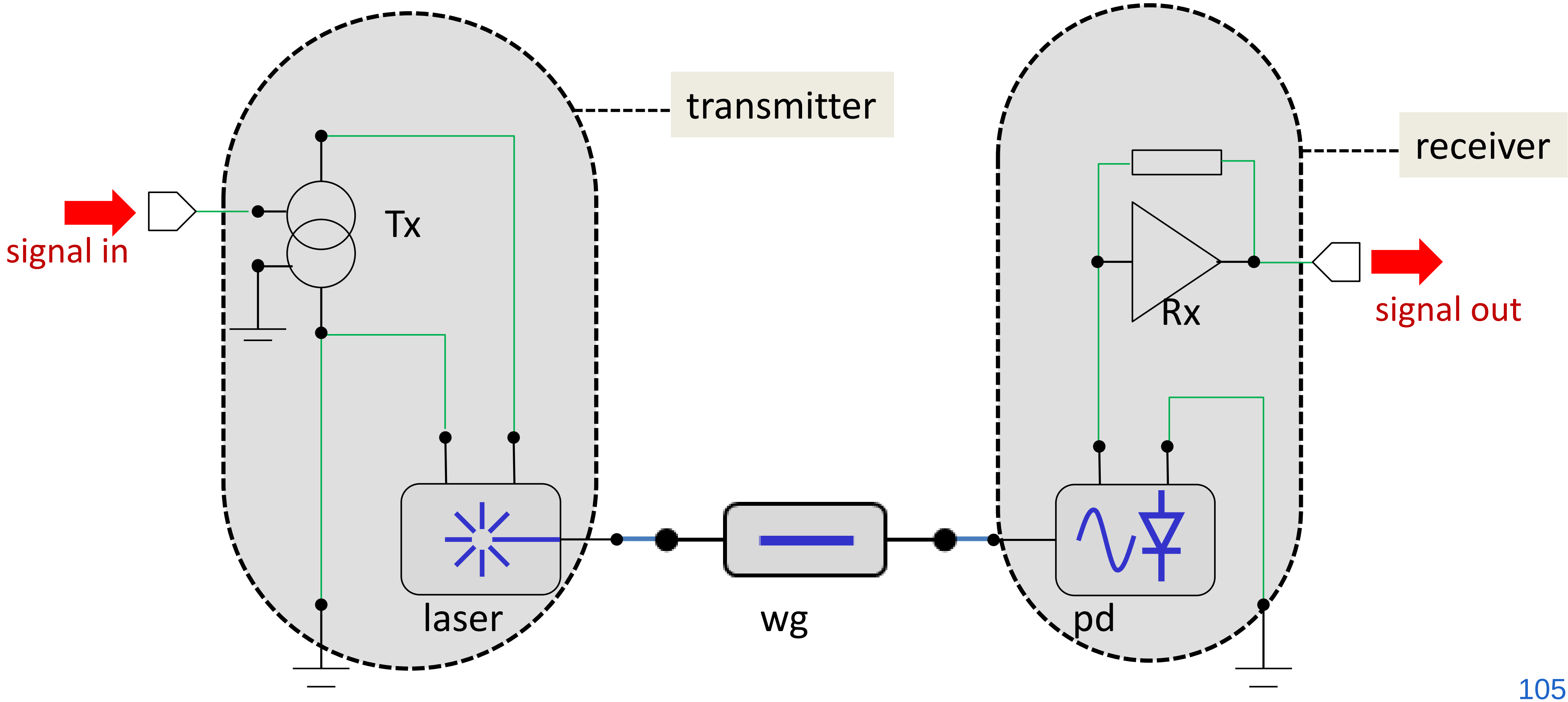


optical:
- Scattering waves

SIMULATING PHOTONICS + ELECTRONICS

Real system: photonics + electronics

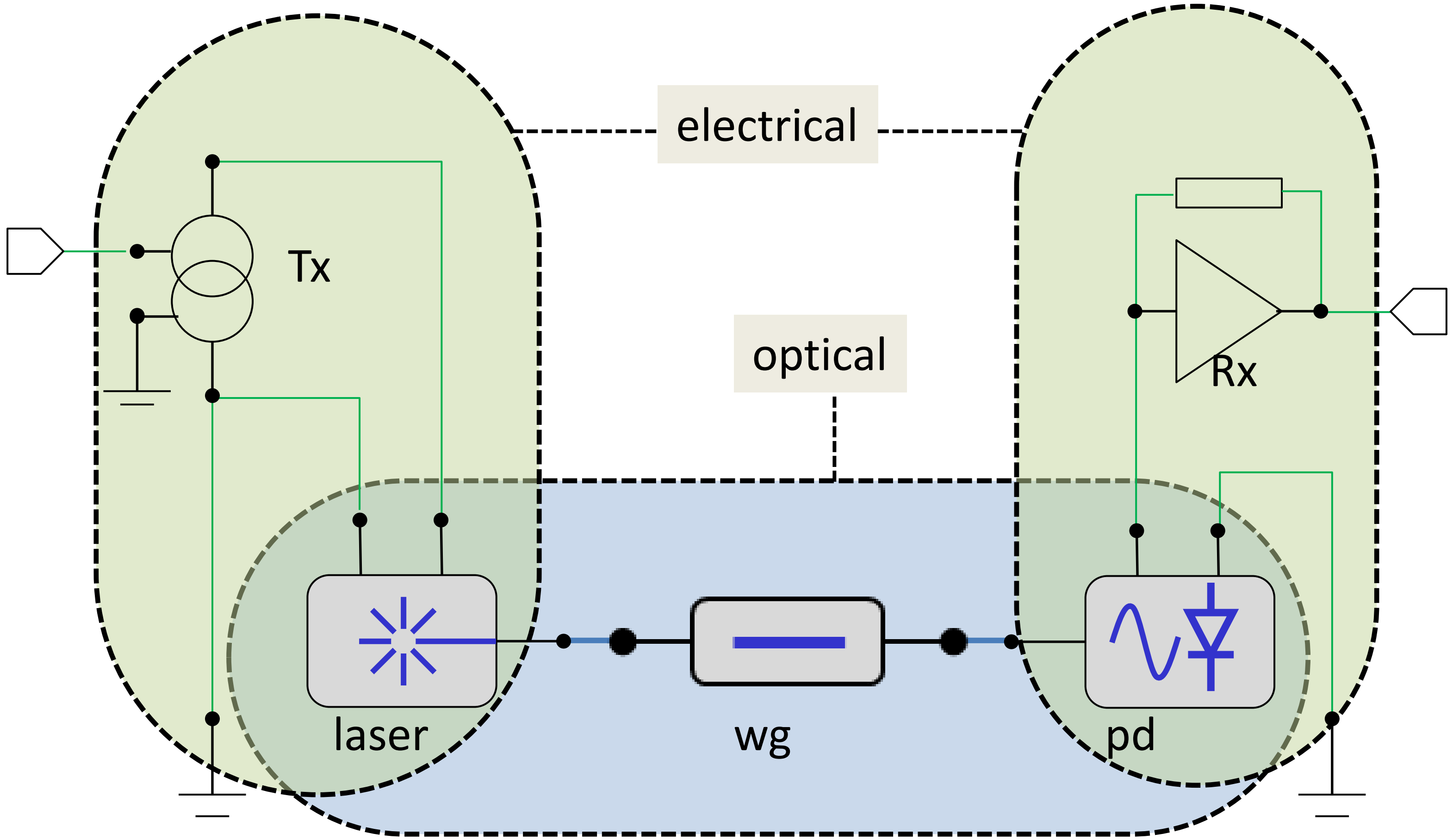
Example: optical link



Simulating Photonics + Electronics

Circuit has optical and electrical parts:

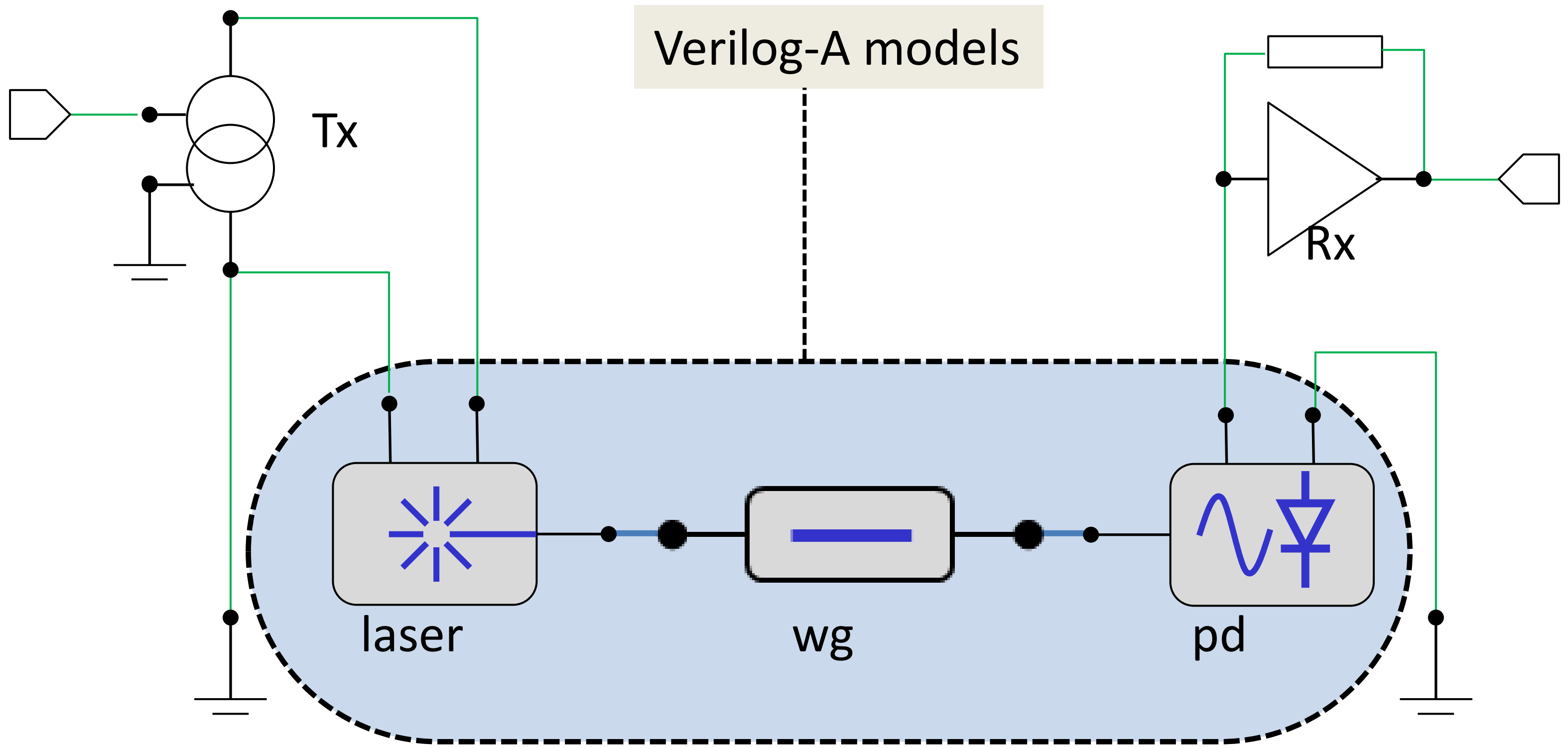
Some components overlap



Simulating Photonics + Electronics

Simulating everything in electrical simulator (SPICE – MNA)

- Use native, verified models for electronics
- Build Verilog-A models for photonics

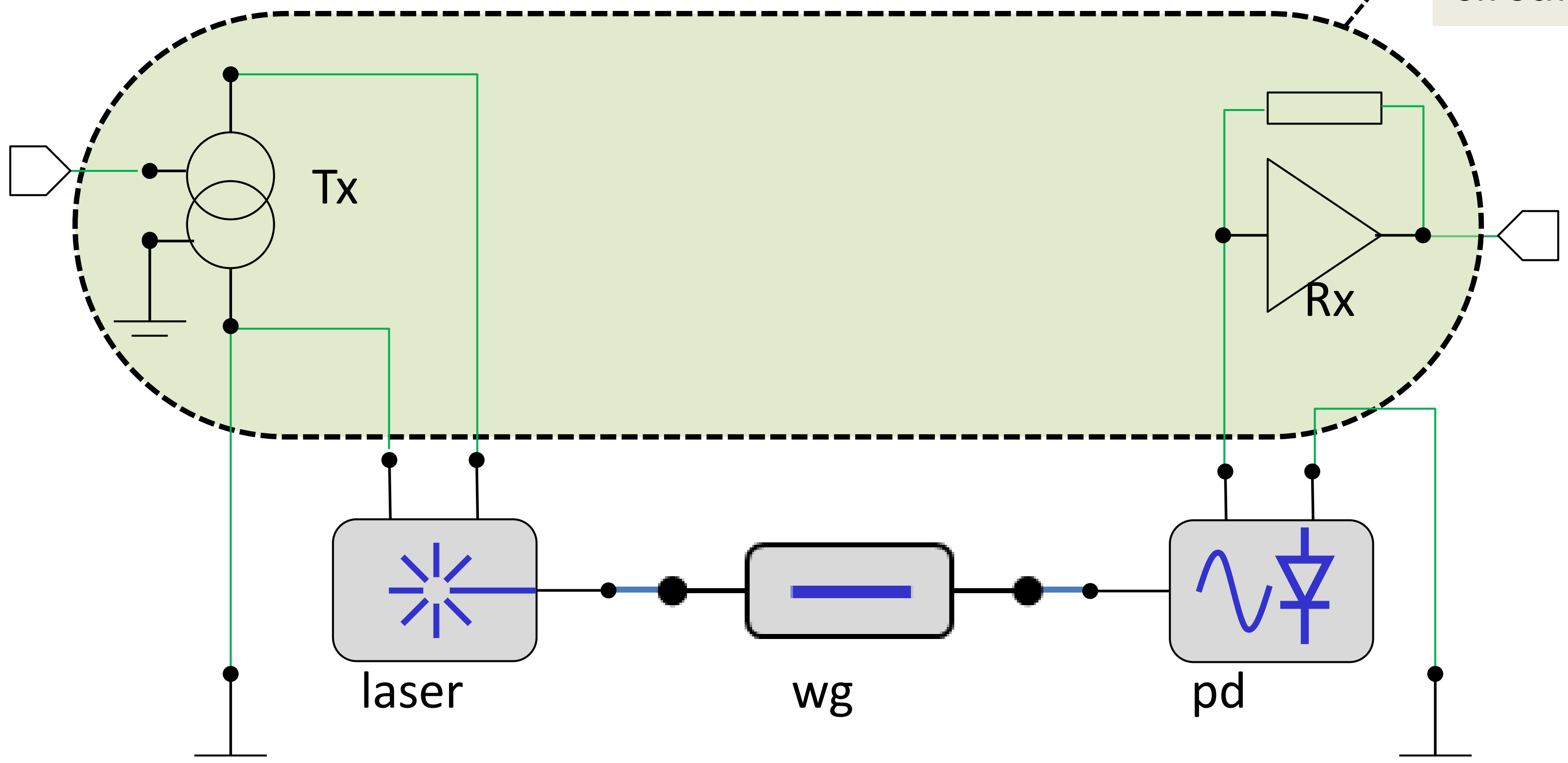


SIMULATING PHOTONICS + ELECTRONICS

Simulate everything in a photonics simulator (Interconnect, Caphe, OptSim)

- Optimized models and formalisms for photonics
- Electronics models need to be mapped. No verified fab models

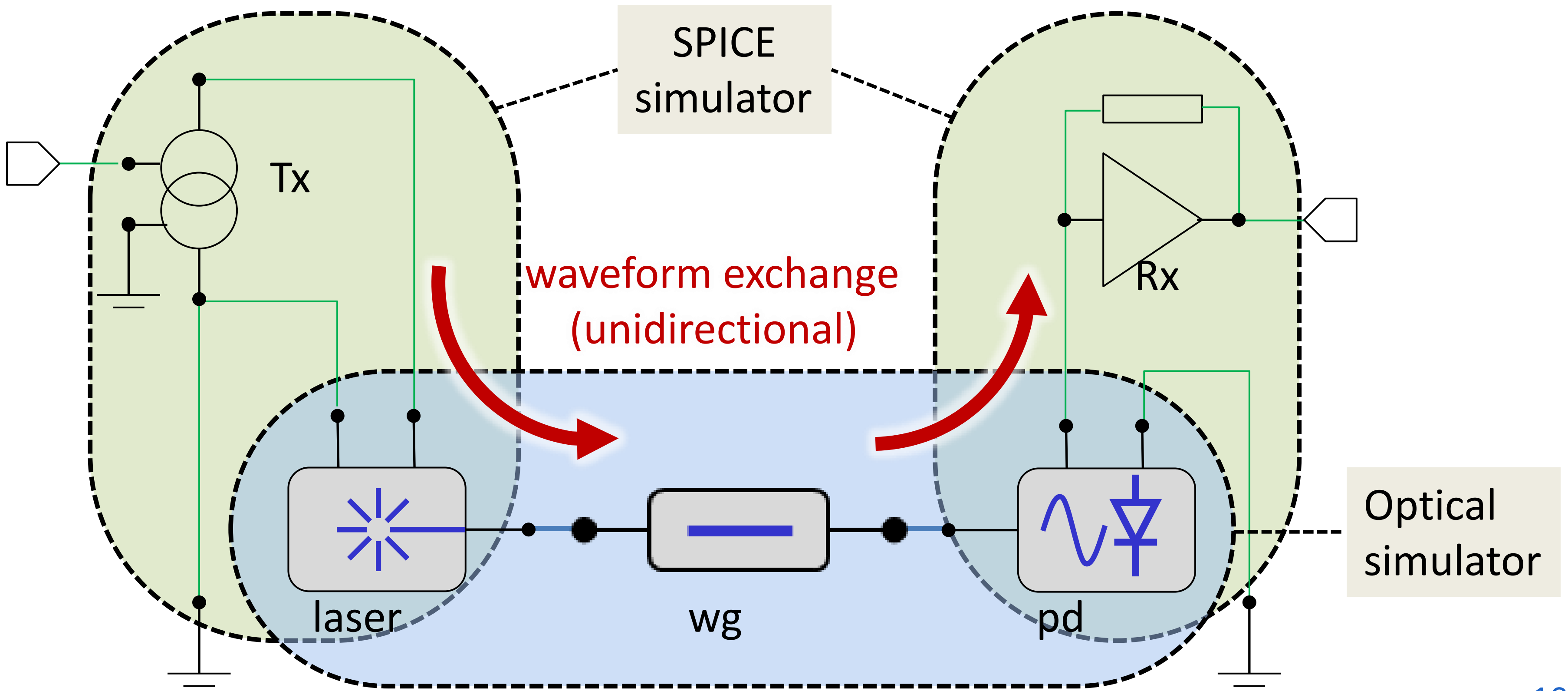
custom models for photonic circuit simulator



SIMULATING PHOTONICS + ELECTRONICS

Co-simulate with waveform exchange

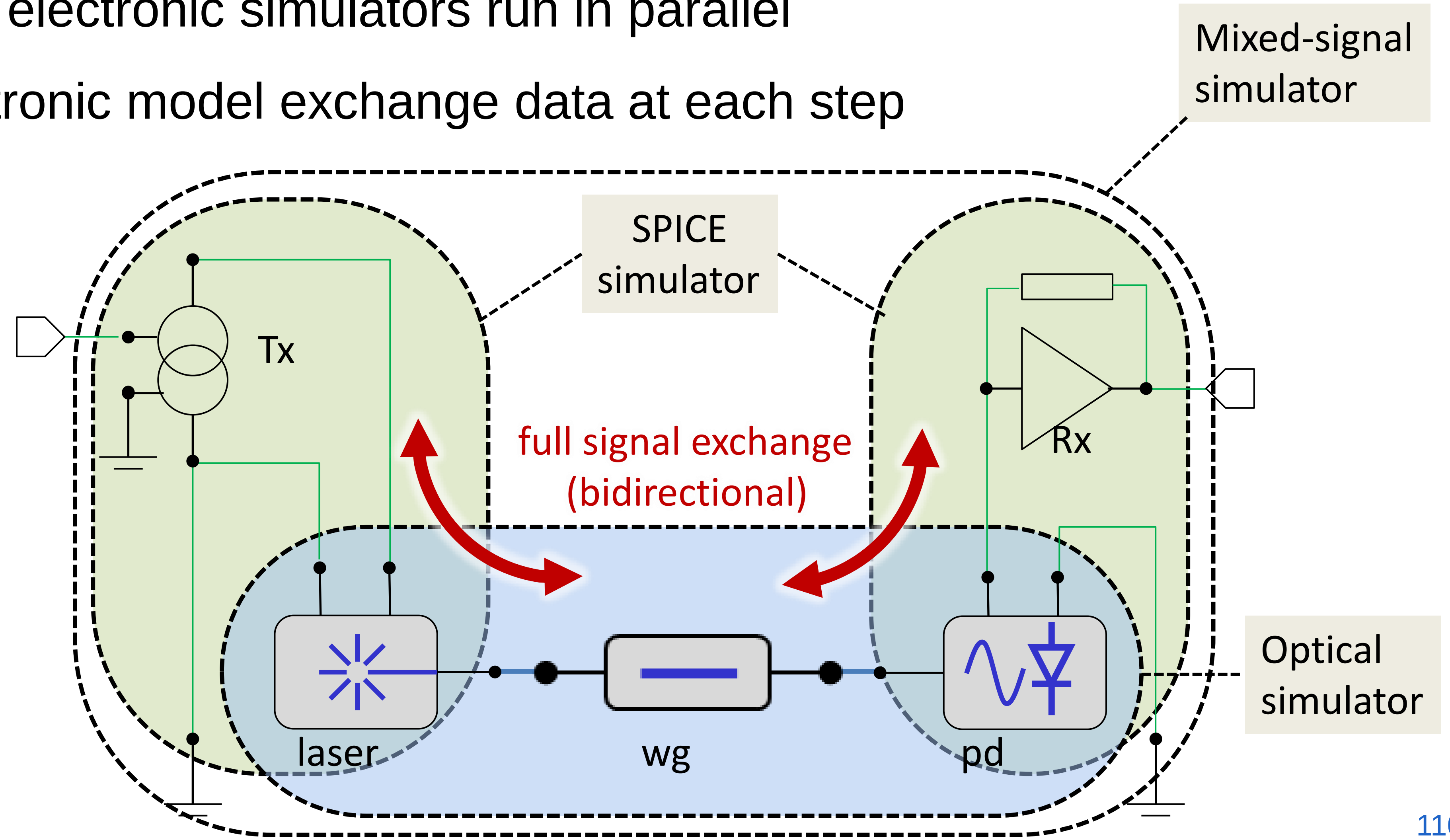
- Photonics and electronics in optimized model, executed sequentially
- Output of one simulation = input of next simulation



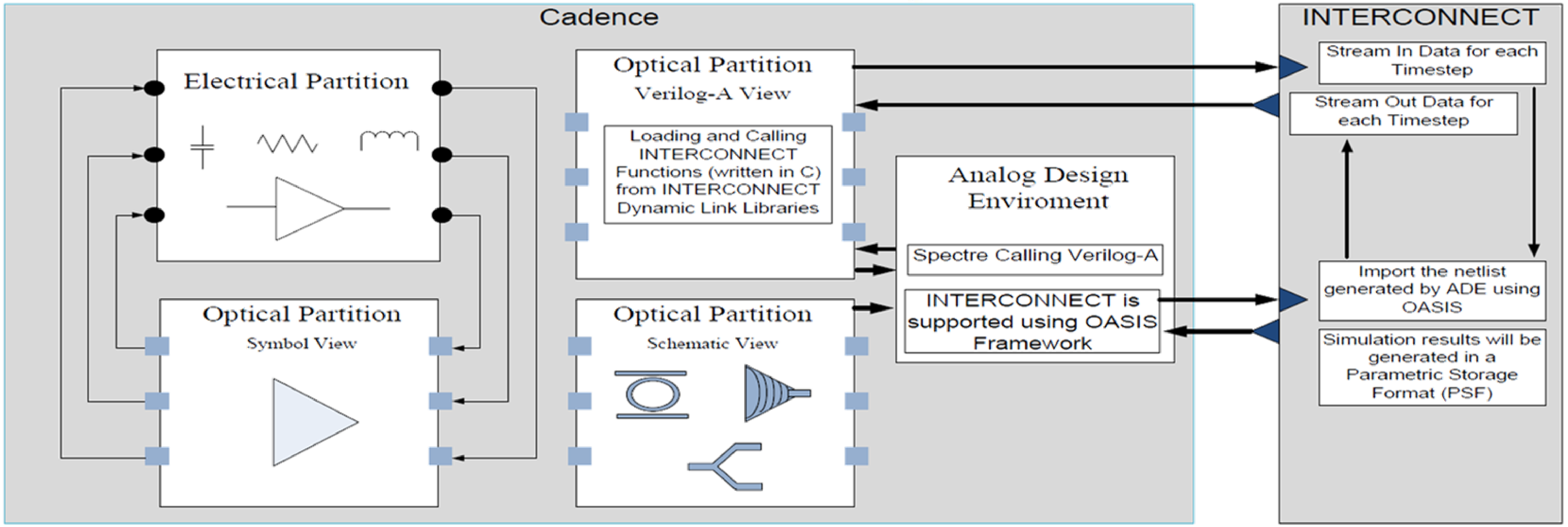
SIMULATING PHOTONICS + ELECTRONICS

True cosimulation (photonics and electronics in lockstep)

- Both photonic and electronic simulators run in parallel
- Photonic and electronic model exchange data at each step



CO-SIMULATION

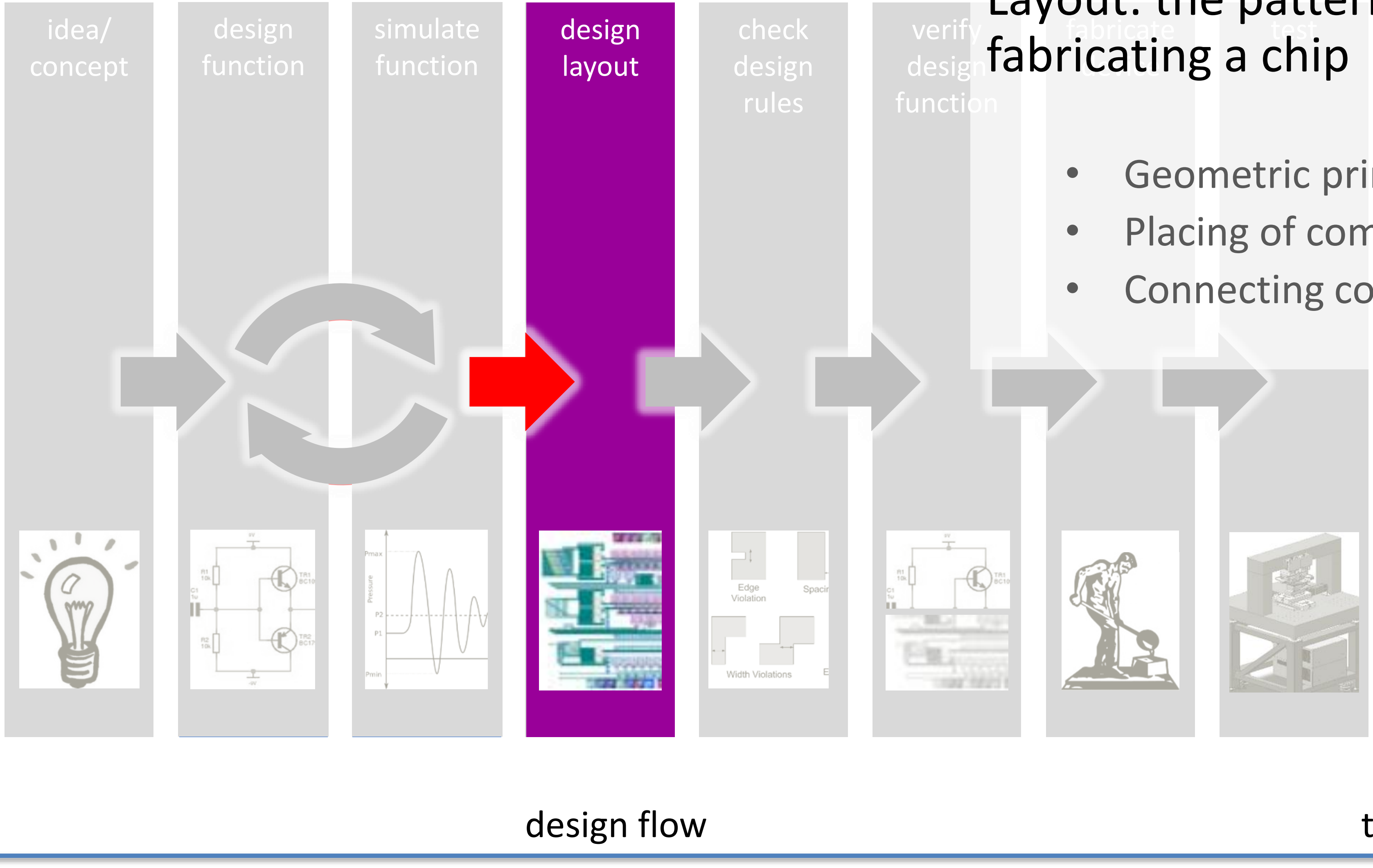

Cadence
Electrical Partition
Optical Partition
Symbol View
Optical Partition
Verilog-A View
Loading and Calling INTERCONNECT Functions (written in C) from INTERCONNECT Dynamic Link Libraries
Optical Partition
Schematic View
Analog Design Enviroment
Spectre Calling Verilog-A
INTERCONNECT is supported using OASIS Framework
INTERCONNECT
Stream In Data for each Timestep
Stream Out Data for each Timestep
Import the netlist generated by ADE using OASIS
Simulation results will be generated in a Parametric Storage Format (PSF)

Optical and electrical co-design in Virtuoso Schematic

Photonic simulation in Lumerical Interconnect

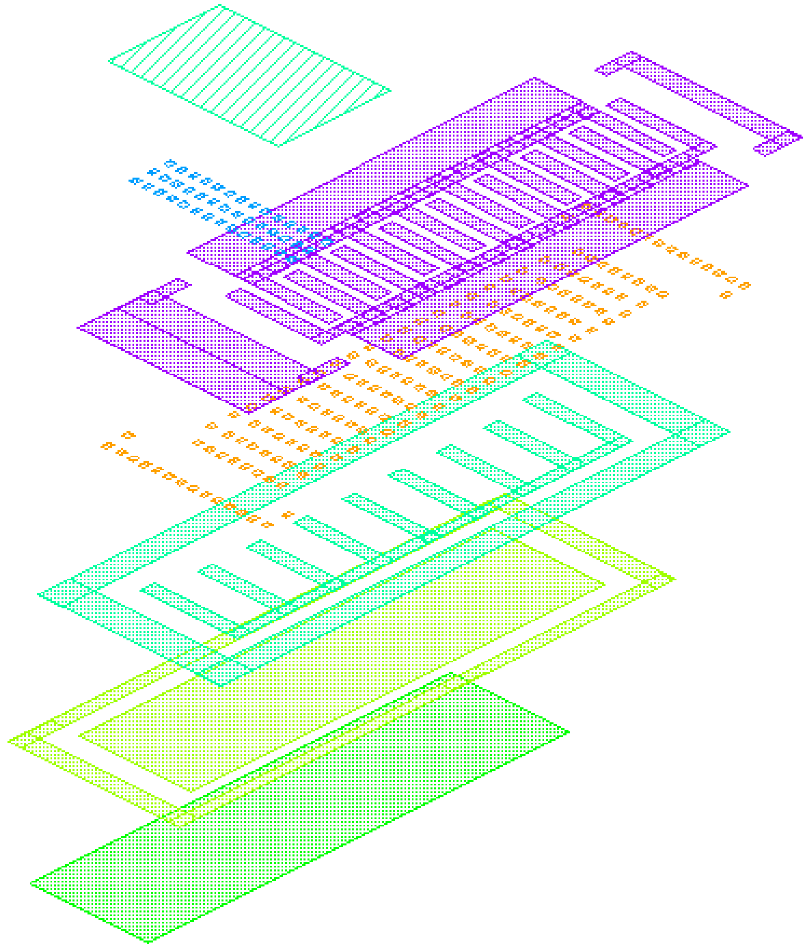
A. Farsaei, APC 2016, JTu4A.1

FROM FUNCTION TO LAYOUT



Layout: the patterns used for fabricating a chip

- Geometric primitives
- Placing of components
- Connecting components



LAYOUT

Geometric patterns

- Originally drawn by hand
- Now drawn by computer
- or programmed using scripts

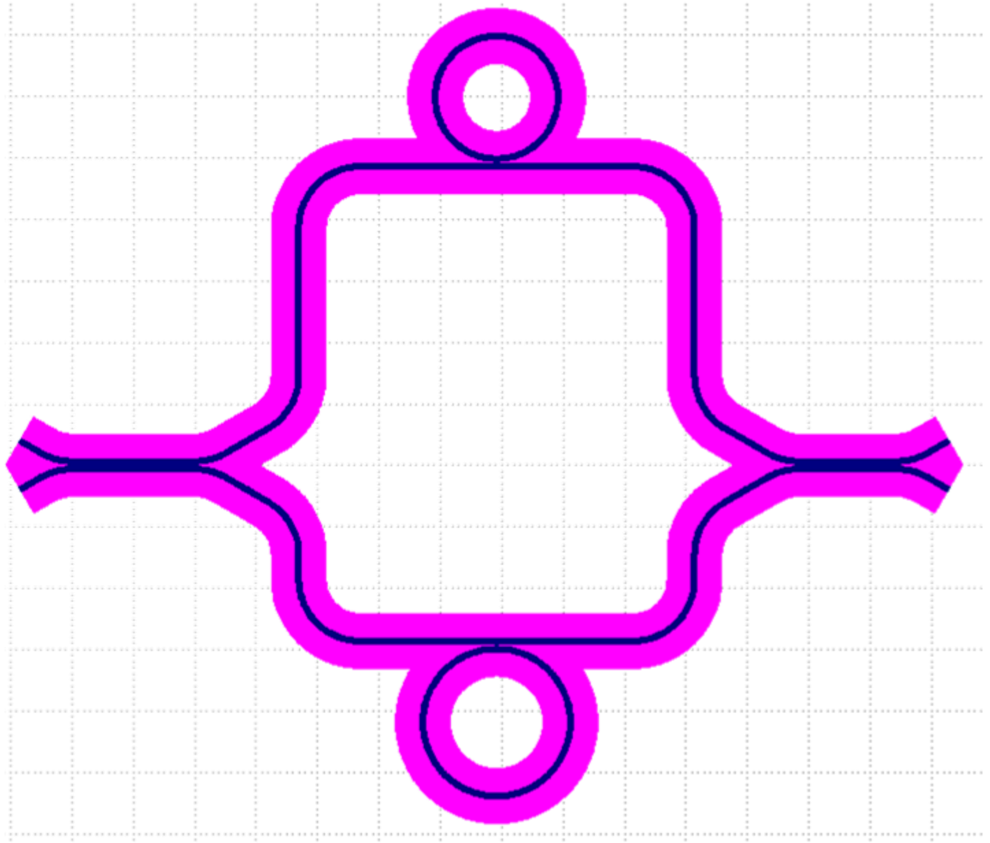
Different layers

- correspond to process steps: Mask layers
- or to logical operations (e.g. Boolean operations)

Different purposes

- Intent of the drawn shape:
  process, exclusion, annotation, …

LAYOUT: CIRCUITS



Organized in (reusable) Cells

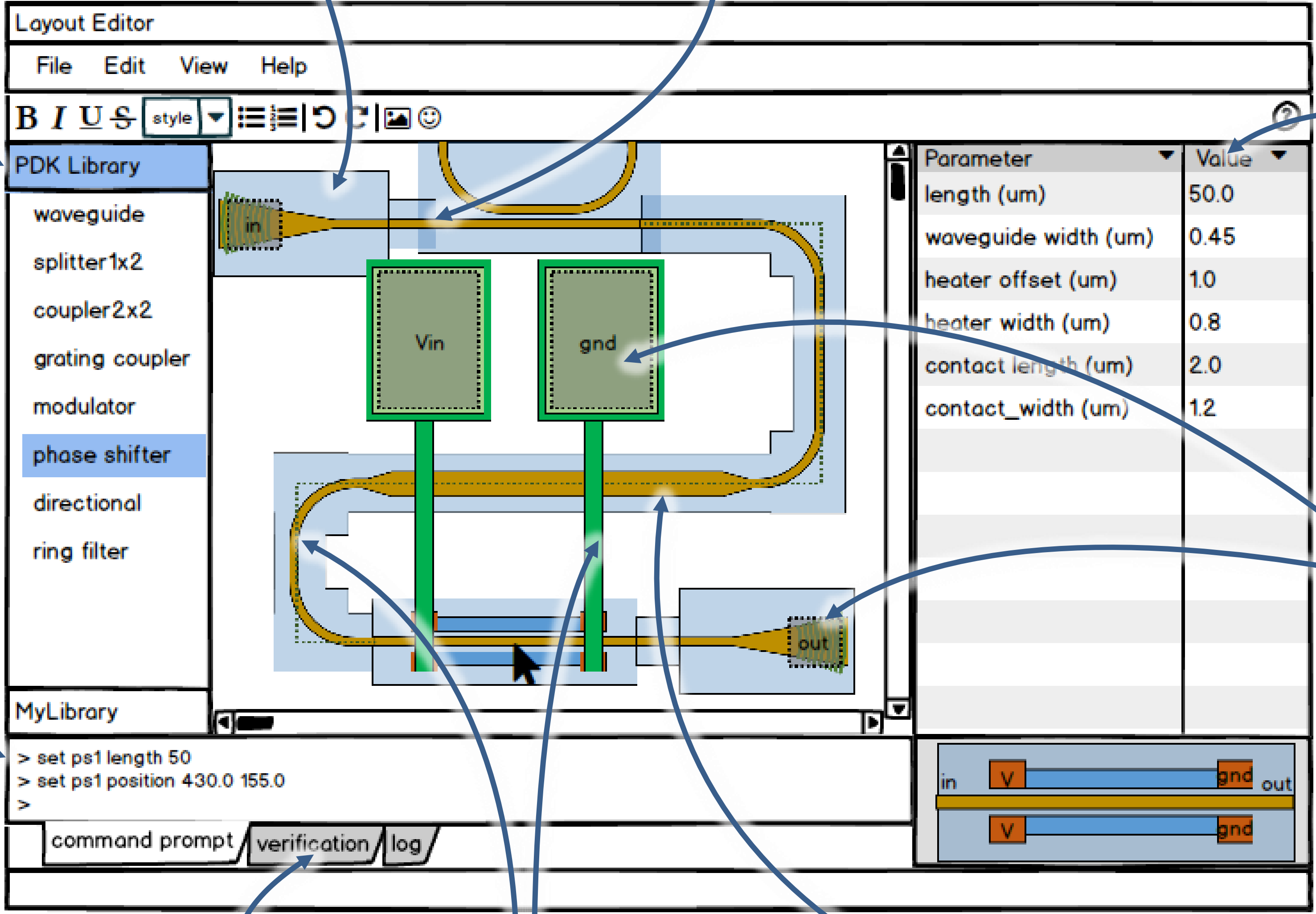
- placement
- transformations

Hierarchy: Cells contain other cells

Routing

- Optical connectivity with waveguides
- Electrical connectivity with metal wiring
- Avoid crossings/shorts/disconnects

LAYOUT EDITORS



```
> set ps1 length 50
> set ps1 position 430.0 155.0
>
```

Parameter	Value
length (um)	50.0
waveguide width (um)	0.45
heater offset (um)	1.0
heater width (um)	0.8
contact length (um)	2.0
contact_width (um)	1.2

drag and dropping components

alignment and snapping at waveguide ports

component libraries

parametrization

scriptability

optical and electrical pins

interface to verification (DRC and LVS)

routing of waveguides and electrical wires

smart waveguide cells with automatic bend radius and flaring in long segments

A DESIGN CELL

combines the different aspects of the design

- symbolic representation
- layout (shapes on mask layers)
- location and orientation of the ports
- a model

Static content: can be stored in a file (e.g. EDIF)

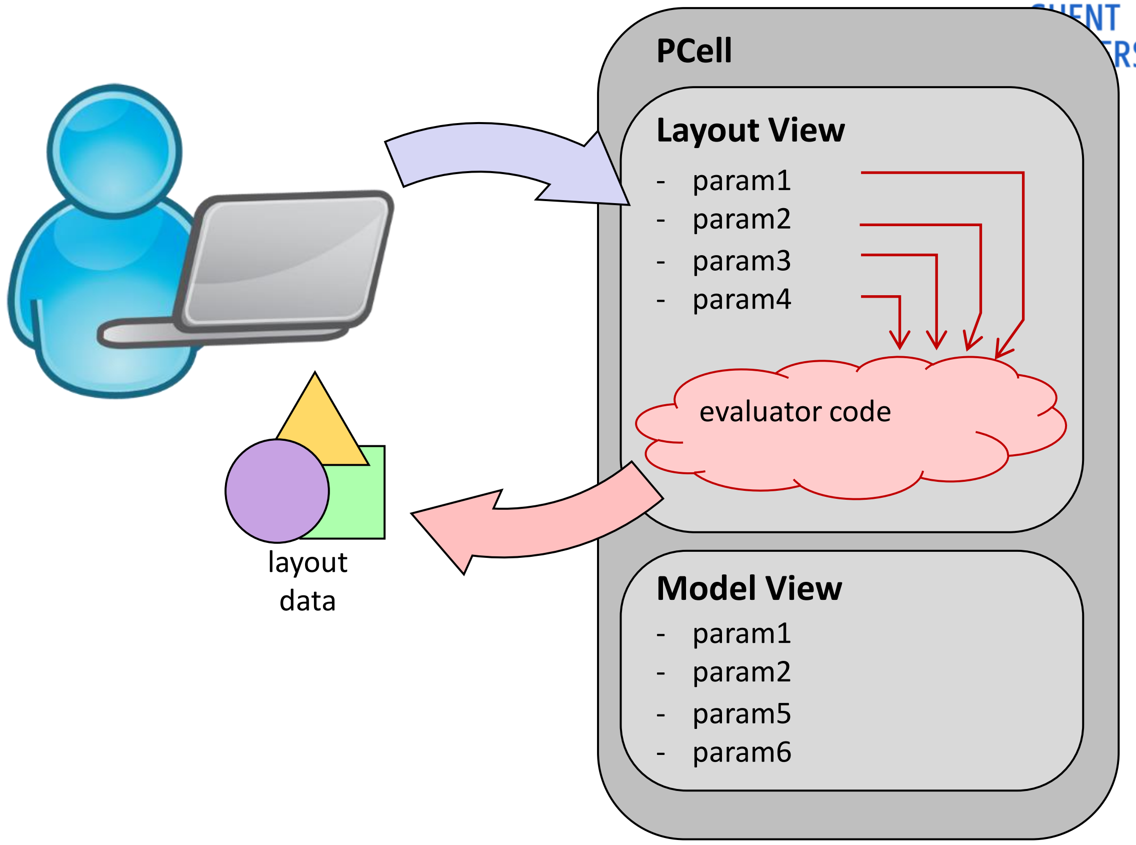
Easy exchange, tool vendor independent

A PARAMETRIC CELL

Same as a cell, but the content is generated based on parameters

Input: user parameters

Output: data

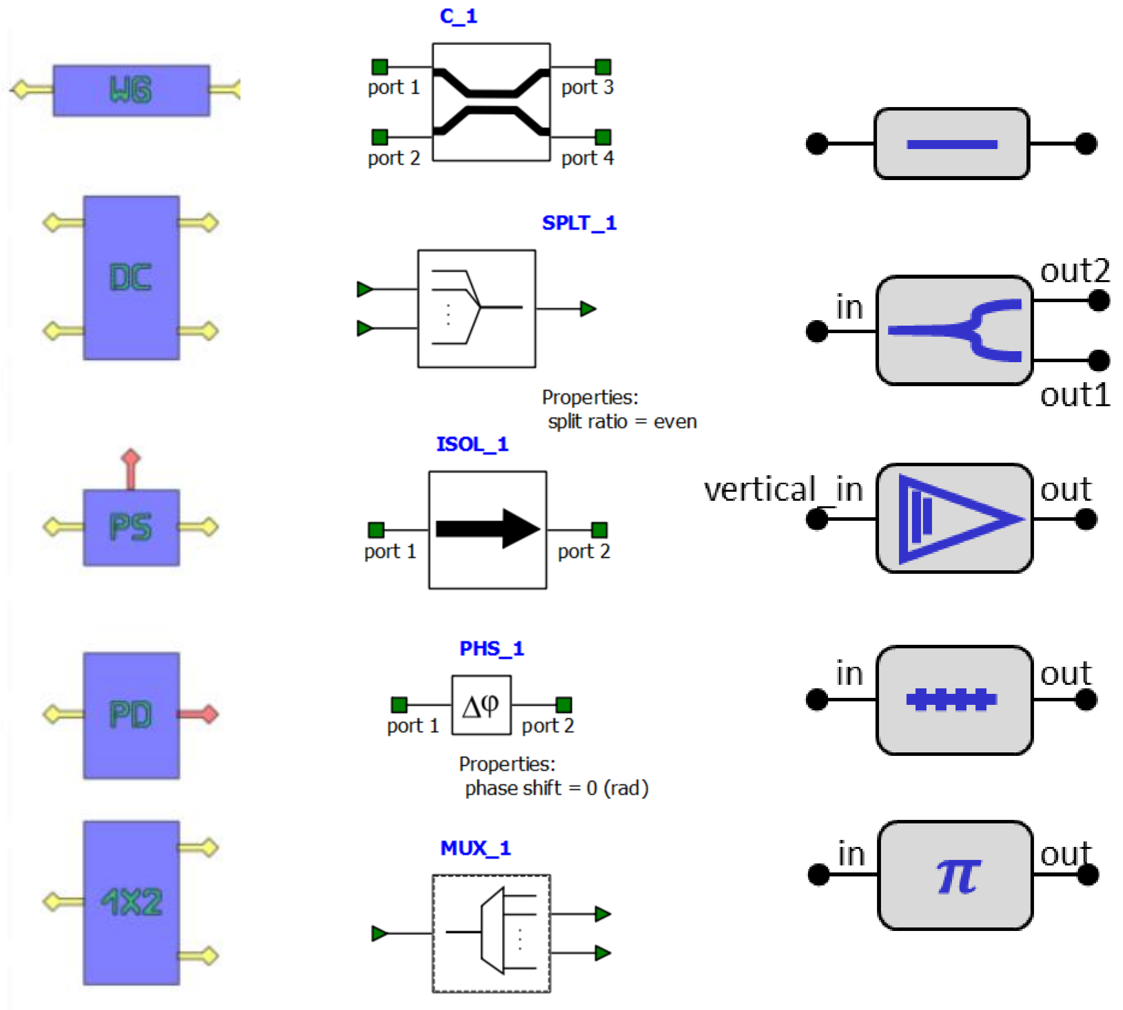


in the middle: an ***evaluator*** function

- a piece of software code
- tool vendor dependent

Storage: in a database

THE SYMBOL VIEW

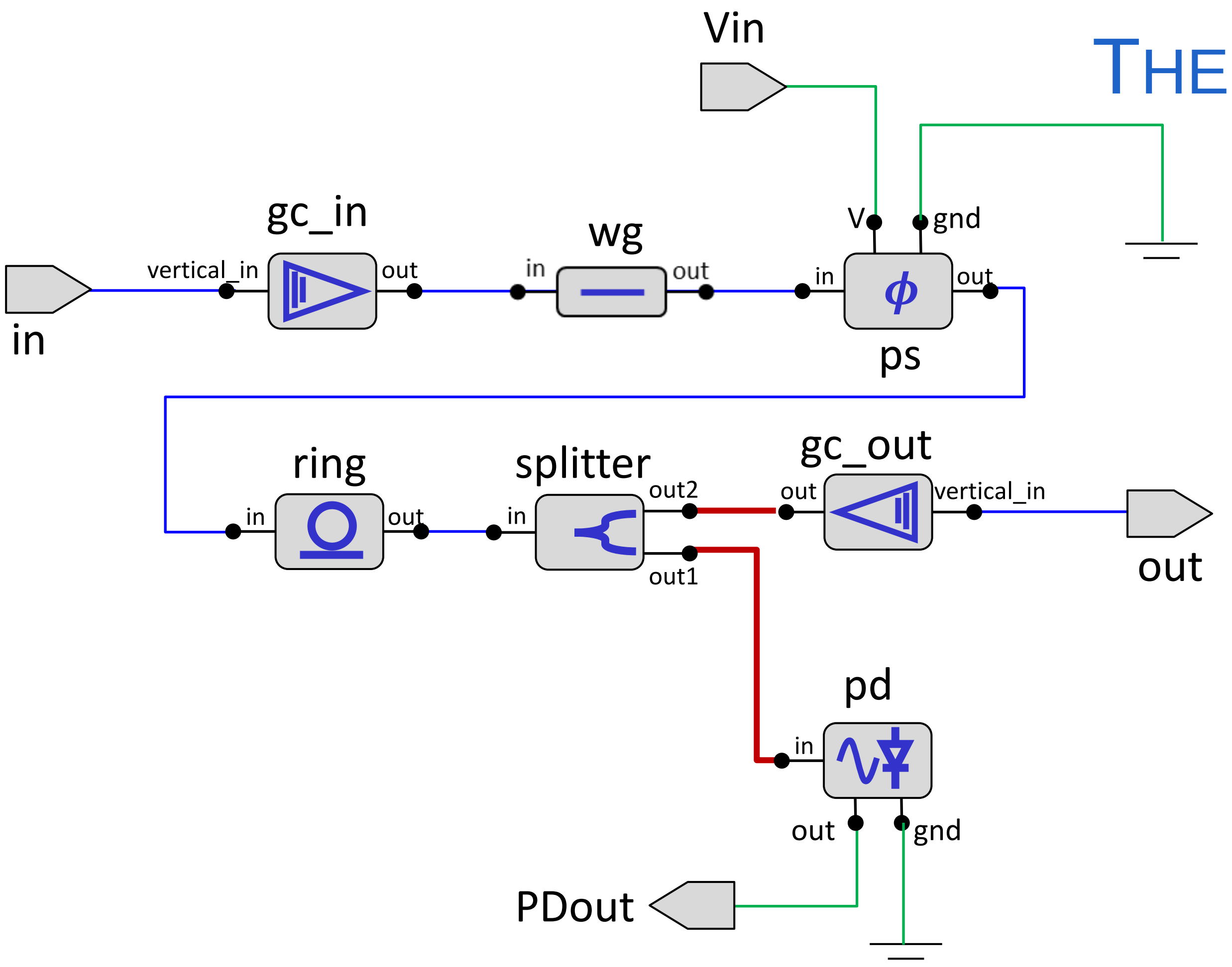


Abstract representation of a component

- Symbolic drawing
- I/O ports/terms (optical/electrical)
- Parameters

There is a standard in electronics (EDIF files) but not between photonics tools.

THE NETLIST/SCHEMATIC VIEW



The netlist describes the internal connectivity of a (sub)circuit

- Circuit elements (instances)

  gc_in, gc_out - grating coupler
  wg - waveguide
  ps - phase shifter
  …

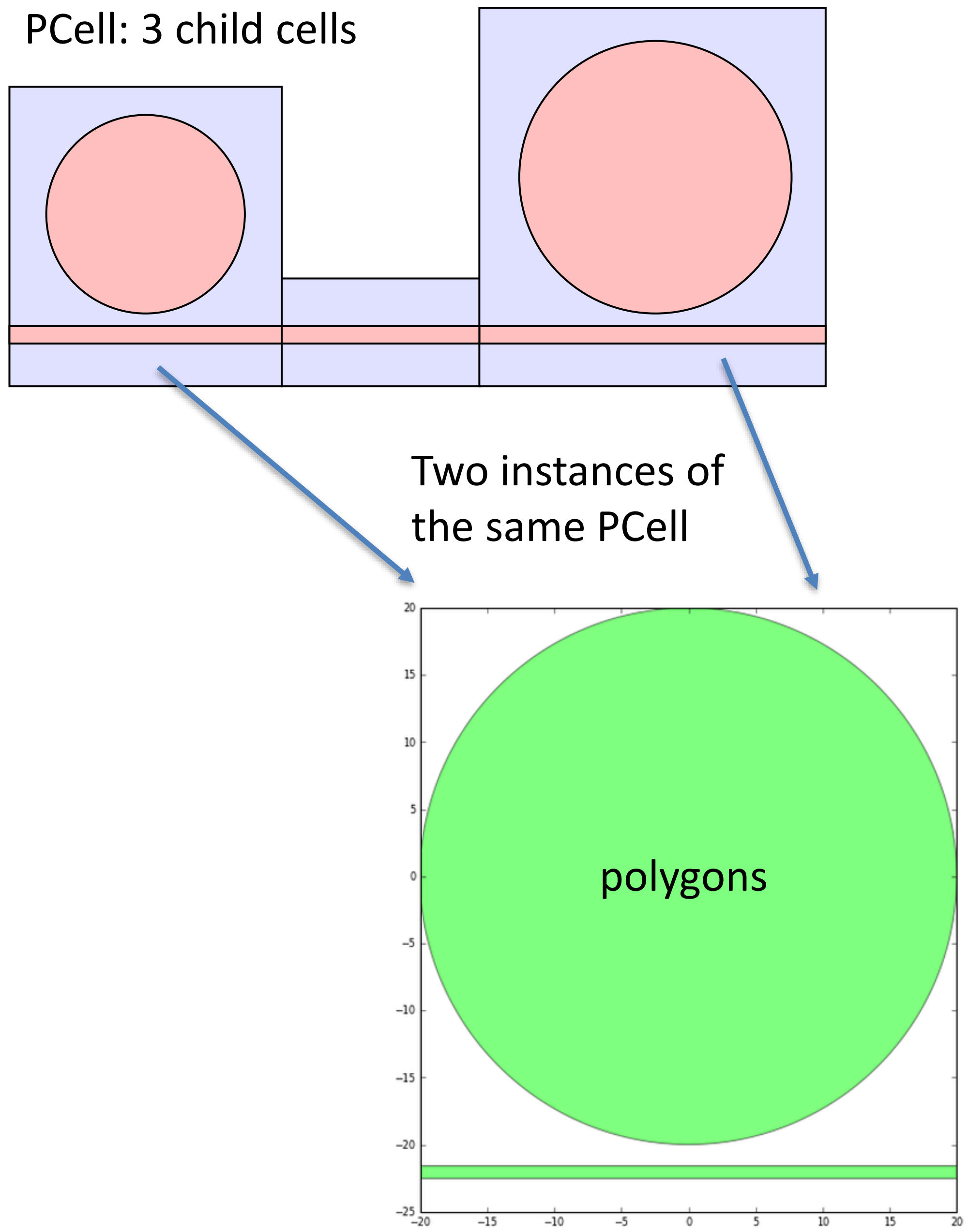
- Connection between ports

  gc_in:out – wg:in
  wg:out – ps:in
  ps:out – ring:in
  ring:out – splitter:in
  …

- Connections with outside world

  gc_in:vertical_in – in
  gc_out:vertical_in – out
  pd:out – PDout
  pg:gnd – GND
  …

THE LAYOUT VIEW



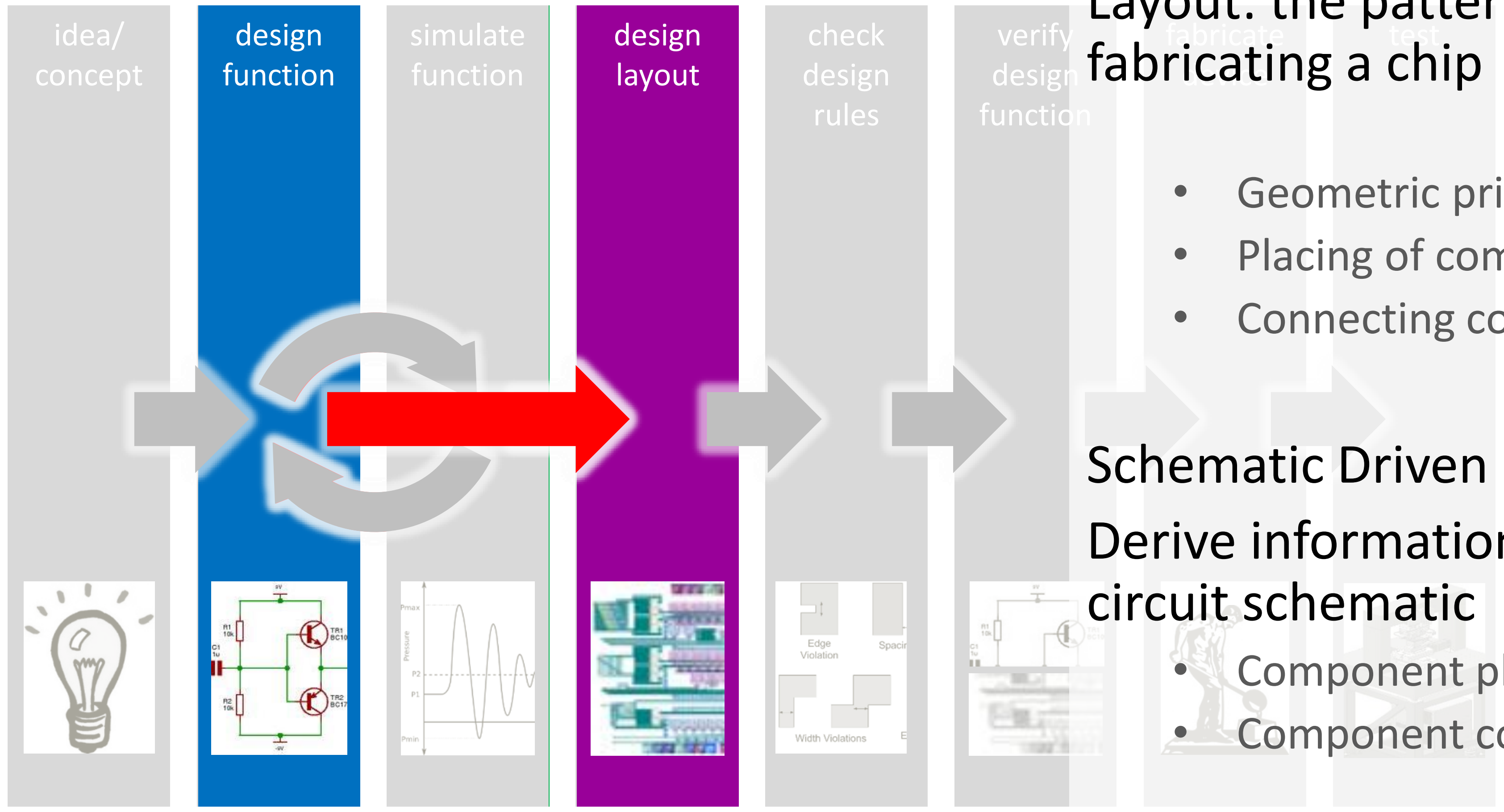
Hierarchical description of polygons on layers

- Raw polygons
- Instances of other cells
    - single
    - array

Here parametrization is used most intensively

- calculate complex shapes
- perform repetitive placements

Schematic Driven Layout (SDL)



Layout: the patterns used for fabricating a chip

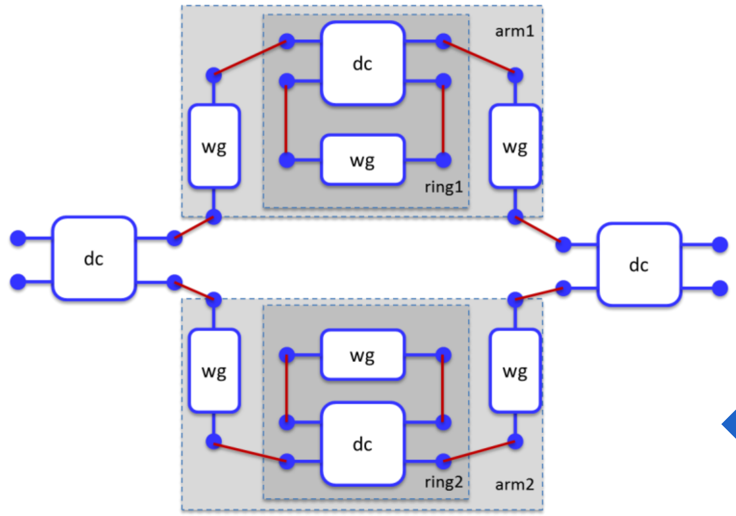
- Geometric primitives
- Placing of components
- Connecting component

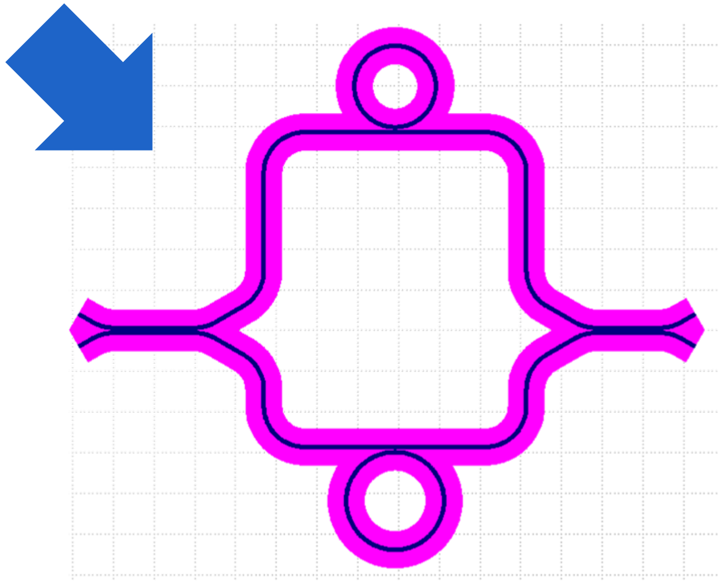
Schematic Driven Layout:

Derive information from circuit schematic

- Component placement
- Component connectivity

SCHEMATIC DRIVEN LAYOUT (SDL)





Derive the physical layout from the schematic

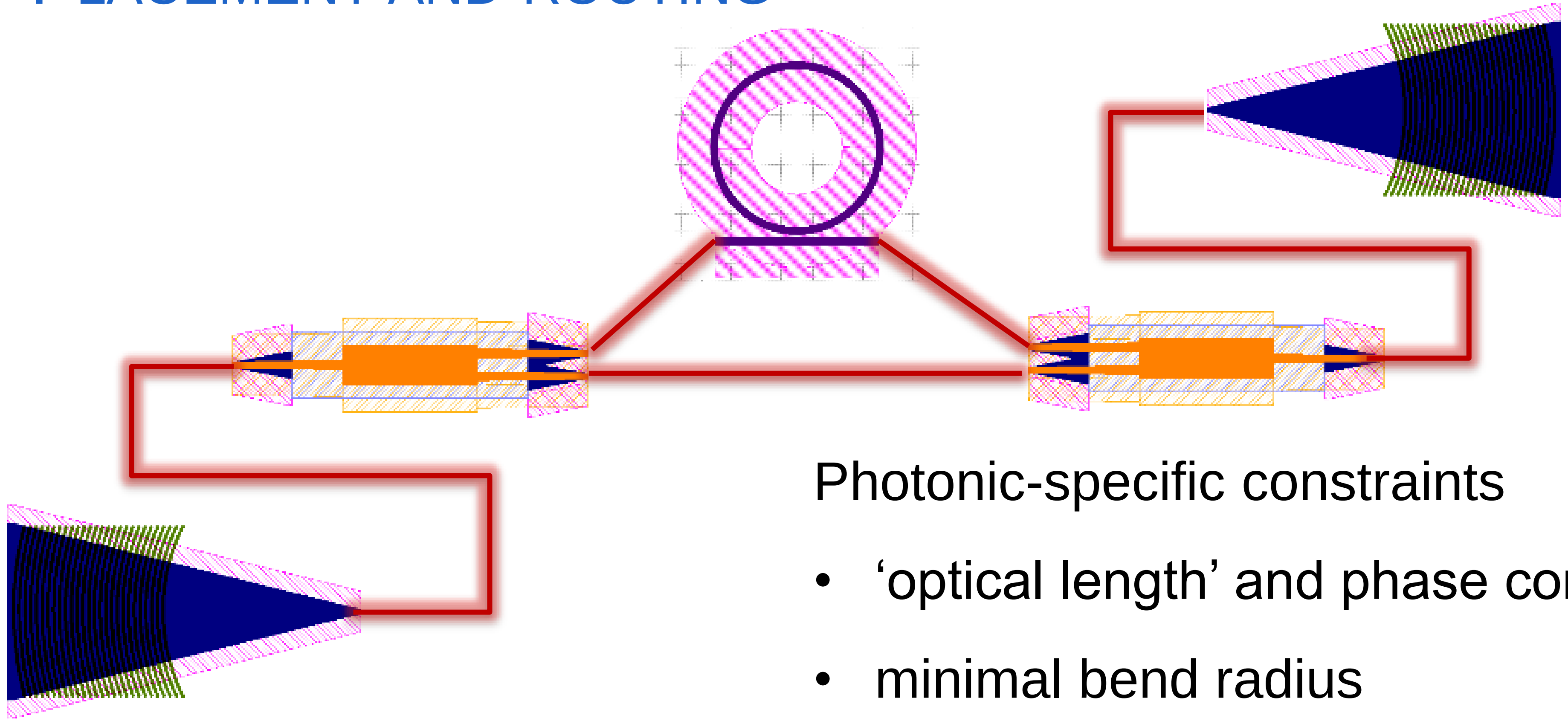
- Generate the Layout (P)Cells
- Place the Layout Cells
- Connect the layout cells together

Not trivial to fully automate

- What is the optimal placement?
- Is the topology possible?
- Constraints for length matching?
- On which layer to route?
- Waveguide bends and crossings?

Combination of manual + auto

PLACEMENT AND ROUTING



Photonic-specific constraints

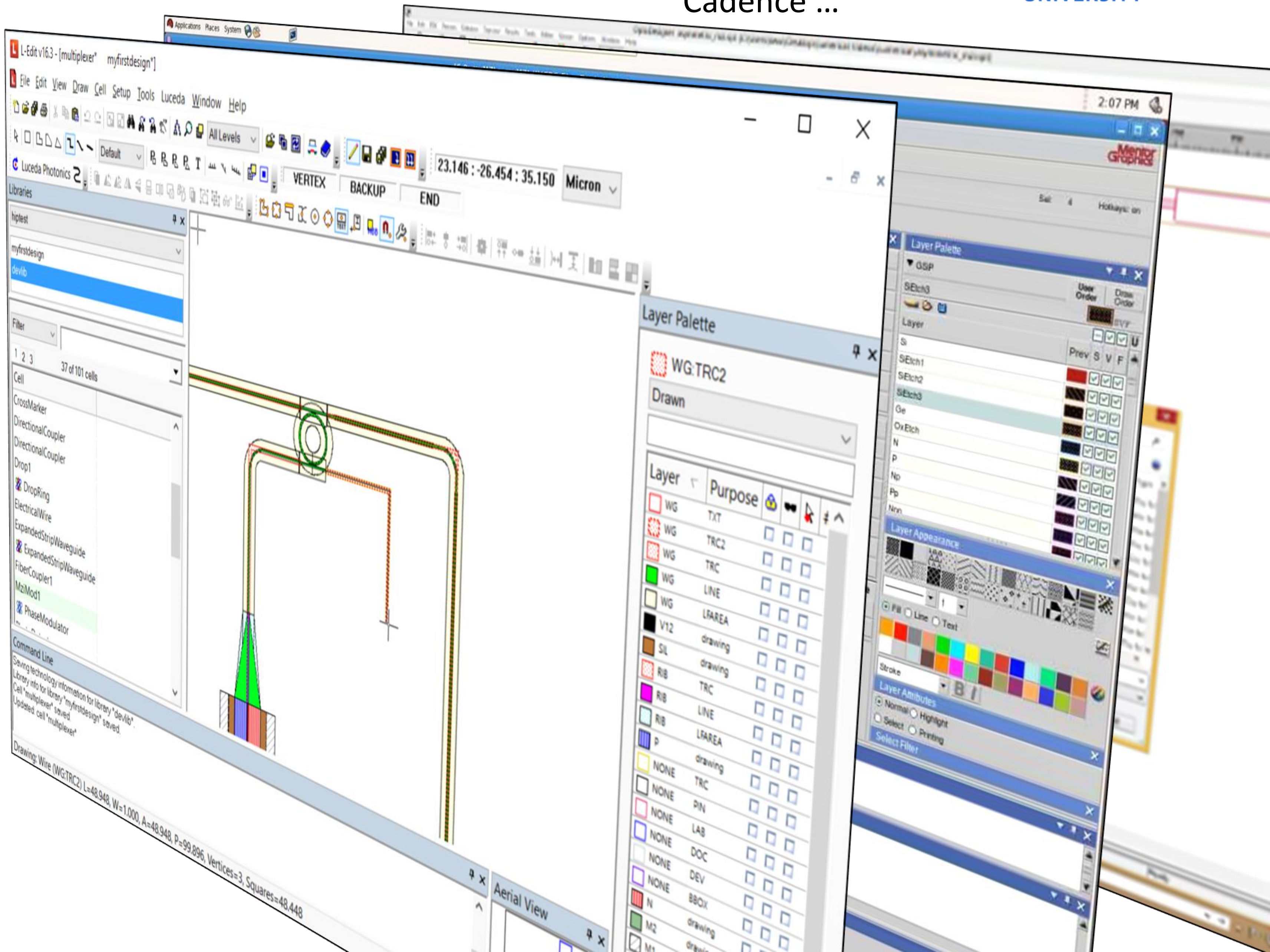
- ‘optical length’ and phase control
- minimal bend radius
- waveguide spacing
- matching port direction
- single routing layer!

Photonic SDL tools are emerging

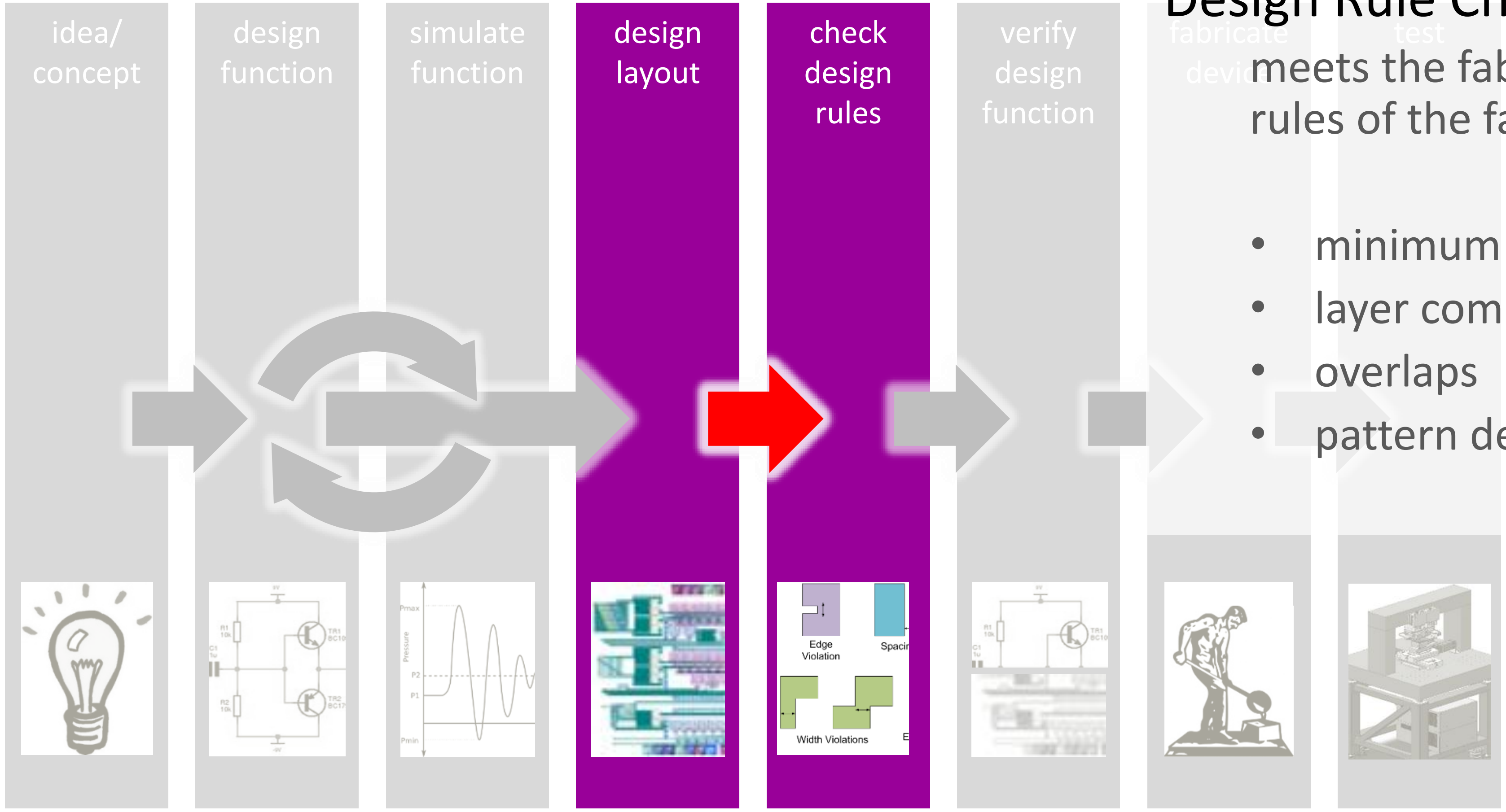
Luceda, Synopsys,
Mentor Graphics,
Cadence ...

Pure photonics
or based on EDA tools

- define connections
- place components
- route waveguides



Is the layout valid?



Design Rule Checking

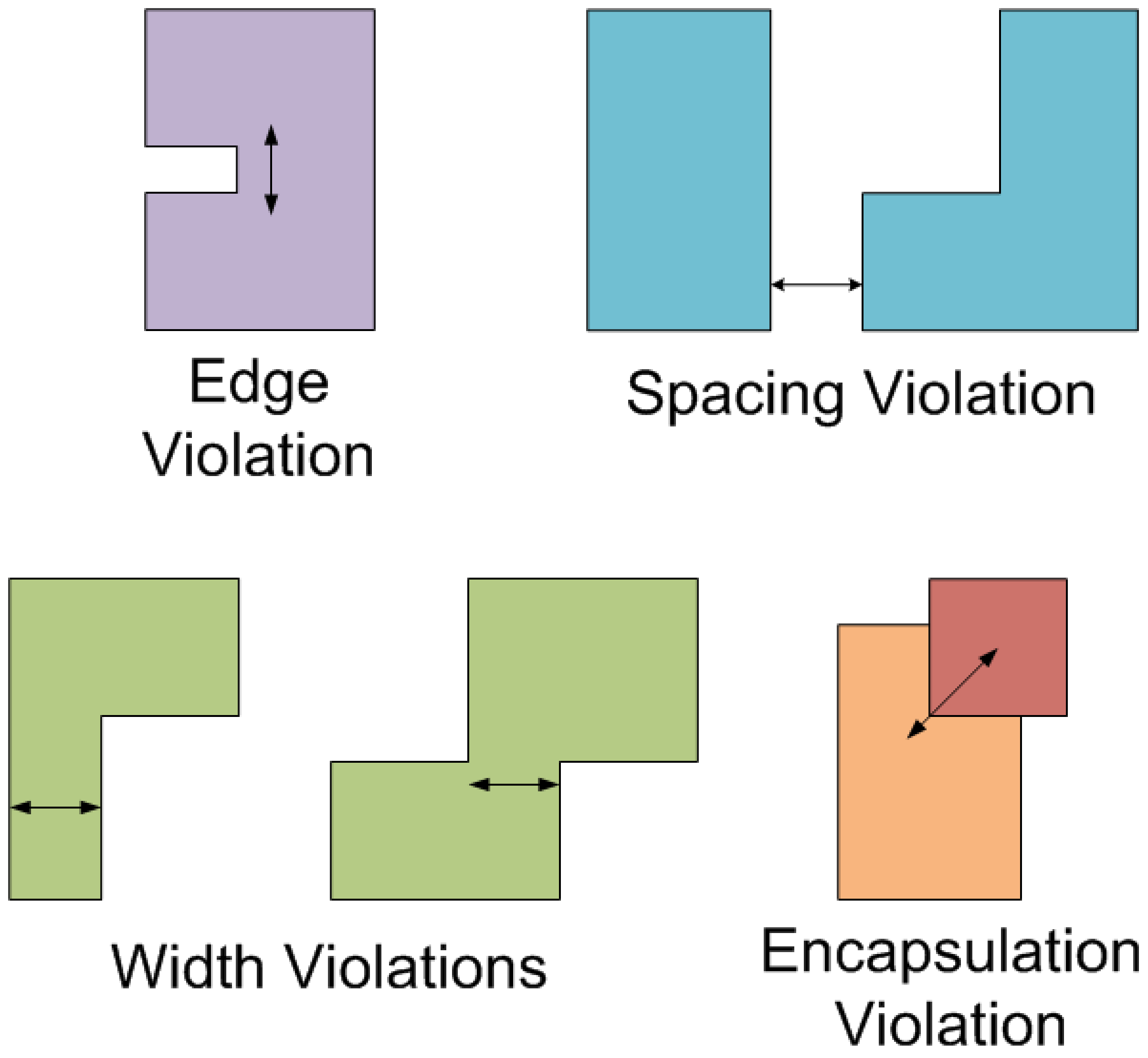
meets the fabrication rules of the fab?

- minimum features
- layer combinations
- overlaps
- pattern density

design flow

time

Design Rule Violations: Examples

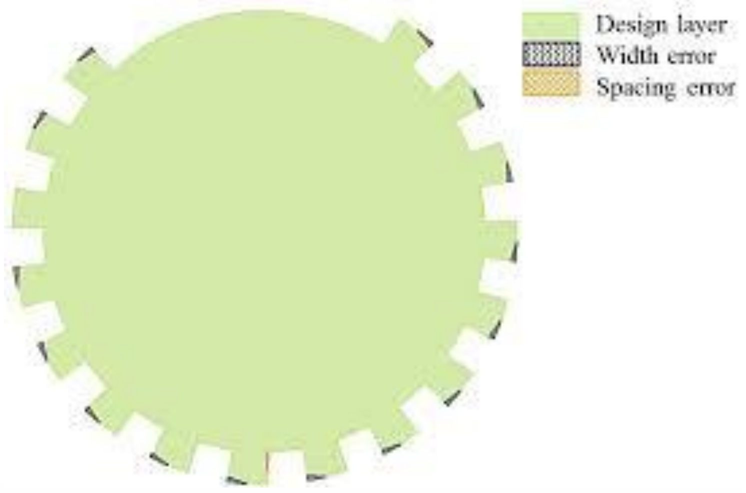


Photonic problems with DRC?

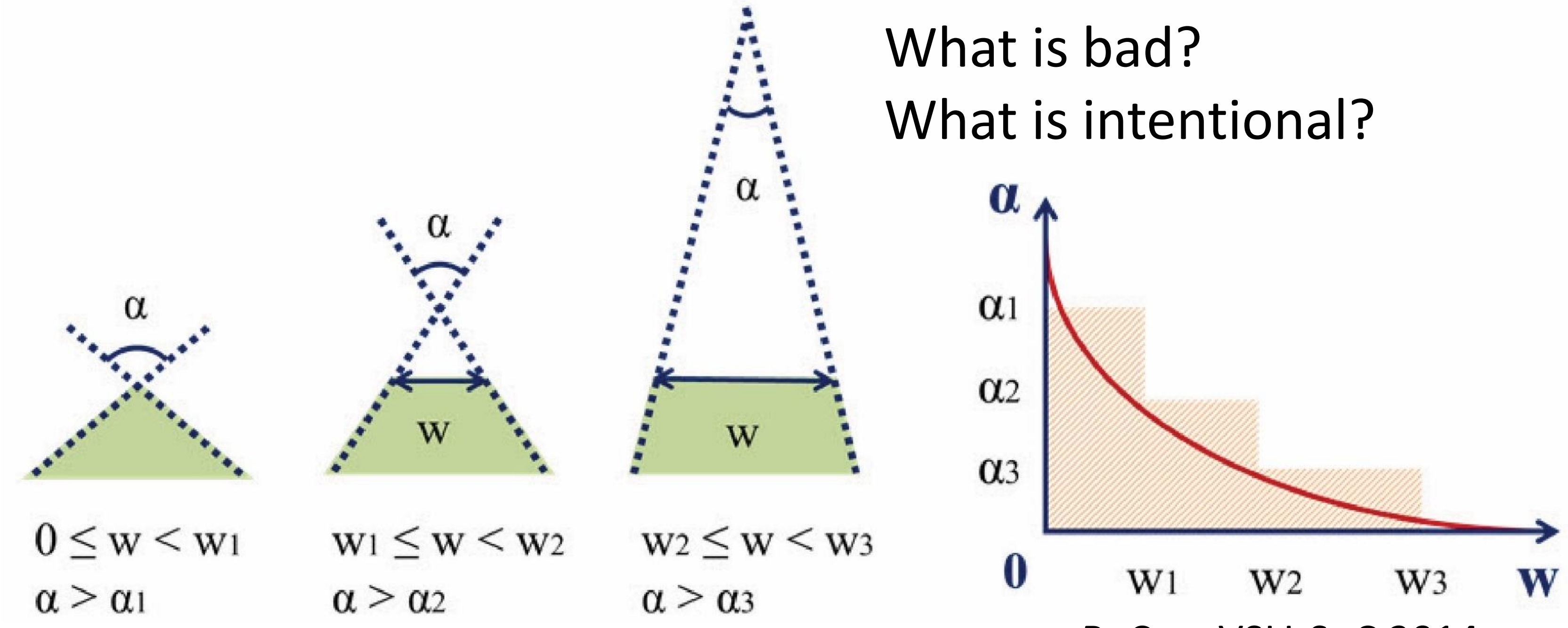
DRC techniques were designed for electronics: 90-degree angles…

Silicon Photonics:

- All-angle waveguides – discretized…
- Nanometer scale sensitivities
- Arbitrary geometries (e.g. slot waveguides, PhC)

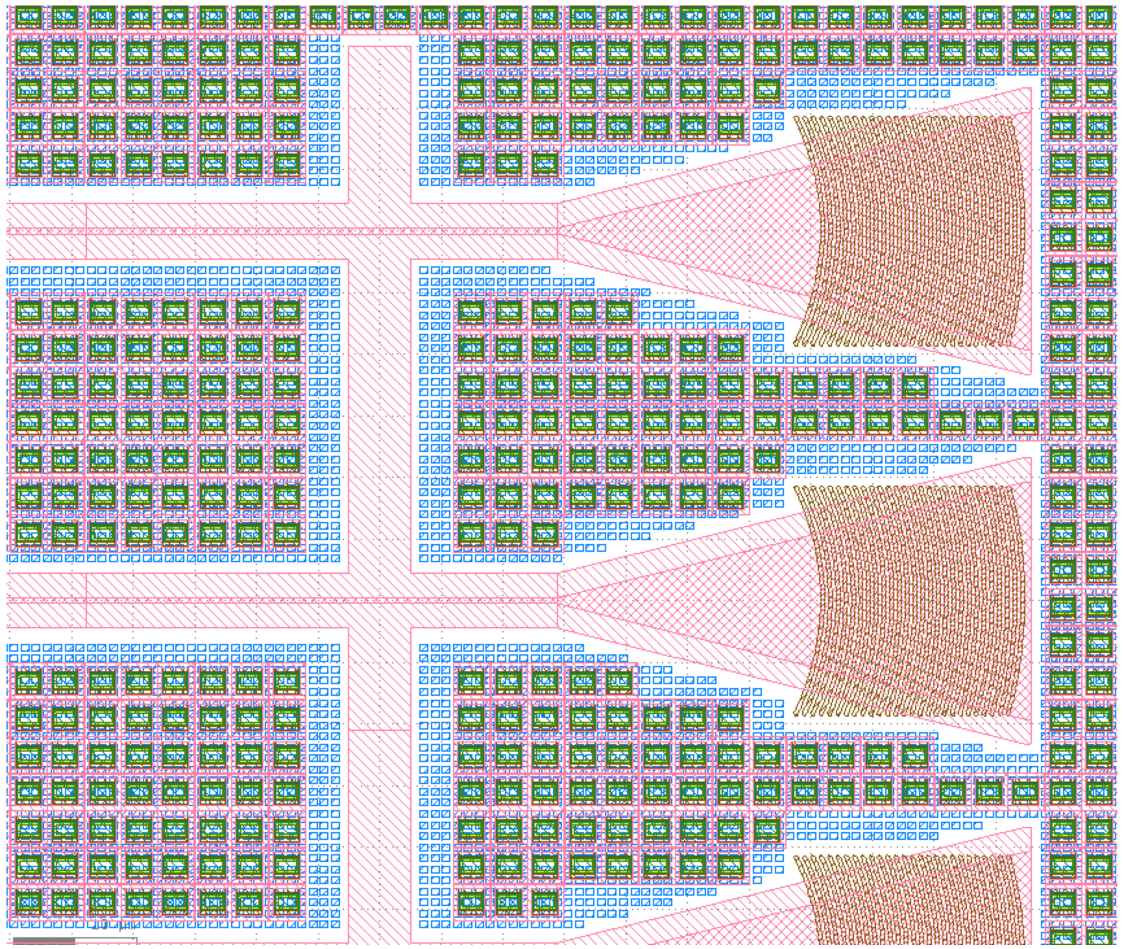


What is bad?
What is intentional?



R. Cao, VSLI-SoC 2014

PATTERN DENSITIES



Pattern density must be sufficiently uniform

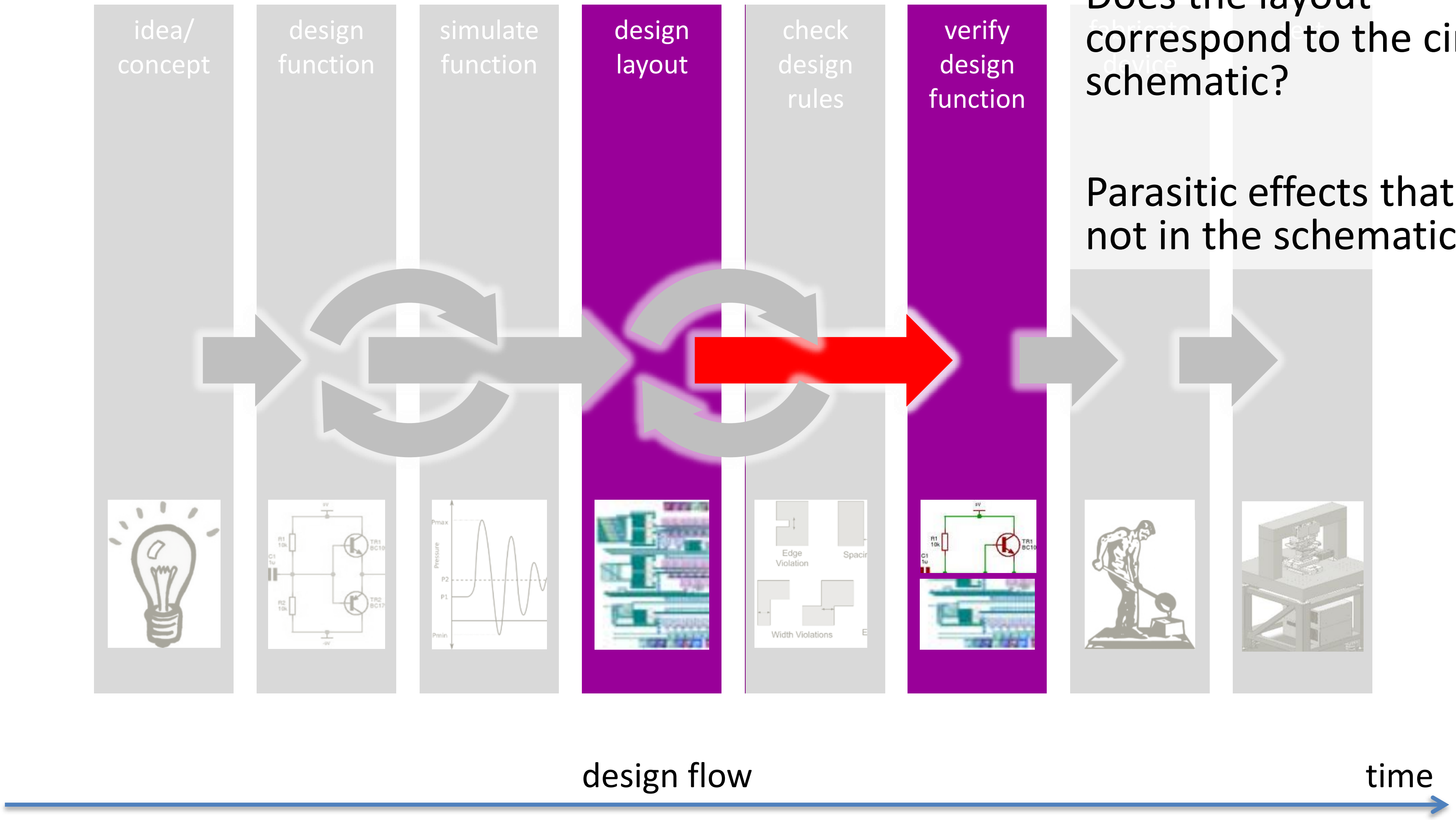
- Etch rate control
- Avoid CMP dishing

Tiles are added

There must be sufficient room to add tiles

- Slab areas (AWG)
- Dense waveguide arrays
- …

FUNCTIONAL VERIFICATION



Does the layout correspond to the circuit schematic?

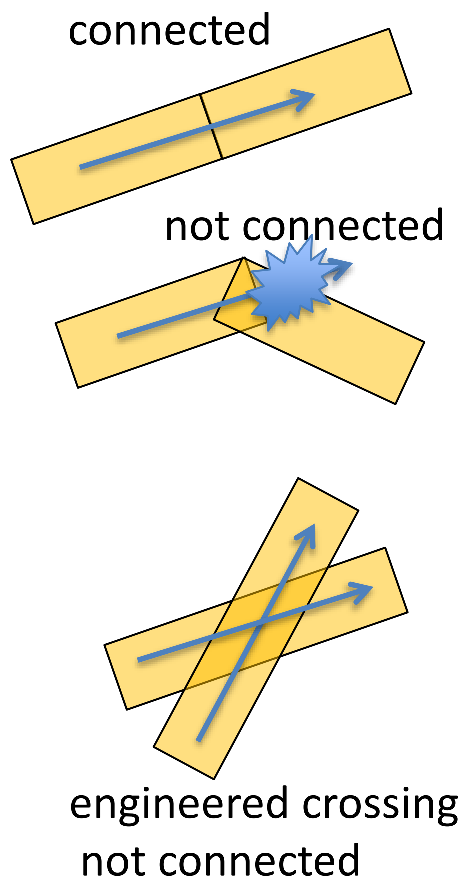
Parasitic effects that were not in the schematic

FUNCTIONAL VERIFICATION: LAYOUT VERSUS SCHEMATIC

Check Connectivity

Are the correct components placed?

Are they properly connected?

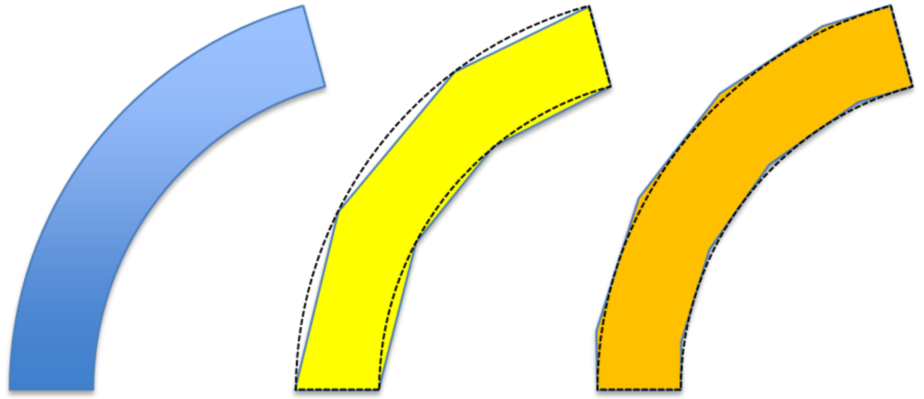


Check functionality

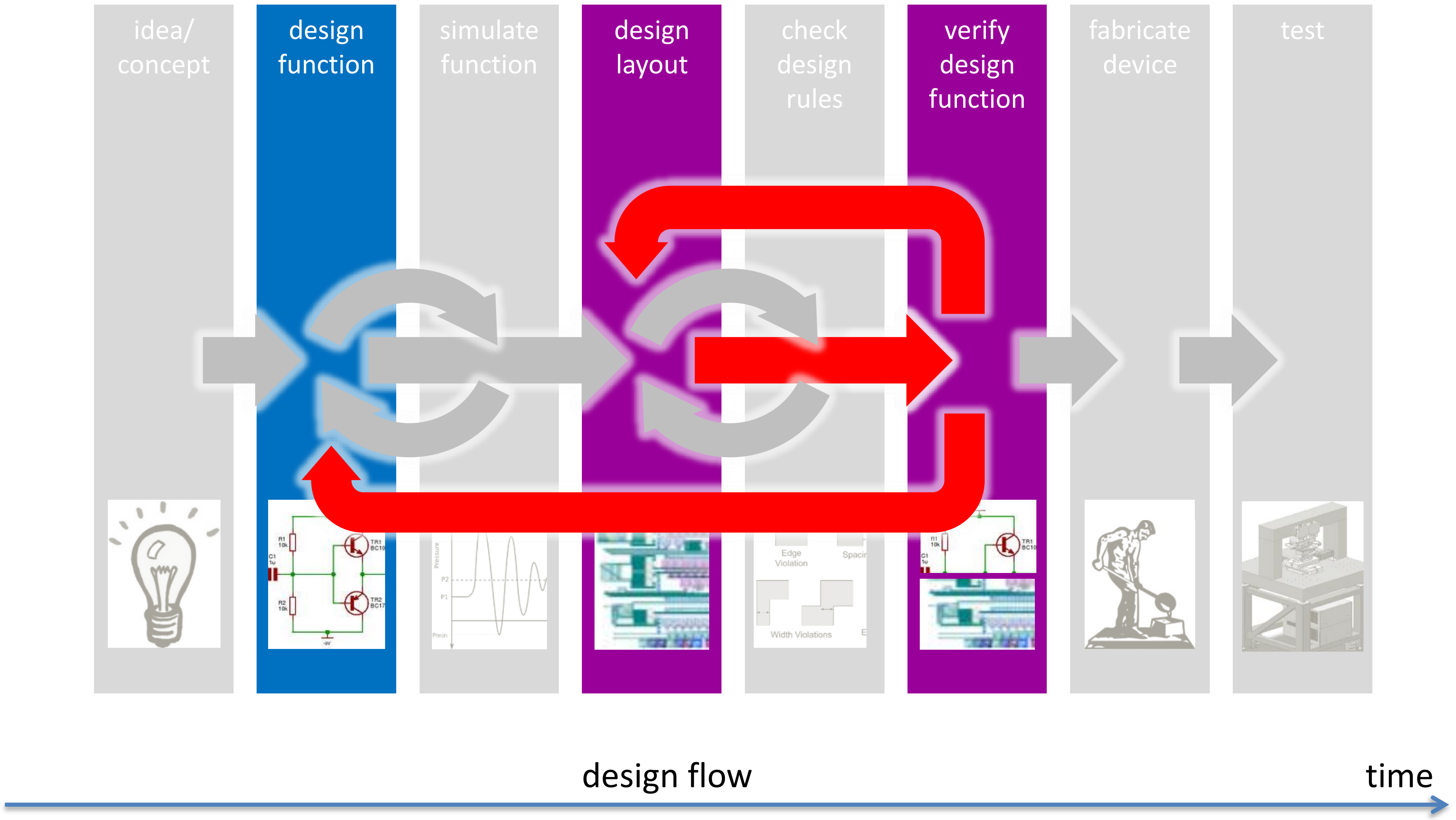
Did we use the right parameters?

Does the layout perform the correct function?

e.g. does the waveguide have the correct width (i.e. optical length)



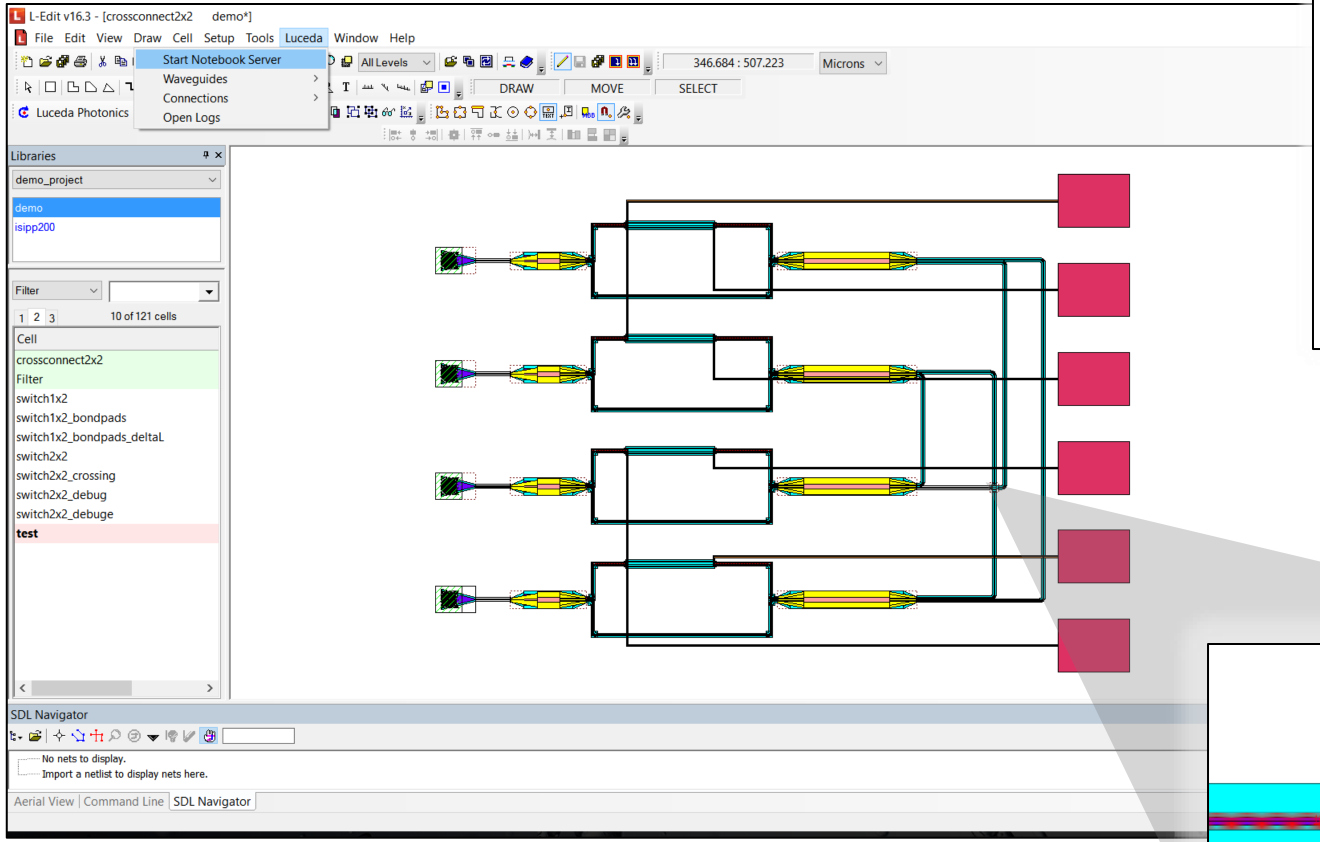
Functional verification

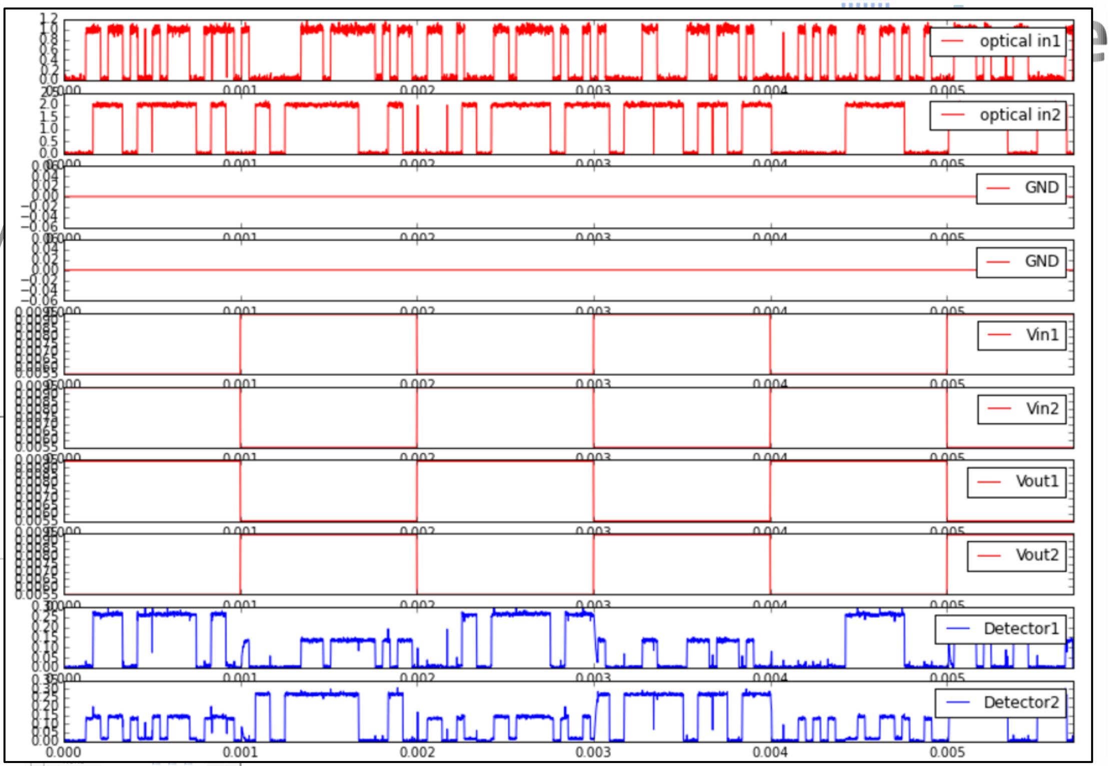


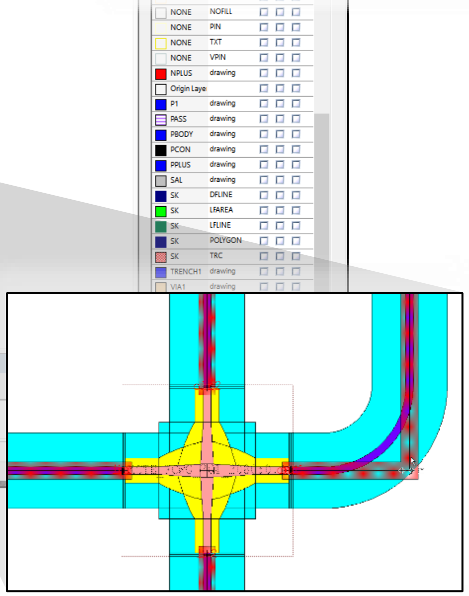
POST-LAYOUT SIMULATION

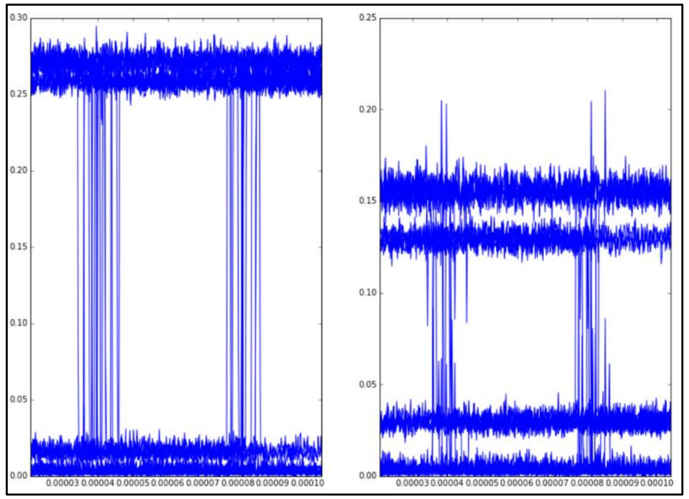
Resimulate the circuit based on the actual lay

Include lengths, crossings, reflections, …





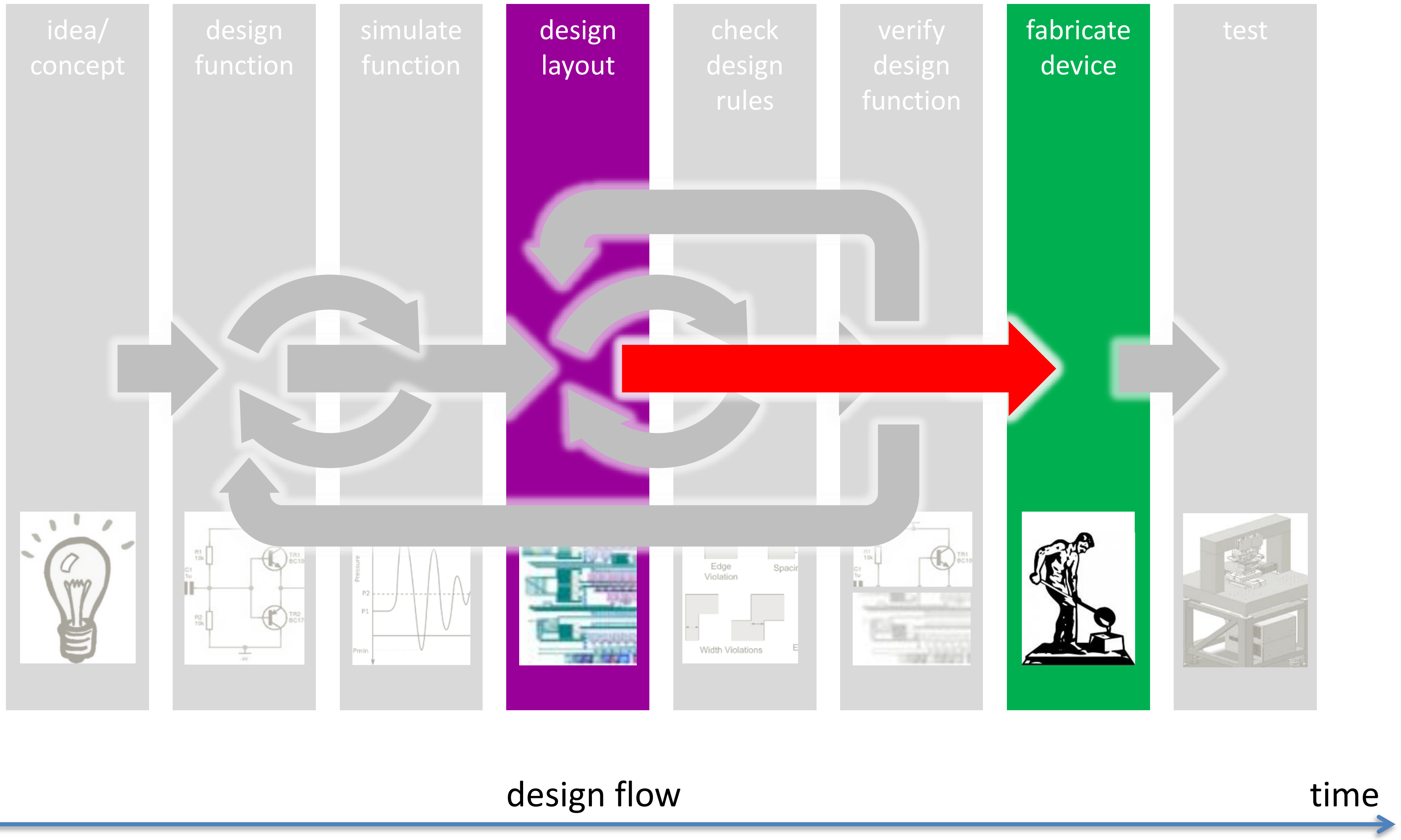




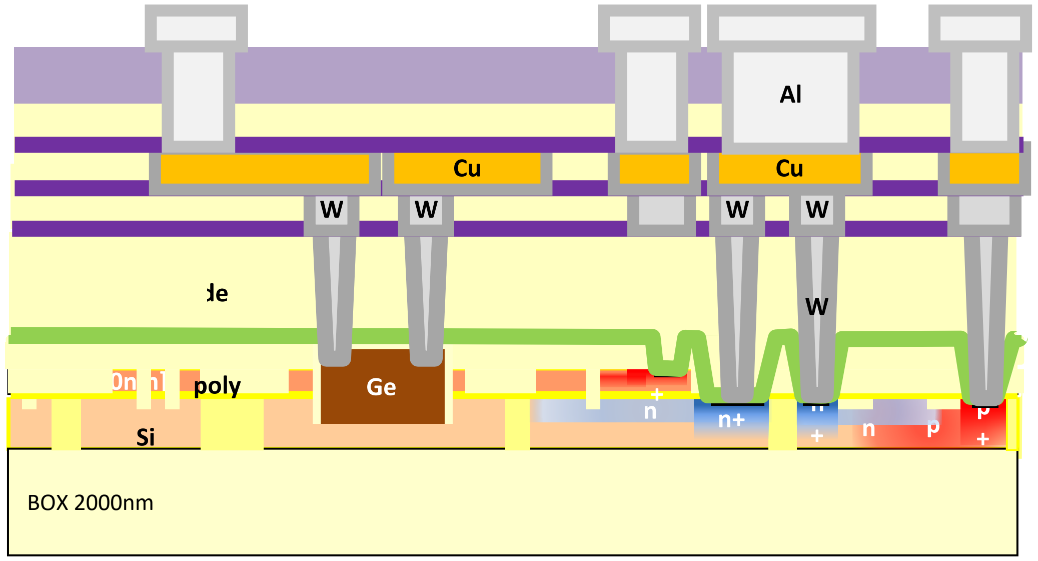
LUCEDA

FABRICATION

"no plan survives contact with the enemy"
H. von Moltke (misquoted)



THE ACTUAL FABRICATION PROCESS



example: IMEC silicon Photonics

Layer depositions

Pattern definition (lithography)

Pattern transfer (etch)

Planarization

Thermal treatment

Doping and implantation

…

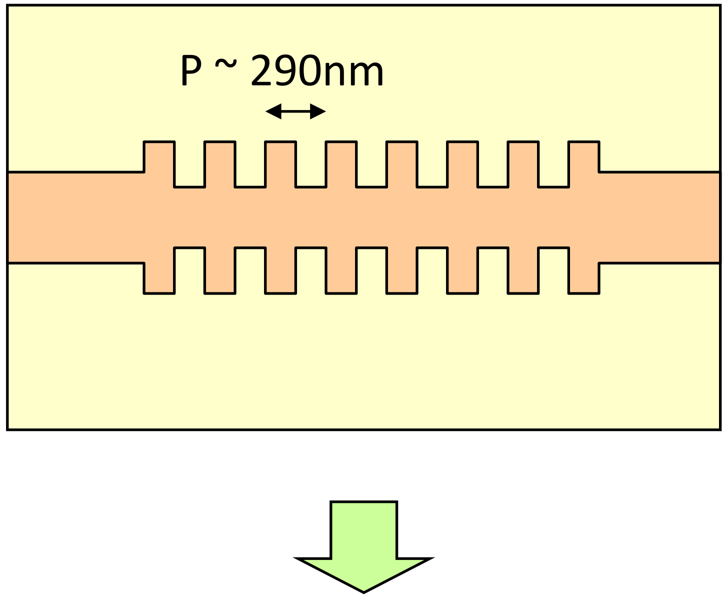
and each step with imperfections and variability

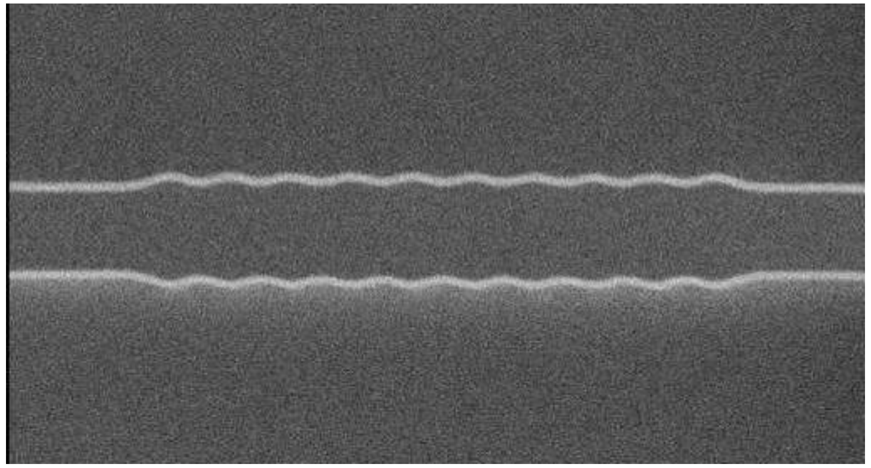
LITHOGRAPHY: NOT PERFECT

Spatial low-pass filter

- Minimum feature size
- Minimum pitch
- Pattern rounding

Example: Bragg grating





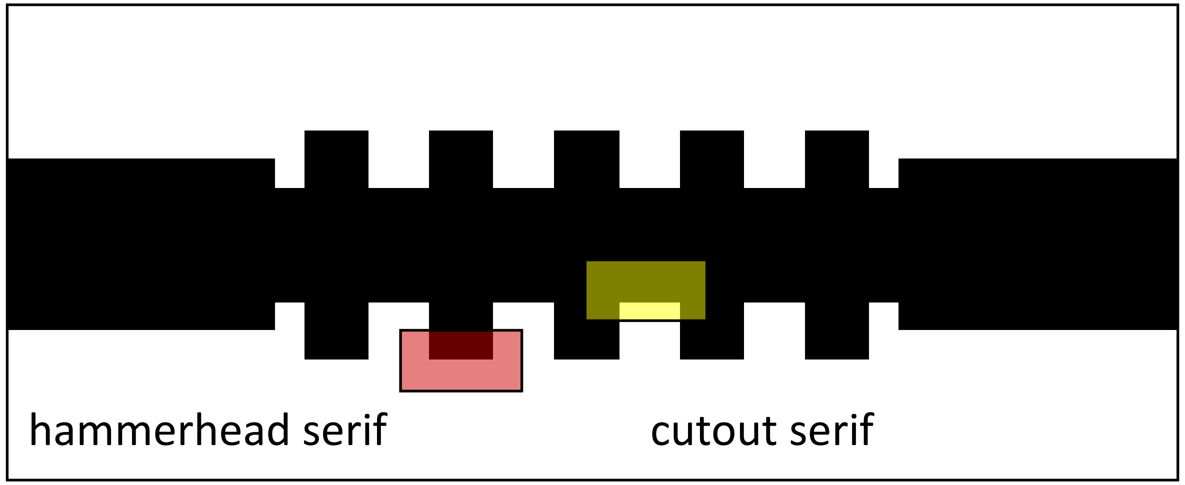
OPTICAL PROXIMITY CORRECTIONS (OPC)

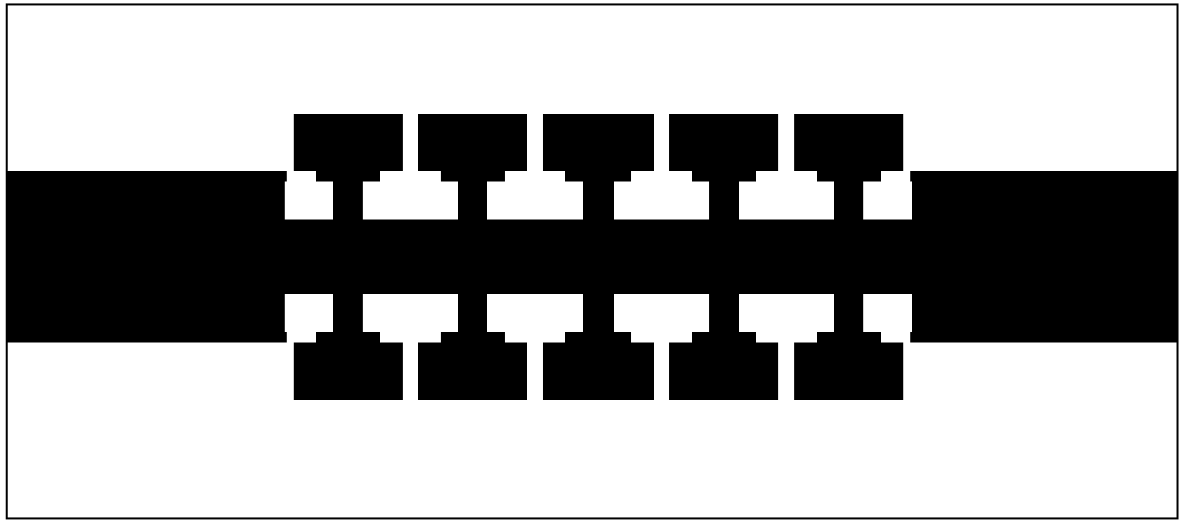
Overcome rounding: add OPC

- — serifs
- — cutouts

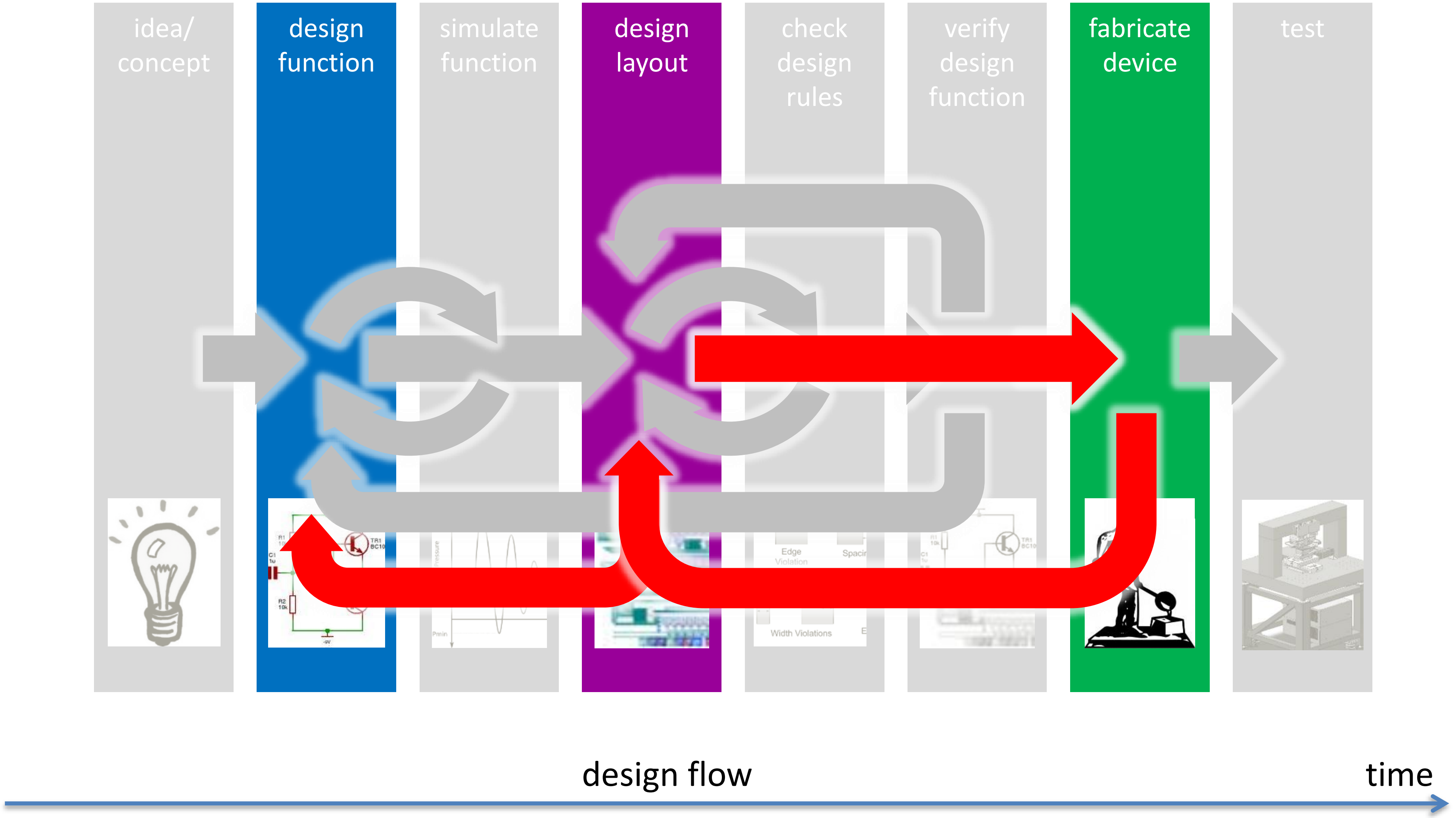
Makes mask more complex (and costly)

Not always possible without violating DR

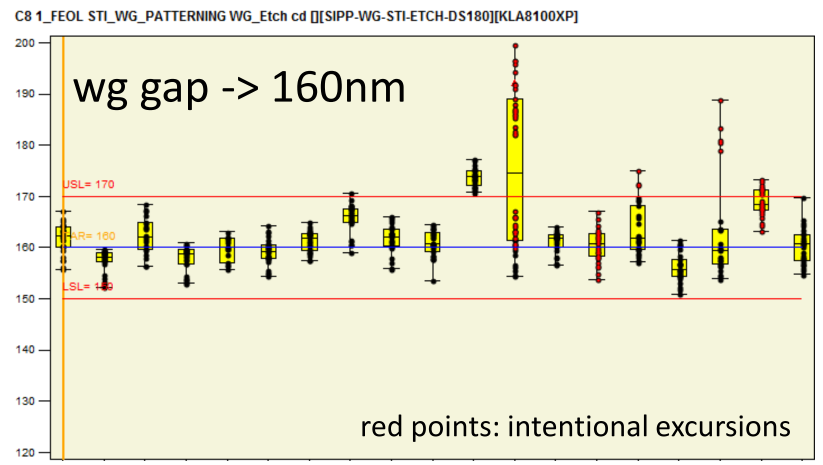


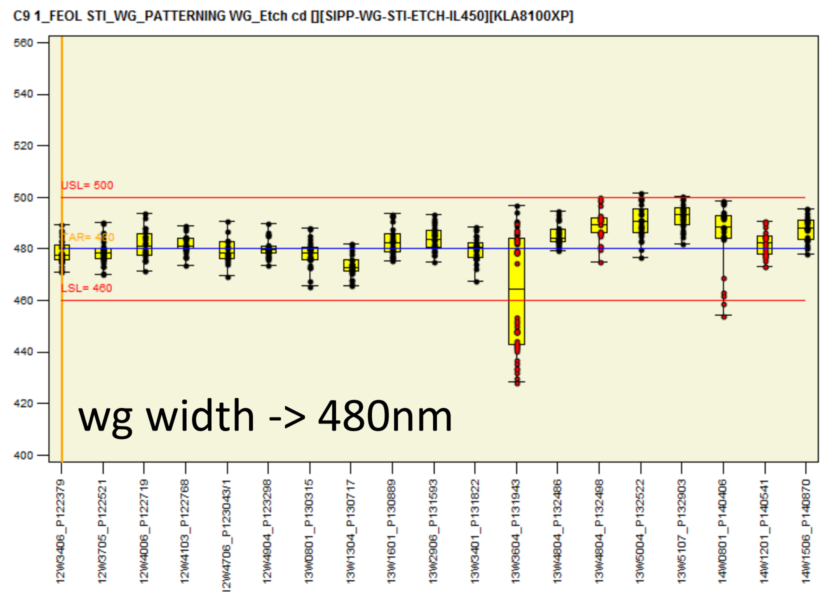


Fabrication: in-line data



IN-LINE PROCESS DATA





Collect data from wafers as they are being processed

- Line width
- Etch depth
- Layer thickness
- …

Feed in design process

- FRONT-END: Predict behavioural change
- BACK-END: Adjust layout

STATISTICS!

THERE ARE MANY SOURCES OF NON-UNIFORMITY

Reticle/Mask
- CD uniformity
- Flatness
- Transmission

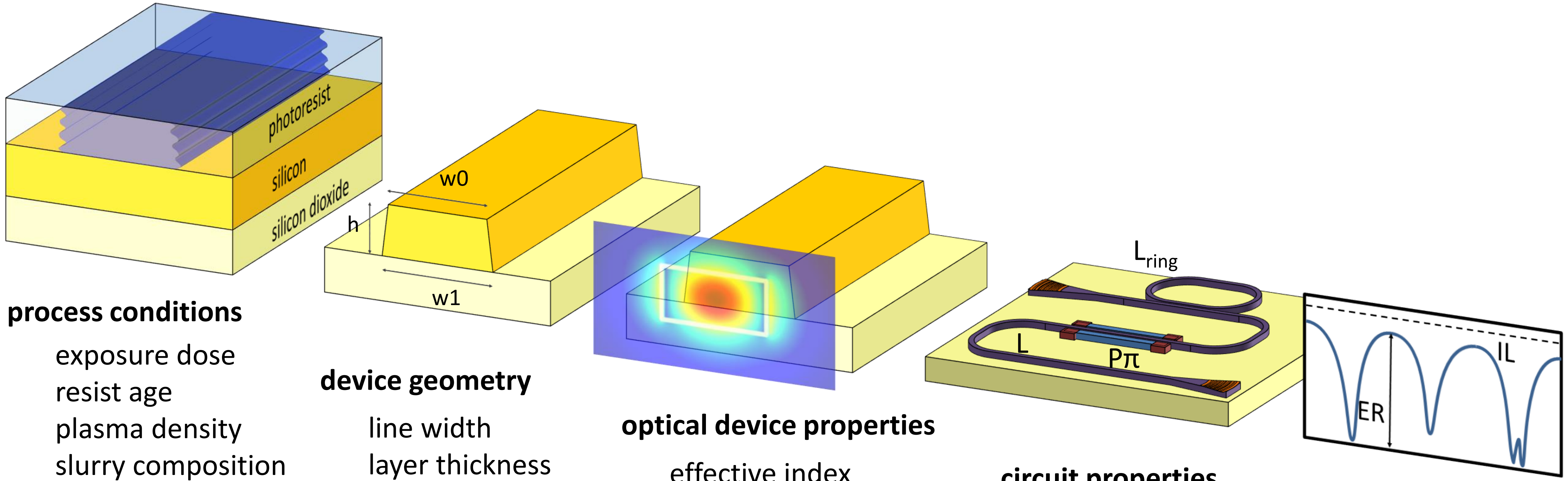
Litho tool
- Exposure dose
- Slit uniformity
- Chuck flatness
- Focus stability
- Scan direction
- Source spectrum

Resist process
- BARC uniformity
- Resist uniformity
- PEB °C Uniformity
- Developer
- Metrology

Wafer
- Wafer flatness
- Stack uniformity
- Topography

Etch process
- Plasma Chemistry
- Coil power stability
- Bias stability
- Resist coverage
- °C stability
- Metrology

DESCRIBING VARIABILITY AT DIFFERENT LEVELS



process conditions

- exposure dose
- resist age
- plasma density
- slurry composition
- …

device geometry

- line width
- layer thickness
- sidewall angle
- doping profile
- …

optical device properties

- effective index
- group index
- coupling coefficients
- center wavelength
- …

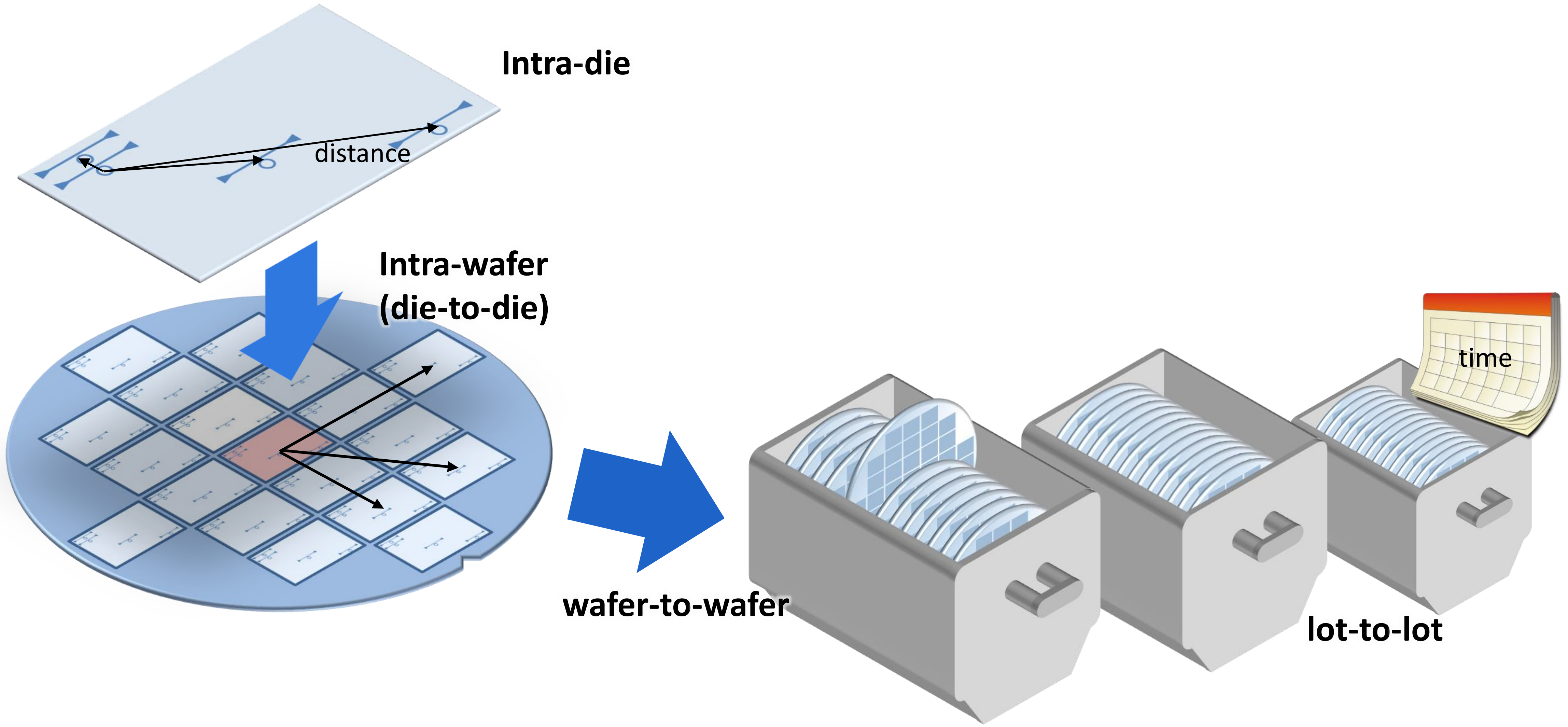
circuit properties

- optical delay
- path imbalance
- tuning curve
- …

system performance

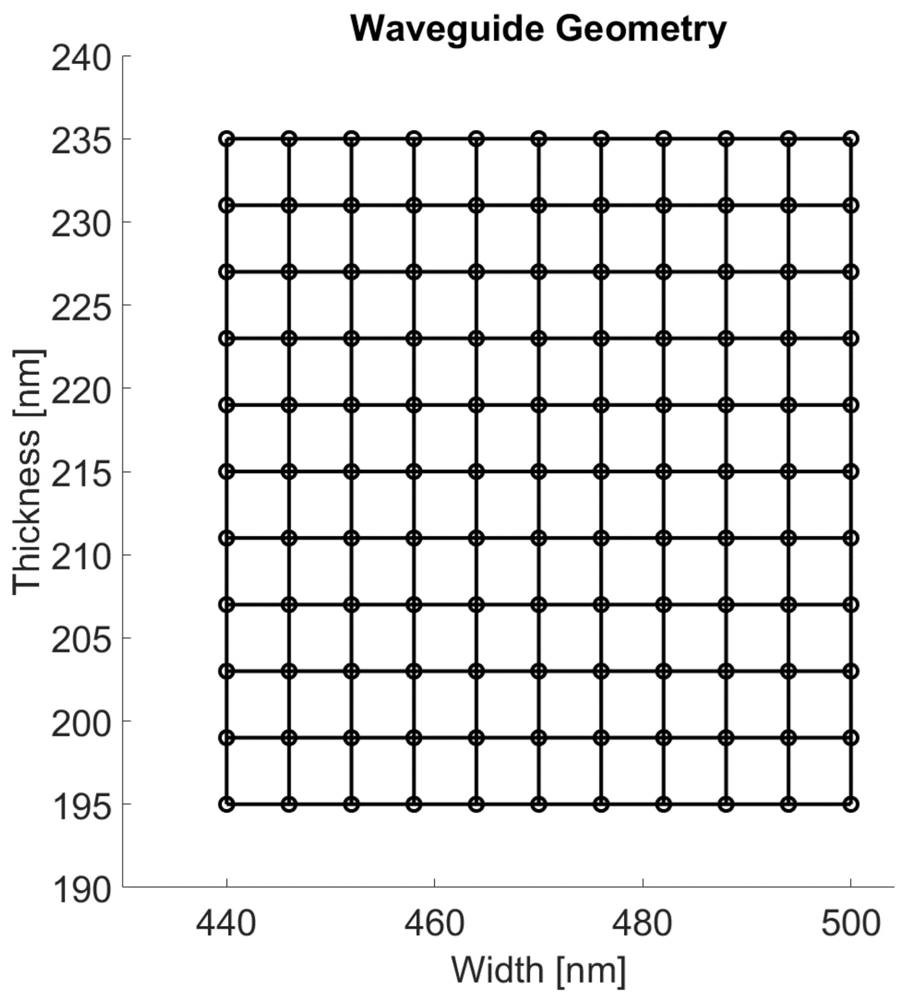
- insertion loss
- crosstalk
- noise figures
- power consumption
- …

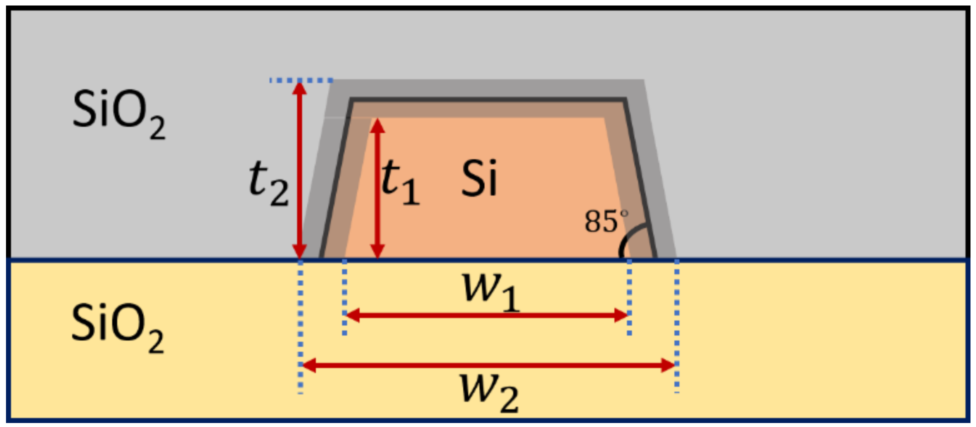
VARIABILITY EFFECTS WORK ON DIFFERENT SCALES



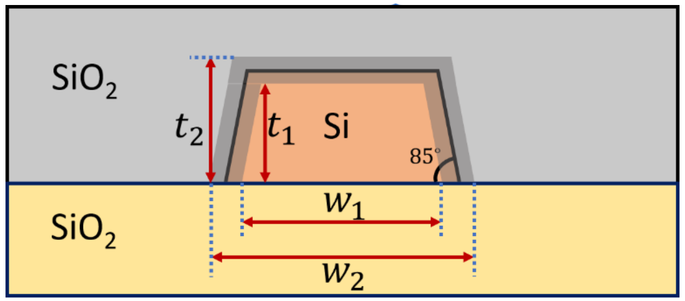
Mapping Geometry on Optical Properties

- width/thickness
- effective/group index

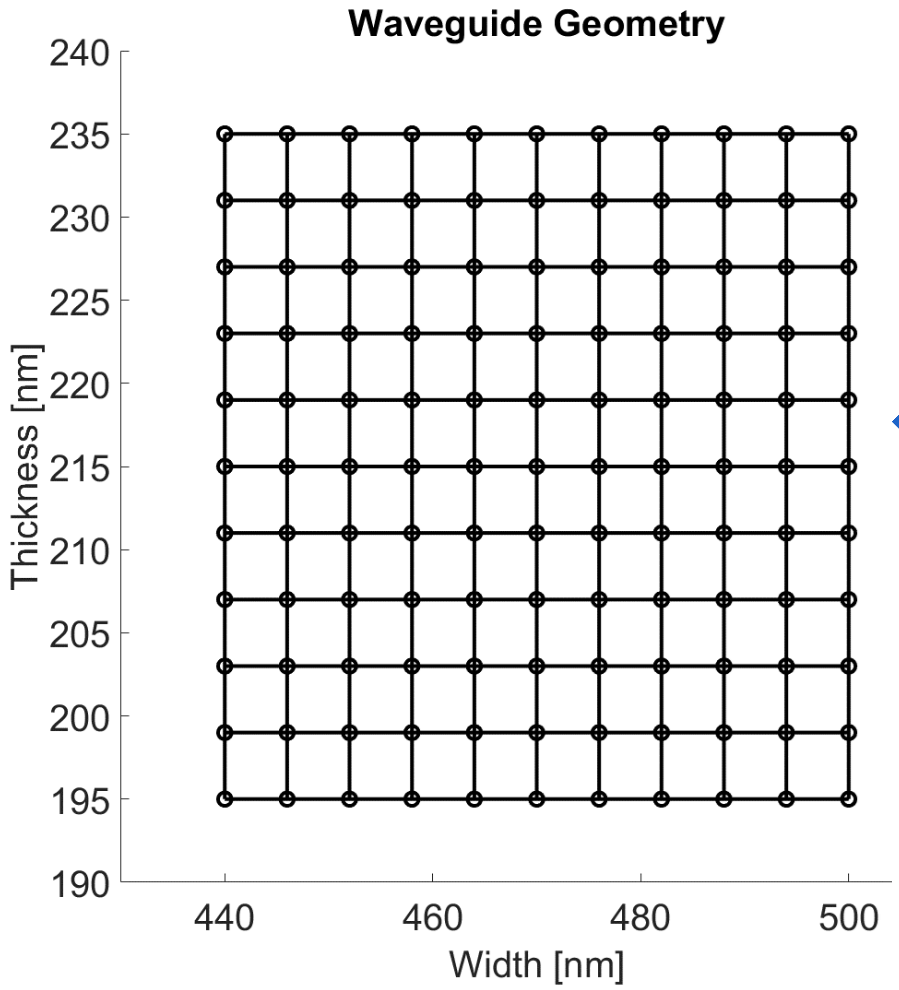


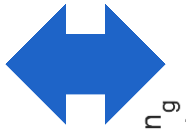


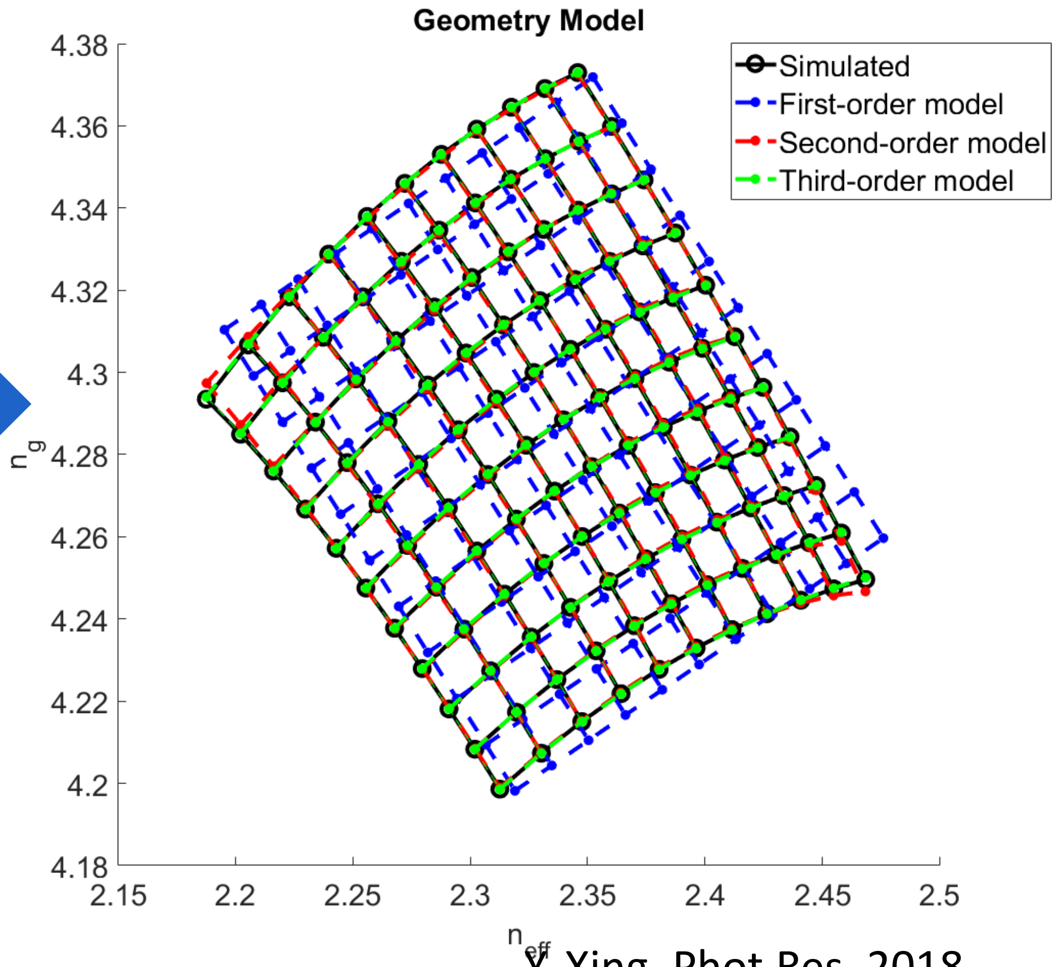
MAPPING GEOMETRY ON OPTICAL PROPERTIES



- width/thickness
- effective/group index

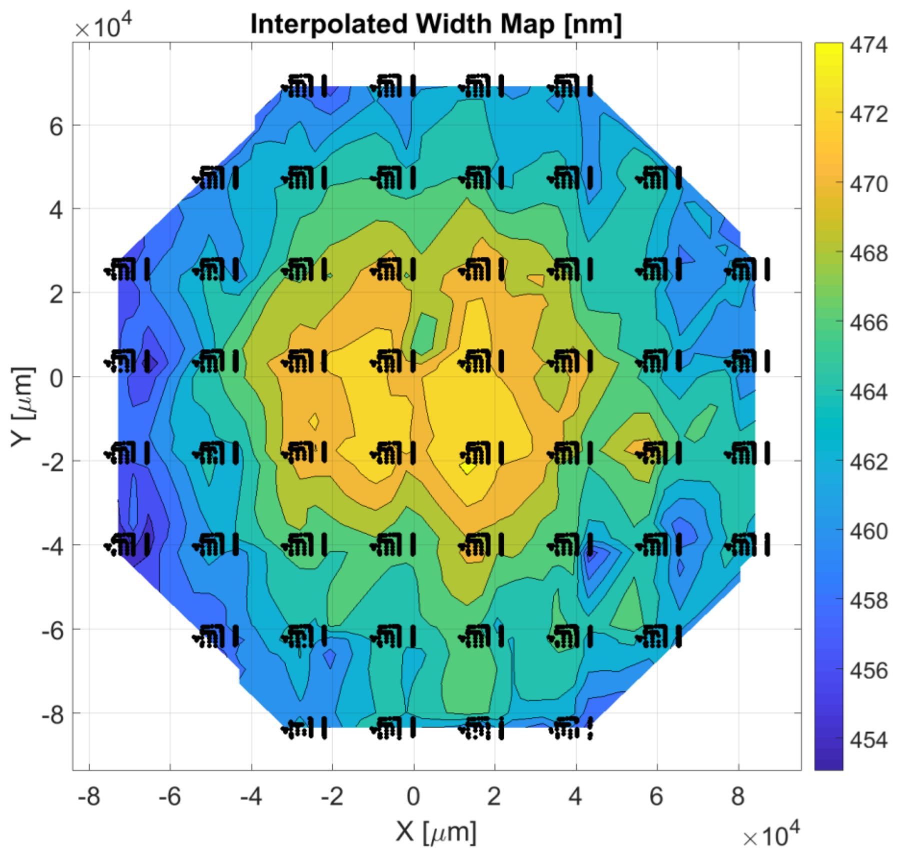


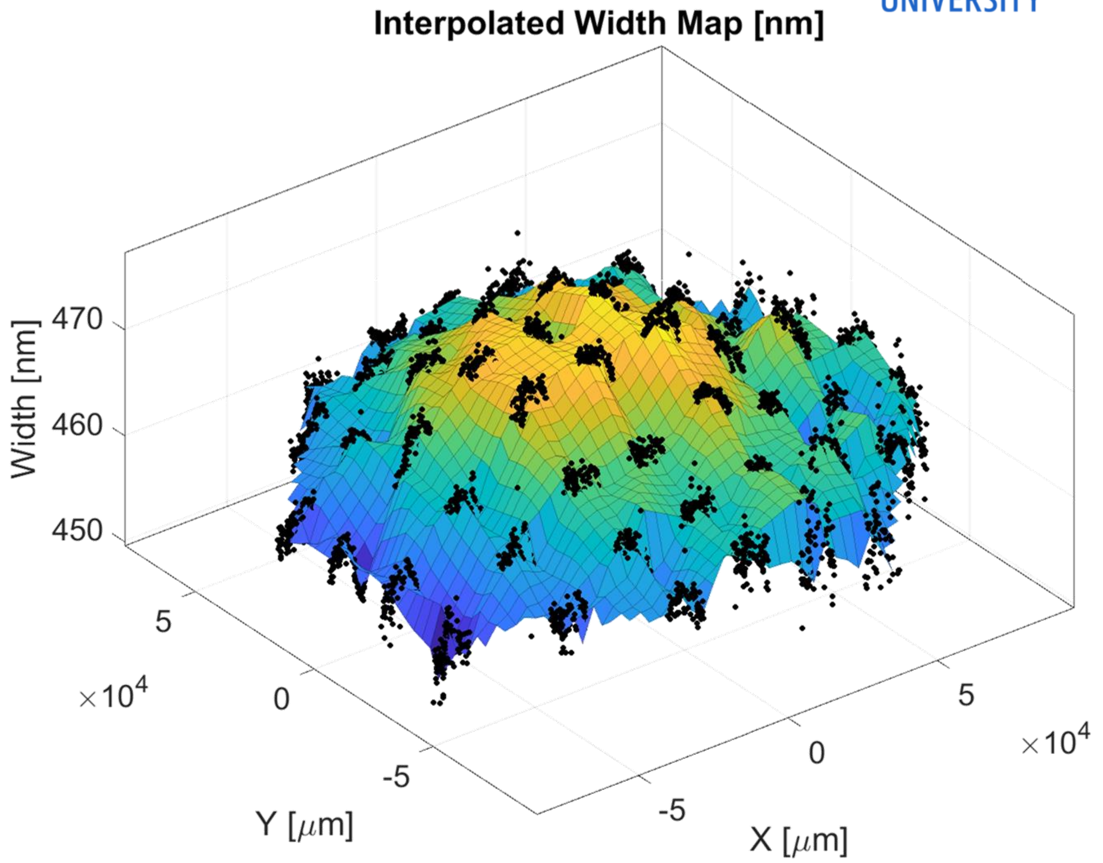




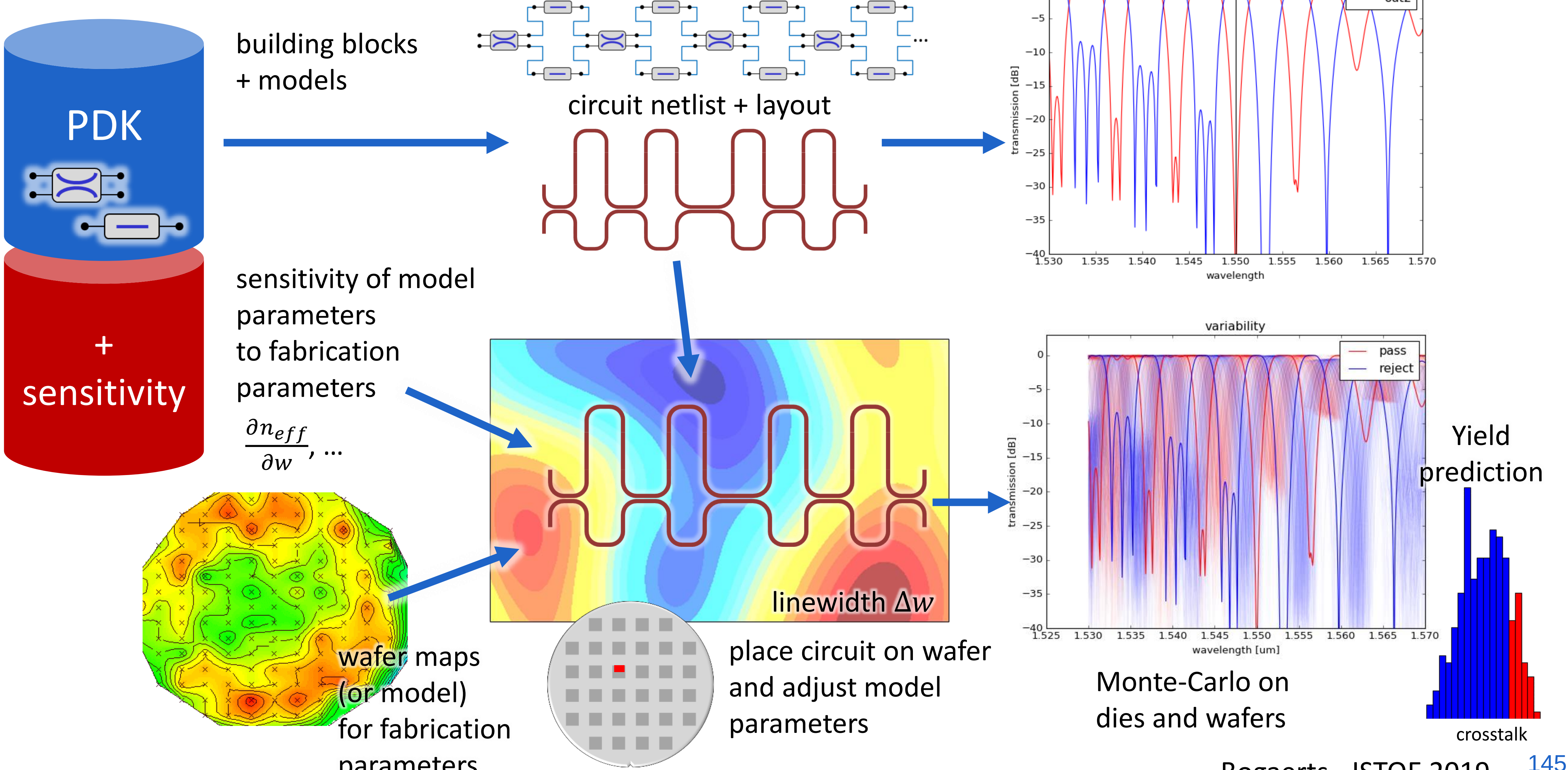
Y. Xing, Phot.Res. 2018

LINEWIDTH MAP





YIELD PREDICTION SCHEME

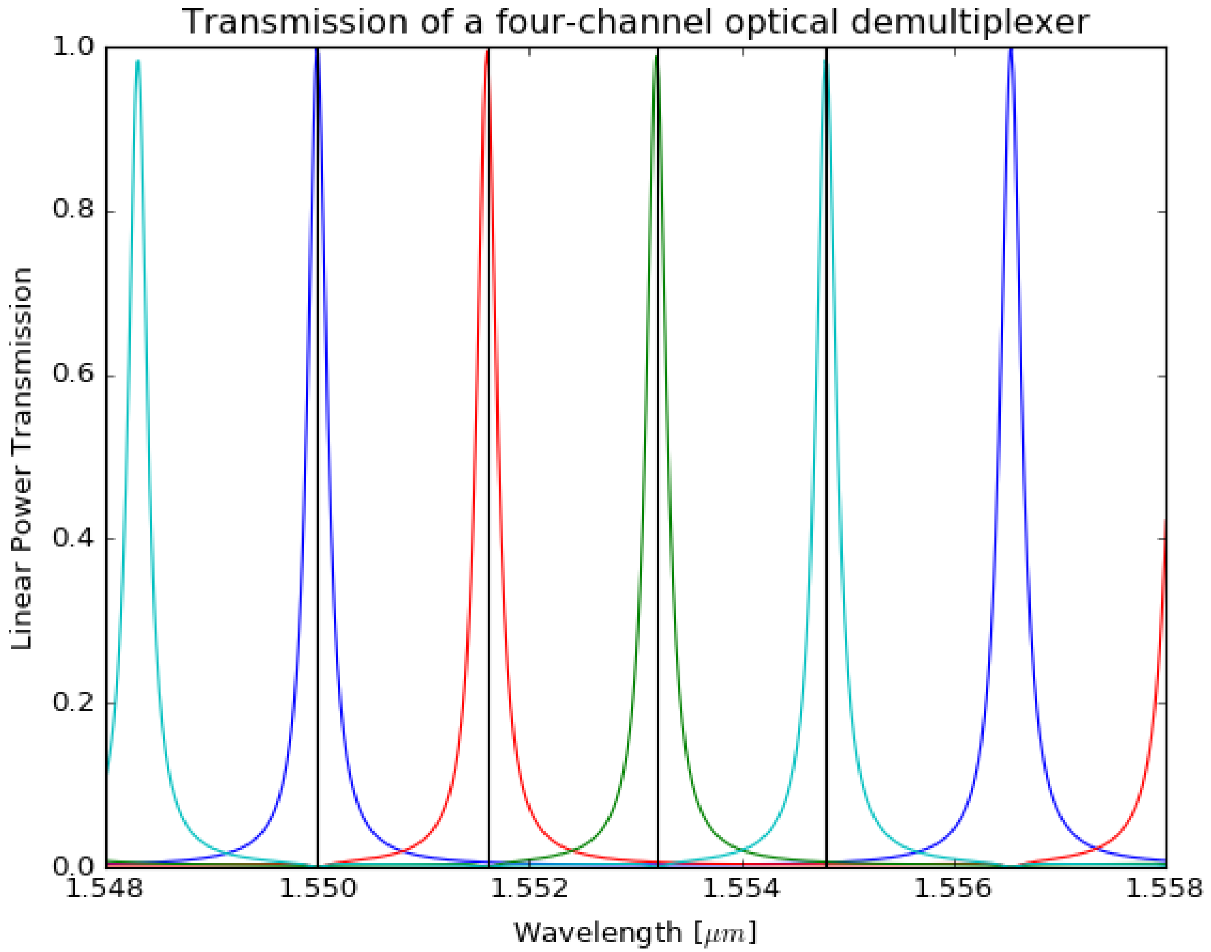


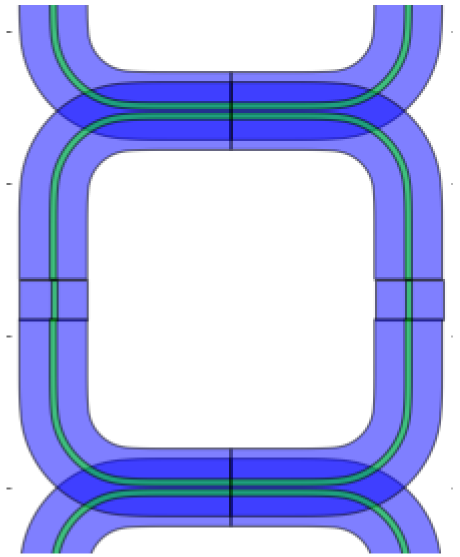
PDK
+
sensitivity
building blocks
+ models
circuit netlist + layout
circuit simulation
out1
out2
transmission [dB]
wavelength
sensitivity of model parameters to fabrication parameters
$\frac{\partial n_{eff}}{\partial w}$, …
linewidth Δw
wafer maps (or model) for fabrication parameters
place circuit on wafer and adjust model parameters
variability
pass
reject
wavelength [um]
Monte-Carlo on dies and wafers
Yield prediction
crosstalk

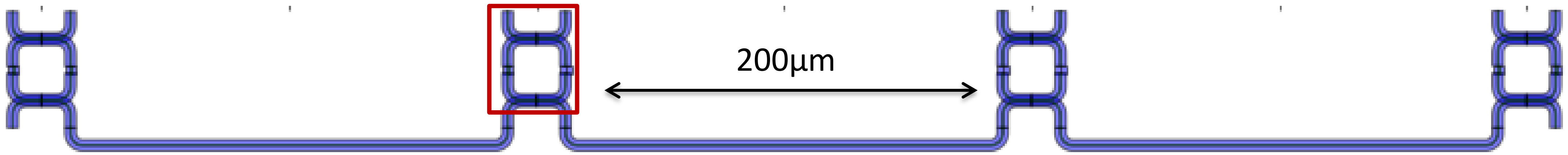
Bogaerts , JSTQE 2019

Other example: 4-ring Demux

- 4 rings with 1.6nm spacing
- $\lambda_1 = 1.55\ \mu m$

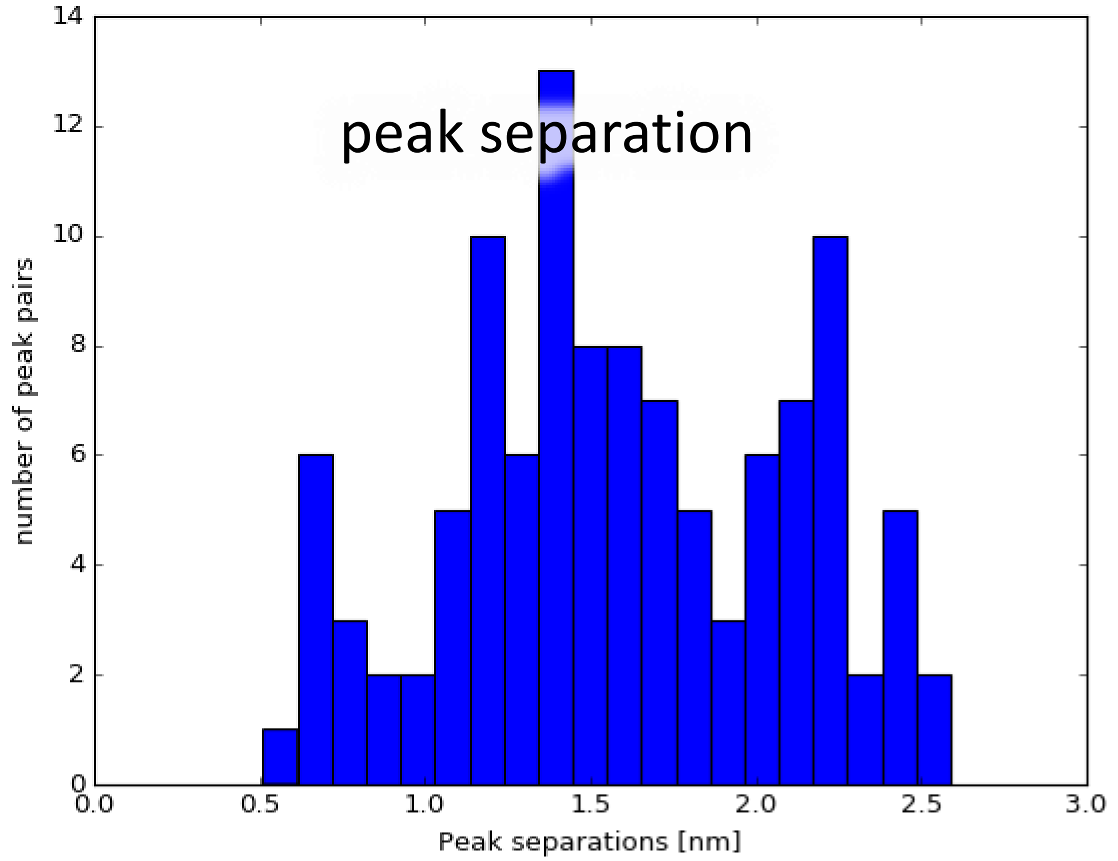


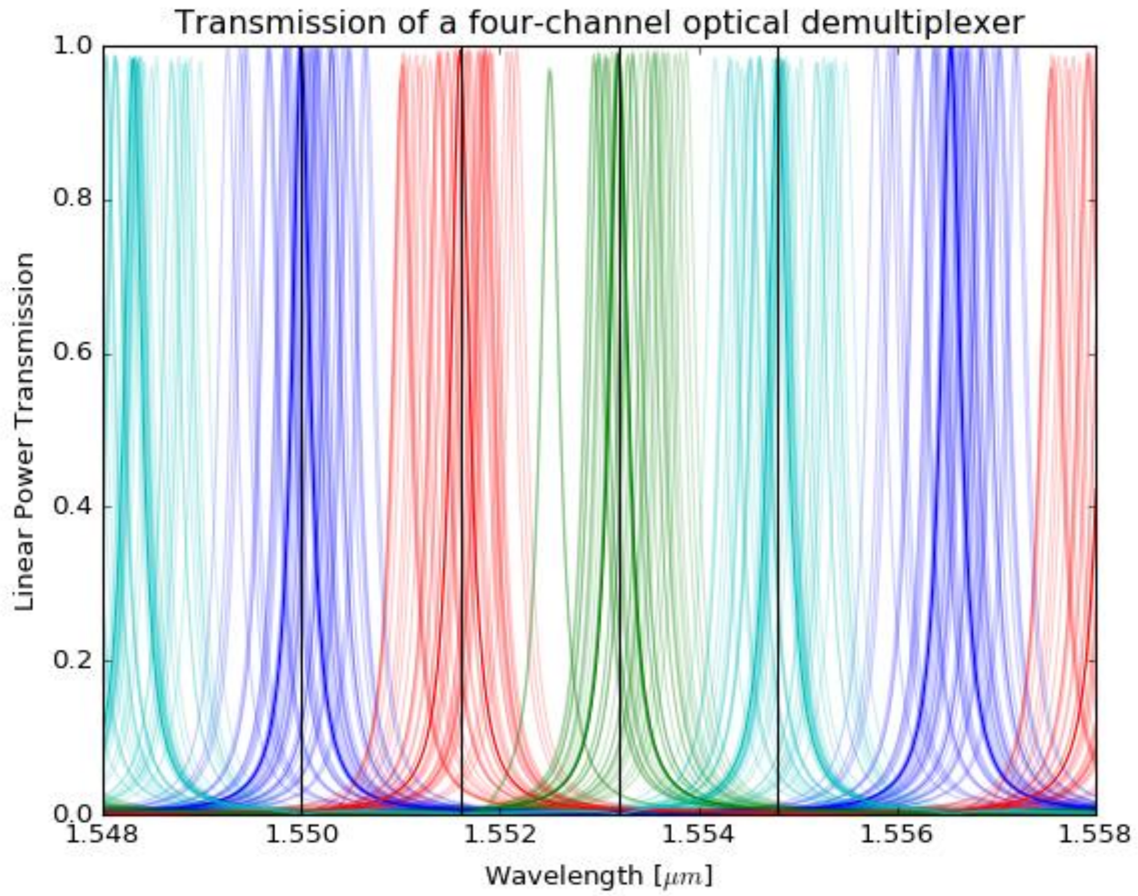


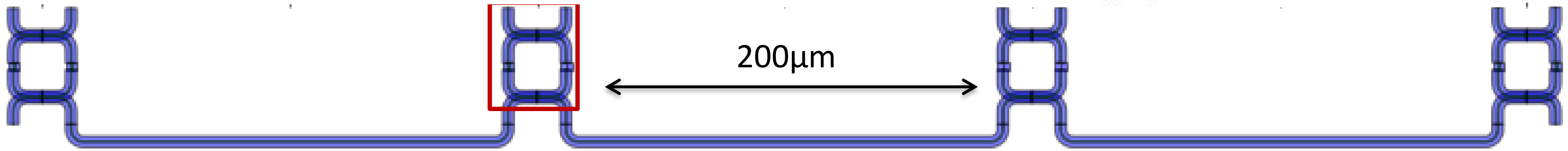


Effect of linewidth variation

fabrication linewidth variation only ($\sigma = 1nm$)

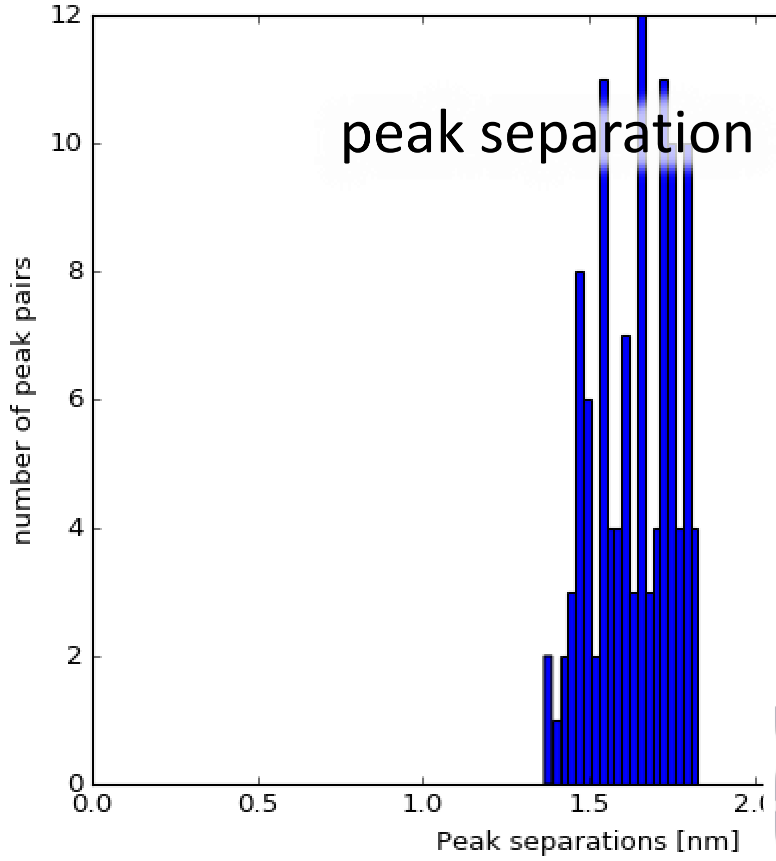


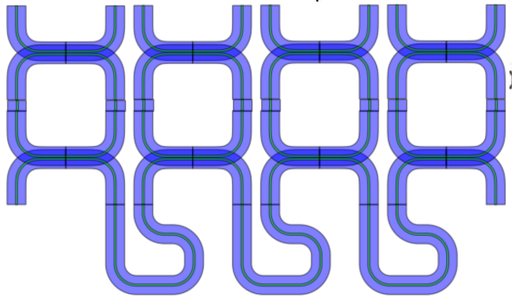


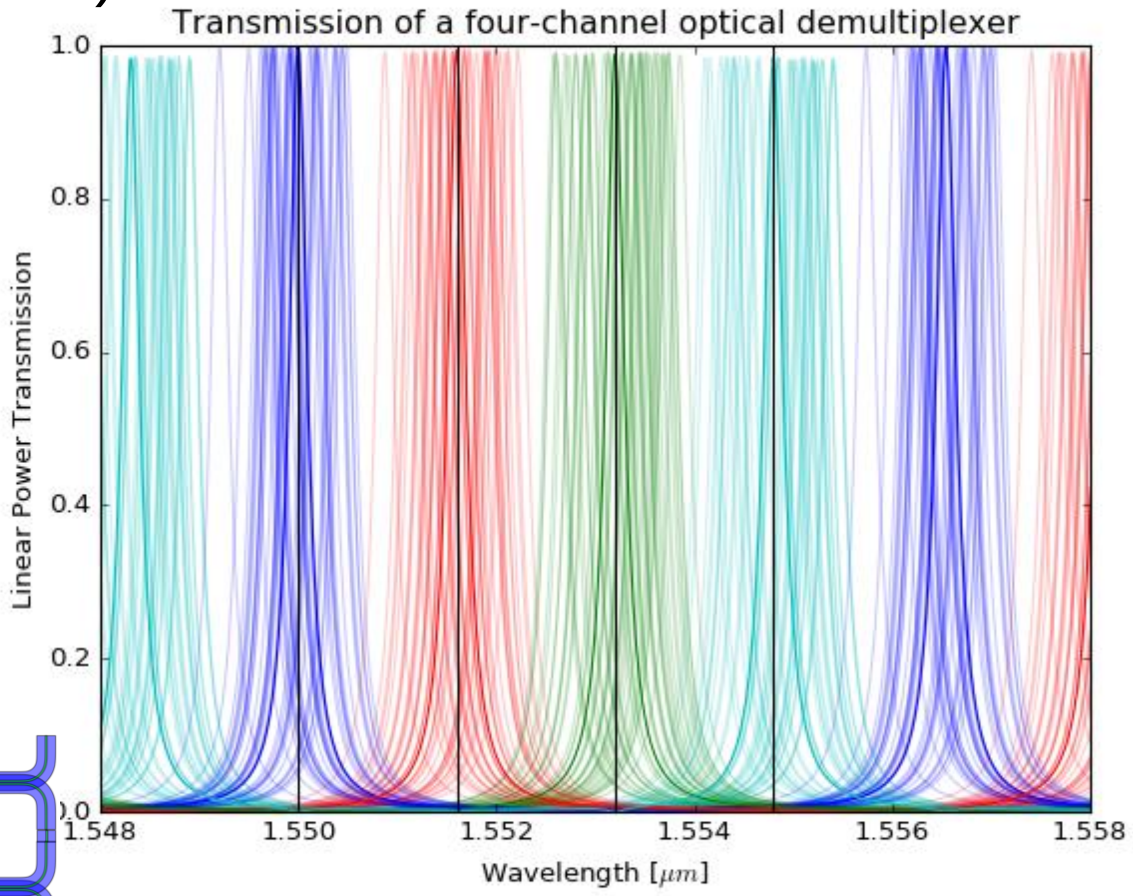


BRINGING THE DEVICES CLOSER TOGETHER

fabrication linewidth variation only ($\sigma = 1nm$)



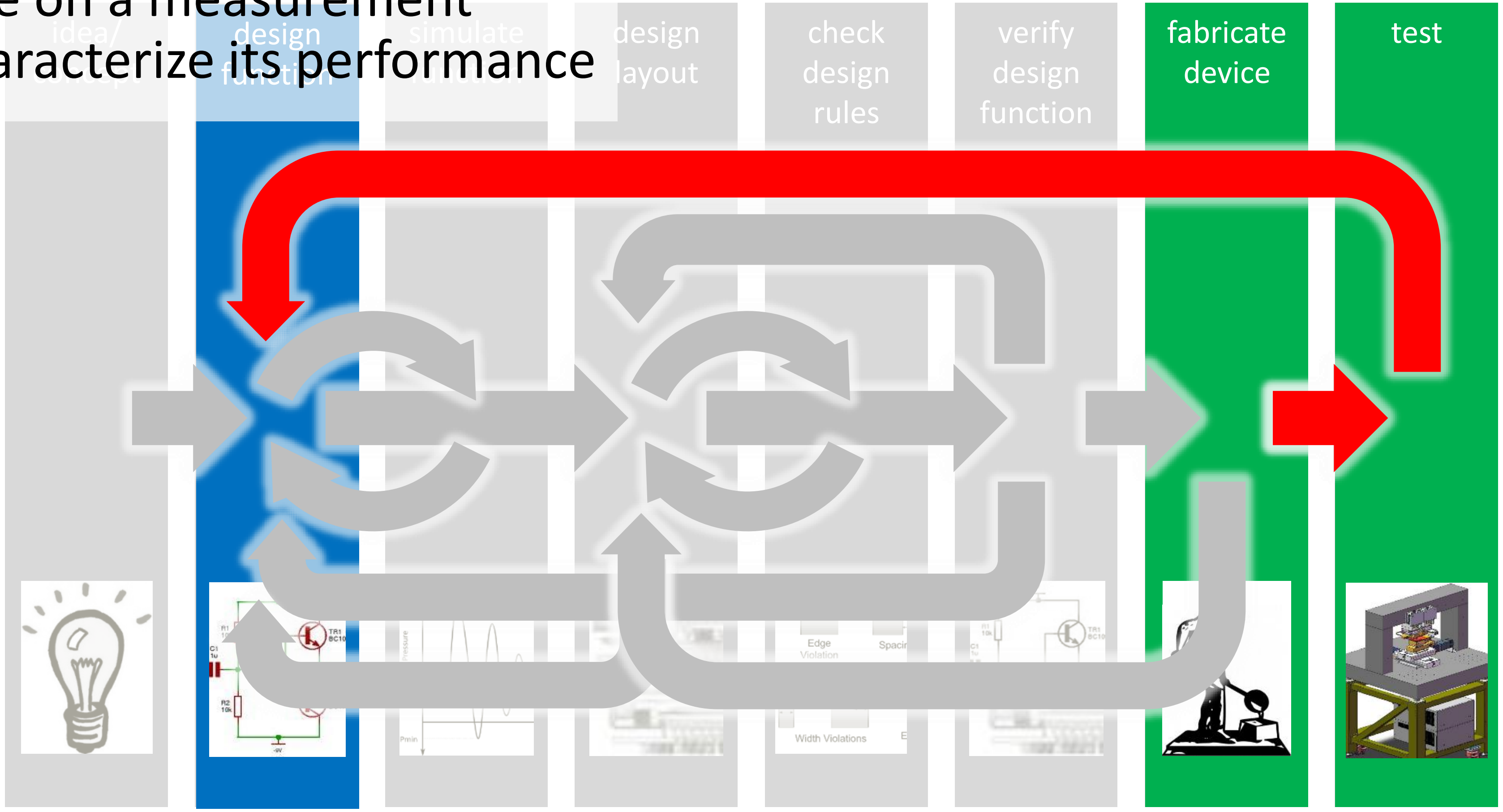



Transmission of a four-channel optical demultiplexer
Linear Power Transmission
Wavelength [μm]
0.0
0.2
0.4
0.6
0.8
1.0
1.548
1.550
1.552
1.554
1.556
1.558

30µm

TESTING

Put the device on a measurement setup and characterize its performance



design flow time

HOW TO TEST?

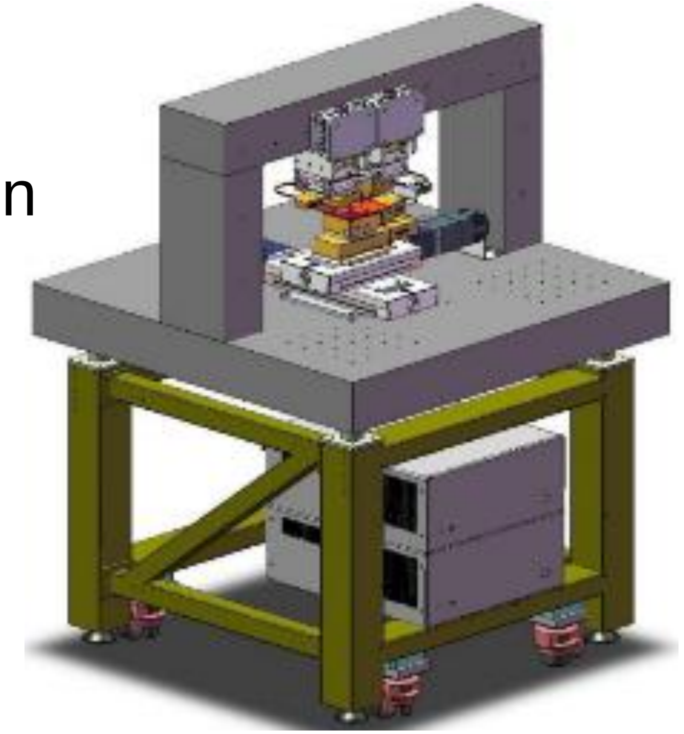
Electrical, optical, or both?

Wafer-scale testing -> grating couplers

Testing after packaging?

Need statistics?

depends on application



CHALLENGE: DEFINING GOOD TESTS

You need to think about tests during the design stage

- Which structures are representative?
- How can I isolate them?
- What parameters do I want to measure?
- How will I analyse/fit the data?

Parameters for your component models!

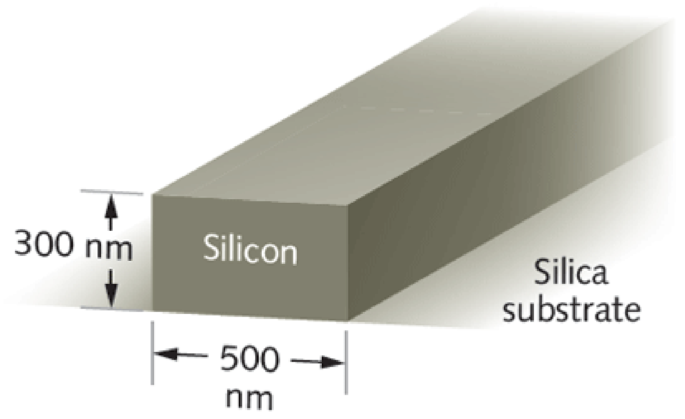
- What makes a good model?

Example: waveguide model

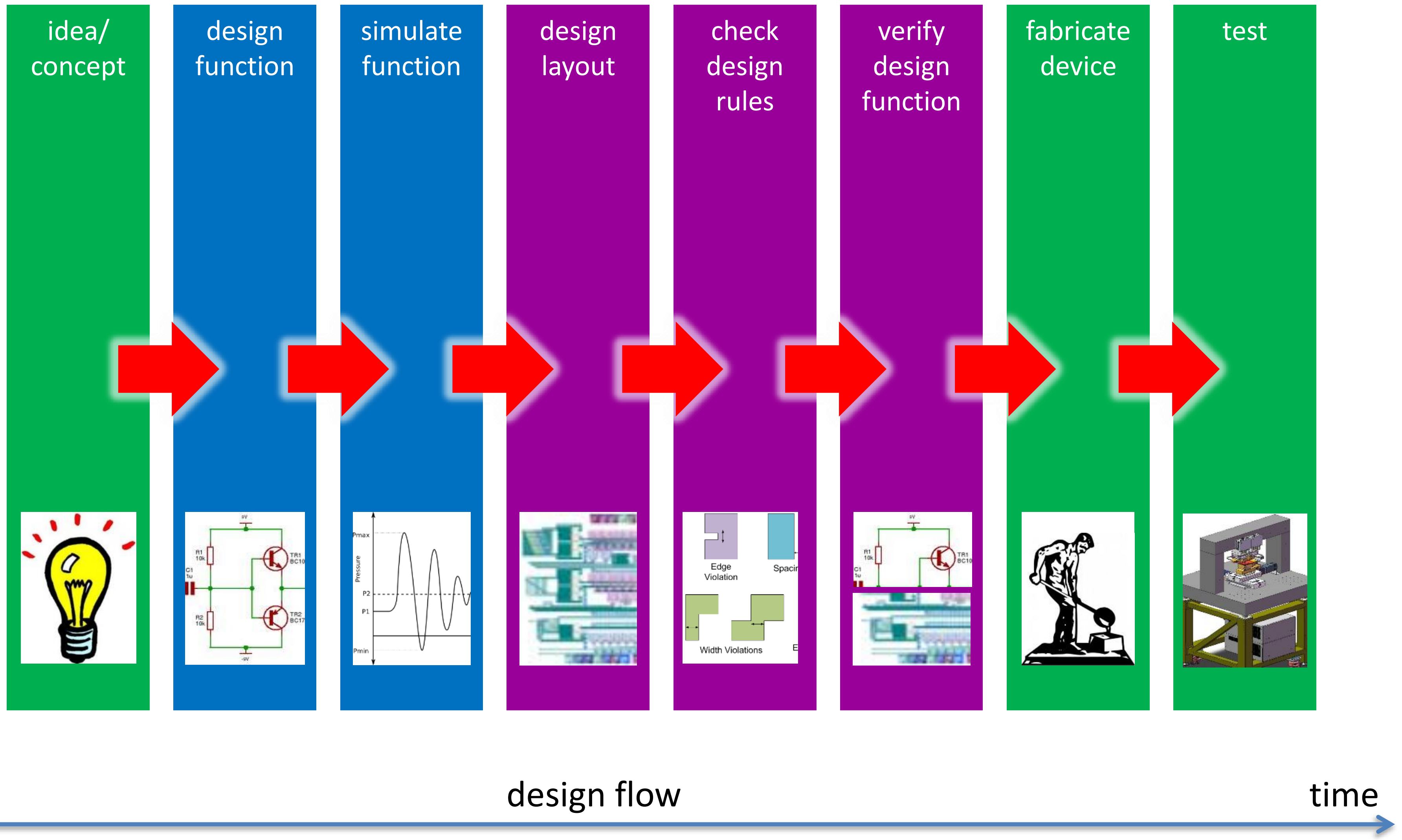
- $n_{eff}(\lambda)$ -> polynomial?
- loss(λ) -> polynomial?
- nonlinearities?

How to measure n_{eff}?

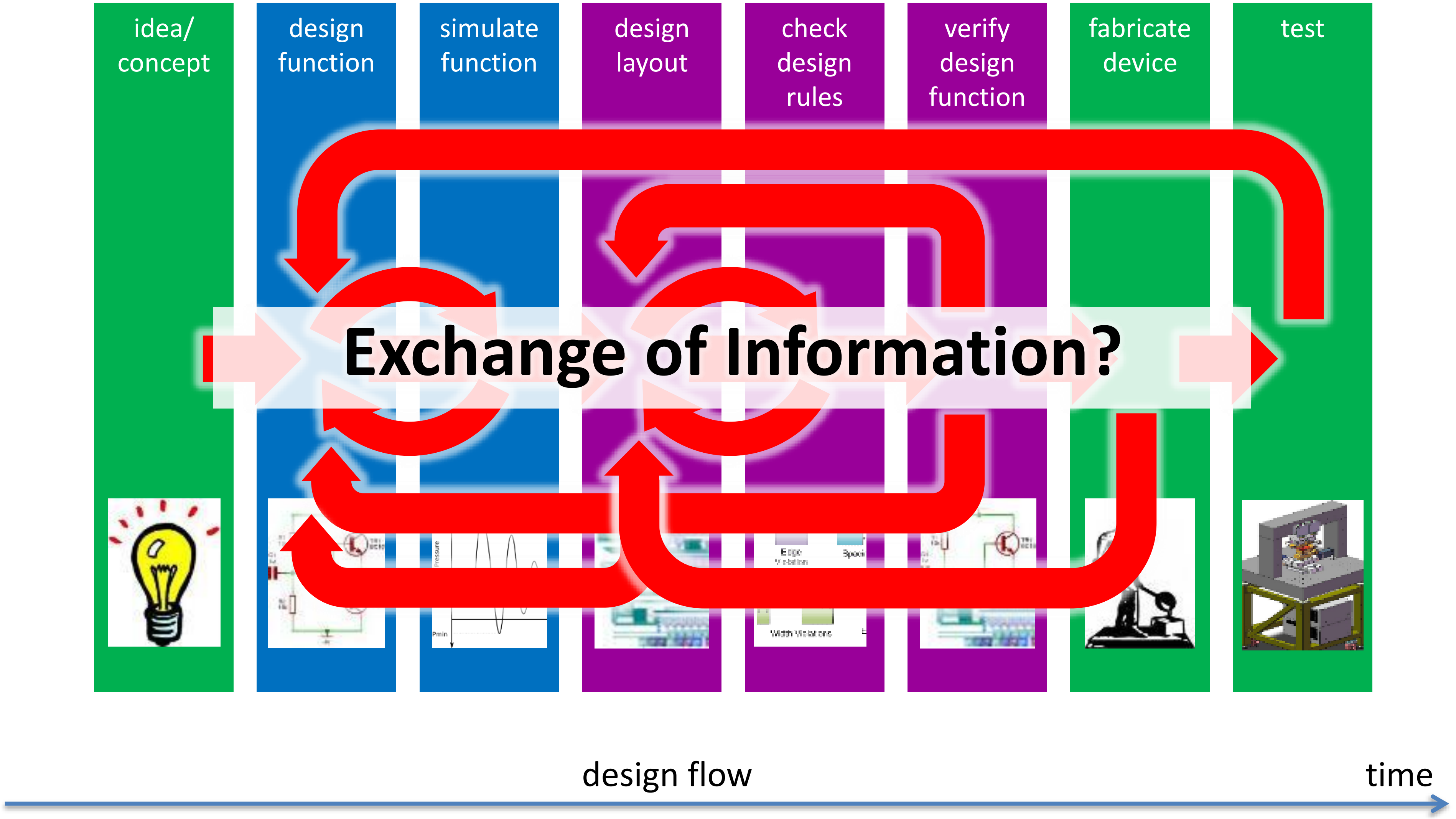
Typical silicon nanowire



Our simple design flow



OUR SIMPLE DESIGN FLOW



EXCHANGE OF INFORMATION

Files

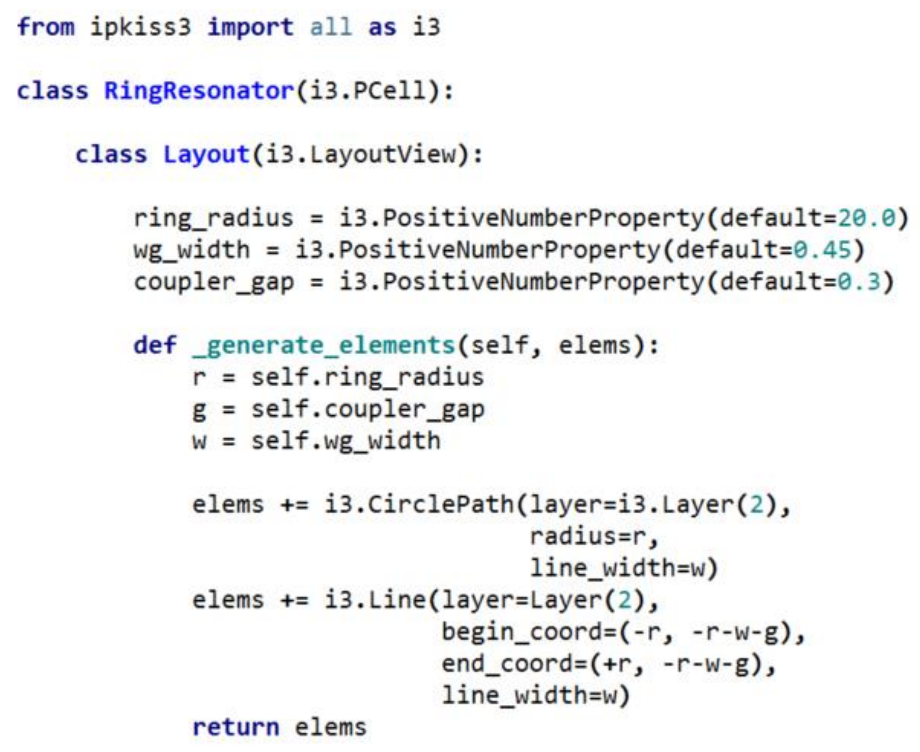
- Layout: GDSII and OASIS
- Netlist/Schematic: Spice, EDIF
- Models: Spice, VerilogA, C++, Python
- PCell code: Skill, Python , Tcl
- Data: Touchstone, XML

Databases

- proprietary
- EDA standard: OpenAccess

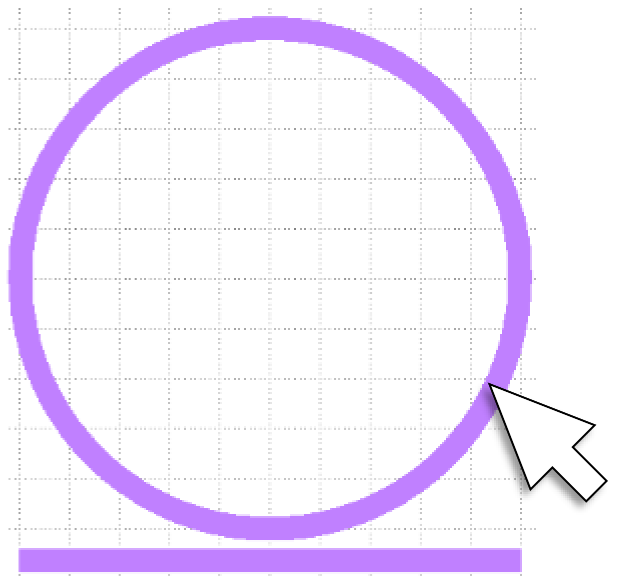
Designing in Code versus GUI

Designing in Code



```python
from ipkiss3 import all as i3

class RingResonator(i3.PCell):

    class Layout(i3.LayoutView):

        ring_radius = i3.PositiveNumberProperty(default=20.0)
        wg_width = i3.PositiveNumberProperty(default=0.45)
        coupler_gap = i3.PositiveNumberProperty(default=0.3)

        def _generate_elements(self, elems):
            r = self.ring_radius
            g = self.coupler_gap
            w = self.wg_width

            elems += i3.CirclePath(layer=i3.Layer(2),
                                   radius=r,
                                   line_width=w)
            elems += i3.Line(layer=Layer(2),
                             begin_coord=(-r, -r-w-g),
                             end_coord=(+r, -r-w-g),
                             line_width=w)
            return elems
```

Designing in GUI



DESIGNING IN CODE VERSUS GUI

Designing in Code

Pro:

- Easy to reuse
- Easy to upgrade design
- Easy to share and version
- Easy parametrize
- Easy to document and make examples
- Everything is numerically correct

Con:

- Harder to learn
- No immediate visual feedback

Designing in GUI

Pro:

- Intuitive quick start
- Visual feedback
- WYSIWYG
- Quick point and click

Con:

- Difficult to make complex things
- No calculations
- A lot of manual work
- Easy make small (invisible) mistakes

DESIGNING IN CODE VERSUS GUI

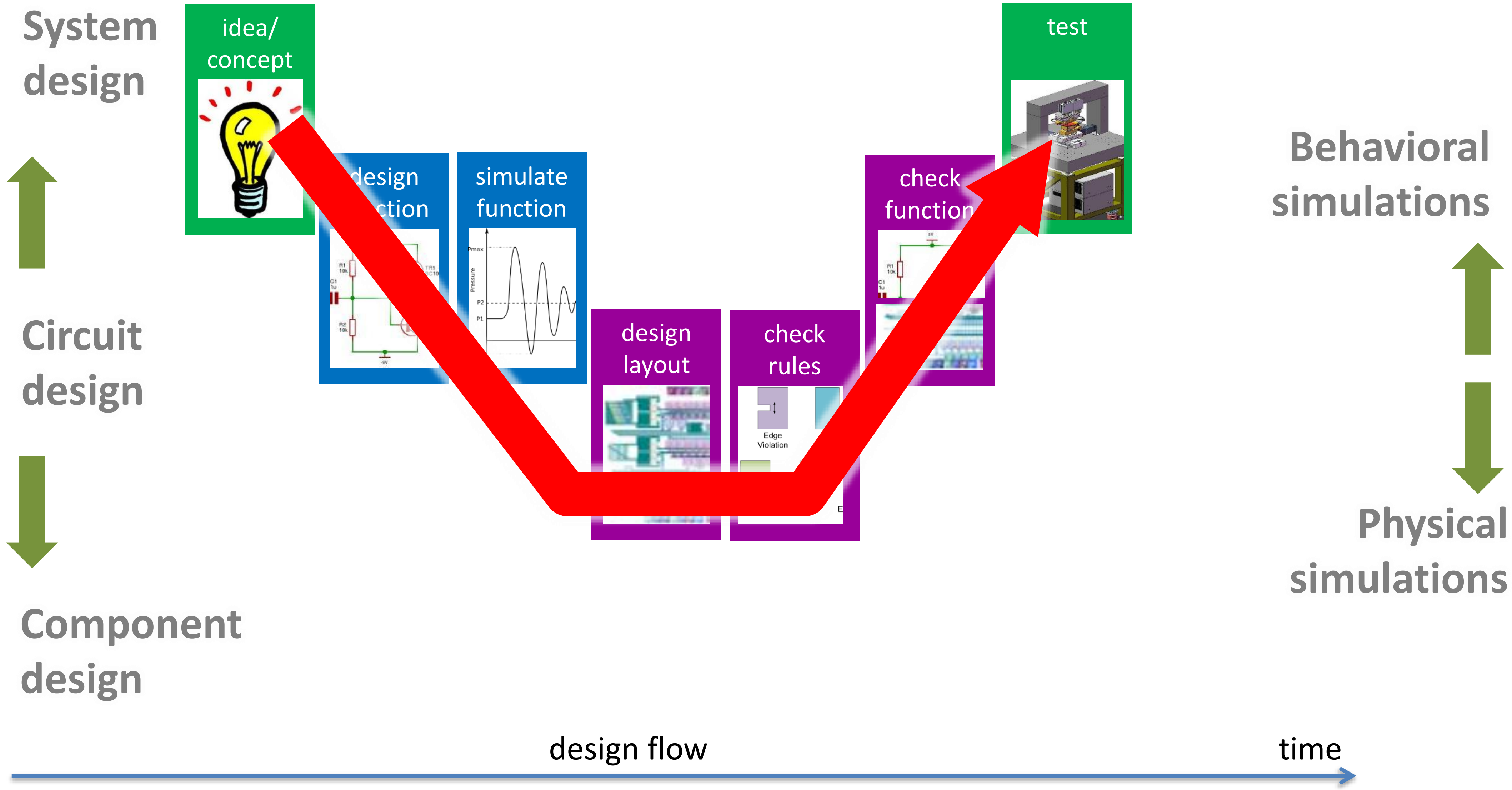
Designing in Code

- parameter sweeps
- calculated geometries
- circuit models
- automatic placement and routing

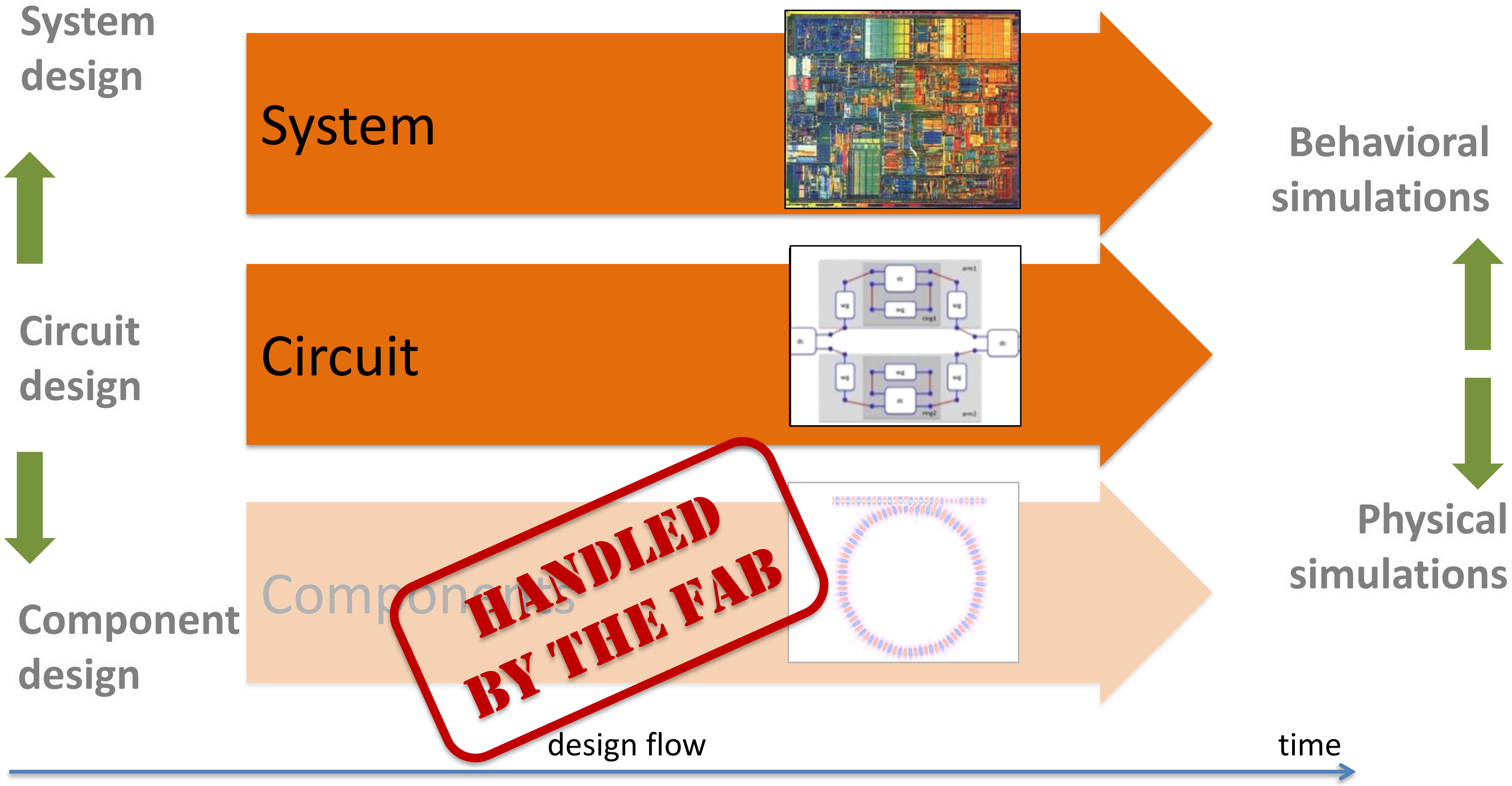
Designing in GUI

- schematic connectivity
- layout positioning (floorplanning)
- fixing the last DRC errors
- quick manual routing

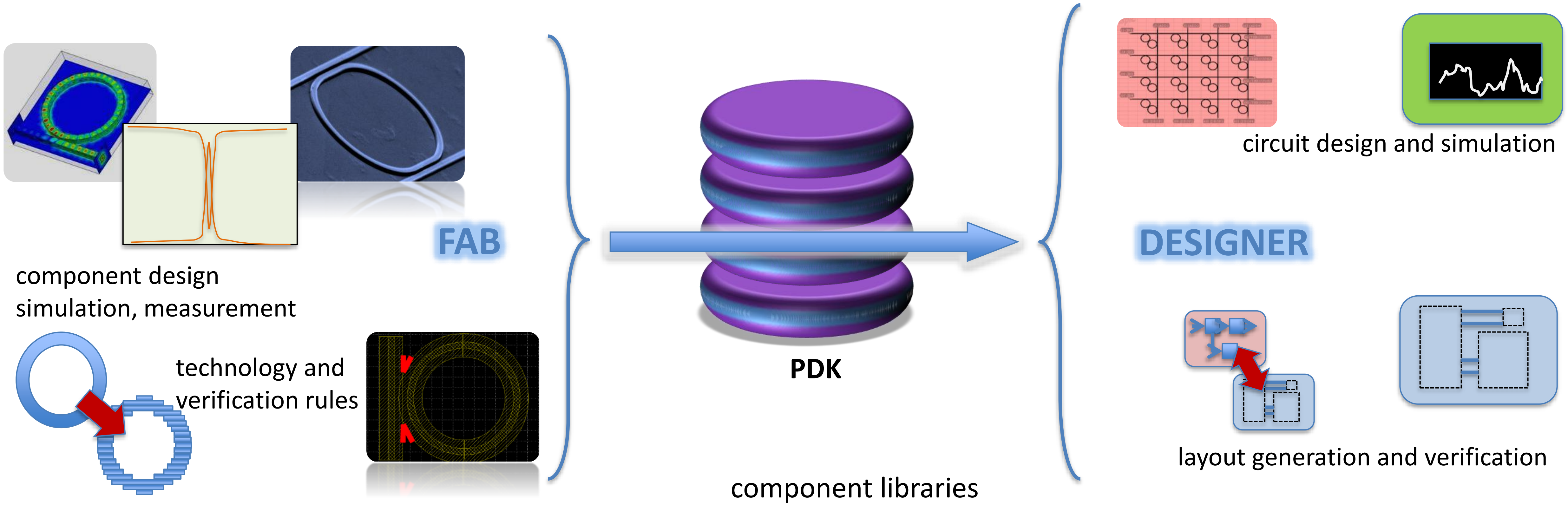
ABSTRACTIONS IN A CIRCUIT DESIGN FLOW



ABSTRACTIONS IN A CIRCUIT DESIGN FLOW



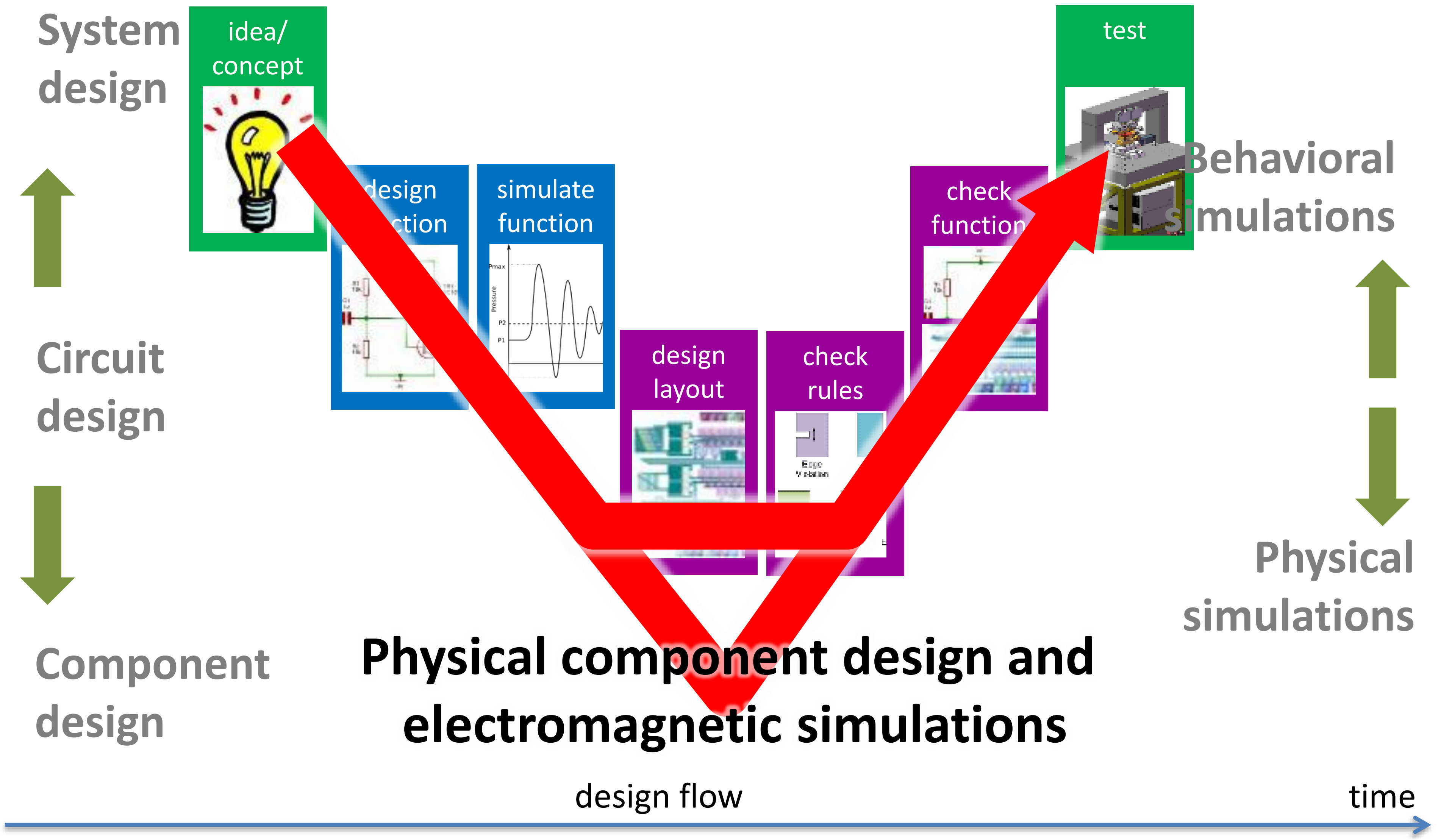
PDK: INTERFACE FROM FAB TO DESIGNER



component libraries
documentation
support scripts

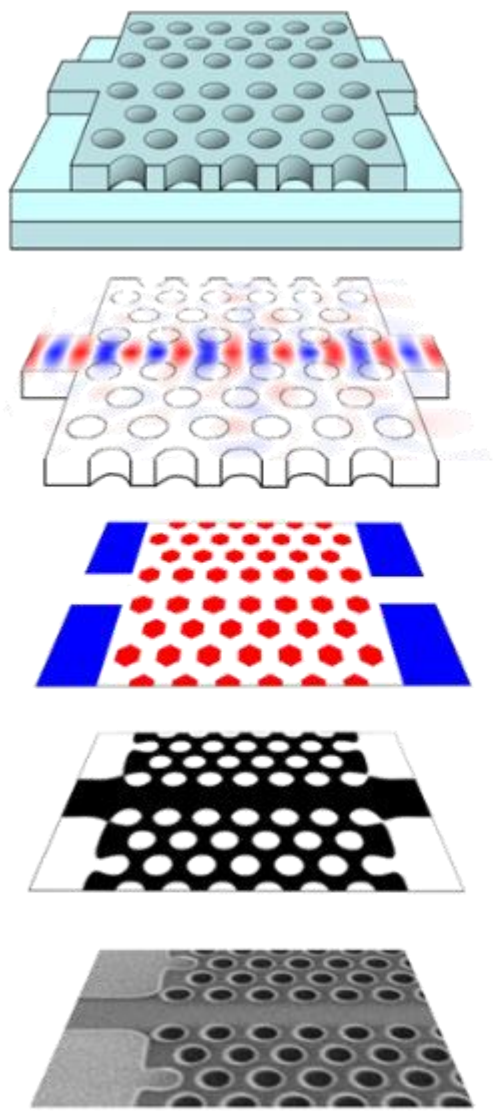
PDK for photonics

ABSTRACTIONS IN A DESIGN FLOW



Physical component design and electromagnetic simulations

WHY MORE COMPONENT DESIGN IN PHOTONICS?



More geometric design freedom

- Photonic Crystals
- Subwavelength gratings
- Arbitrary holographic functions
- …

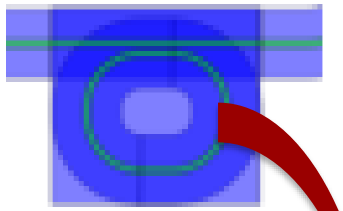
More complex behaviour

- Phase: interference effects
- Wavelength dependence
- Nonlinearities
- …

Requires accurate physical modelling

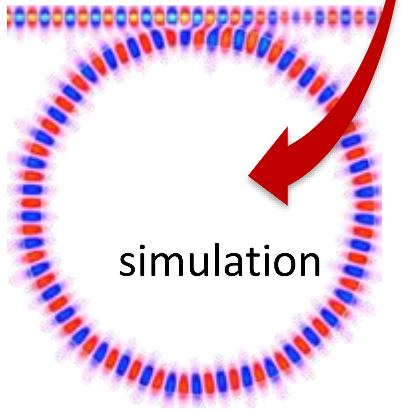
COMPONENT DESIGN VS. CIRCUIT DESIGN

Component Design



layout

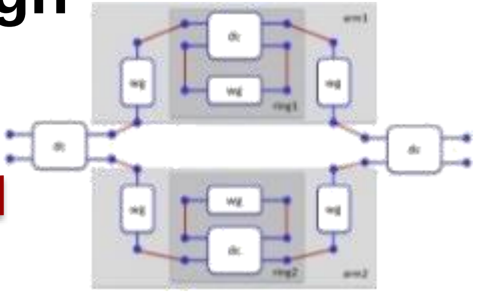
geometry

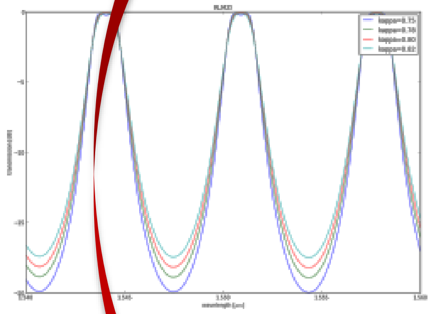


simulation

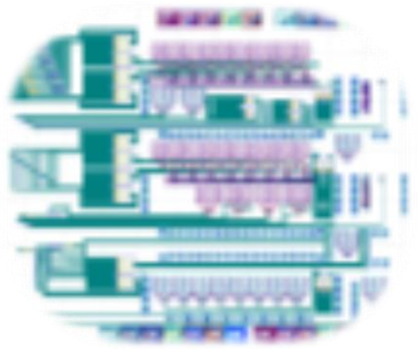
Circuit design

Circuit capture





simulation



layout

GHENT UNIVERSITY imec

Photonic Circuit design tools

Optiwave

SIMULIA

VPIphotonics DESIGN AUTOMATION

FILARETE SOFTWARE & SERVICES FOR PHOTONICS

LUCEDA

Photon Design®

Ansys lumerical illuminating the way

SYNOPSYS®

SIEMENS Mentor®

cadence™

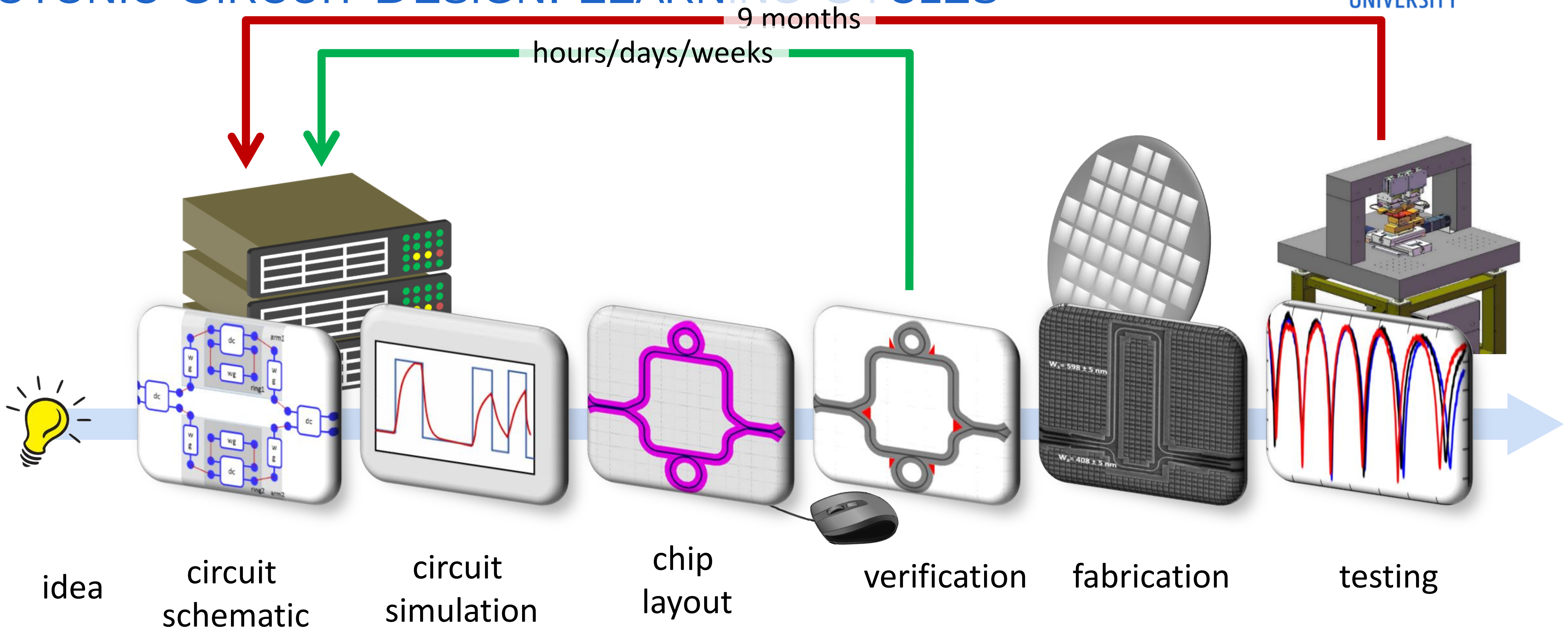
Nazca design™ photonic IC design framework

Tool capabilities

★ links to other tools (documented)

	Component sim	Circuit Sim	Component Layout	Circuit Layout	Verification
SYNOPSYS	🟩 Fullwave Beamprop	🟩 OptSim	🟩 Optodesigner	🟩 Optodesigner	🟨 Optodesigner
cādence	🟥	🟨 Spectre AMS ★	🟩 Virtuoso	🟩 Virtuoso	🟨 Assura
SIEMENS Mentor	🟥	🟨 Eldo	🟩 L-Edit	🟩 L-Edit LightSuite	🟩 Calibre
LUCEDA	🟨 Camfr ★	🟩 Caphe	🟩 IPKISS	🟩 IPKISS	🟧 ★
VPIphotonics	🟨 ModeDesigner	🟩 ComponentMaker	🟥	🟥	🟥
lumerical / Ansys	🟩 FDTDSolutions ModeSolutions Device	🟩 Interconnect	🟥	🟥	🟥
Nazca design	🟧 ★	🟧	🟩 Nazca	🟩 Nazca	🟥
Photon Design	🟩 Fimmwave Omnisim	🟩 PICwave	🟥	🟥	🟥
SIMULIA	🟩 CST studio	🟥	🟥	🟥	🟥

Photonic Circuit Design: Learning Cycles



Learning cycles through fabrication take a lot of time.

PROTOTYPING A NEW (SILICON) PHOTONIC IC

Design (4M)

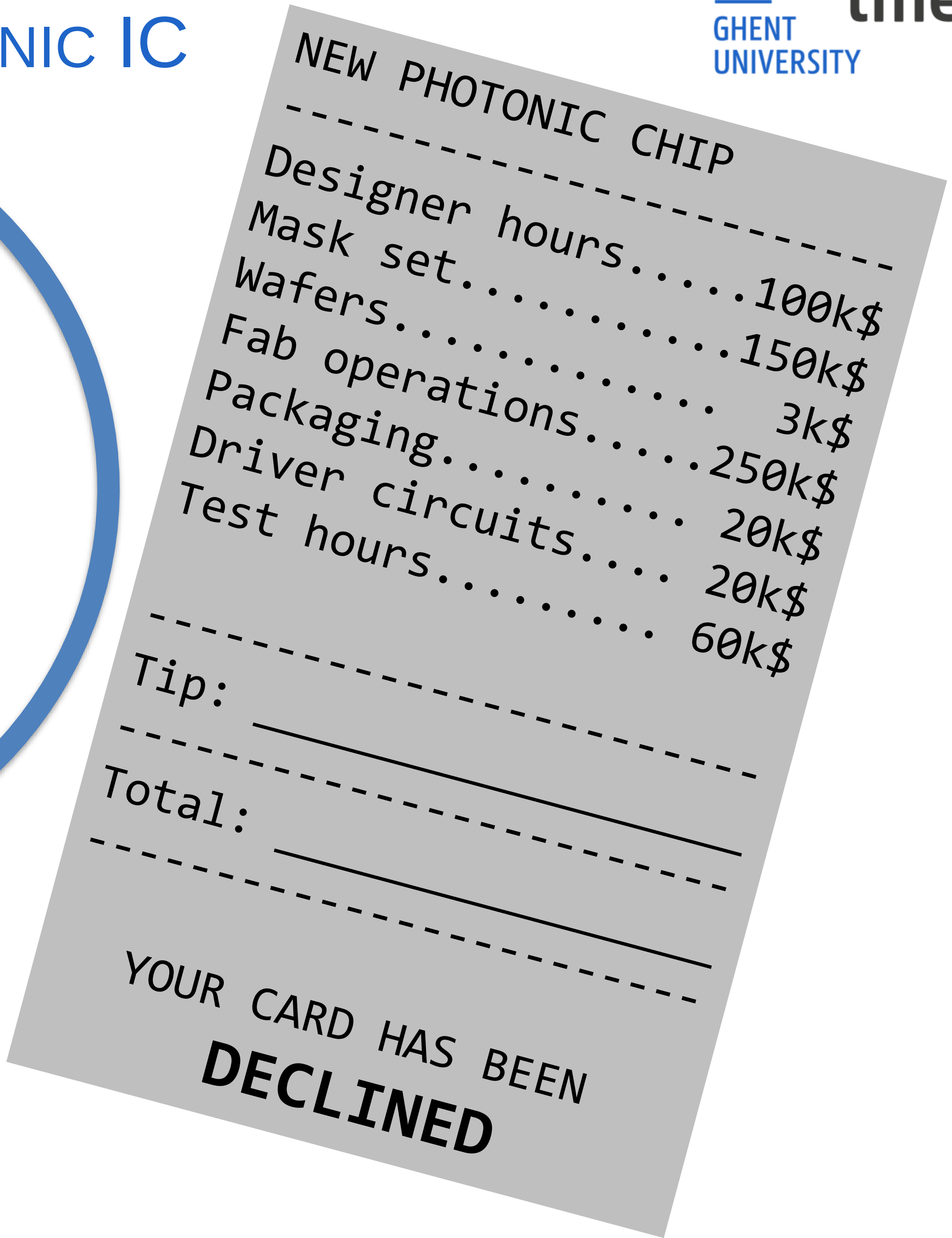
Fabrication (6M)

Package (1M)

Test (2M)

Then you discover the bugs…

Repeat!



```
NEW PHOTONIC CHIP
------------------------
Designer hours.....100k$
Mask set...........150k$
Wafers.........      3k$
Fab operations.....250k$
Packaging........    20k$
Driver circuits....  20k$
Test hours.........  60k$
------------------------
Tip: ___________________
------------------------
Total: _________________
------------------------

YOUR CARD HAS BEEN
DECLINED
```

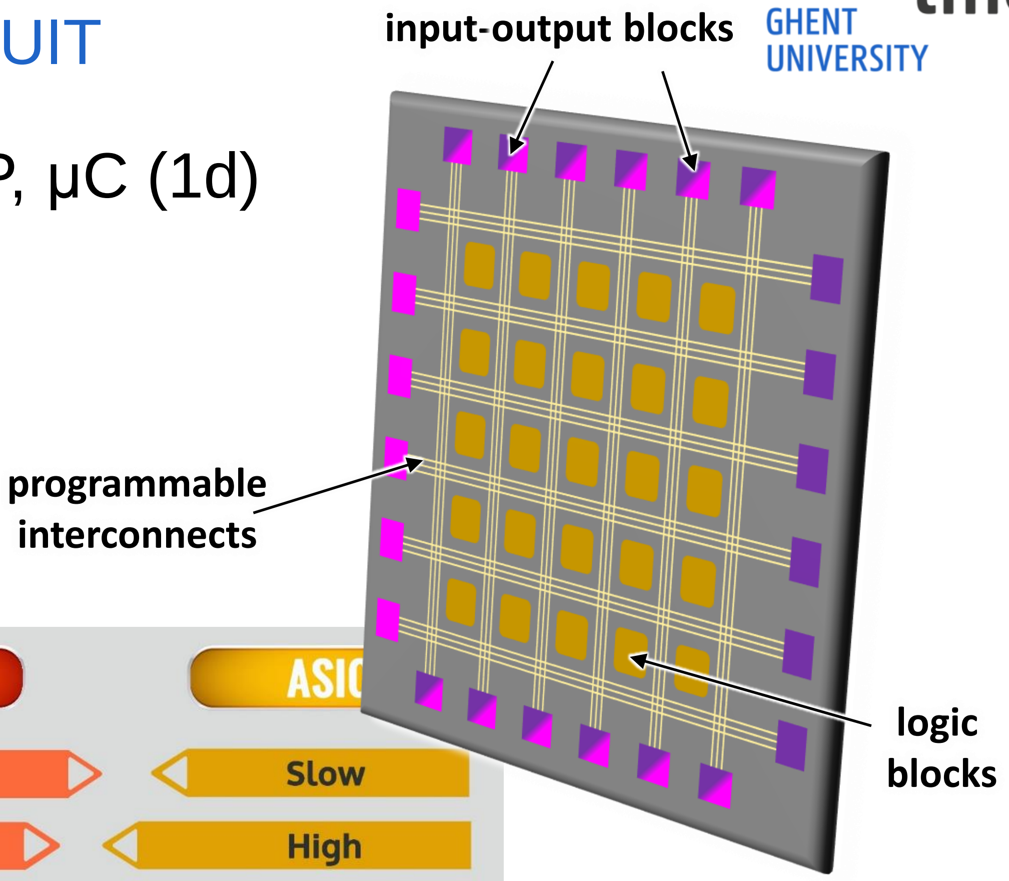
PROTOTYPING A NEW ELECTRONIC CIRCUIT

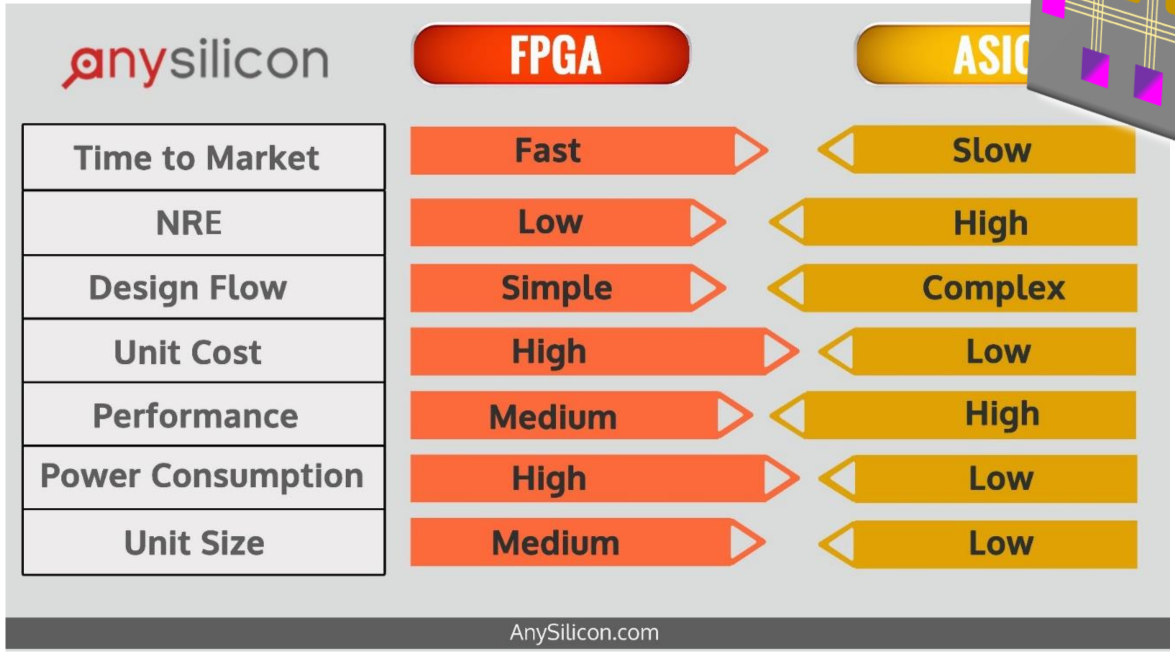
Select a suitable programmable IC: FPGA, DSP, µC (1d)

Program and test the chip (1-4w)

Only then, if needed:

- Design ASIC …




anysilicon

	FPGA	ASIC
Time to Market	Fast	Slow
NRE	Low	High
Design Flow	Simple	Complex
Unit Cost	High	Low
Performance	Medium	High
Power Consumption	High	Low
Unit Size	Medium	Low

AnySilicon.com

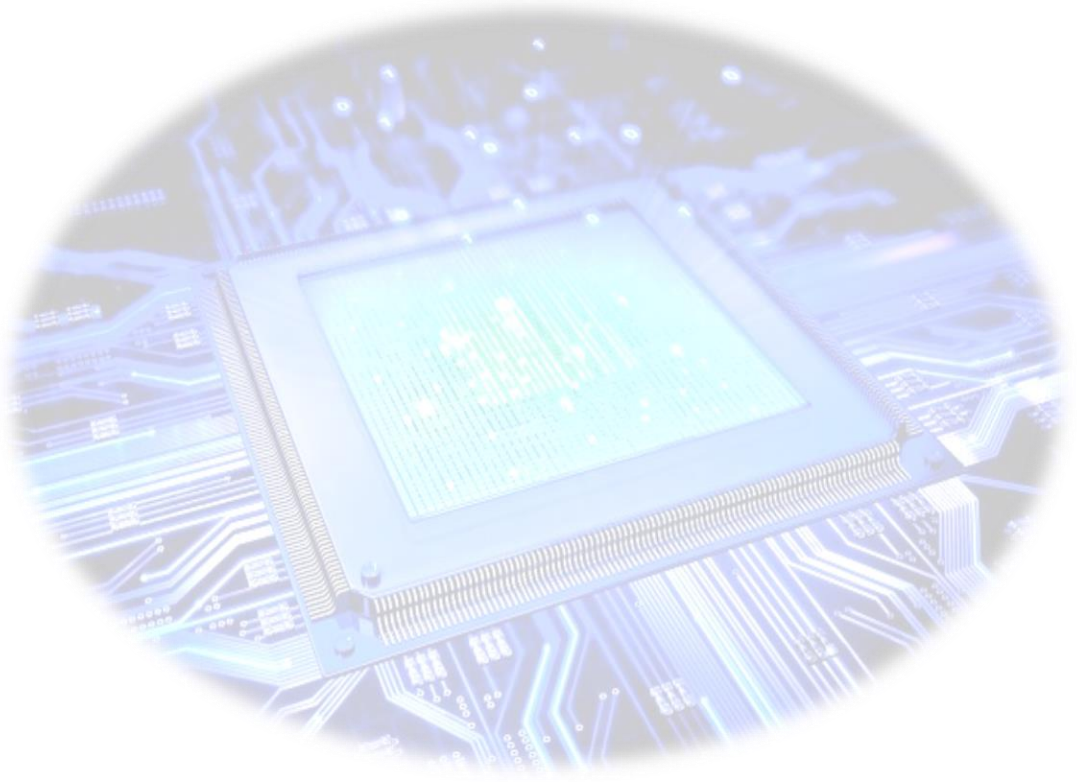
The Photonic FPGAs?

or programmable photonics

reconfigurable photonics

photonic processors

universal photonic circuits …



Photonic Integrated Circuits

that **can be reconfigured**

using **software**

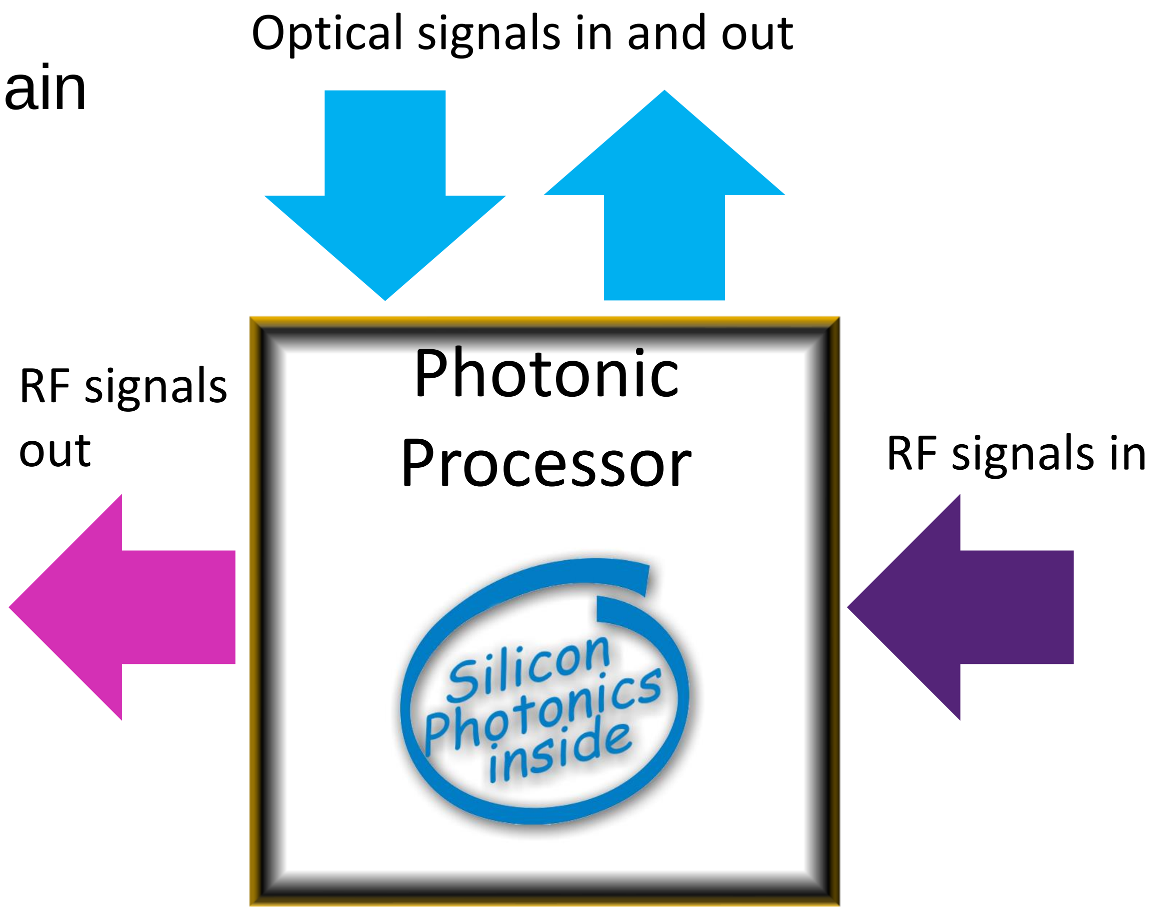
to perform **different functions**.

Programmable Photonic Chip

Can process signals in the optical domain

- balancing
- filtering
- transformations

Both on Optical and RF signals



GENERIC PROGRAMMABLE OPTICAL PROCESSOR

Optical inputs and outputs

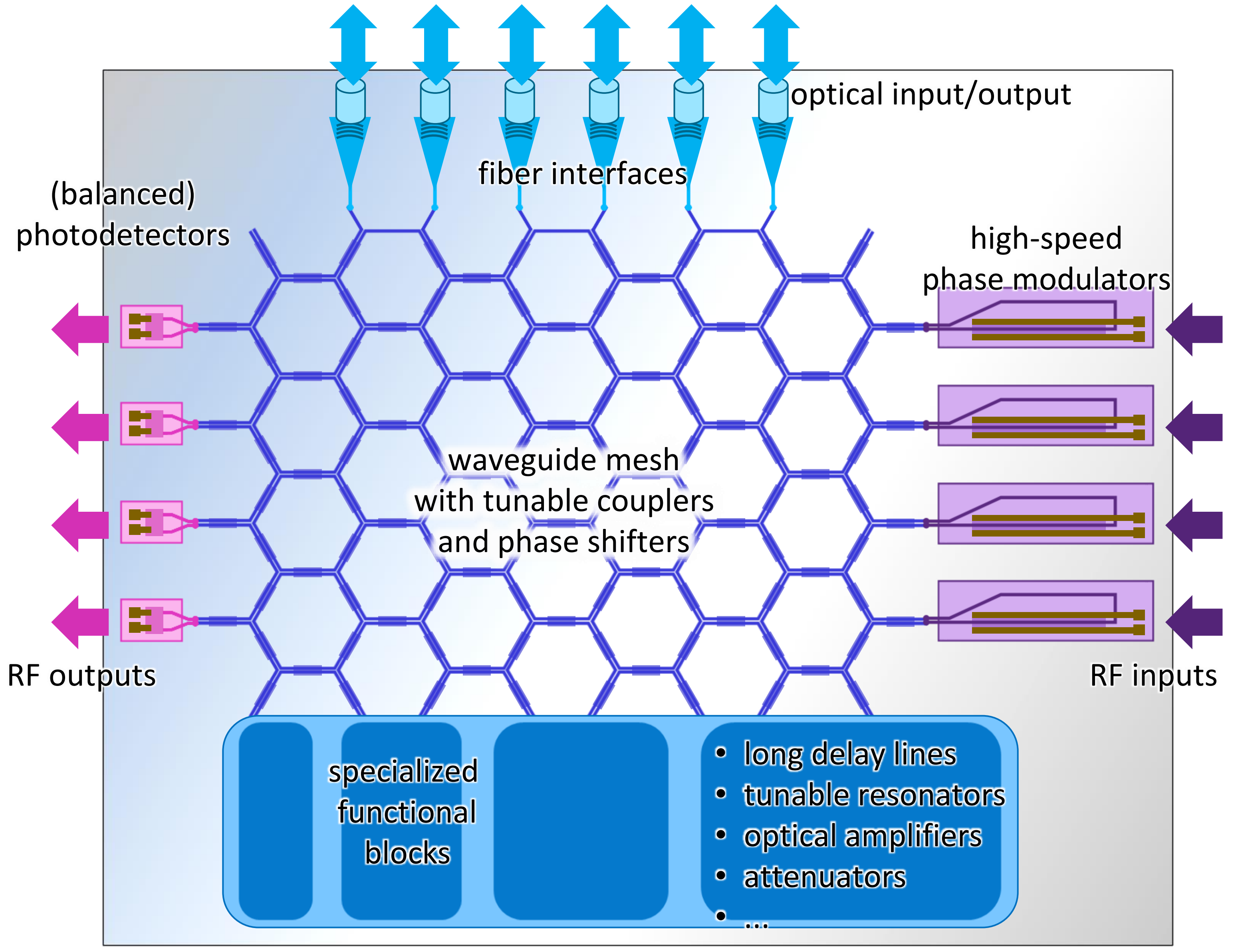
RF inputs: modulators

RF outputs: balanced PDs

Specialized high performance blocks

Connected by a programmable

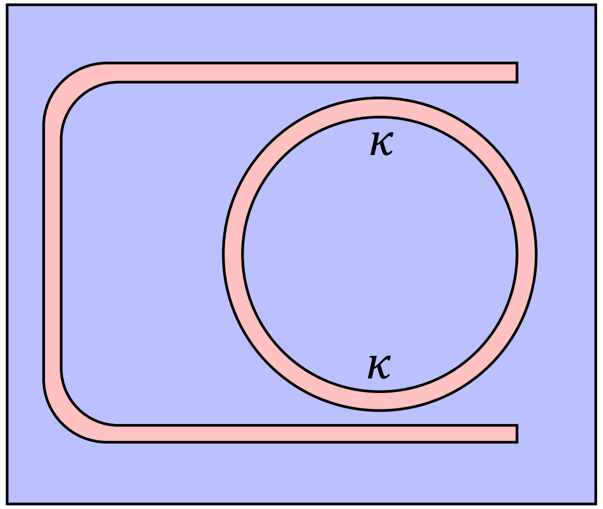
linear optical circuit


optical input/output
fiber interfaces
(balanced) photodetectors
high-speed phase modulators
waveguide mesh with tunable couplers and phase shifters
RF outputs
RF inputs
specialized functional blocks
• long delay lines
• tunable resonators
• optical amplifiers
• attenuators
• ...

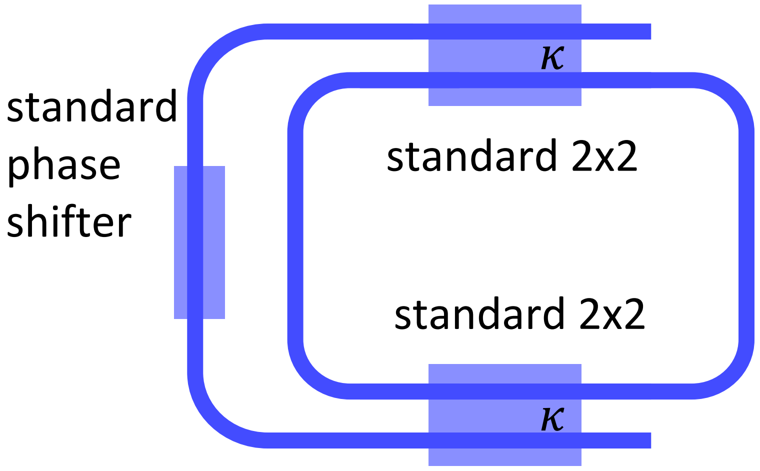
A NEW WAY OF DESIGNING FUNCTIONALITY

Full Custom design

geometry design

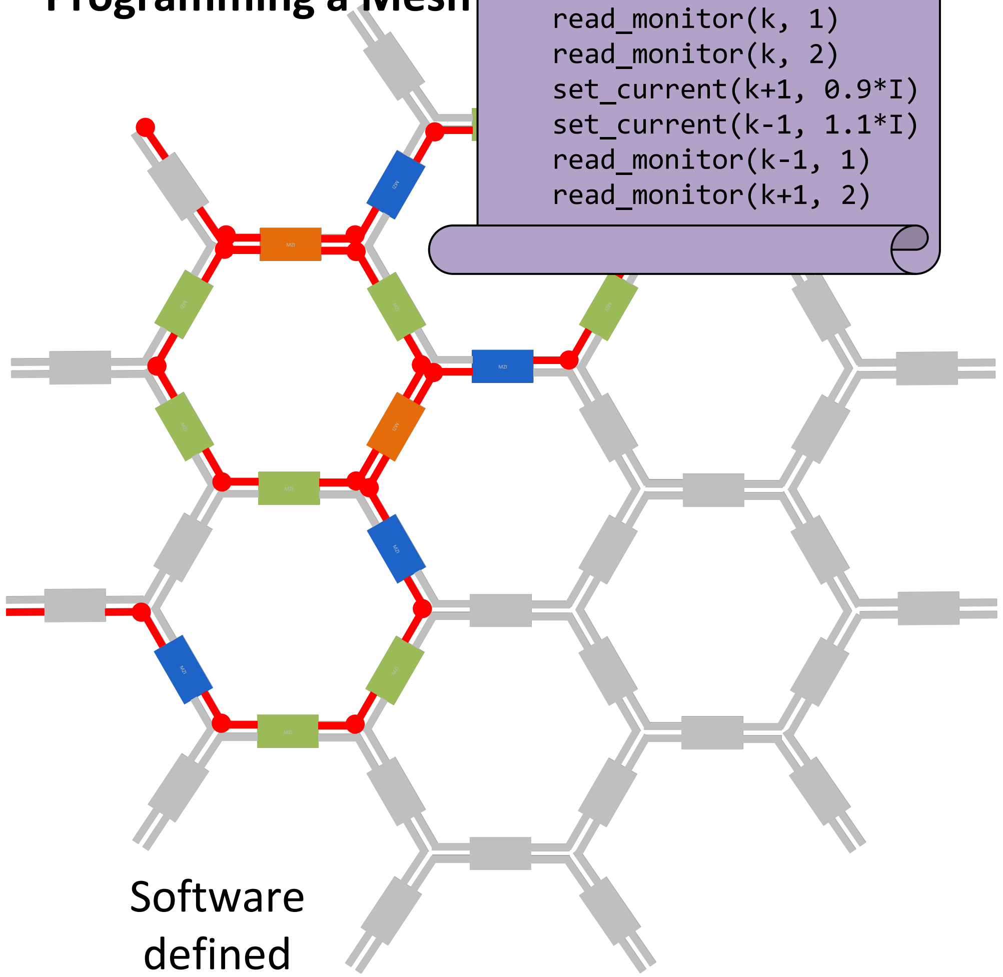


PDK-based Circuit Design



Custom circuit design with standard tunable couplers and phase shifters

Programming a Mesh



```
for k in range(N):
    set_current(k, I)
    read_monitor(k, 1)
    read_monitor(k, 2)
    set_current(k+1, 0.9*I)
    set_current(k-1, 1.1*I)
    read_monitor(k-1, 1)
    read_monitor(k+1, 2)
```

Software defined functionality

NEW TYPES OF IP

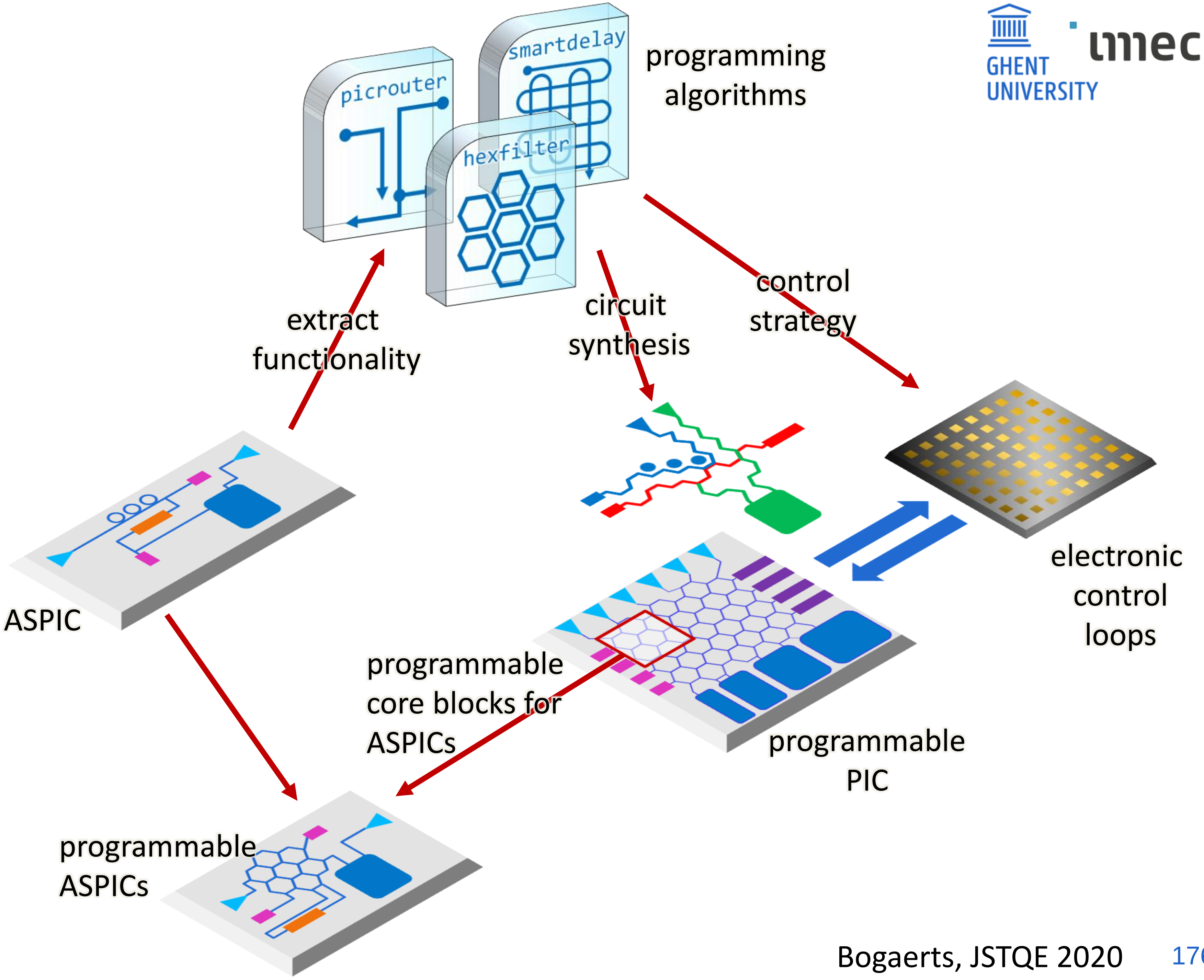
Programming routines

Circuit synthesis

Control strategies

Pluggable design IP

- linear cores
- electronic controls



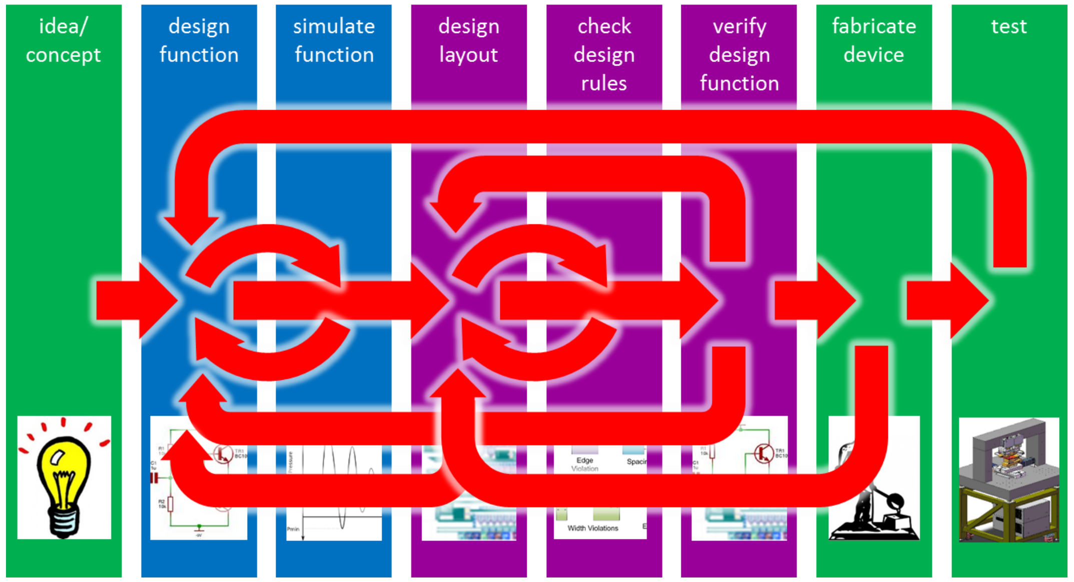
SUMMARY

(Silicon) Photonics is growing towards a circuit platform

- Technology supports larger circuits
- A circuit-oriented design flow is emerging (similar to electronics)
- Fabs are building PDKs

Challenges

- Schematic-driven Layout for photonics
- Variability: fabrication, performance, models
- Verification: DRC and LVS
- Design for manufacturability
- Photonic-electronic-software stacks
- New design methods for programmable photonics



THE SIEPIC TECHNOLOGY

FABRICATE A DEVICE

Get your design fabricated

- coordinated by Lukas Chrostowski of the University of British Columbia (Short Course SC432 - **Cancelled**)
- e-beam lithography at Applied Nanotools (http://www.appliednt.com/nanosoi/)
- chips are measured at UBC or Maple Leaf Photonics
- you can analyse the measurement data (you will not receive the actual chip)

POSTPONED OR CANCELLED

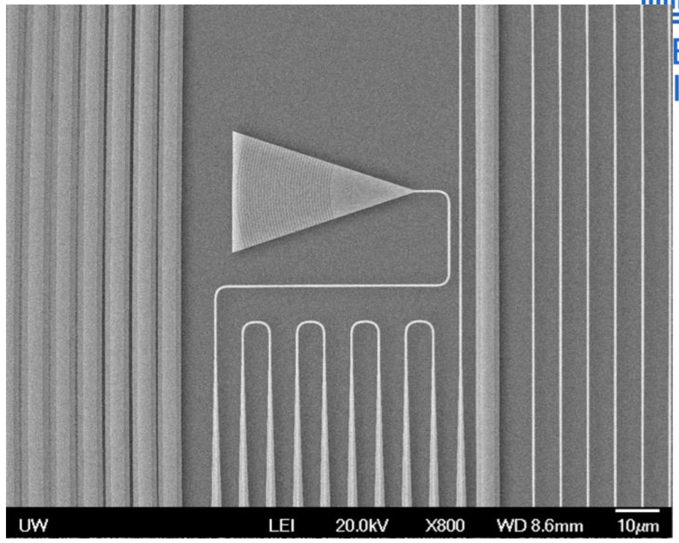
SIEPIC - EBEAM

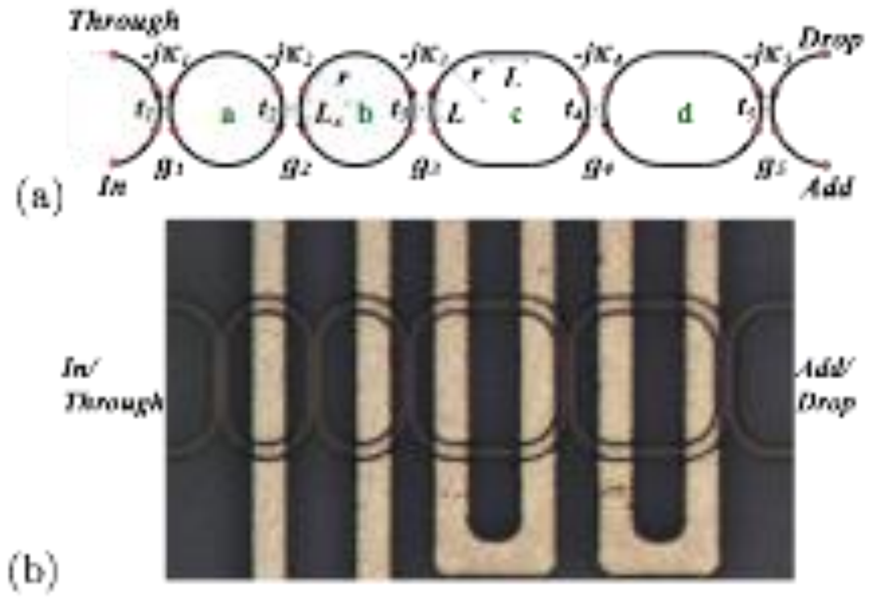
Mature processes at

- University of Washington (2011-)
- Applied Nanotools (2014-).

Organized by Lukas Chrostowski (UBC)

- over 40 MPW runs
- used extensively in courses
- low marginal costs
- automated measurements at UBC



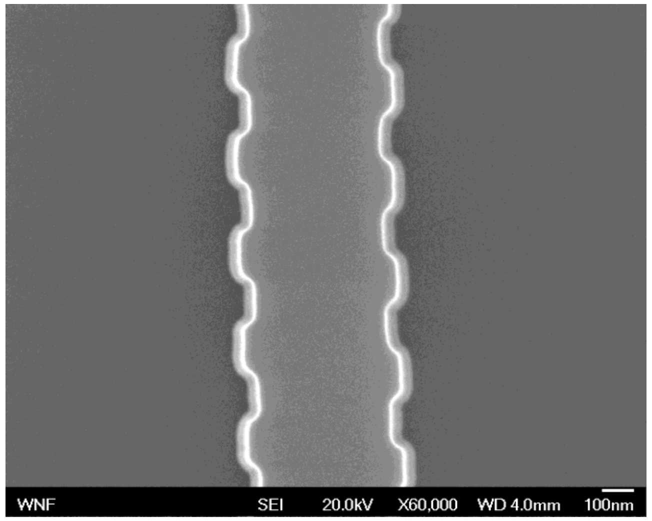


Robert Boeck, et al, *Optics Letters*, 07/2013

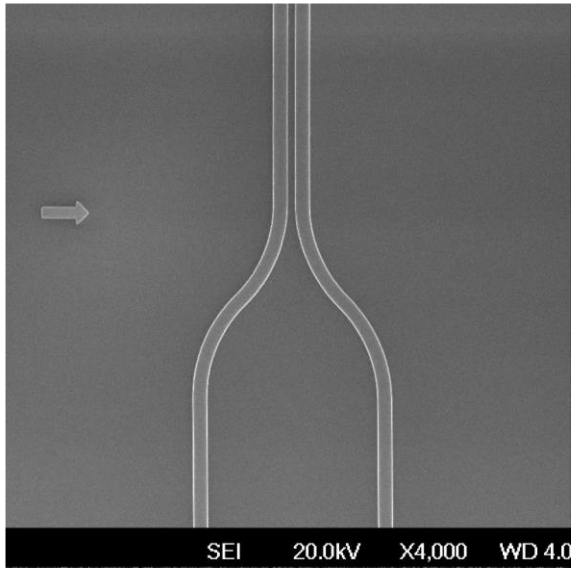
Si-EPIC

Updated: 2014/07/05

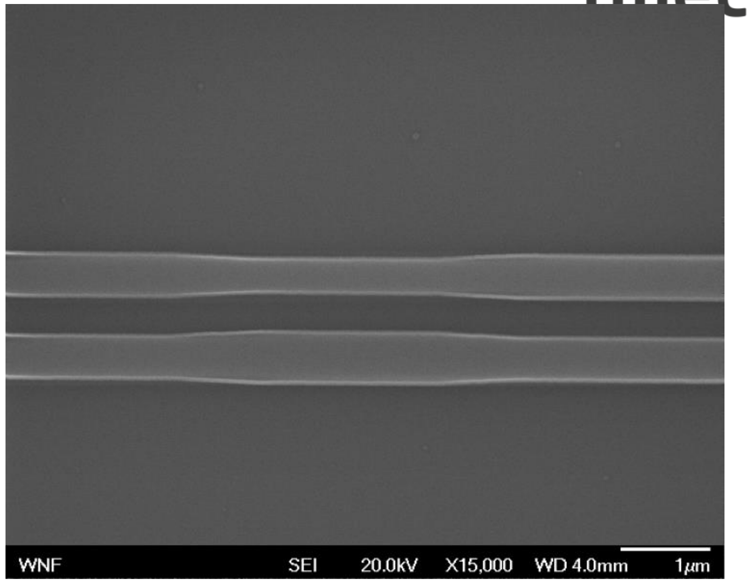
COMPONENTS



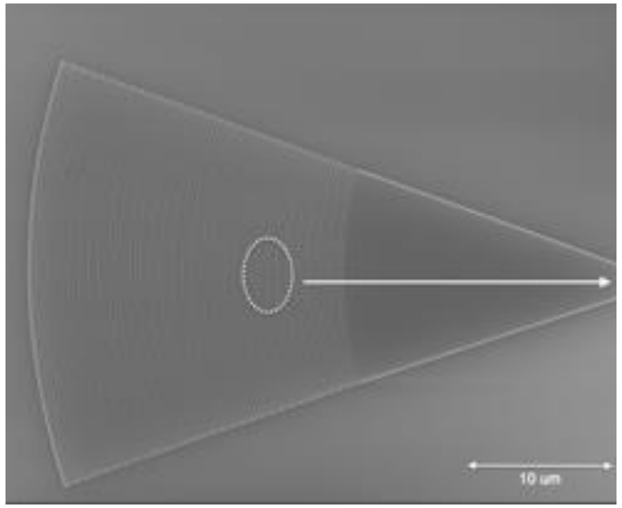
Bragg grating

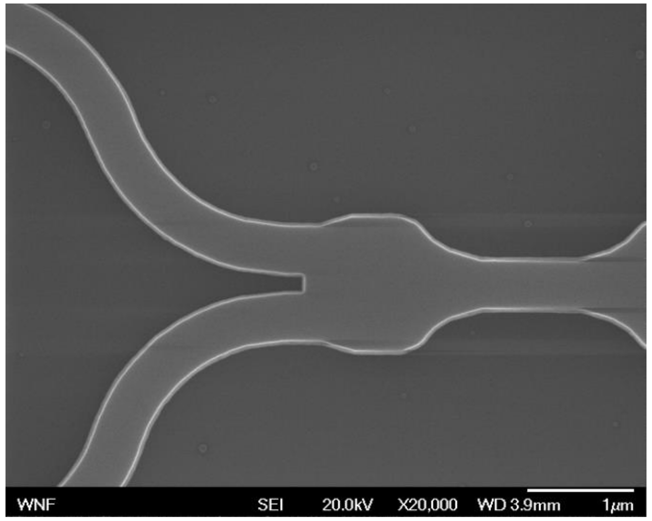


Directional Coupler

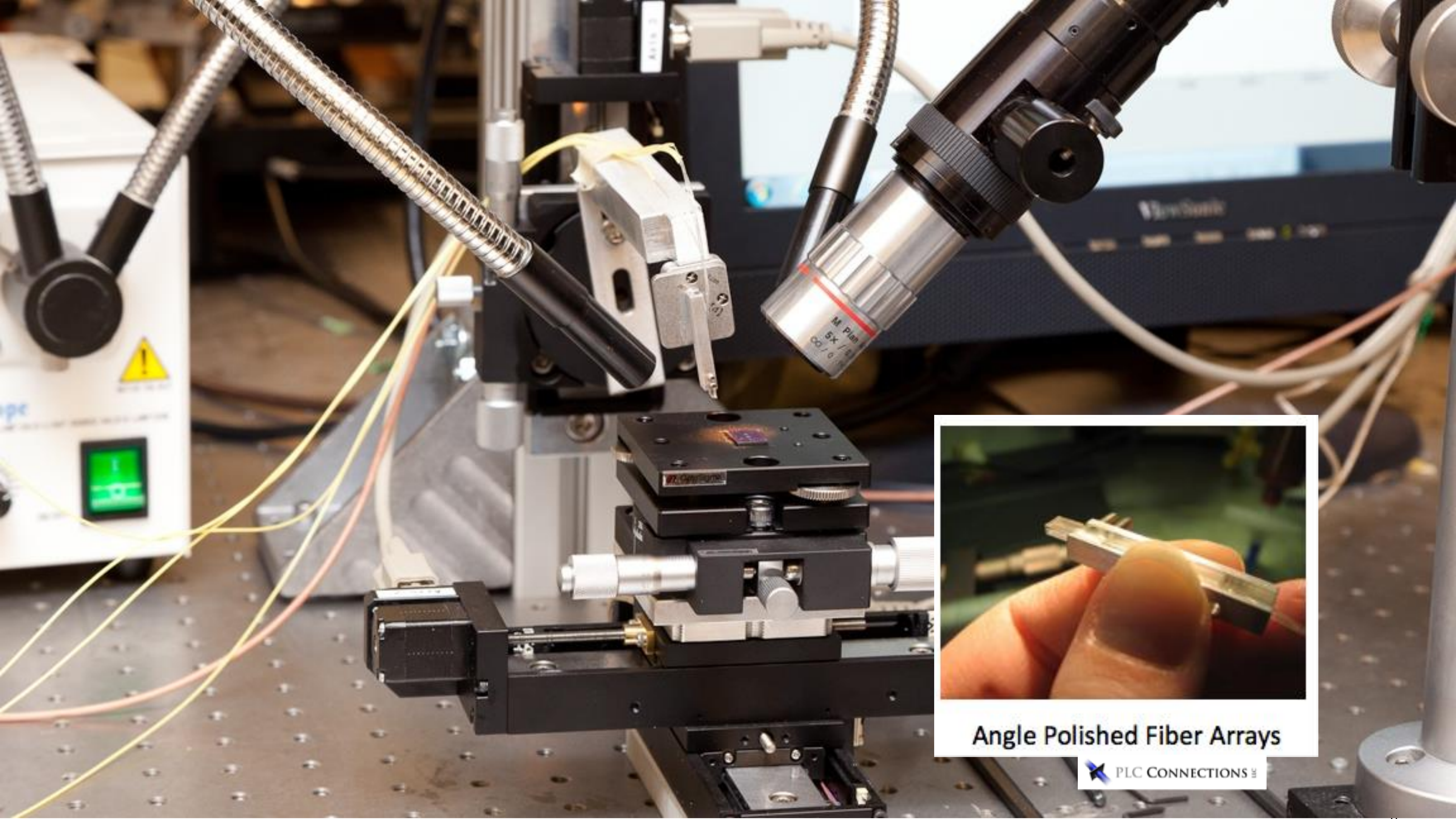


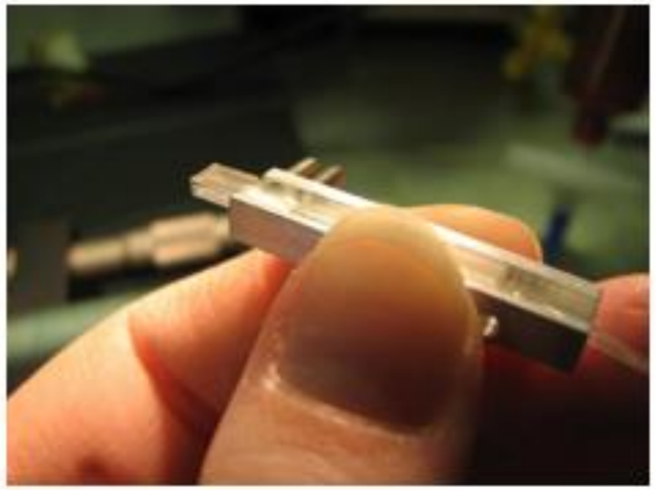
Broad-band Directional Couplers





Splitters





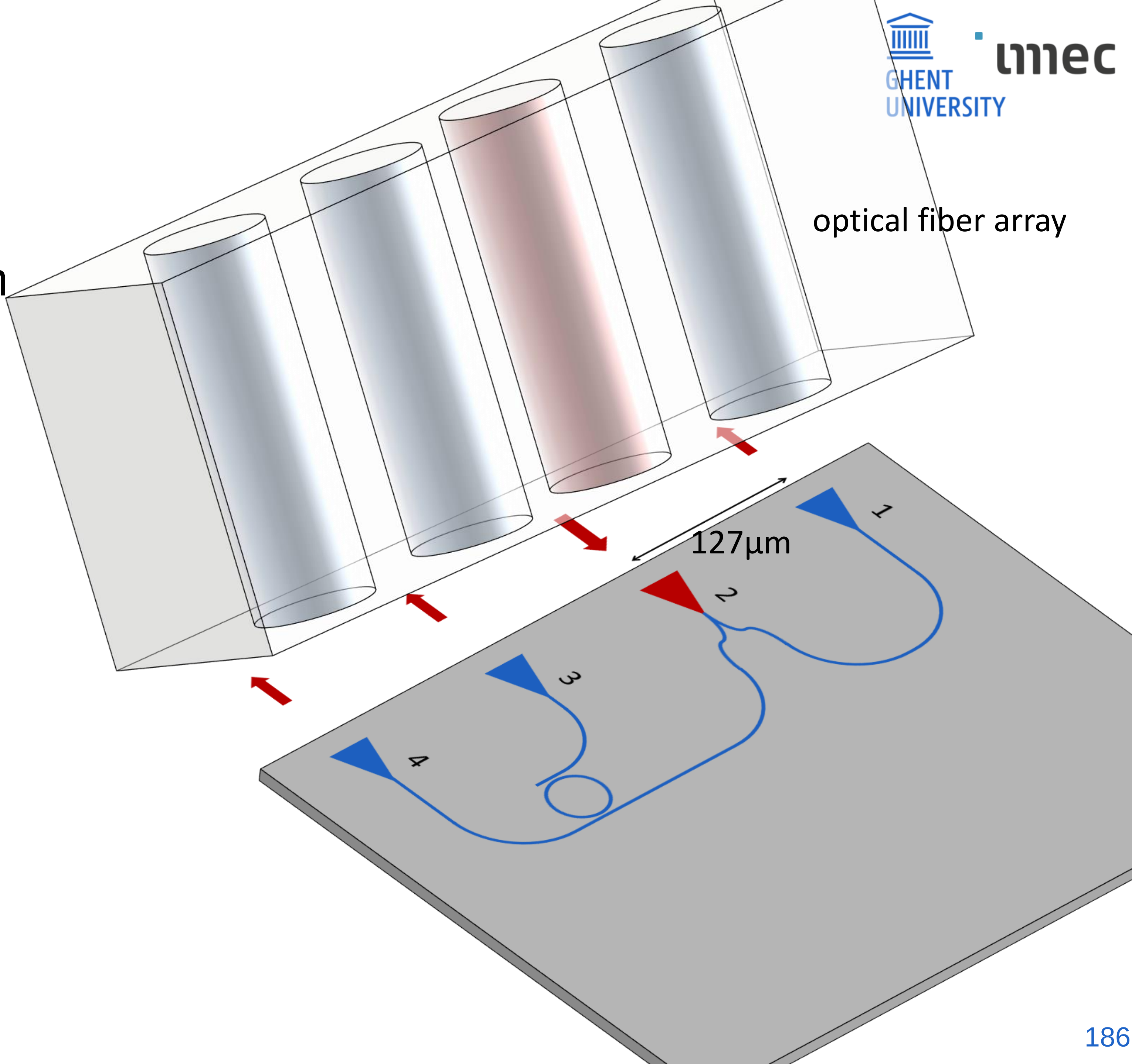
Angle Polished Fiber Arrays

PLC CONNECTIONS

MEASUREMENTS

Using a fiber array

- wavelength transmission from input (2) to all outputs (1,3,4)



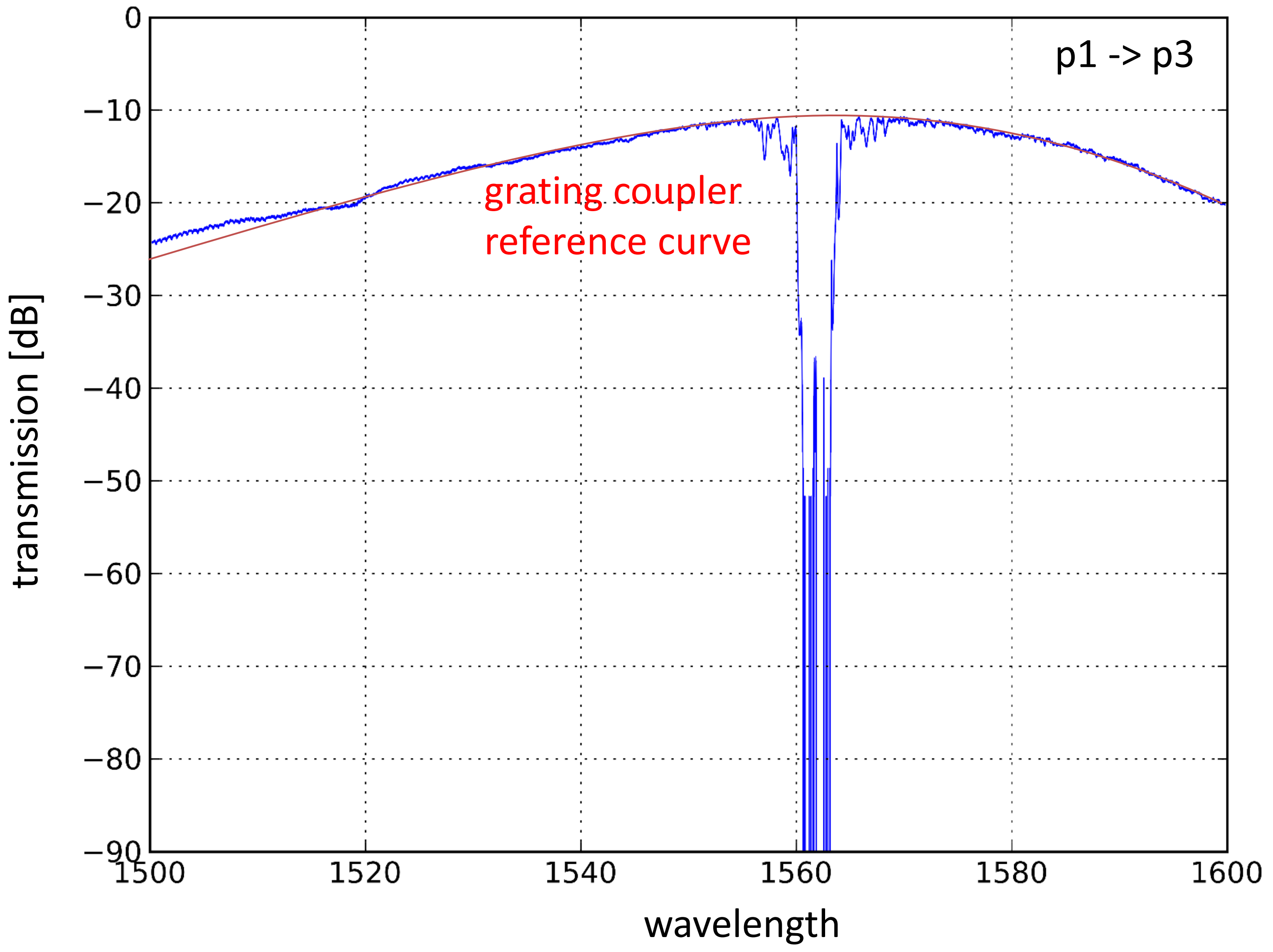
MEASUREMENT RESULTS

Transmission data

- wavelength
- transmitted power

No Optical phase

(very hard to measure)



Practical Setup

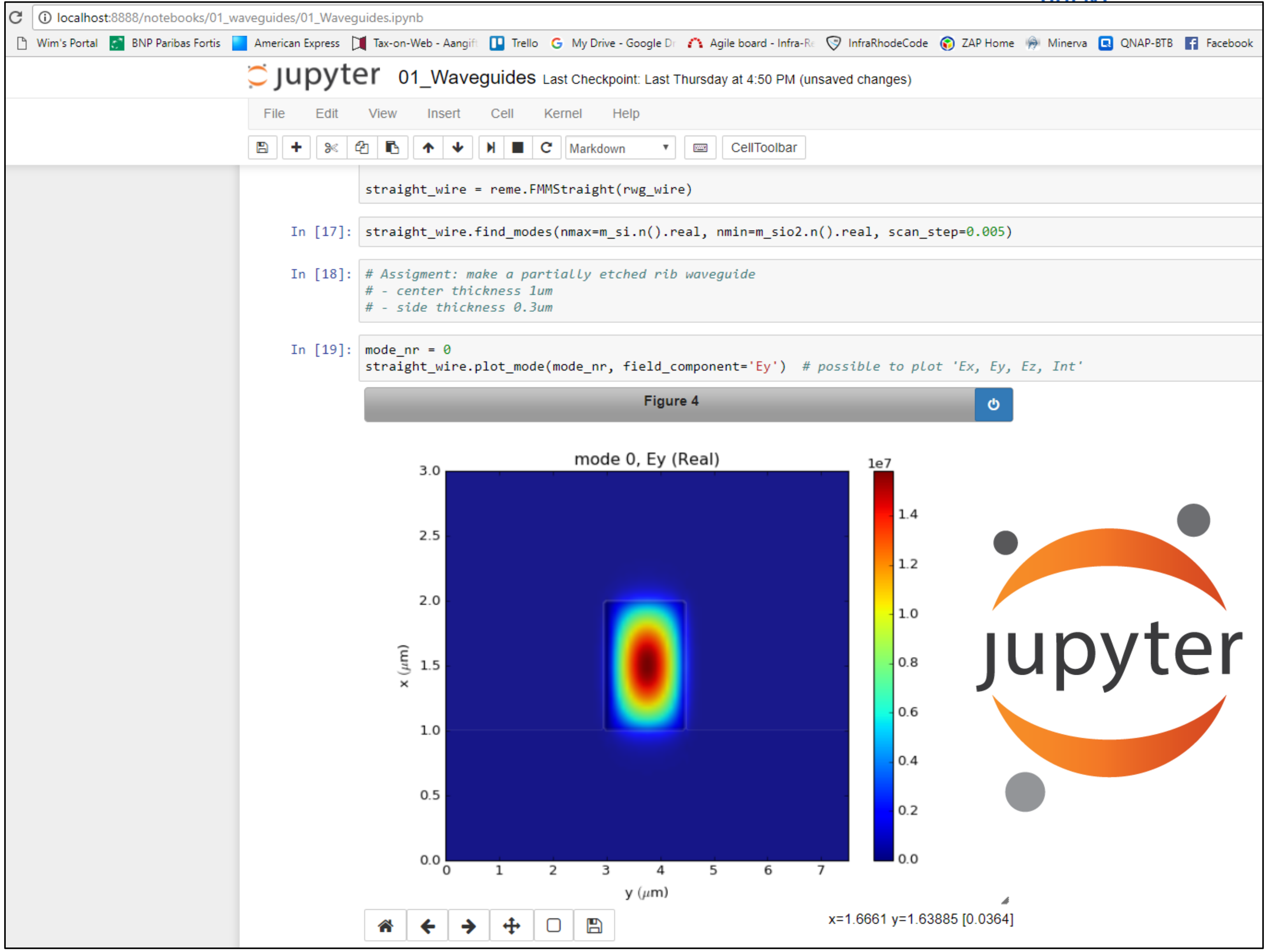
JUPYTER NOTEBOOKS

interactive notebook

- text, figures
- formulas
- python code

simulation and design

- built-in IPKISS



```
straight_wire = reme.FMMStraight(rwg_wire)

In [17]: straight_wire.find_modes(nmax=m_si.n().real, nmin=m_sio2.n().real, scan_step=0.005)

In [18]: # Assigment: make a partially etched rib waveguide
         # - center thickness 1um
         # - side thickness 0.3um

In [19]: mode_nr = 0
         straight_wire.plot_mode(mode_nr, field_component='Ey')  # possible to plot 'Ex, Ey, Ez, Int'
```

THE IPKISS DESIGN FRAMEWORK

Design framework for Photonic Integrated Circuits

- Parametric design
- Focus on reuse and automation

History

- Developed at Ghent University – imec in 2000-2014
- Spin-off into Luceda Photonics in 2014
- Currently thousands of users worldwide

The IPKISS Design framework

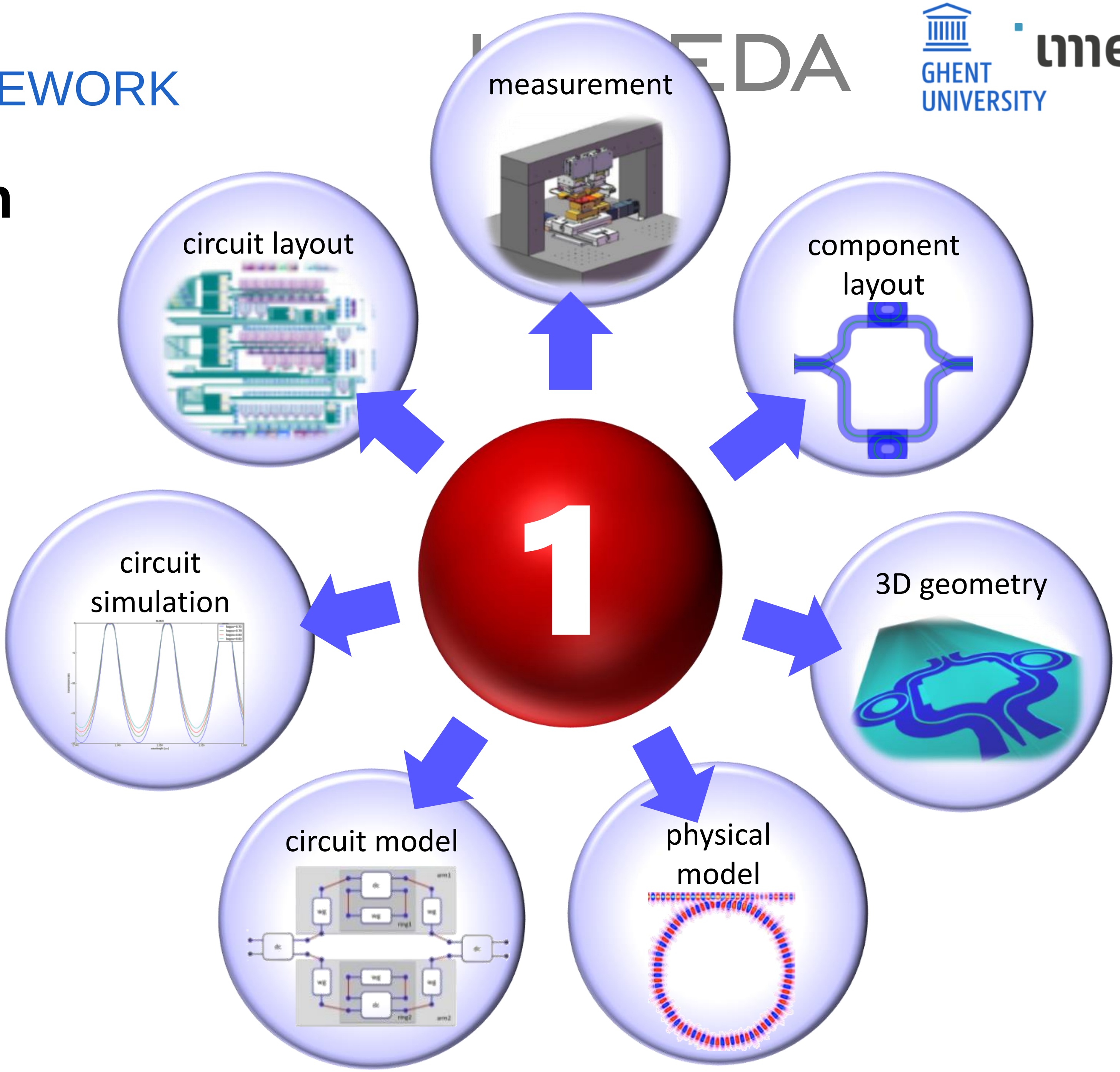
One component definition

for

Circuit design

Layout

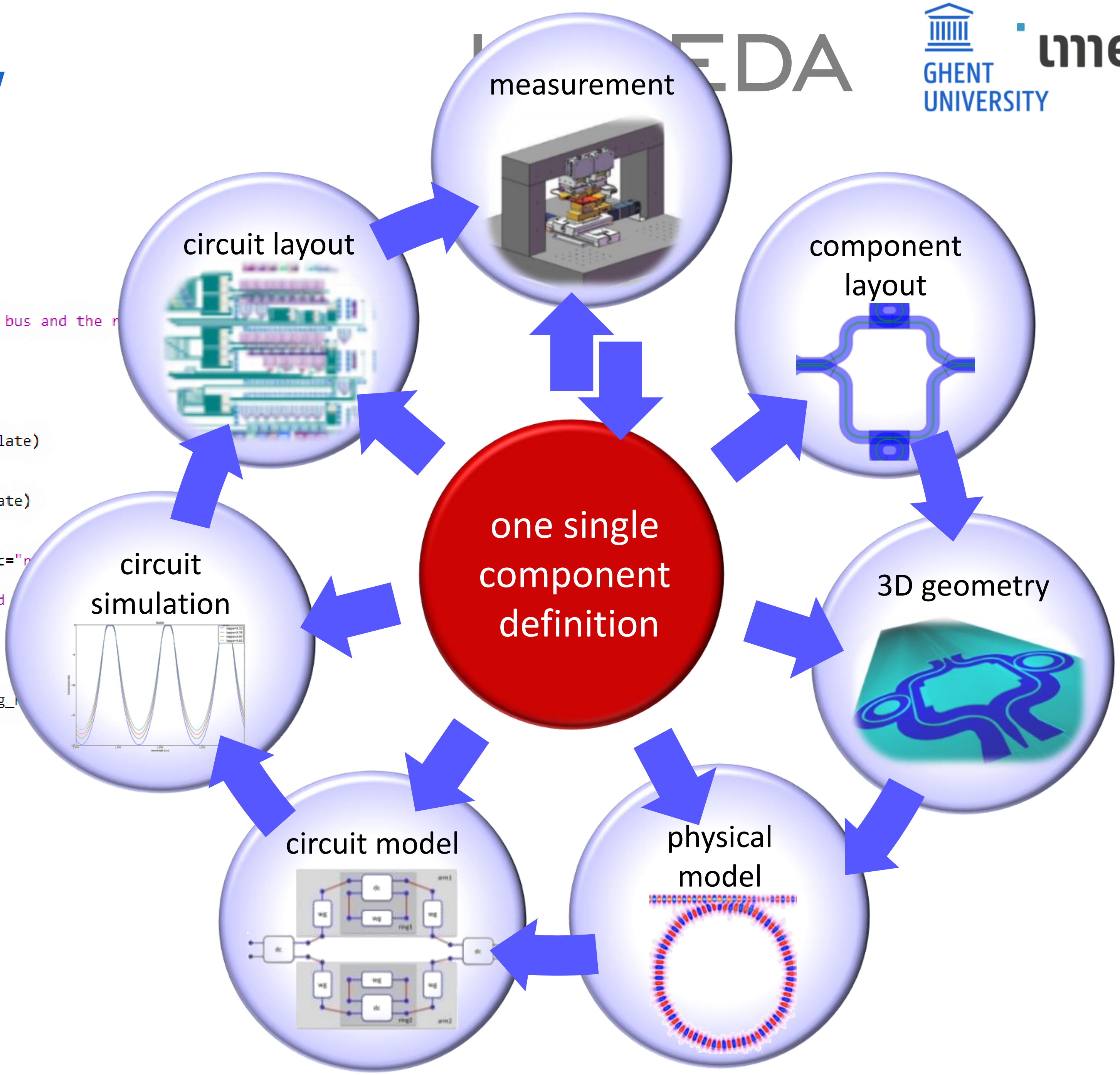
Simulation



THE IPKISS DESIGN FLOW

Python script based

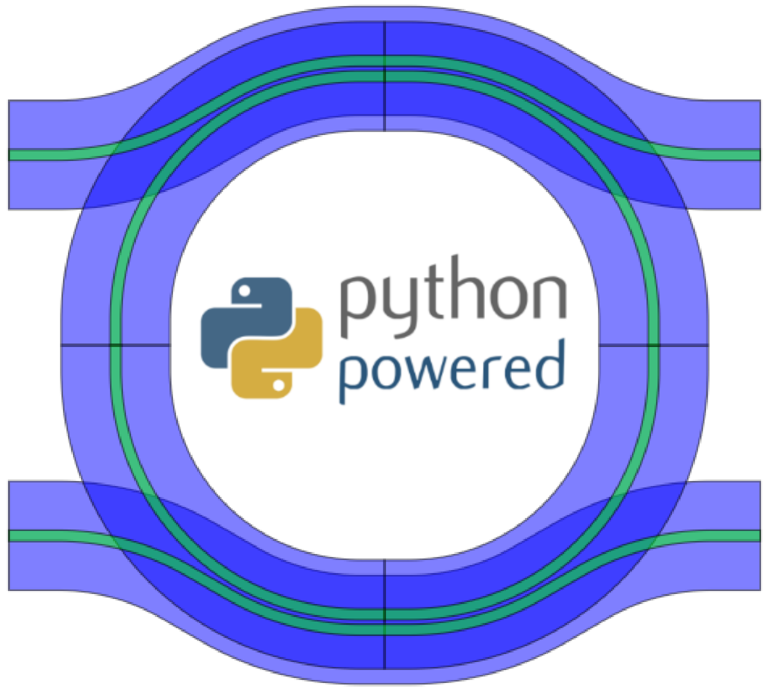
```
class RingResonator(i3.PCell):
    """A generic ring resonator class."""

    wg_template = i3.WaveguideTemplateProperty(default=TECH.PCELLS.WG.DEFAULT,
                                               doc="trace template used for the bus and the r

    bus = i3.ChildCellProperty(doc="bus waveguide")
    ring = i3.ChildCellProperty(doc="ring waveguide")

    def _default_ring(self):
        return i3.Waveguide(name=self.name+"_ring", trace_template=self.wg_template)

    def _default_bus(self):
        return i3.Waveguide(name=self.name+"_bus", trace_template=self.wg_template)

    class Layout(i3.LayoutView):
        ring_radius = i3.PositiveNumberProperty(default=TECH.WG.BEND_RADIUS, doc="r
        coupler_spacing = i3.PositiveNumberProperty(default=TECH.WG.DC_SPACING,
                                                    doc="spacing between bus and

        def _default_ring(self):
            ring_layout = self.cell.ring.get_default_view(i3.LayoutView)
            ring_layout.set(trace_template=self.wg_template,
                            shape=i3.ShapeCircle(center=(0, 0), radius=self.ring_r
            return ring_layout

        def _default_bus(self):
            r, s = self.ring_radius, self.coupler_spacing
            bus_layout = self.cell.bus.get_default_view(i3.LayoutView)
            bus_layout.set(trace_template=self.wg_template,
                           shape=[(-r, -r-s), (+r, -r-s)])
            return bus_layout

        def _generate_instances(self, insts):
            insts += i3.SRef(name="ring", reference=self.ring)
            insts += i3.SRef(name="bus", reference=self.bus)
            return insts

        def _generate_ports(self, ports):
            ports += self.instances["bus"].ports
            return ports
```



THE IPKISS DESIGN FLOW

Python script based

```python
class RingResonator(i3.PCell):
    """A generic ring resonator class."""

    wg_template = i3.WaveguideTemplateProperty(default=TECH.PCELLS.WG.DEFAULT,
                                               doc="trace template used for the bus and the ring")

    bus = i3.ChildCellProperty(doc="bus waveguide")
    ring = i3.ChildCellProperty(doc="ring waveguide")

    def _default_ring(self):
        return i3.Waveguide(name=self.name+"_ring", trace_template=self.wg_template)

    def _default_bus(self):
        return i3.Waveguide(name=self.name+"_bus", trace_template=self.wg_template)

    class Layout(i3.LayoutView):
        ring_radius = i3.PositiveNumberProperty(default=TECH.WG.BEND_RADIUS, doc="radius of ring")
        coupler_spacing = i3.PositiveNumberProperty(default=TECH.WG.DC_SPACING,
                                                    doc="spacing between bus and ring waveguide")

        def _default_ring(self):
            ring_layout = self.cell.ring.get_default_view(i3.LayoutView)
            ring_layout.set(trace_template=self.wg_template,
                            shape=i3.ShapeCircle(center=(0, 0), radius=self.ring_radius))
            return ring_layout

        def _default_bus(self):
            r, s = self.ring_radius, self.coupler_spacing
            bus_layout = self.cell.bus.get_default_view(i3.LayoutView)
            bus_layout.set(trace_template=self.wg_template,
                           shape=[(-r, -r-s), (+r, -r-s)])
            return bus_layout

        def _generate_instances(self, insts):
            insts += i3.SRef(name="ring", reference=self.ring)
            insts += i3.SRef(name="bus", reference=self.bus)
            return insts

        def _generate_ports(self, ports):
            ports += self.instances["bus"].ports
            return ports
```



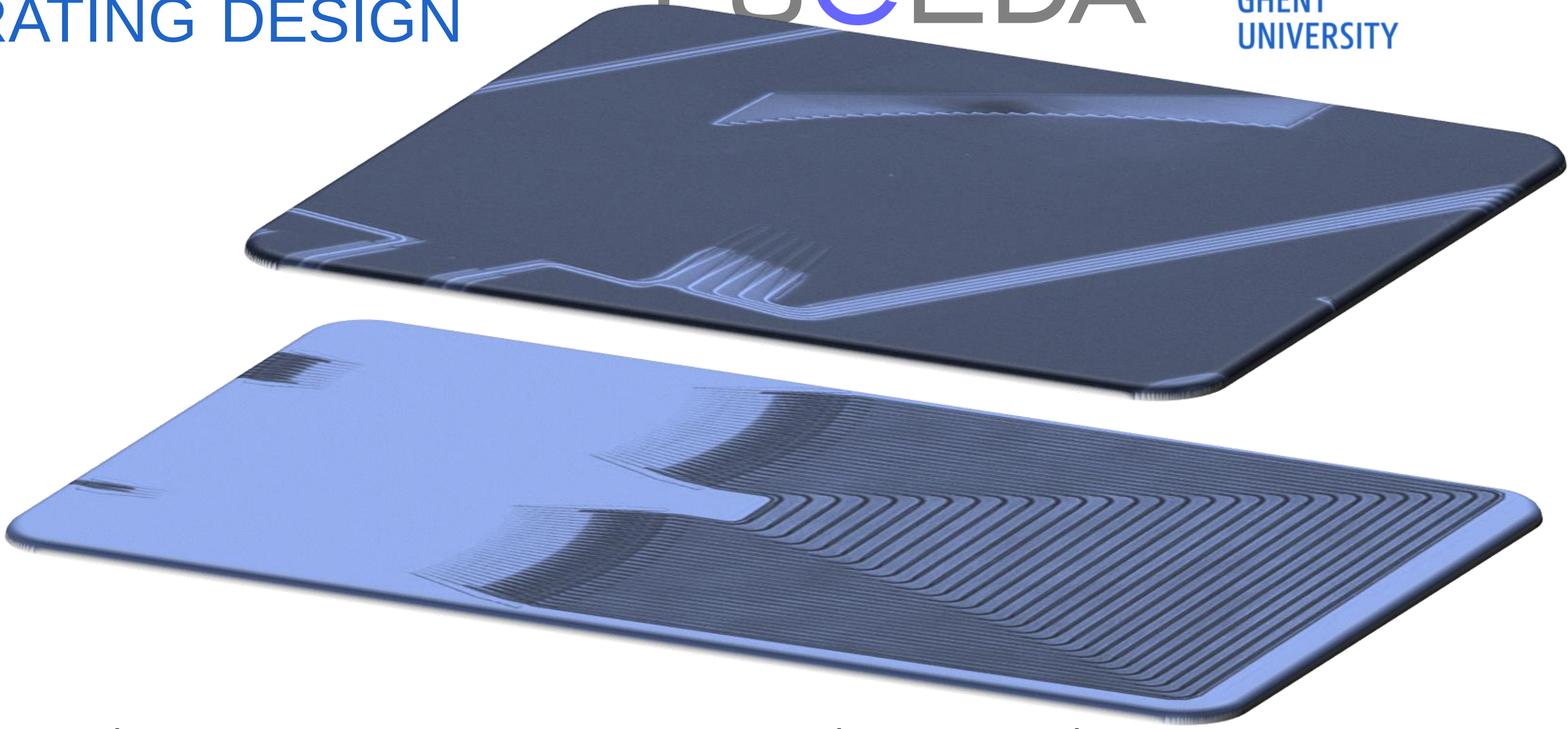
- extremely flexible
- easy-to-read
- powerful engineering libraries
- industry standard

ARRAYED WAVEGUIDE GRATING DESIGN

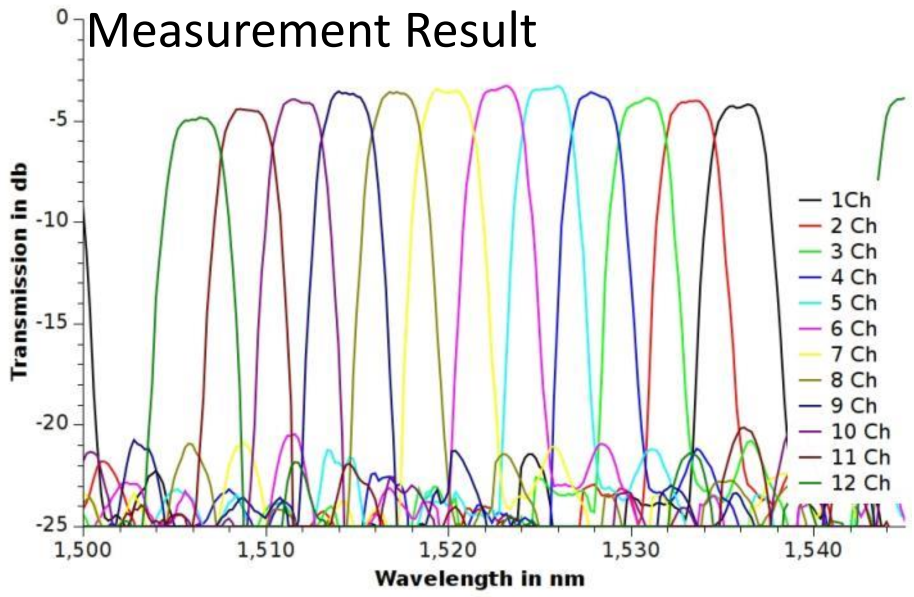
Arrayed Waveguide Gratings

Echelle Gratings

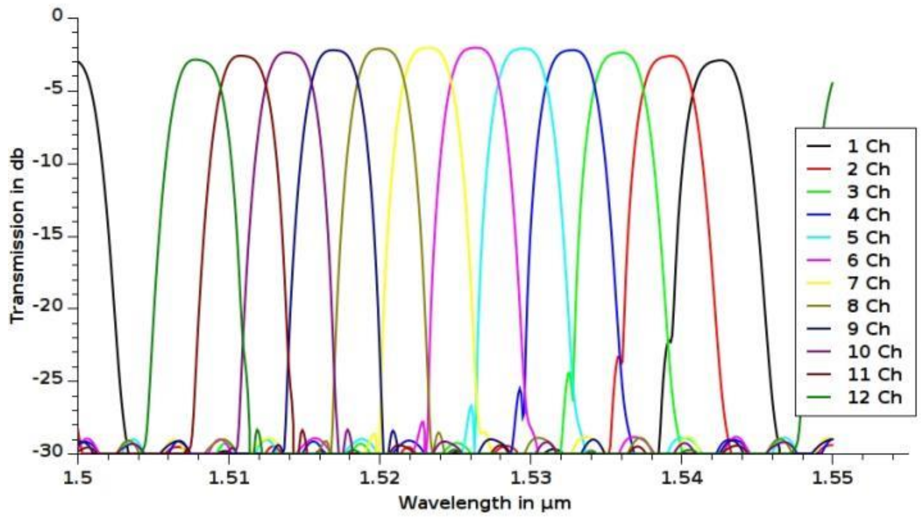
- Fully parametric
- Design from specifications
- Integrated layout and simulation
- Validated on fabricated devices



Measurement Result



Simulation Result



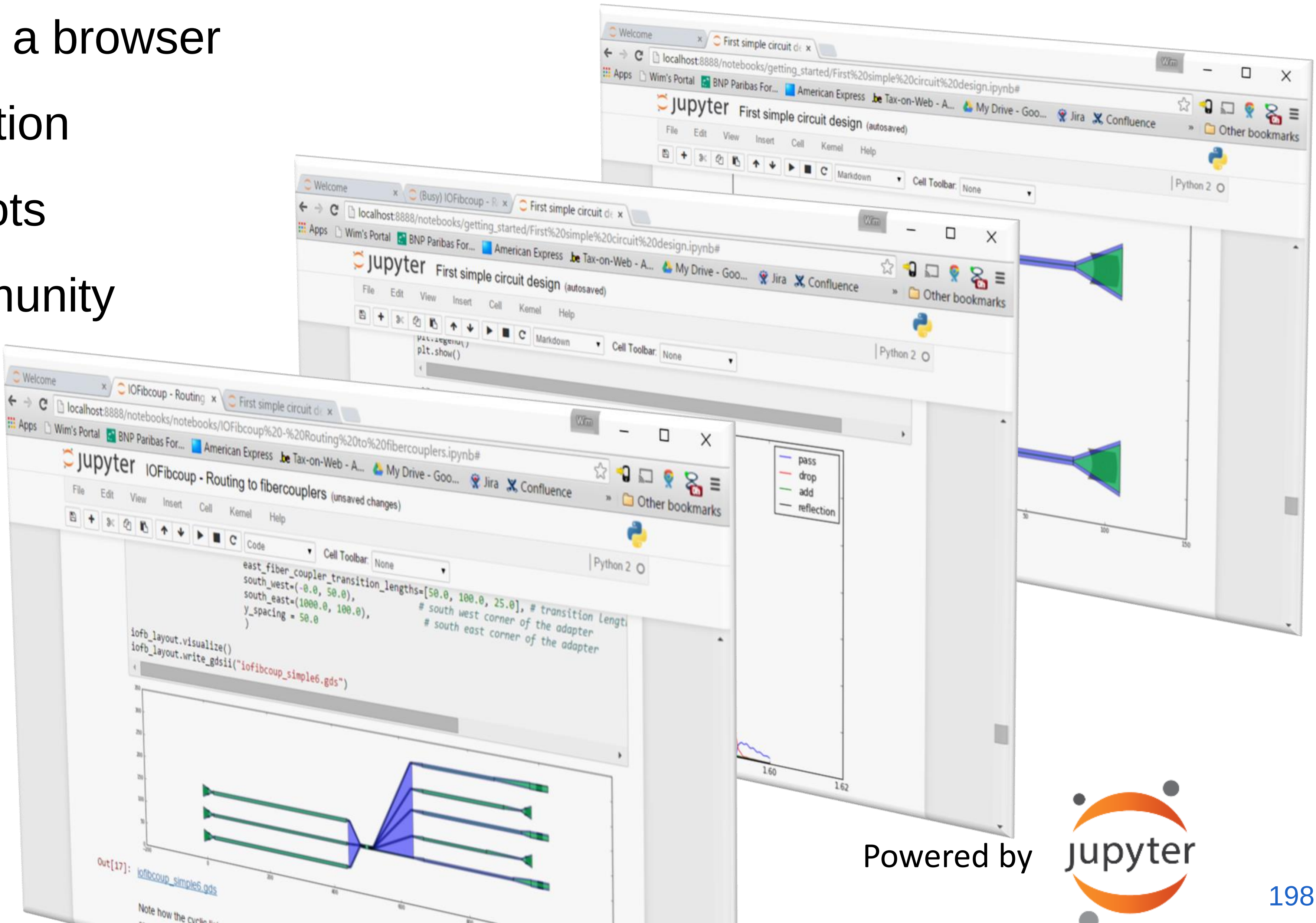
IPKISS NOTEBOOKS

Explore your designs in a browser

Very rapid experimentation

Interactive code and plots

Widely supported community


jupyter
First simple circuit design (autosaved)
IOFibcoup - Routing to fibercouplers (unsaved changes)
localhost:8888/notebooks/notebooks/IOFibcoup%20-%20Routing%20to%20fibercouplers.ipynb#
localhost:8888/notebooks/getting_started/First%20simple%20circuit%20design.ipynb#
File Edit View Insert Cell Kernel Help
Cell Toolbar: None
Python 2
pass
drop
add
reflection

```
        east_fiber_coupler_transition_lengths=[50.0, 100.0, 25.0], # transition lengt
        south_west=(-0.0, 50.0),                    # south west corner of the adapter
        south_east=(1000.0, 100.0),                 # south east corner of the adapter
        y_spacing = 50.0
        )
iofb_layout.visualize()
iofb_layout.write_gdsii("iofibcoup_simple6.gds")
```

```
plt.legend()
plt.show()
```

Out[17]: iofibcoup_simple6.gds

Note how the cycle

Powered by jupyter

FIRST NOTEBOOKS

Unfamiliar with Python?

/01_01_jupyter_notebooks: *How to use a notebook*

/01_02_python_getting_started: *basic Python tutorial*

/01_03_ numpy_and_plotting: *Numpy and Matplotlib*

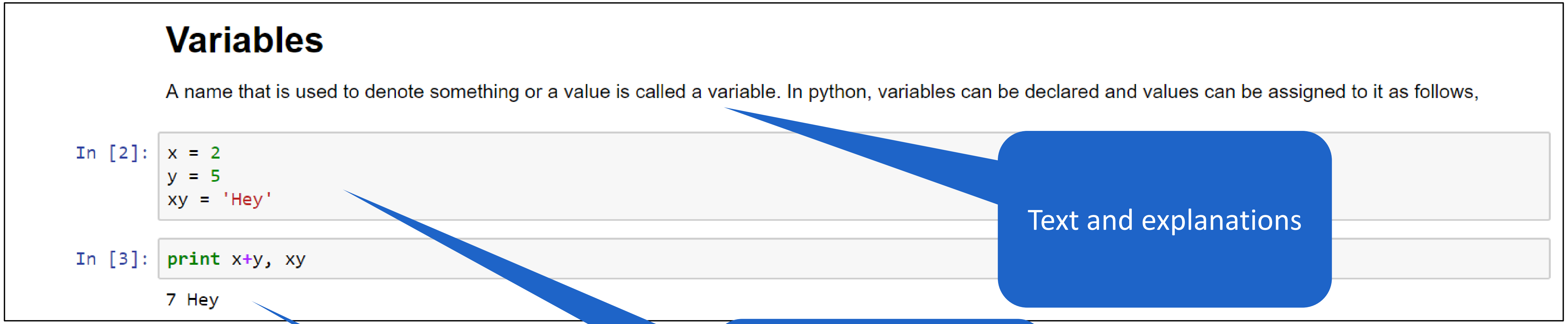
Check if everything works and if you find your way around the notebook interface.

PRACTICAL

1. Open web browser (Chrome, Firefox, Opera)
2. Connect to Jupyter server:

   https://wscarapils.intec.ugent.be
3. Log in with your personal ID/password

NOTEBOOK: INTERACTIVE ENVIRONMENT



Variables

A name that is used to denote something or a value is called a variable. In python, variables can be declared and values can be assigned to it as follows,

```
In [2]: x = 2
        y = 5
        xy = 'Hey'

In [3]: print x+y, xy

        7 Hey
```

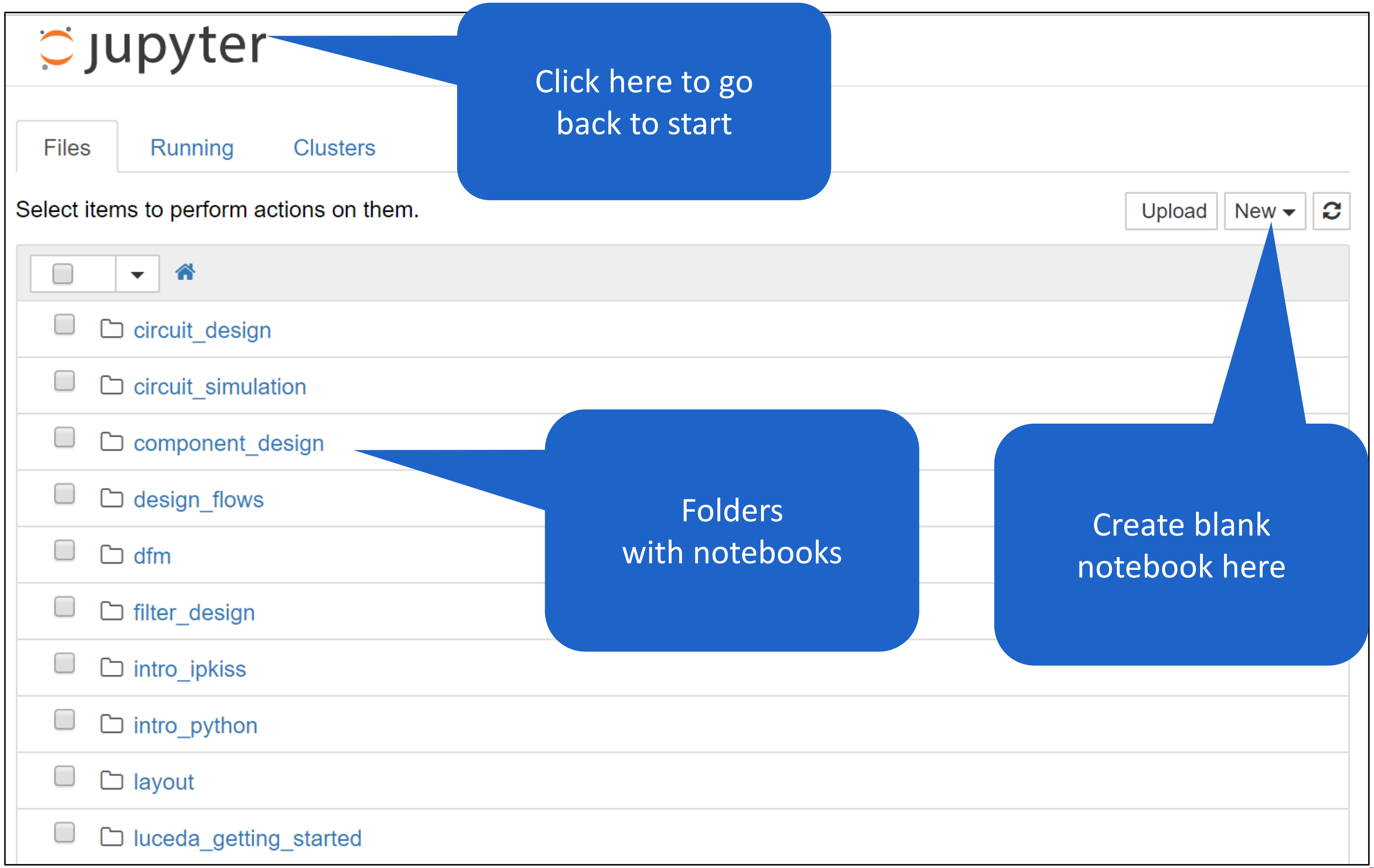
Text and explanations

Executable python code

SHIFT+ENTER

to execute

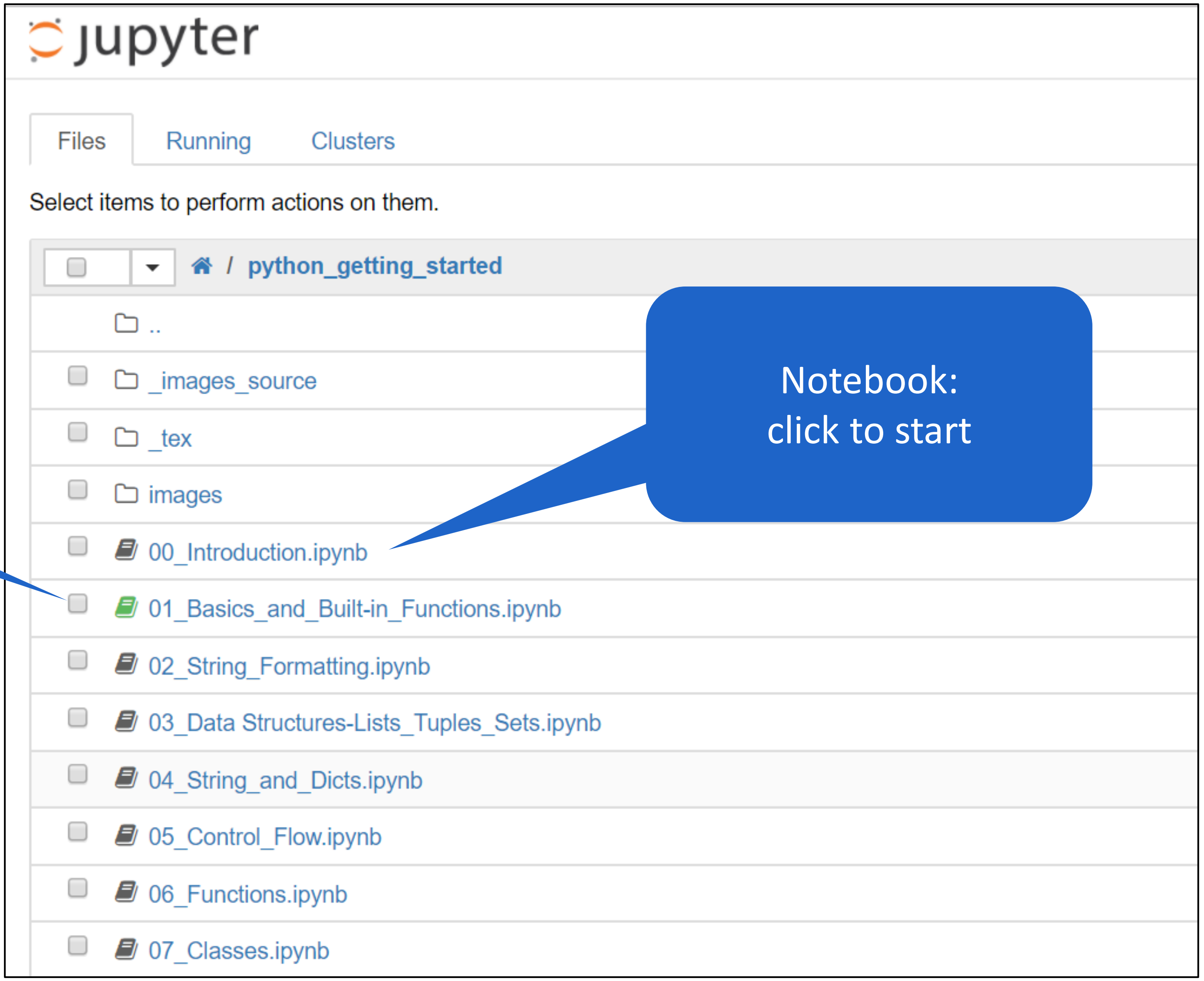
Navigating



Select items to perform actions on them.

circuit_design
circuit_simulation
component_design
design_flows
dfm
filter_design
intro_ipkiss
intro_python
layout
luceda_getting_started

NAVIGATING



jupyter

Files Running Clusters

Select items to perform actions on them.

/ python_getting_started

..
_images_source
_tex
images
00_Introduction.ipynb
01_Basics_and_Built-in_Functions.ipynb
02_String_Formatting.ipynb
03_Data Structures-Lists_Tuples_Sets.ipynb
04_String_and_Dicts.ipynb
05_Control_Flow.ipynb
06_Functions.ipynb
07_Classes.ipynb

Running Notebook

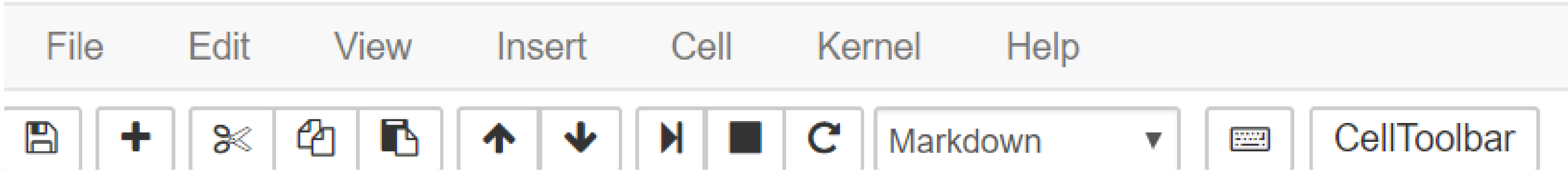
Notebook: click to start

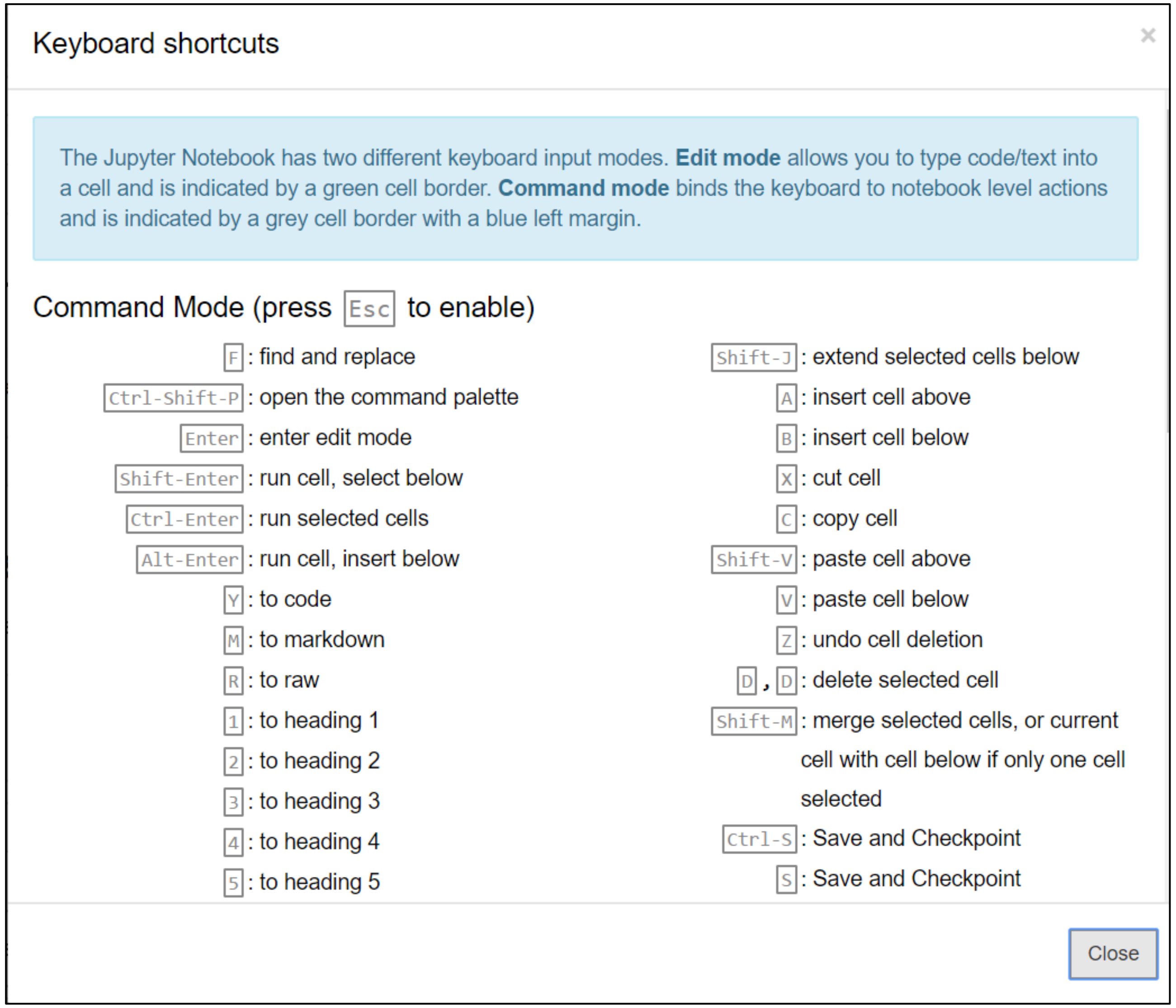
PRESS H FOR 'HELP'

Useful menu and toolbar

Keyboard shortcuts

are extremely powerful





Keyboard shortcuts

The Jupyter Notebook has two different keyboard input modes. **Edit mode** allows you to type code/text into a cell and is indicated by a green cell border. **Command mode** binds the keyboard to notebook level actions and is indicated by a grey cell border with a blue left margin.

Command Mode (press `Esc` to enable)

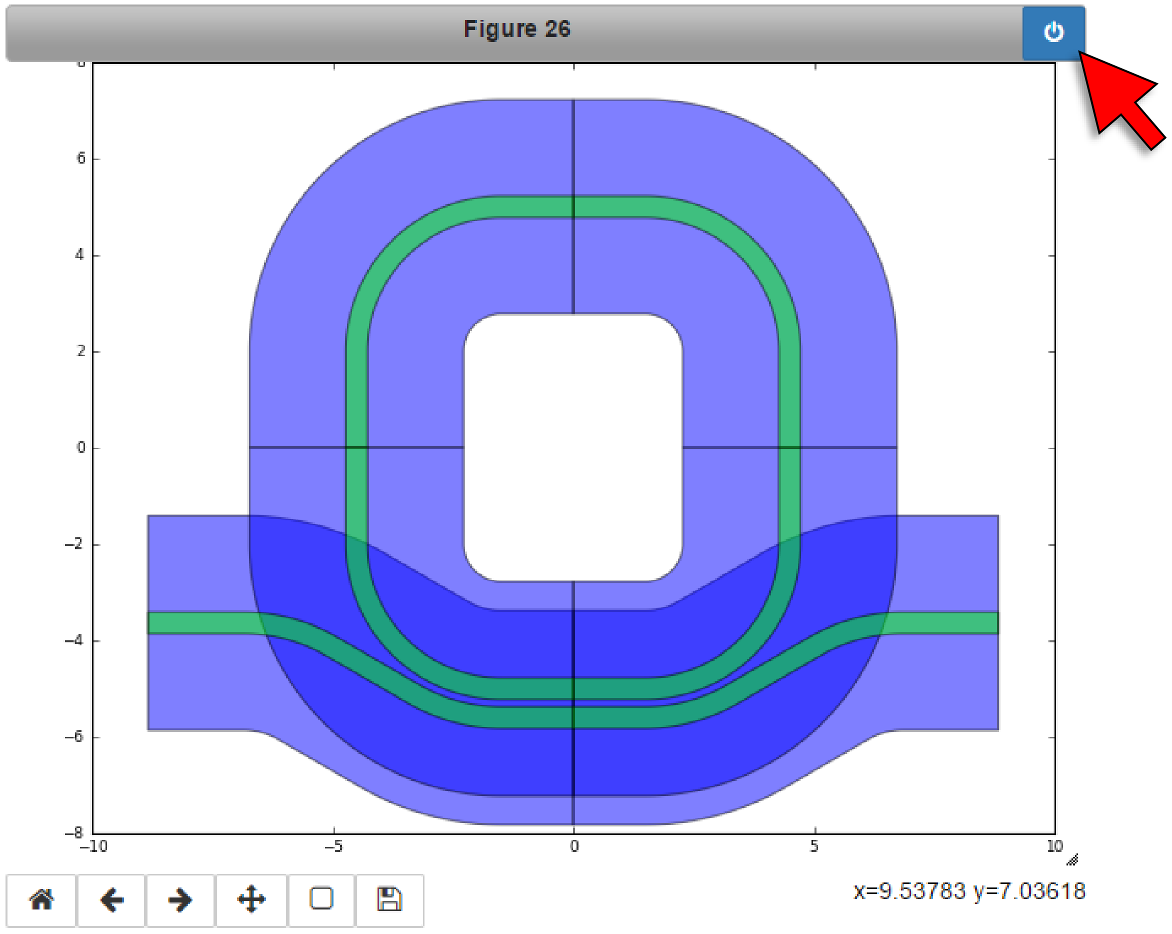
`F`	find and replace	`Shift-J`	extend selected cells below
`Ctrl-Shift-P`	open the command palette	`A`	insert cell above
`Enter`	enter edit mode	`B`	insert cell below
`Shift-Enter`	run cell, select below	`X`	cut cell
`Ctrl-Enter`	run selected cells	`C`	copy cell
`Alt-Enter`	run cell, insert below	`Shift-V`	paste cell above
`Y`	to code	`V`	paste cell below
`M`	to markdown	`Z`	undo cell deletion
`R`	to raw	`D`,`D`	delete selected cell
`1`	to heading 1	`Shift-M`	merge selected cells, or current cell with cell below if only one cell selected
`2`	to heading 2	`Ctrl-S`	Save and Checkpoint
`3`	to heading 3	`S`	Save and Checkpoint
`4`	to heading 4		
`5`	to heading 5		

TAKE CARE OF MEMORY

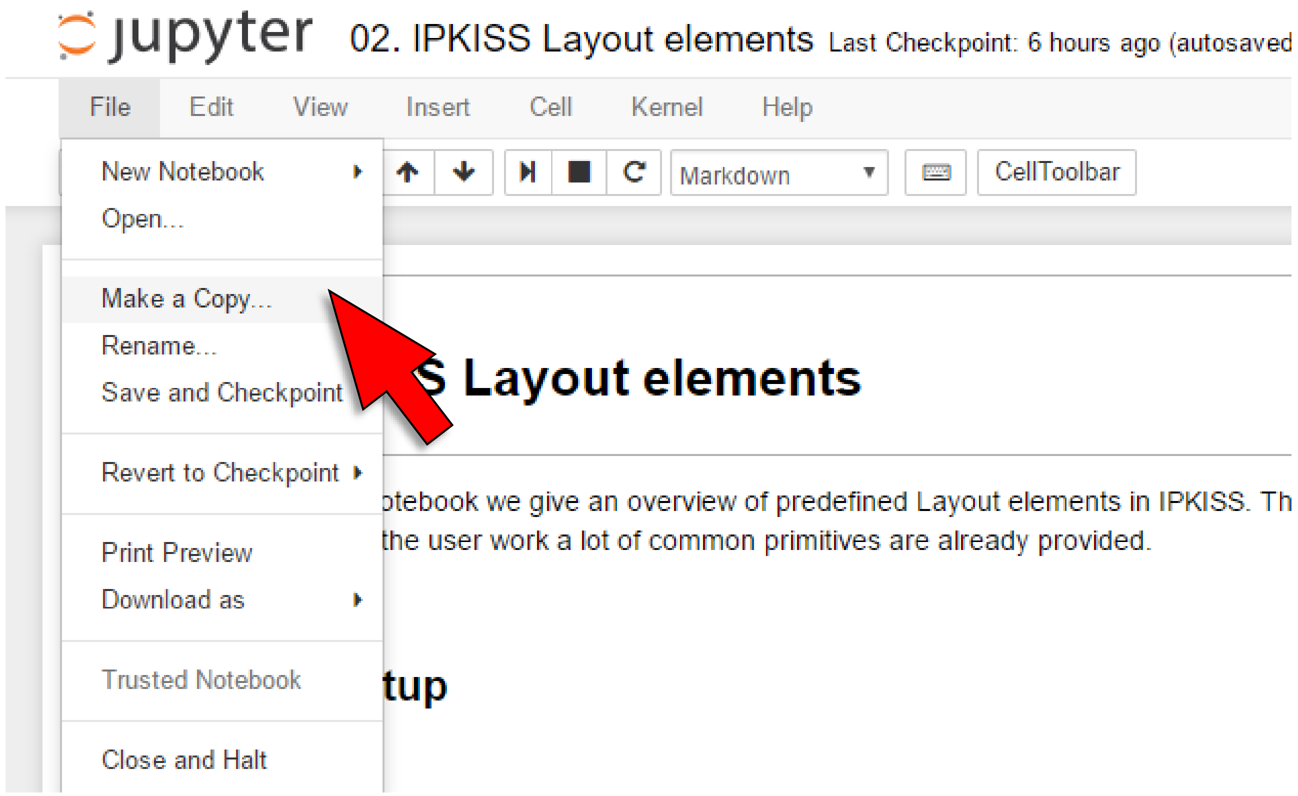
Interactive plots consume resources.

Close them when ready.

```
C:\luceda\ipkiss_311\python\envs\ipkiss3\lib\site-packages\matplotlib\pyplot.py:516: RuntimeWarning: More than 20 figures have
 been opened. Figures created through the pyplot interface (`matplotlib.pyplot.figure`) are retained until explicitly closed an
d may consume too much memory. (To control this warning, see the rcParam `figure.max_open_warning`).
  max_open_warning, RuntimeWarning)
```



GETTING STARTED…



- open browser (Chrome, Firefox)
- connect to notebook server: https://wscarapils.intec.ugent.be
- notebook login / password

Launch a notebook

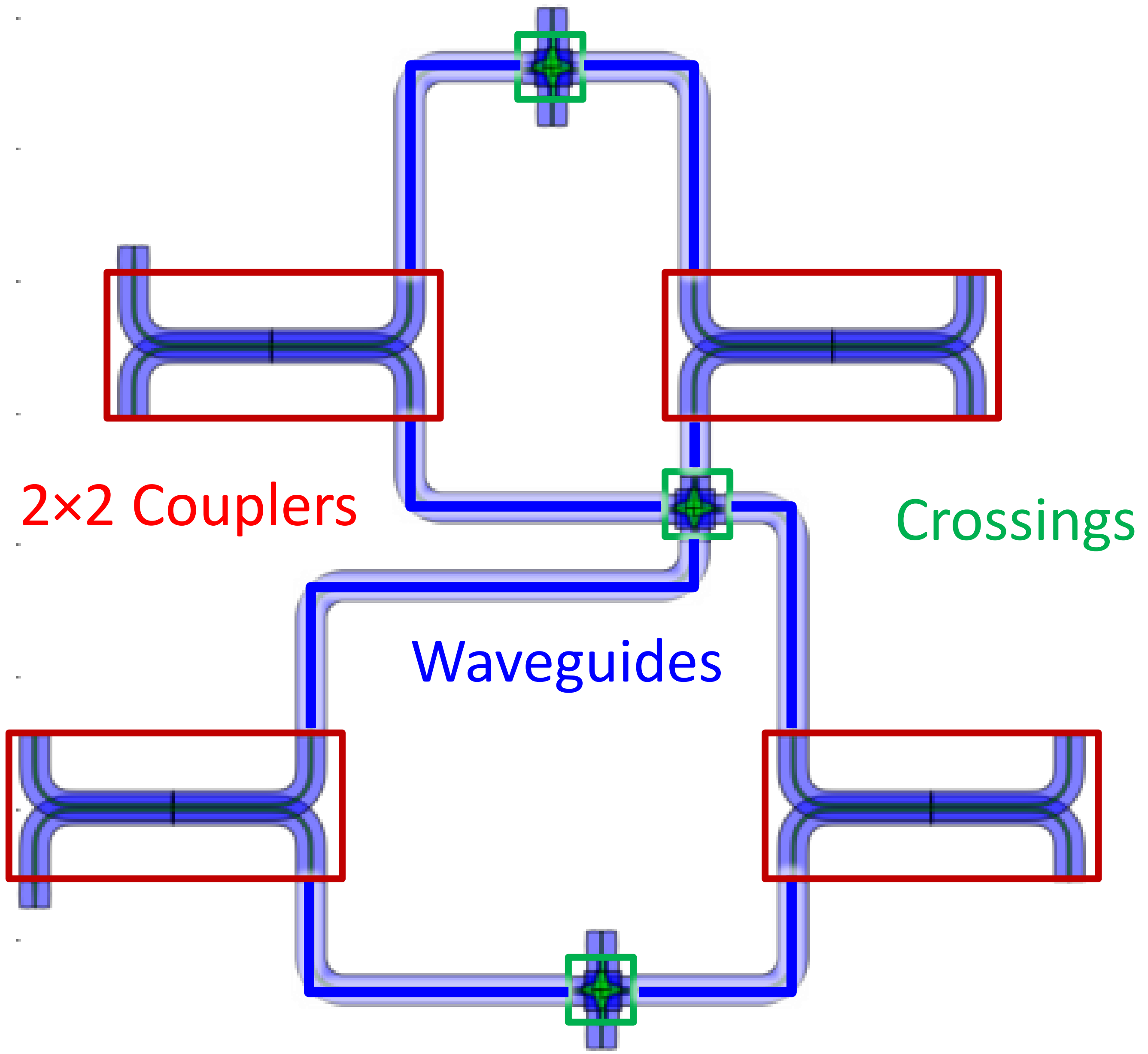
Step 1:
Copy the notebook

GHENT UNIVERSITY imec

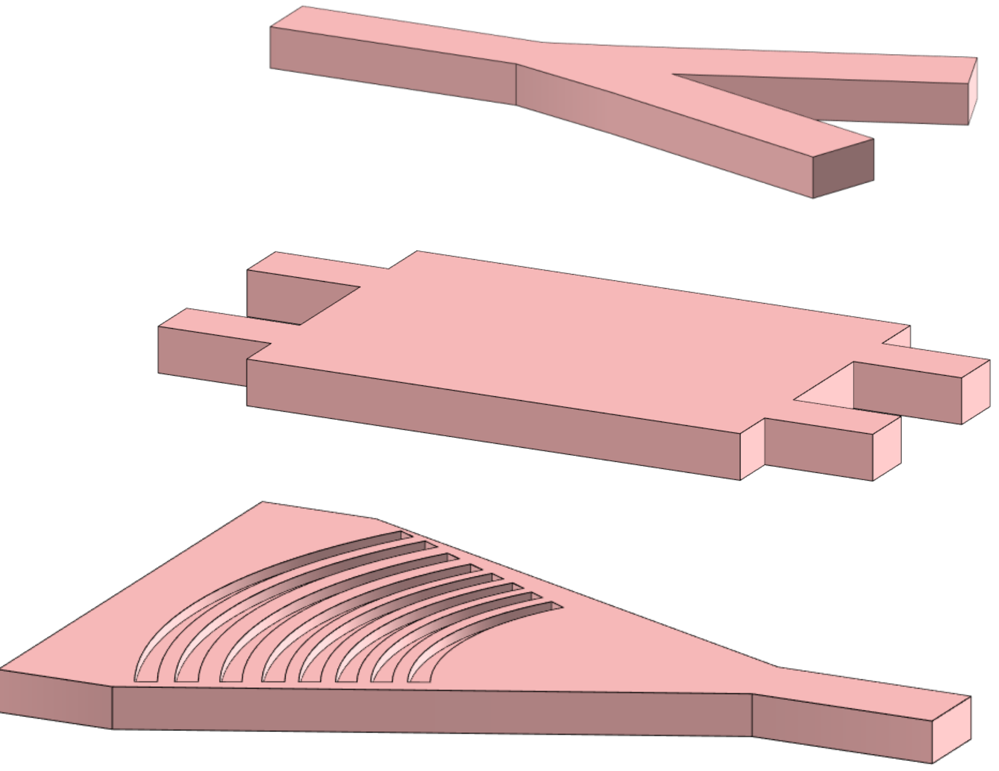
BUILDING YOUR FIRST PHOTONIC CIRCUITS

A SIMPLE PASSIVE CIRCUIT

- Four 2×2 couplers
- 3 Crossings
- Connection waveguides



BUILDING BLOCKS



Passive: waveguides, splitters, couplers, crossings

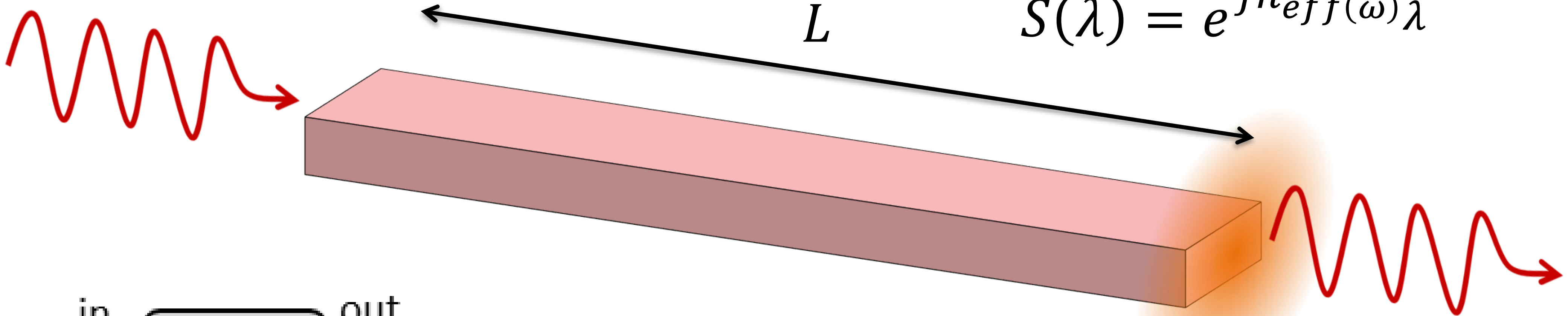
Active: modulators, detectors, tuners

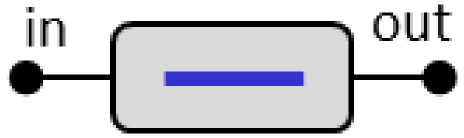
Where do they come from?

- Make them yourself
- Use existing blocks
  - From a shared library
  - From the fab : **Process Design Kit (PDK)**
- Building blocks are **process-specific**

WAVEGUIDES

$$S(\omega) = e^{j2\pi c n_{eff}(\omega) L/\omega}$$
$$S(\lambda) = e^{j n_{eff}(\omega)\frac{L}{\lambda}}$$





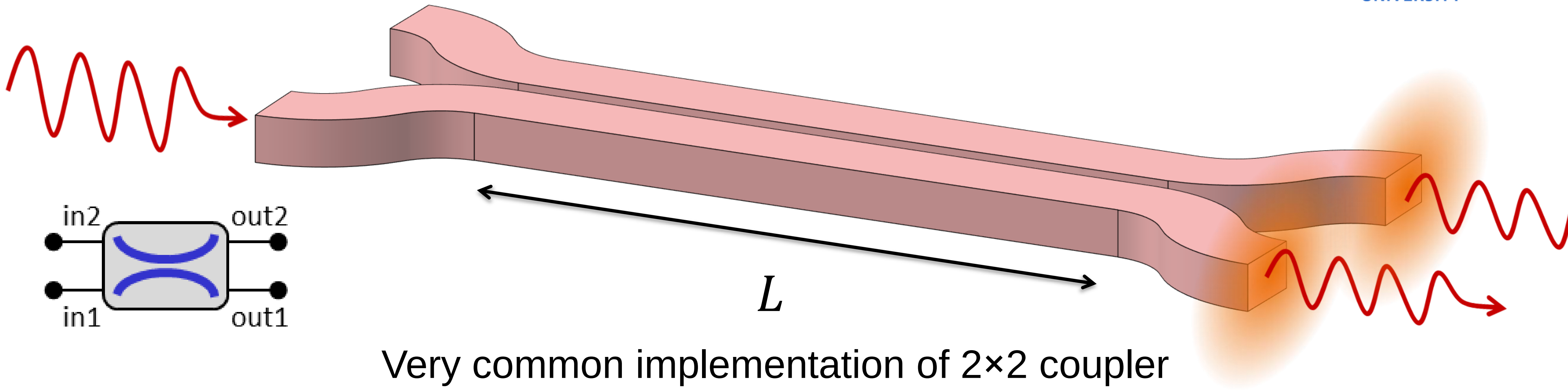
guided mode: $n_{eff}(\lambda)$

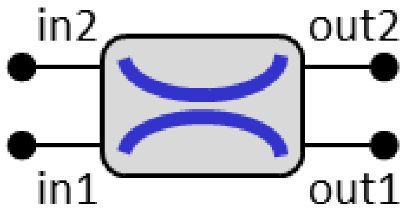
Propagate light from the input to the output

- wavefronts propagate with velocity $v_{ph}(\lambda) = \frac{c}{n_{eff}(\lambda)}$

  ($n_{eff}(\lambda)$= effective refractive index)
- Dispersion: $n_{eff}(\lambda)$ is wavelength dependent
- Group velocity: time delay of a wave packet: $v_g(\lambda) = \frac{c}{n_g(\lambda)}$

DIRECTIONAL COUPLER





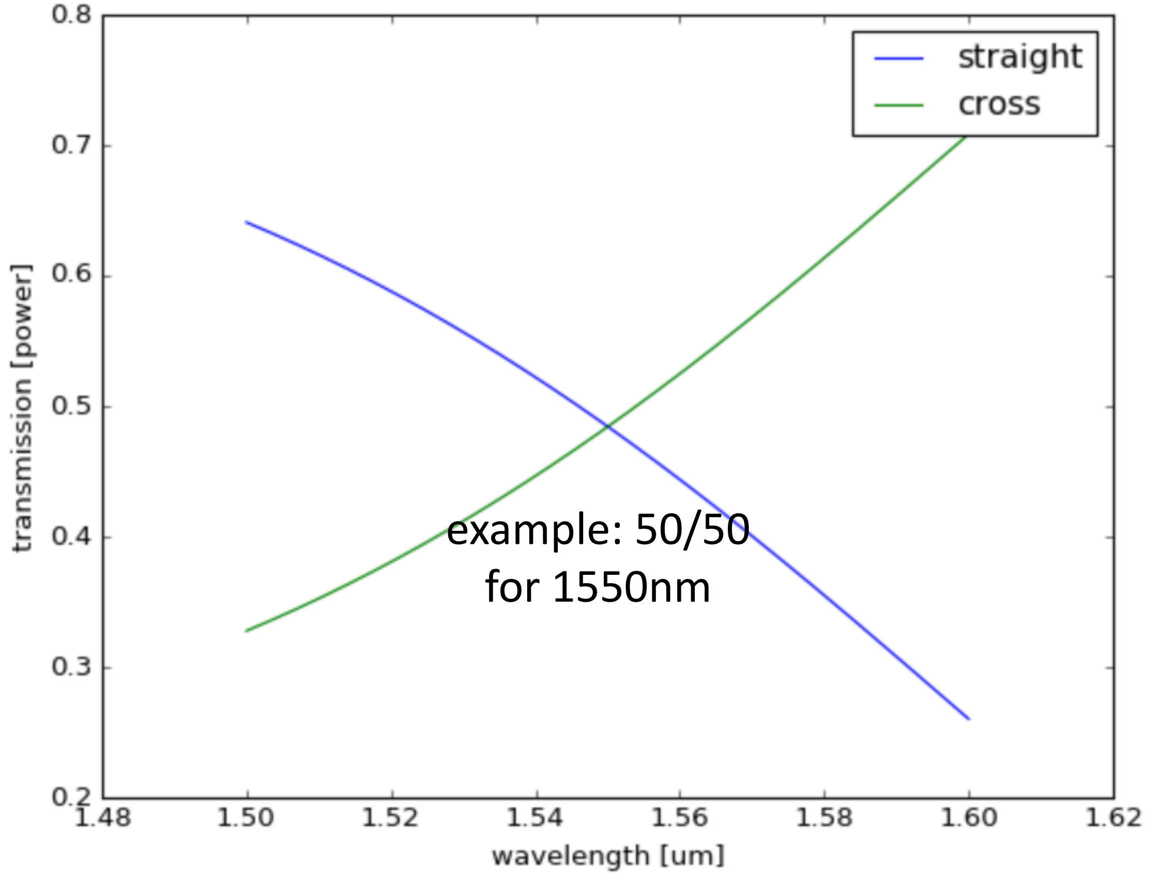
Very common implementation of 2×2 coupler

- based on interference of even and odd mode in waveguide pair
- Power coupling: $K = \sin^2(\kappa_0 + L.\kappa')$

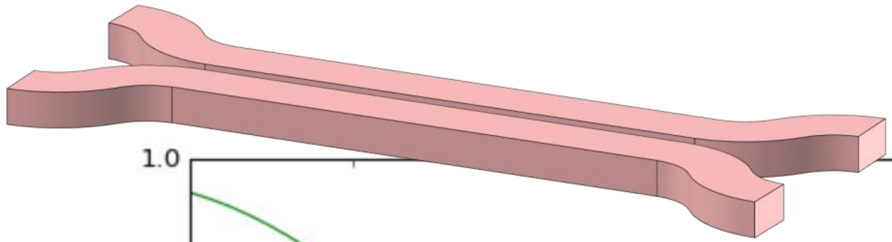
coupling in the bends

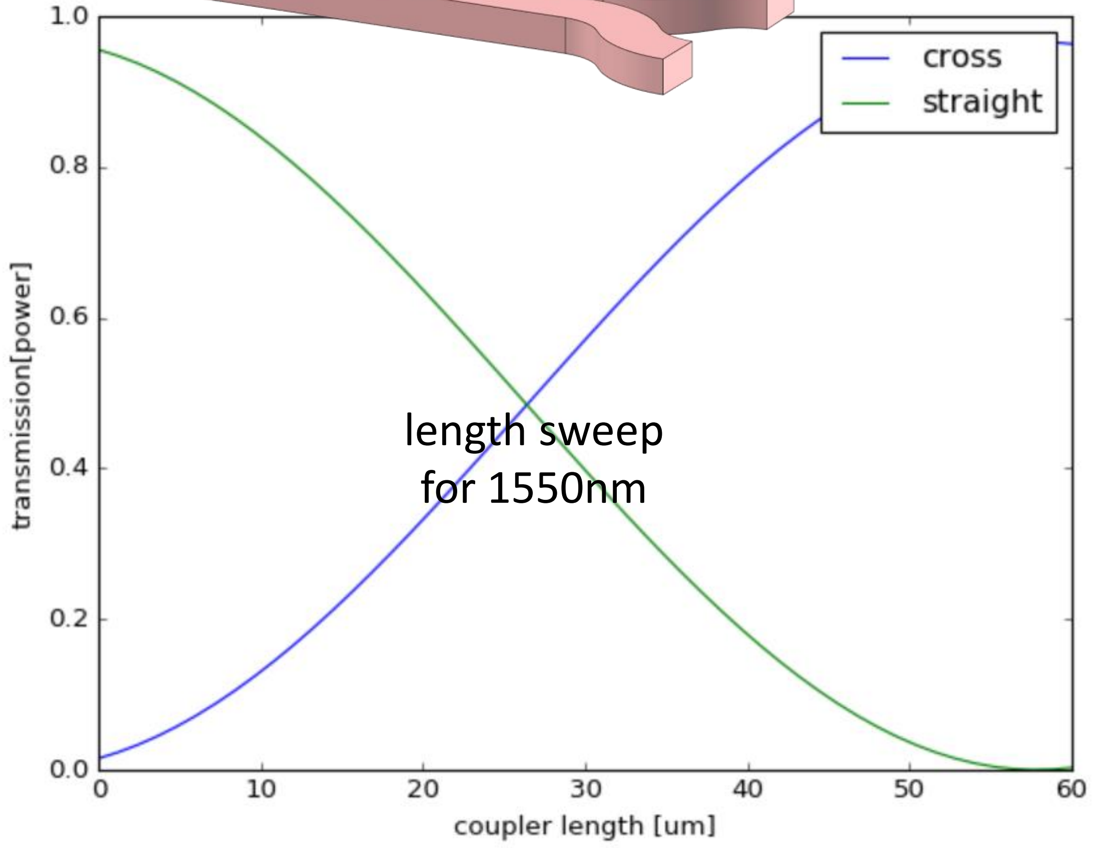
coupling in straight section

DIRECTIONAL COUPLER



example: 50/50 for 1550nm





length sweep for 1550nm

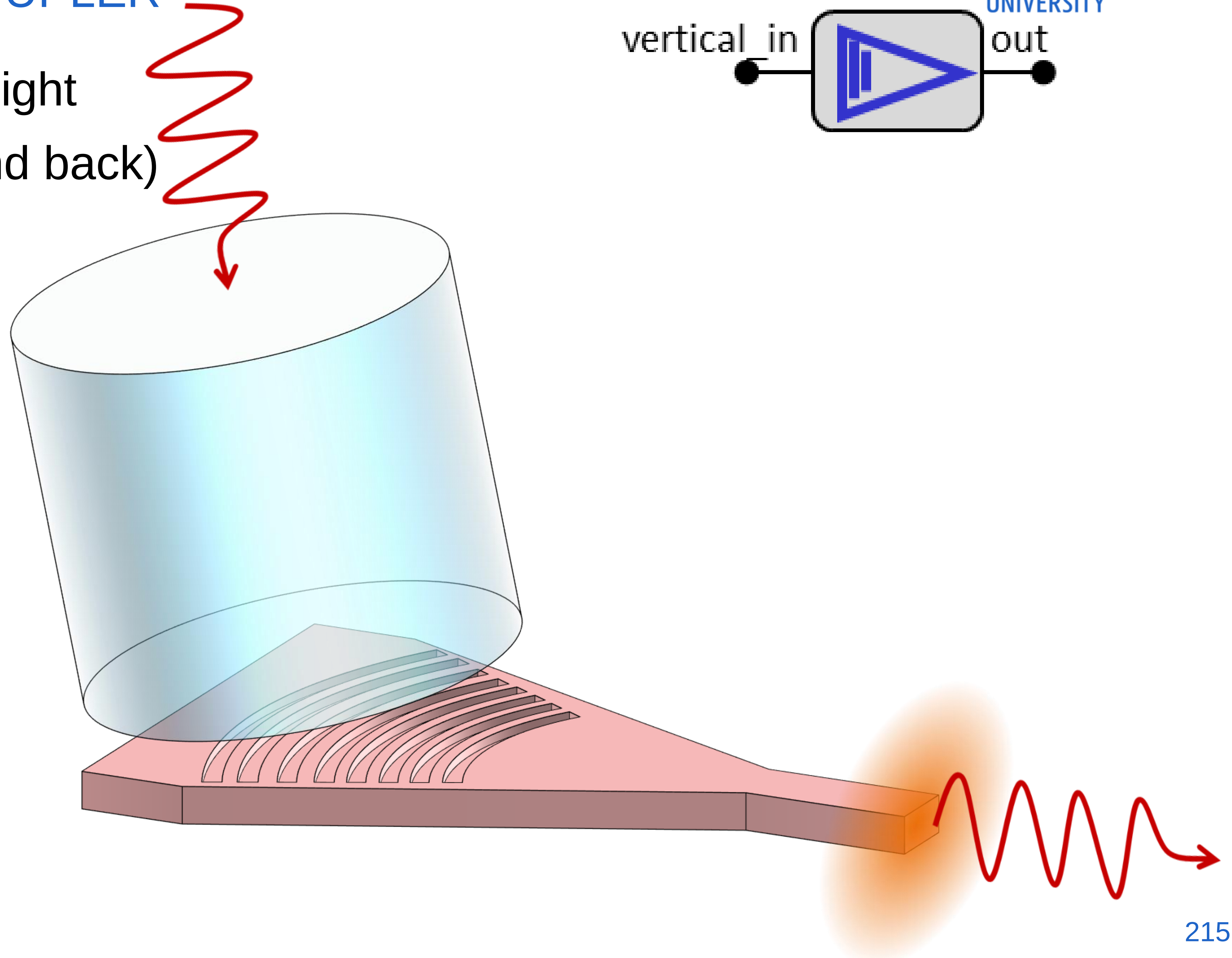
$$K = \sin^2(\kappa_0 + L.\kappa')$$

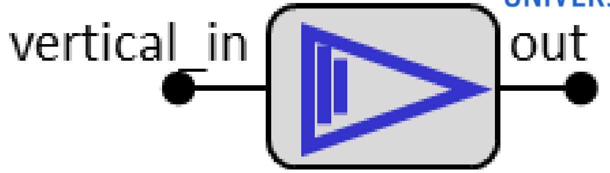
κ_0 and κ' are wavelength dependent

EXAMPLE: GRATING COUPLER

Diffraction grating couples light from fiber to waveguide (and back)

- wavelength dependent

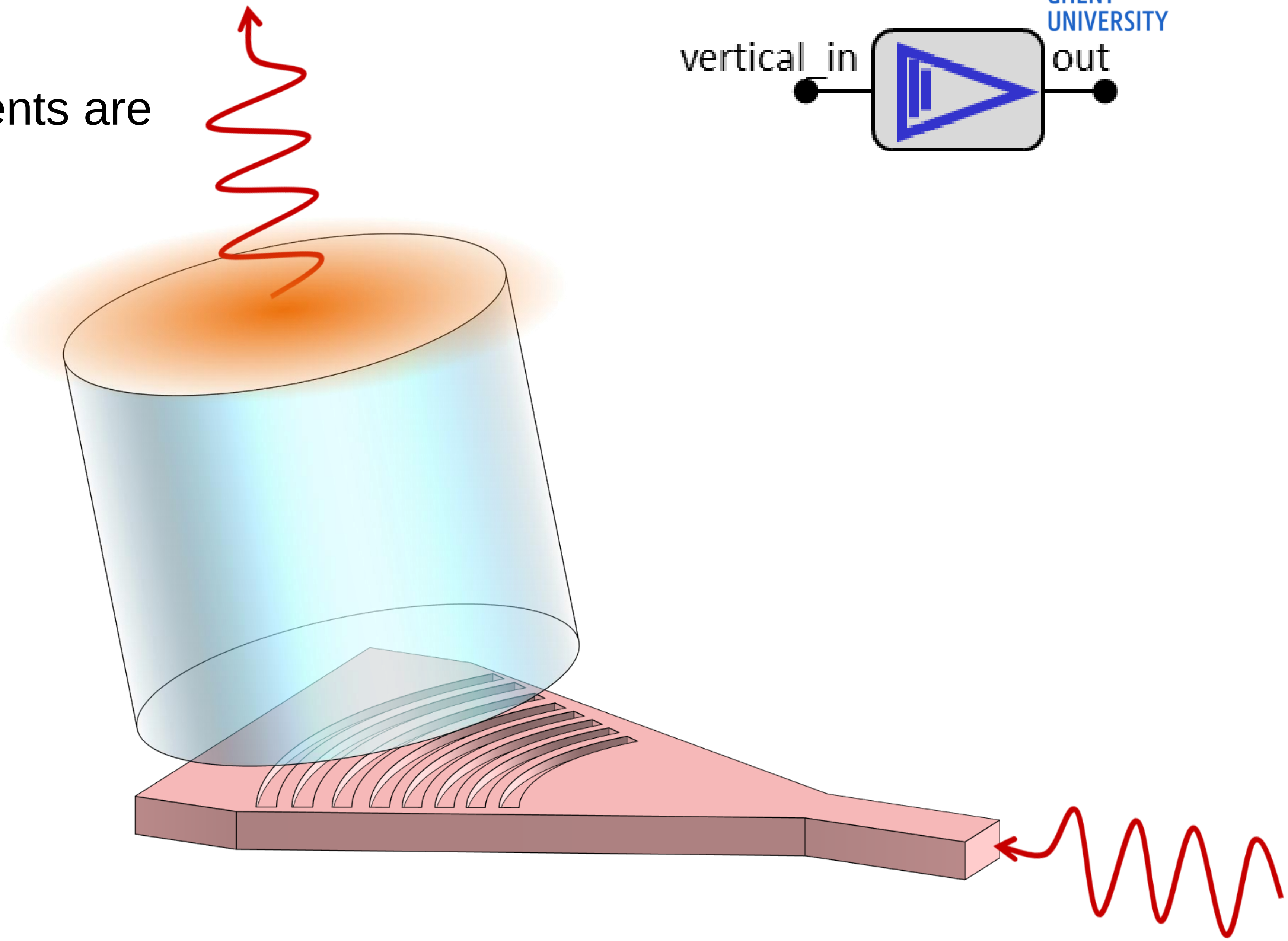


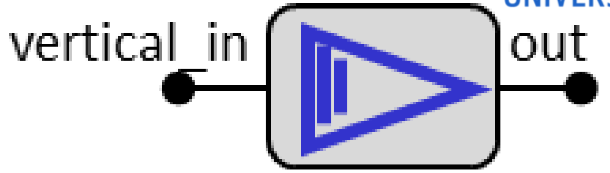


Reciprocity

Linear, passive components are reciprocal

$$S_{21}(\omega) = S_{12}(\omega)$$



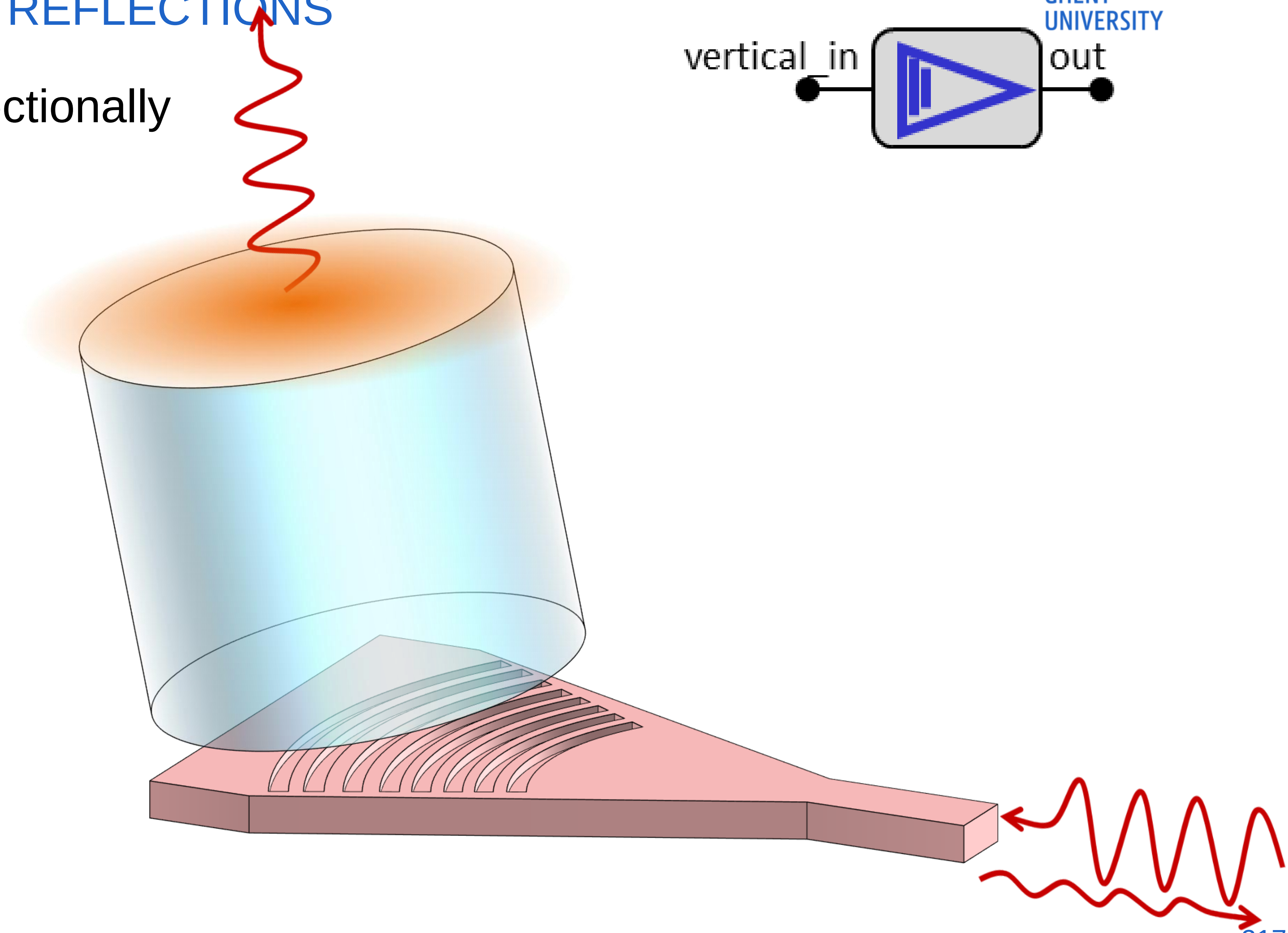


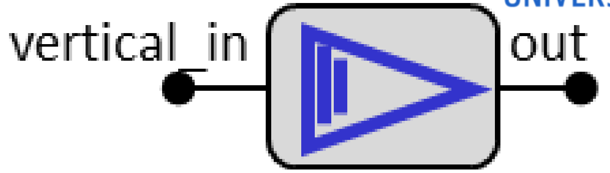
S-matrix includes reflections

circuits propagate bidirectionally

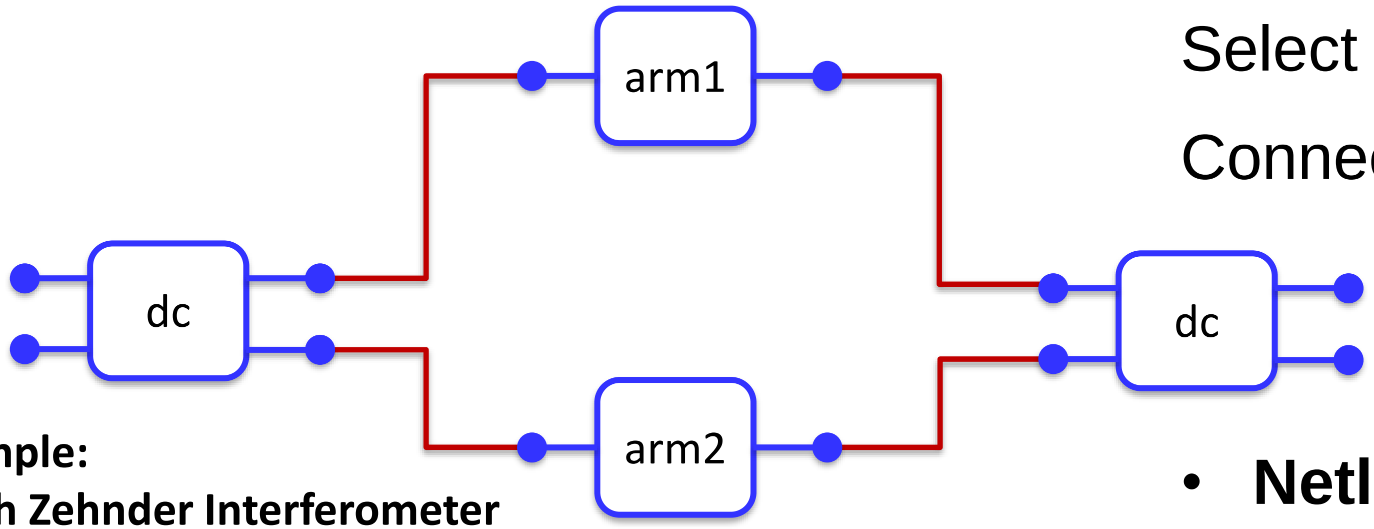
e.g. Grating coupler

has reflections





Connecting Components into Circuits



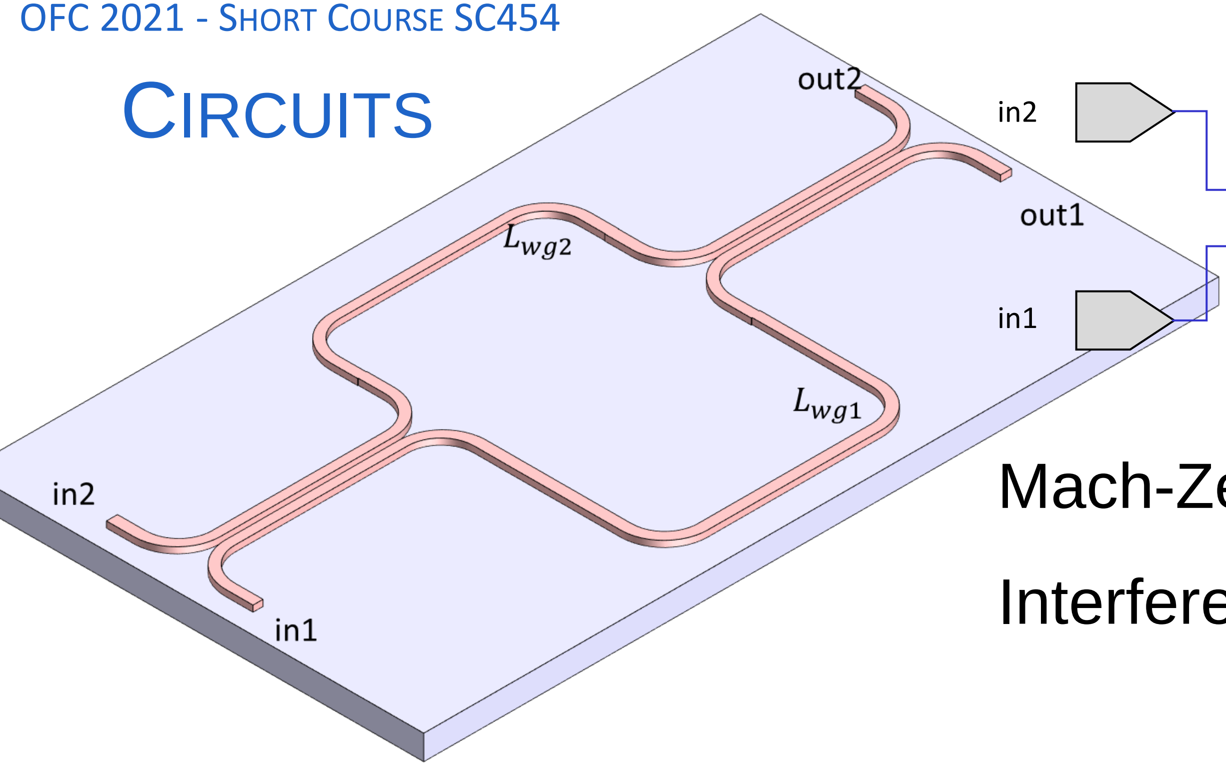
Example:
Mach Zehnder Interferometer

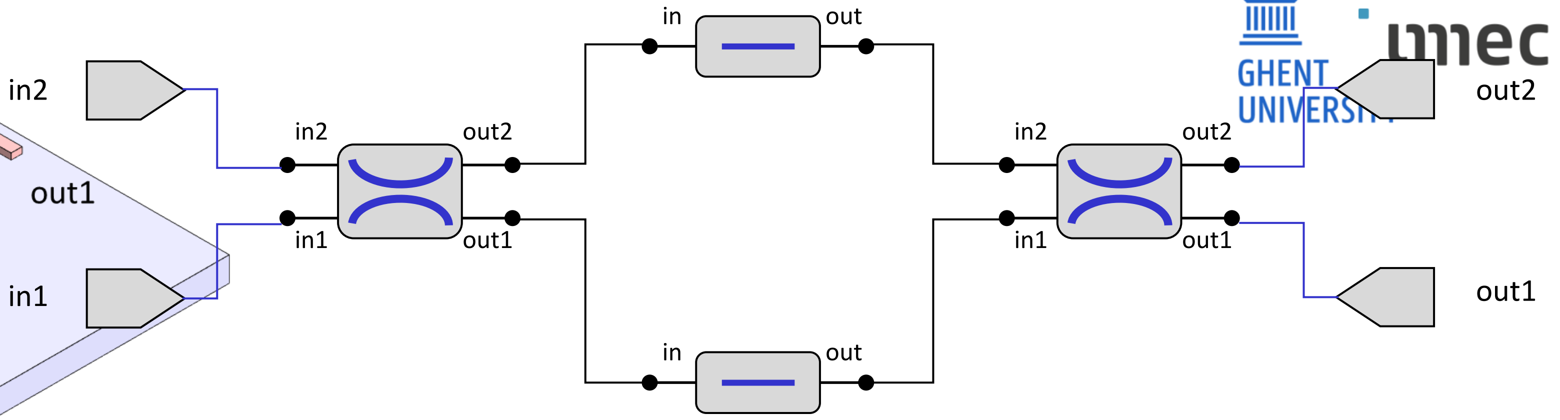
Select functional blocks

Connect them together

- **Netlist**:
  list of connections (“Nets”) and which components the nets are attached to.
- **Schematic**:
  graphical representation of a netlist, with placements

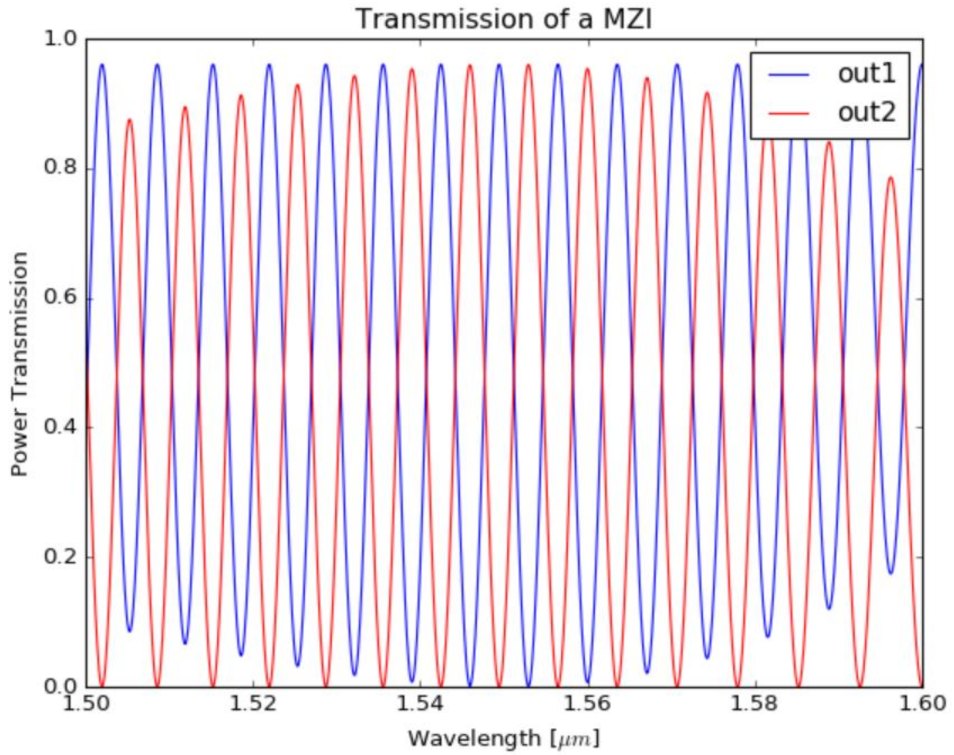
CIRCUITS



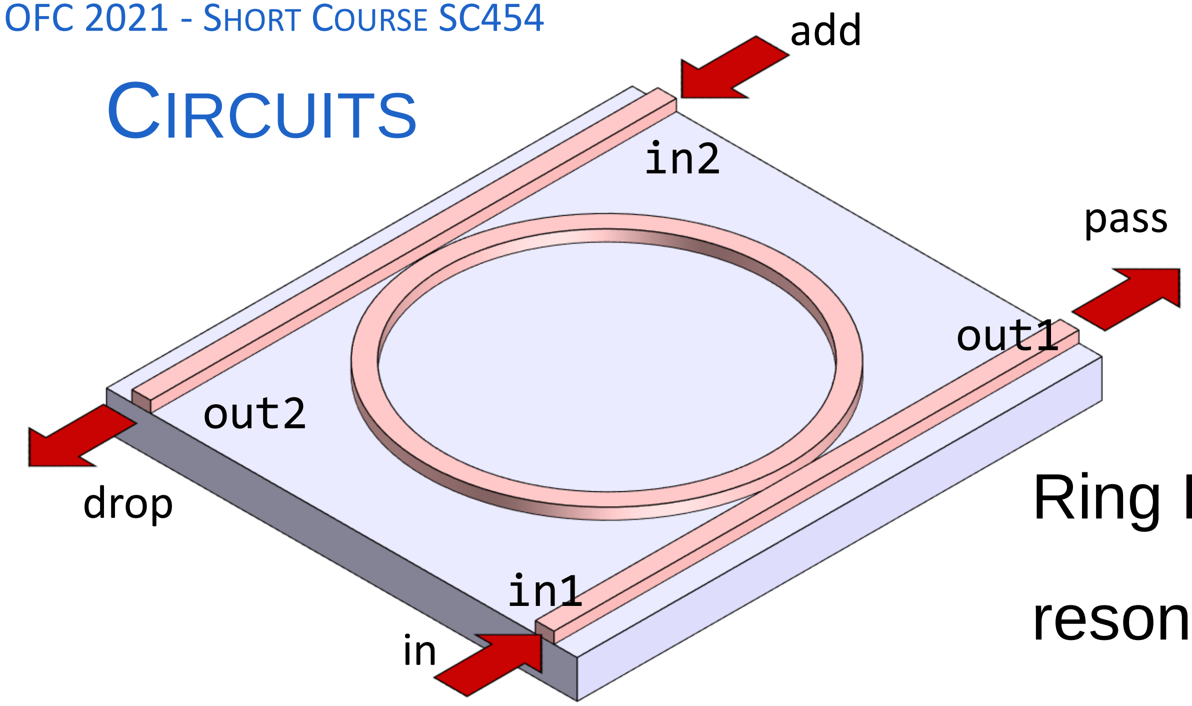


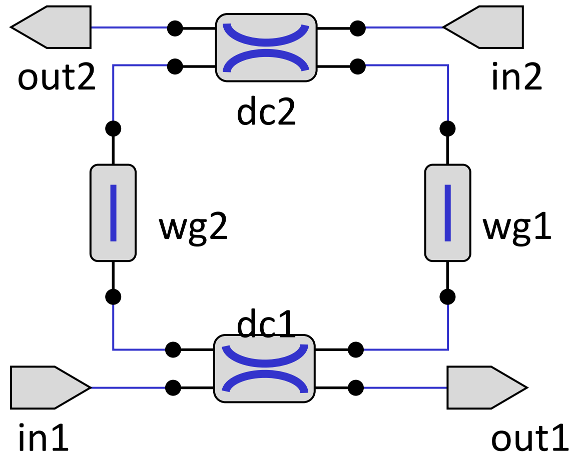
Mach-Zehnder interferometer

Interference with a delay: Periodic response



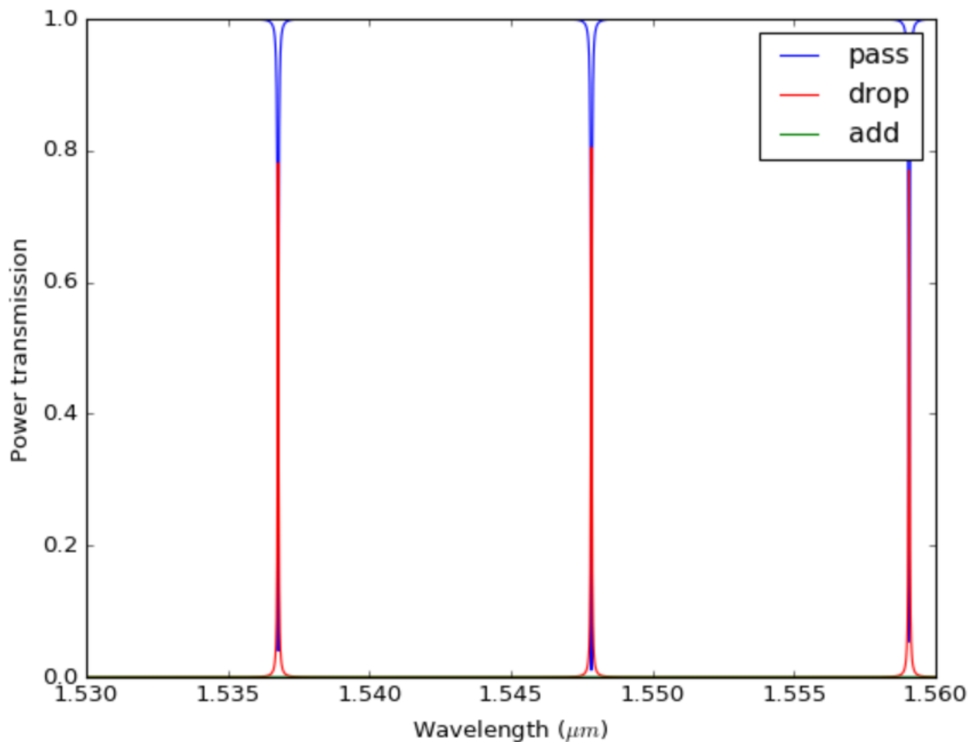
Circuits





Ring Resonator: light circulates in the ring

resonance when $L.n_{eff}(\lambda) = m.\lambda$

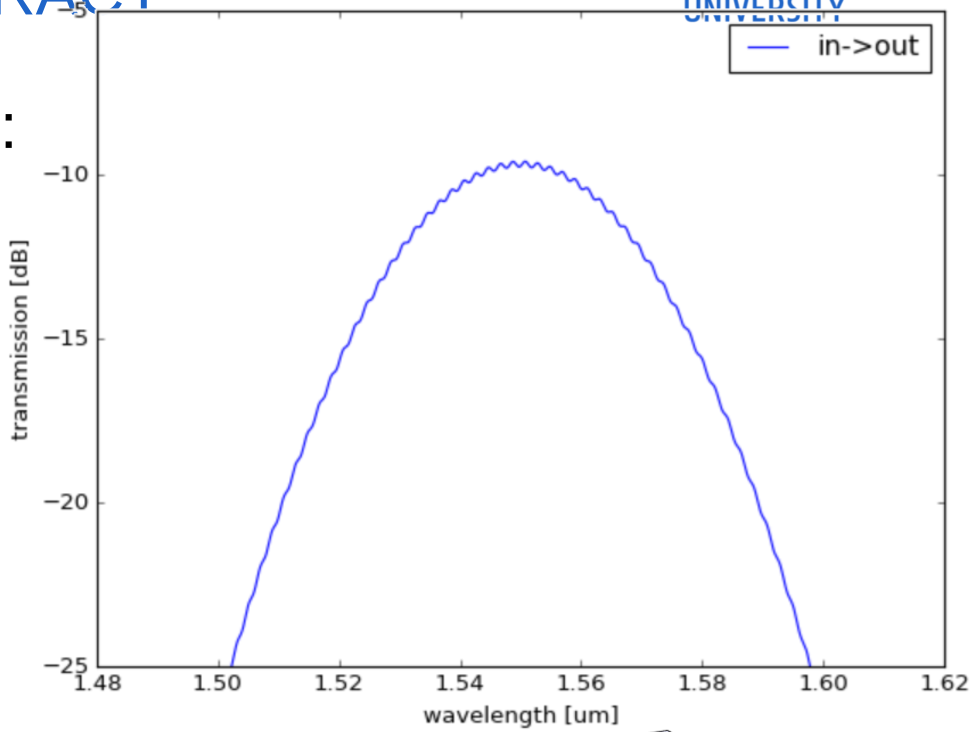


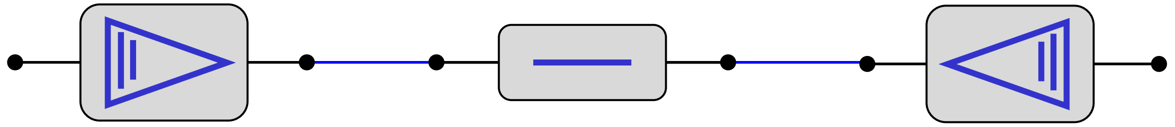
CIRCUIT EFFECTS: COMPONENTS CAN INTERACT

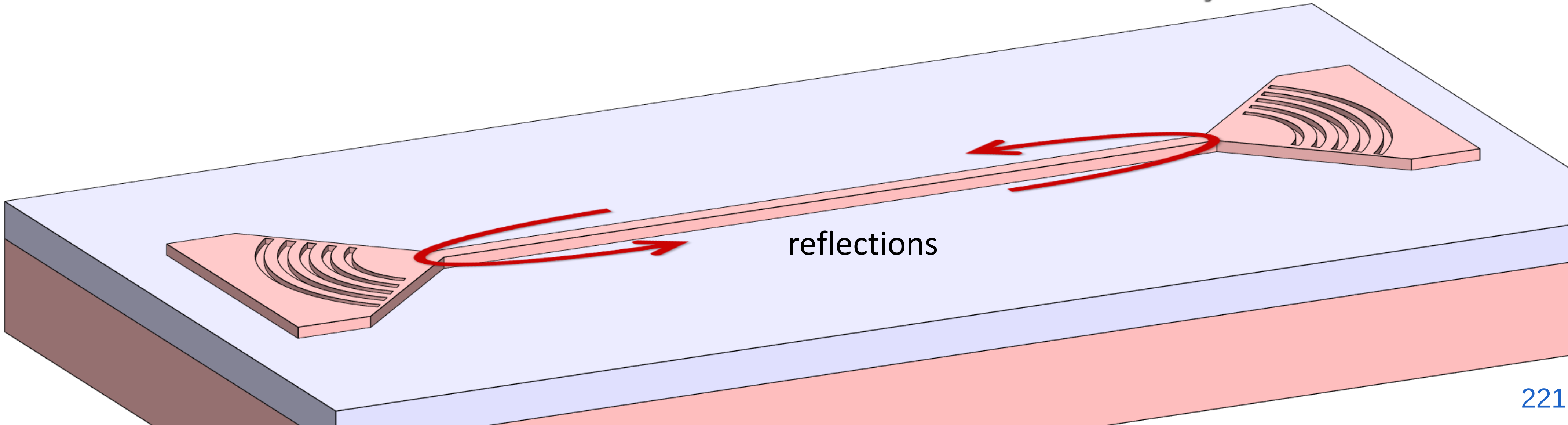
Example: weak reflections on two grating couplers:

A Fabry-Perot cavity is formed

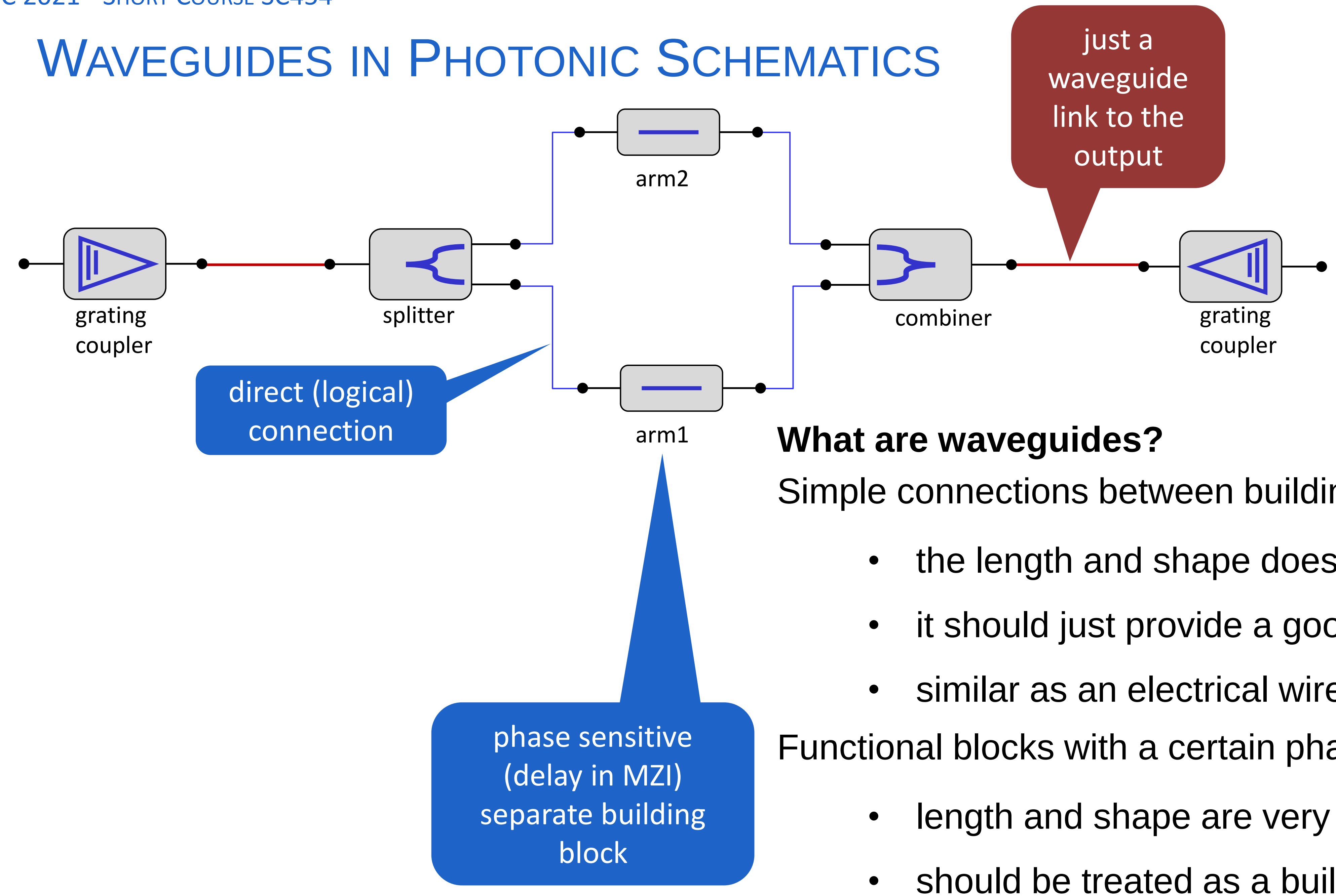
Interference causes ripple on the transmission







WAVEGUIDES IN PHOTONIC SCHEMATICS


just a waveguide link to the output
arm2
grating coupler
splitter
combiner
grating coupler
direct (logical) connection
arm1
phase sensitive (delay in MZI) separate building block

What are waveguides?

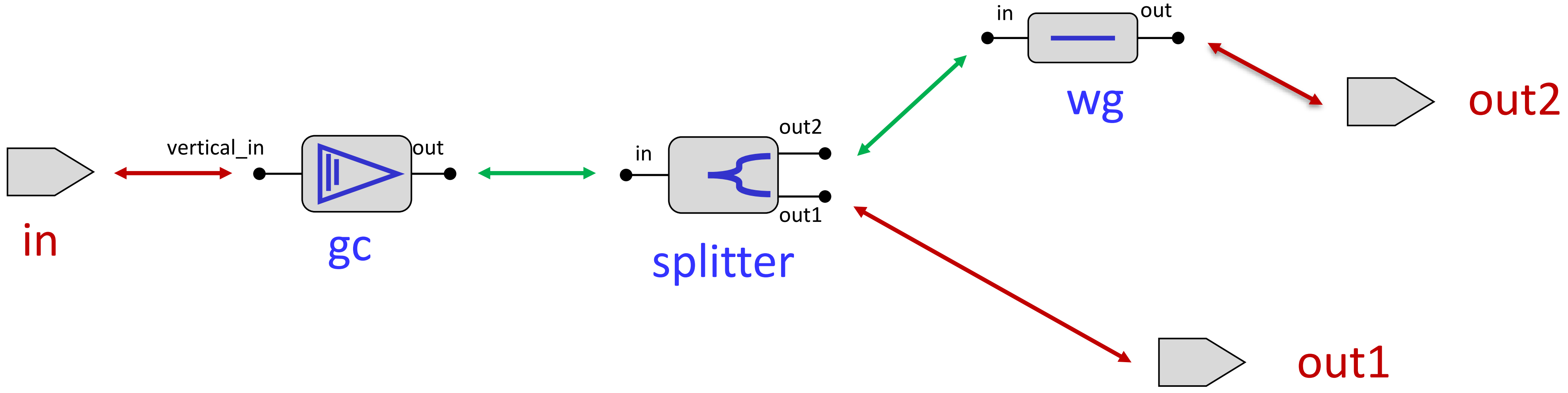
Simple connections between building blocks

- the length and shape does not really matter
- it should just provide a good connection
- similar as an electrical wire

Functional blocks with a certain phase/time delay

- length and shape are very important
- should be treated as a building block

Building Circuits in a Jupyter Notebook



Define schematics in python code

- List building blocks (or subcircuits)
    - gc, splitter, wg
- List internal connections
    - gc:out↔splitter:in, splitter:out2↔wg:in
- List external ports
    - in ↔gc:vertical_in, out1 ↔ splitter:out1, out2 ↔wg:out

BUILDING CIRCUITS: AUTOPLACEANDCONNECT

Circuits with direct connections: no waveguide generation

```
from addon_luceda.auto_place_and_connect import AutoPlaceAndConnect

child_cells = {"dc1": my_dircoup,
               "dc2": my_dircoup,
               "wg1": my_wg,
               "wg2": my_wg}

links = [("dc1:out2", "wg1:in"),
         ("wg1:out" , "dc2:in2"),
         ("dc2:out2", "wg2:in"),
         ("wg2:out", "dc1:in2")]

external_port_names = {"dc1:in1"  : "in1",
                       "dc1:out1" : "out1",
                       "dc2:in1"  : "in2",
                       "dc2:out1" : "out2"}

my_ring = AutoPlaceAndConnect(child_cells=child_cells,
                              links=links,
                              external_port_names=external_port_names)
my_ring_lo = my_ring.Layout()
my_ring_lo.visualize(annotate=True)
```

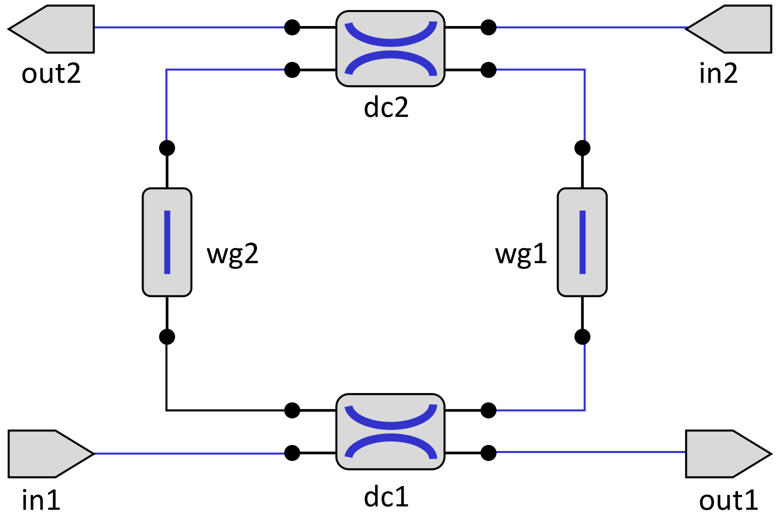
4 components

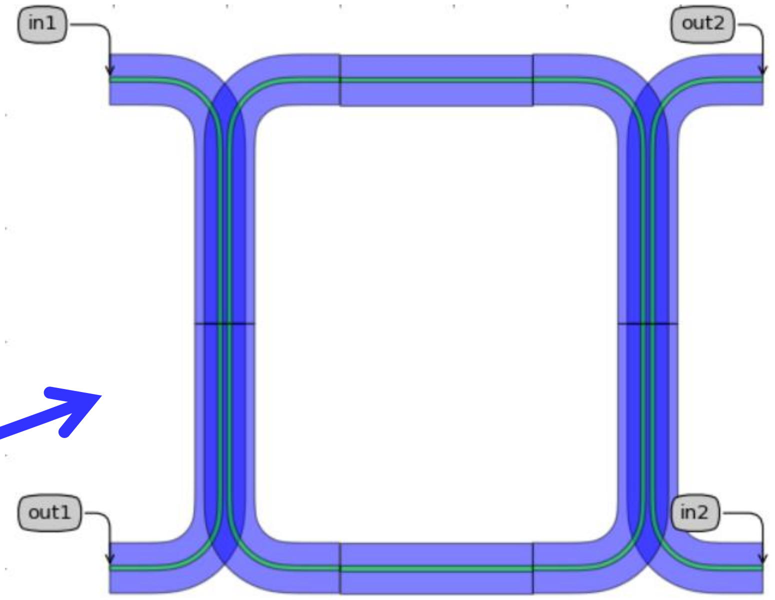
4 internal connections

4 input/output ports

automatic placement

auto-generate layout





BUILDING CIRCUITS: PLACEANDAUTOROUTE

Generate waveguides for connections

```
from picazzo3.routing.place_route import PlaceAndAutoRoute

dc_circuit = PlaceAndAutoRoute(name="dc_with_gc",
                               child_cells={"dc": my_dc,
                                            "gc_in" : fc,
                                            "gc_out_bar" : fc,
                                            "gc_out_cross" : fc,
                                            "gc_reflection" : fc
                                           },
                               links=[("gc_in:out", "dc:in1"),
                                      ("gc_reflection:out", "dc:in2"),
                                      ("dc:out1", "gc_out_bar:out"),
                                      ("dc:out2", "gc_out_cross:out"),
                                      ],
                               external_port_names={"gc_in:vertical_in": "in",
                                                    "gc_out_bar:vertical_in": "out_bar",
                                                    "gc_out_cross:vertical_in": "out_cross",
                                                    "gc_reflection:vertical_in": "reflection"}
                               )

transformations = {"gc_in": i3.Translation((-100,-20)),
                   "gc_out_cross": i3.Rotation(rotation=180) + i3.Translation((100, +20)),
                   "gc_out_bar": i3.Rotation(rotation=180) + i3.Translation((100, -20)),
                   "gc_reflection": i3.Rotation(rotation=180) + i3.Translation((100, +60)),
                  }

dc_circuit_layout = dc_circuit.Layout(child_transformations=transformations,
                   bend_radius=10.0)
dc_circuit_layout.visualize(annotate=True)
```

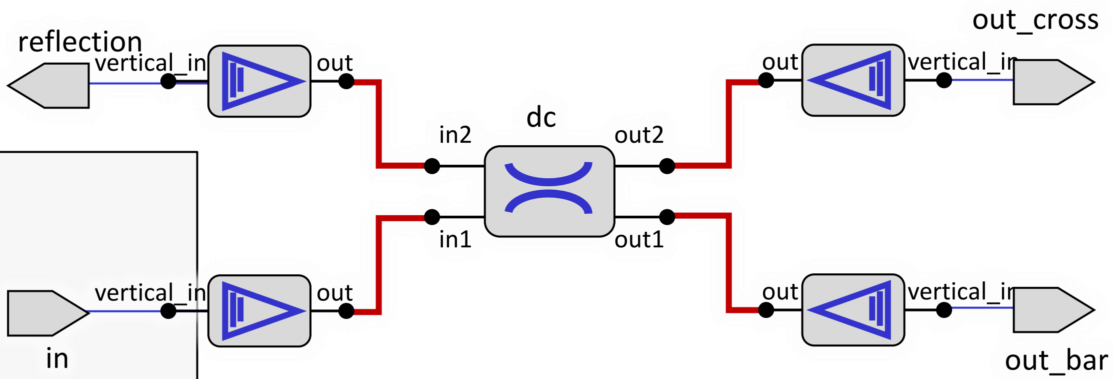
5 components

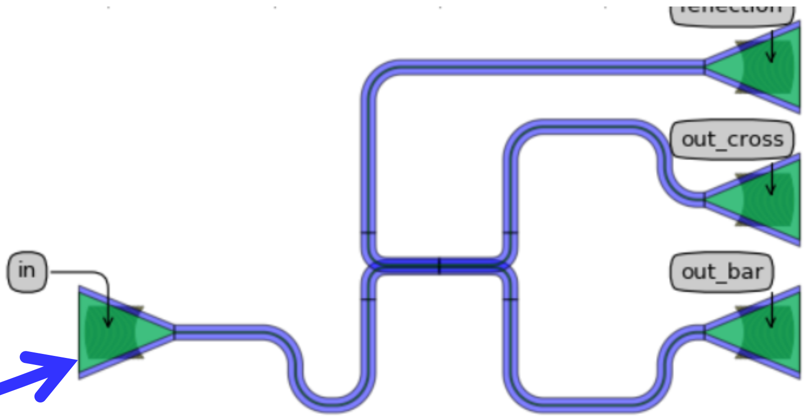
4 internal connections

4 input/output ports

manual placement

auto-generate layout

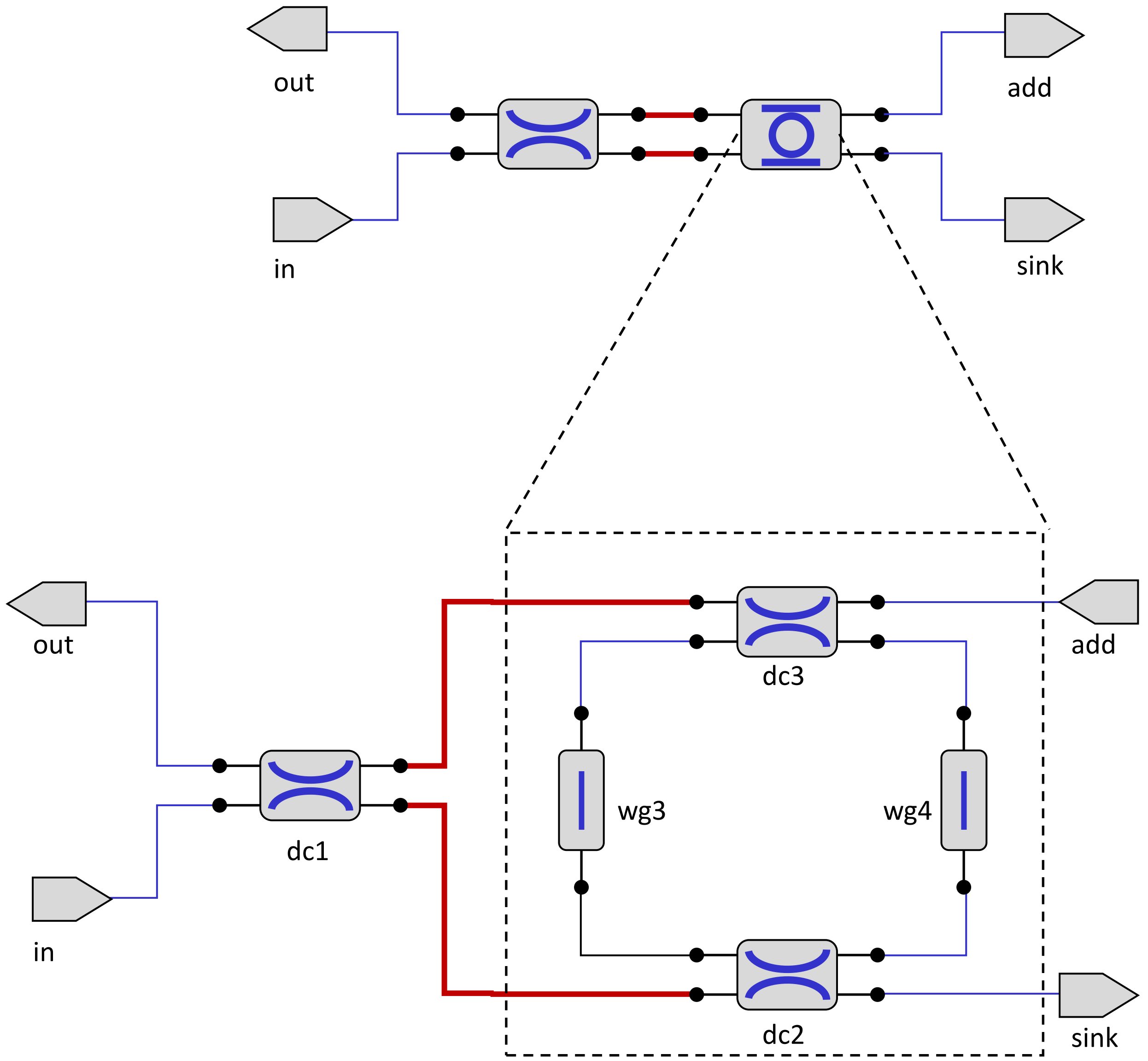




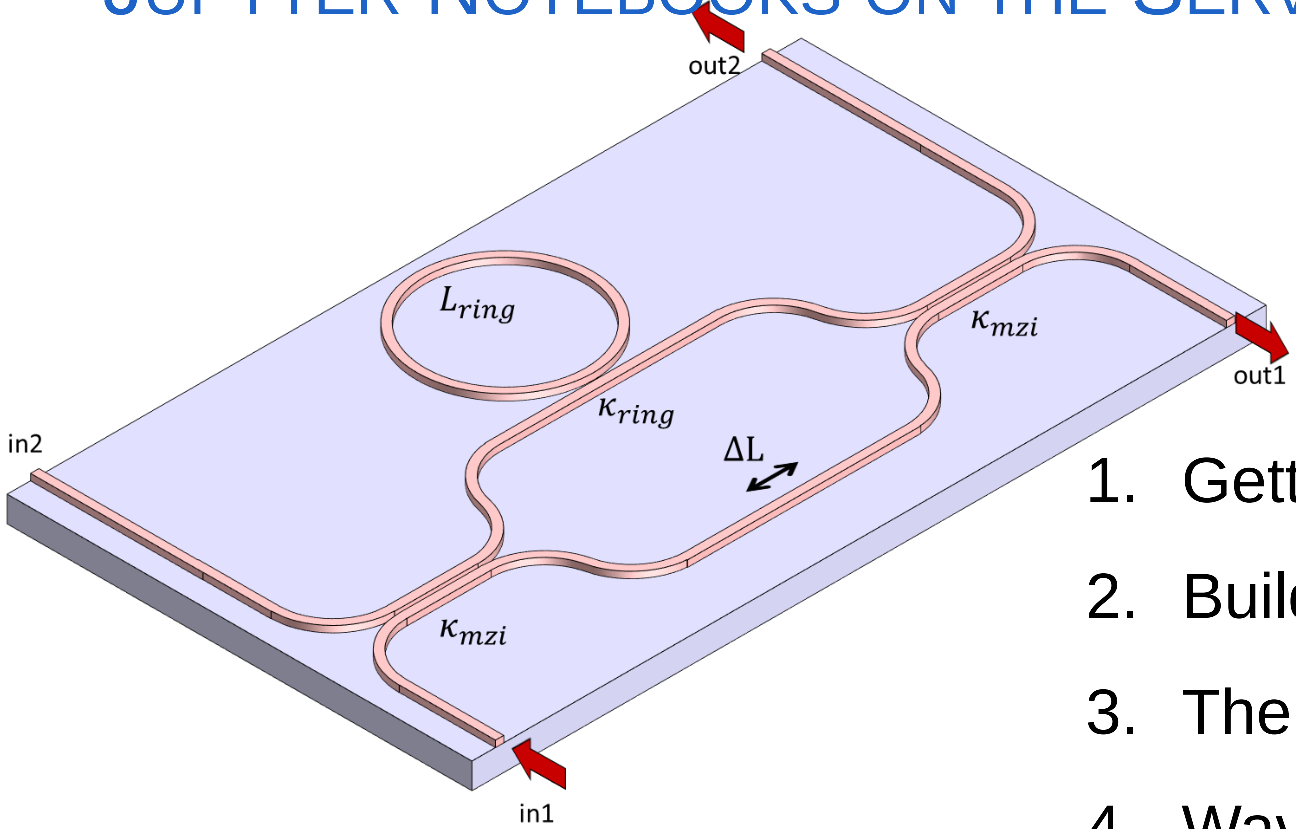
USE HIERARCHY: YOU CAN USE A CIRCUIT AS A BUILDING BLOCK

Circuits can be nested

Break up circuits into reusable parts



JUPYTER NOTEBOOKS ON THE SERVER



1. Getting started: Python, notebooks, IPKISS
2. Building a first circuit
3. The Design Kit
4. Wavelength Filters: Rings, MZIs, AWGs
5. Example designs
6. Submitting your design

The small print on Copyright

The material on the server is copyrighted

- The IPKISS toolset
- The addon libraries
- The notebooks

Please do not download the material to your own PC. It will probably not work as the server has a specific set of pre-configured utilities.

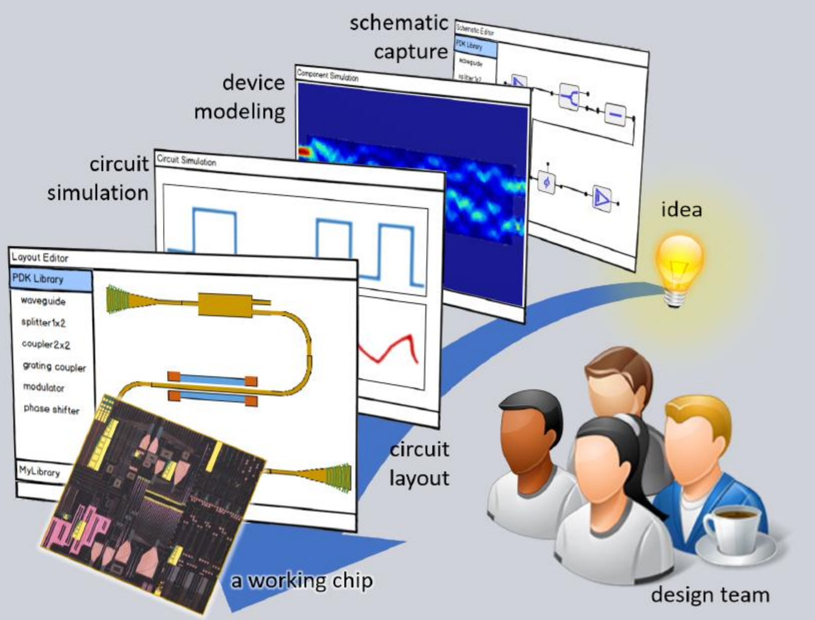
Interested in using IPKISS, contact info@lucedaphotonics.com

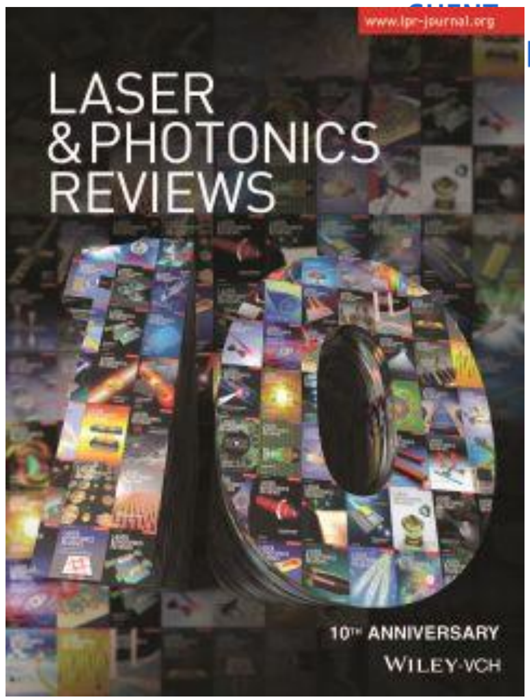
Interested in using the course material, contact wim.bogaerts@ugent.be

You can continue to use the server until 30 September 2021.

Further reading

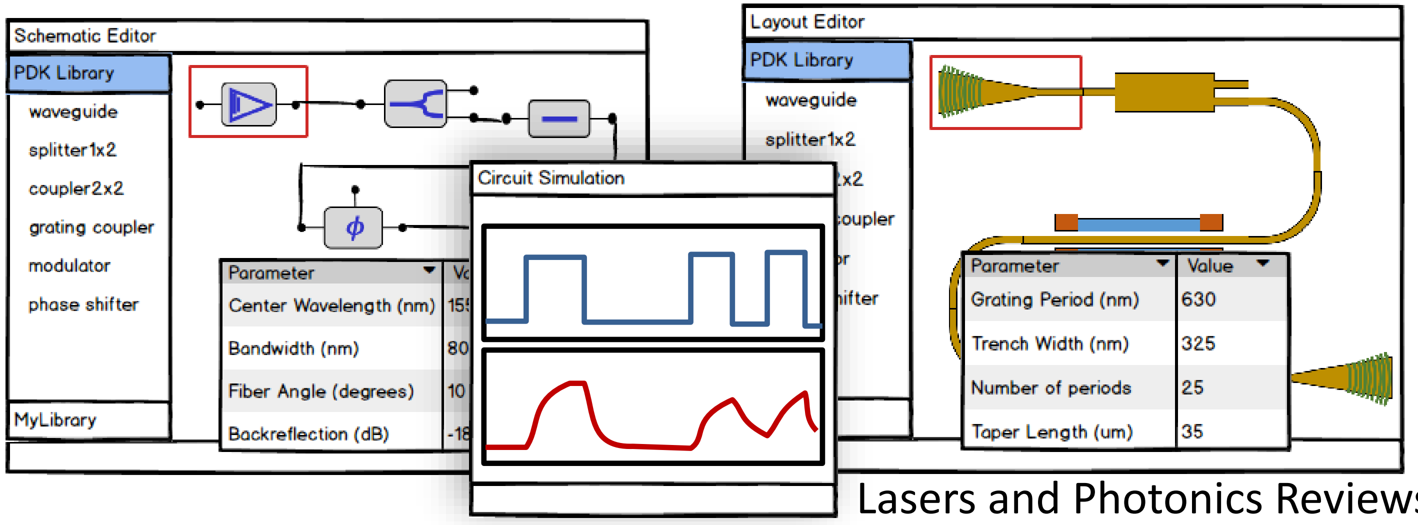
Abstract Silicon Photonics technology is rapidly maturing as a platform for larger-scale photonic circuits. As a result, the associated design methodologies are also evolving from component-oriented design to a more circuit-oriented design flow, that makes abstraction from the very detailed geometry and enables design on a larger scale. In this paper, we review the state of this emerging photonic circuit design flow and its synergies with electronic design automation (EDA). We cover the design flow from schematic capture, circuit simulation, layout and verification. We discuss the similarities and the differences between photonic and electronic design, and the challenges and opportunities that present themselves in the new photonic design landscape, such as variability analysis, photonic-electronic co-simulation and compact model definition.





Silicon Photonics Circuit Design: Methods, Tools and Challenges

Wim Bogaerts[1,2,*] and Lukas Chrostowski[3]



Parameter	Va
Center Wavelength (nm)	155
Bandwidth (nm)	80
Fiber Angle (degrees)	10
Backreflection (dB)	-18

Parameter	Value
Grating Period (nm)	630
Trench Width (nm)	325
Number of periods	25
Taper Length (um)	35

Lasers and Photonics Reviews 12(4)

PHOTONICS RESEARCH GROUP

Wim Bogaerts

Professor in Silicon Photonics

wim.bogaerts@ugent.be

+32 9 264 3324

@PhotonicsUGent

@WimBogaerts

www.photonics.intec.ugent.be