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 \( v_{ph}(\lambda) = \frac{c}{n_{eff}(\lambda)} \)
  
  \((n_{eff}(\lambda) = \text{effective refractive index})\)

- 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

\[ K = \kappa^2 \]
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 $\Delta L$
- sinusoidal spectral response

Can be cascaded for more complex filters

$$T(\lambda) \sim \frac{1}{\Delta L}$$

wavelength

splitter

$\Delta L$

combiner
RING RESONATOR

Light resonates in ring cavity:

- $L_{optical} = L_{physical} \cdot n_{eff} = m \cdot \lambda$

- Quality factor $Q \sim$ cavity losses (internal + coupling)
VERTICAL FIBER INTERFACES

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

• wavelength dependent
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)
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

Glass, polymers, III-V semiconductors, Silicon

Light transport

Wavelength filtering

Lithium Niobate
Polymers
III-V semiconductors

Signal modulation

Detection

III-V semiconductors
(GaAs, InP)

Germanium

Light source

III-V semiconductors
(GaAs, InP)
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.

**Trends & Latest Highlights**

READ THE REPORT: GLOBAL SILICON PHOTONICS MARKET 2017-2027

source: emergen research
WHAT IS SILICON PHOTONICS?

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
- **Light transport:**
- **Wavelength filtering:**
- **Signal modulation:**
- **Detection:**
- **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?

Scale

Large scale manufacturing

Submicron-scale waveguides
Scaling on-chip waveguides

Glass waveguide
index contrast ~ 0.1%

Higher index contrast
Smaller waveguides

III-V semiconductors
index contrast ~ 10%

Silicon wire:
index contrast ~ 200%
SILICON PHOTONIC WAVEGUIDES

\[ n_{\text{core}} = 3.45 \]
\[ n_{\text{cladding}} = 1.45 \]

High intensity on sidewalls
**Silica on silicon**

- Contrast ~ 0.01 – 0.1
- Mode diameter ~ 8μm
- Bend radius ~ 5mm
- Size ~ 10 cm²

**Indium Phosphide**

- Contrast ~ 0.2 – 0.5
- Mode diameter ~ 2μm
- Bend radius ~ 0.5mm
- Size ~ 10mm²

**Silicon on insulator**

- Contrast ~ 1.0 – 2.5
- Mode diameter ~ 0.4μm
- Bend radius ~ 5μm
- Size ~ 0.1mm²
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

Silicon (220nm)

Oxide (2µm)

Silicon substrate
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
Waveguide losses dominated by scattering.

Use better litho + etch
DIMENSIONAL DEPENDENCE OF A WAVEGUIDE

\[ n_{\text{eff}}(w) \]

\[ \frac{\partial n_{\text{eff}}}{\partial w} \]

\[ n_{\text{eff}}(h) \]

\[ \frac{\partial n_{\text{eff}}}{\partial h} \]
Sensitivity of Silicon Photonics Wavelength Filters

Especially wavelength filters are sensitive:

• geometry
• stress
• temperature

\[
\frac{\partial \lambda}{\partial w} \approx 1 \text{nm/nm}
\]

\[
\frac{\partial \lambda}{\partial h} \approx 2 \text{nm/nm}
\]

\[
\frac{\partial \lambda}{\partial T} \approx 0.08 \text{nm/K}
\]
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: \( P_\pi \approx 5 - 30 \text{mW} \)

(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
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
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

Butt Coupling

Evanescent Coupling

Two level

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

Challenging
- lattice constant mismatch
- polar vs. apolar material

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

First lasing demonstrated (optically pumped)
SMALL BUILDING BLOCKS → LARGE CIRCUITS

μm-scale building blocks

cm-scale chips

thousands – millions components

Photonic Very Large Scale Integration (VLSI)
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

- 100s optical IOs
- 1000s electronic feedback loops
- 1000s electrical IOs
- 100s optical IOs
- 10000s optical elements
- 10s RF signals
- Software configuration

1000s optical elements
THE PHOTONIC CHIP IS JUST A PART OF THE SYSTEM

- photonics
- analog electronics
- digital electronics
- software

integrated package

user
PACKAGING TECHNOLOGY

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

“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

Front-End

- idea/concept
- design function
- simulate function
- design layout
- check design rules
- verify design function
- fabricate device
- test

Back-End

design flow

time
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**
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**

- Drag and dropping components and drawing connections
- Make waveguides explicit if needed
- Component libraries
- Scriptability
- Parametrization
- Different connections (waveguides, direct optical, electrical)
- Interface to circuit simulation
- Specify I/O ports

---

**Additional Text:**

- Specify I/O ports
- Parametrization
- 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**

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

---

- grating coupler
- splitter
- combiner
- grating coupler

**phase sensitive (delay in MZI)**

**direct (logical) connection**

**just a waveguide link to the output**
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

\[ S_{in}(t) \xrightarrow{H} S_{out}(t) \]
**Black-box vs. White-box model**

**White-box:**
- knows the circuit
- captures the physics

\[ S_{in}(t) \rightarrow S_{out}(t) \]

**Black-box:**
- internals unknown
- mathematical ‘fit’

\[ S_{in}(t) \rightarrow S_{out}(t) \]
**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

\[
a_i = A_i e^{-j\omega t}
\]

\[
b_j = B_j e^{-j\omega t}
\]

Frequency (wavelength) dependent

\[S_{21}(\omega)\]
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
Orthogonal states

- Physically separated waveguides
- Each mode in the waveguide

Example: 6 “ports” $\rightarrow$ $6 \times 6$ 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

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

\[ a(t) = A(t) e^{-j\omega t} \]
\[ b(t) = B(t) e^{-j\omega t} \]
**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
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
An optical link carries an optical signal. The signal propagates in two directions: mode/polarization (M modes), wavelength (N channels), and power/phase. Not all simulators support all combinations.
Example: Single-\(\lambda\) link

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

**Optical Signals: Example**

Two directions

Complex number

Power

Phase

Wavelength: N channels

Single

WDM

Spectrum

Mode/polarization: M modes

\(TE_0\)

\(TM_0\)

\(TE_1\)
**Example: WDM bidirectional link**

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

**Wavelength:**
- N channels

**Mode/polarization:**
- TE0
- TM0
- TE1

**Spectrum:**

- Single wavelength
- WDM spectrum

**Complex number:**
- Power
- Phase

- $2 \times 2 \times 32 \times 1$
**Optical Signals: Example**

Example: DWDM multimode link

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

- complex number
- power
- phase
- wavelength: N channels
- single
- WDM
- spectrum
- mode/polarization: M modes
- TE0
- TM0
- TE1
**Optical Signals: Example**

Example: Spectrometer

- two directions (parasitic reflections)
- wavelength filtering: phase
- continuous spectrum: 100nm @ 5pm
- one mode
SIMULATING LINEAR CIRCUITS

Photonics does not fit easily in Spice

Effort-flow systems

<table>
<thead>
<tr>
<th>Effort-flow</th>
<th>Electrical</th>
<th>Fluidic</th>
<th>Thermal</th>
<th>Mechanical</th>
<th>Photonic?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td>Pressure</td>
<td>Temperature</td>
<td>Force</td>
<td>E-field</td>
<td></td>
</tr>
<tr>
<td>Current</td>
<td>Flow</td>
<td>Heat Flow*</td>
<td>Motion</td>
<td>H-field</td>
<td></td>
</tr>
</tbody>
</table>

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
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

![Diagram of cosimulation](image)

- **Optical simulator**
- **SPICE simulator**
- **Mixed-signal simulator**
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

Design flow:
- idea/concept
- design function
- simulate function
- design layout
- check design rules
- verify design function
- fabricate device
- test
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
Component libraries

Drag and dropping components

Alignment and snapping at waveguide ports

Parametrization

Scriptability

Optical and electrical pins

Routing of waveguides and electrical wires

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

Interface to verification (DRC and LVS)

Alignment and snapping at waveguide ports
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 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
  - ...
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
Photonic-specific constraints

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

Pure photonics
or based on EDA tools

• define connections
• place components
• route waveguides
**IS THE LAYOUT VALID?**

---

**Design Rule Checking**

- minimum features
- layer combinations
- overlaps
- pattern density

---

**design flow**

**time**
DESIGN RULE VIOLATIONS: EXAMPLES

Edge Violation

Spacing Violation

Width Violations

Encapsulation Violation
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?
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

- idea/concept
- design function
- simulate function
- design layout
- check design rules
- verify design function

Parasitic effects that were not in the schematic

design flow
time
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 layout. Include lengths, crossings, reflections, …
“no plan survives contact with the enemy”

H. von Moltke (misquoted)
THE ACTUAL FABRICATION PROCESS

Layer depositions
Pattern definition (lithography)
Pattern transfer (etch)
Planarization
Thermal treatment
Doping and implantation

... and each step with imperfections and variability

e.g., IMEC silicon Photonics
LITHOGRAPHY: NOT PERFECT

Spatial low-pass filter
- Minimum feature size
- Minimum pitch
- Pattern rounding

Example: Bragg grating

P ~ 290nm
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

- Idea/concept
- Design function
- Simulate function
- Design layout
- Check design rules
- Verify design function
- Fabricate device
- Test

Design flow

Time
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!

wg gap -> 160nm

wg width -> 480nm
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

Intra-die

Intra-wafer (die-to-die)

wafer-to-wafer

lot-to-lot
MAPPING GEOMETRY ON OPTICAL PROPERTIES

- width/thickness
- effective/group index

Waveguide Geometry

Y. Xing, Phot.Res. 2018
**Mapping Geometry on Optical Properties**

- width/thickness
- effective/group index

![Waveguide Geometry](image1)

![Geometry Model](image2)

Y. Xing, Phot.Res. 2018
LINEWIDTH MAP

Interpolated Width Map [nm]

Interpolated Width Map [nm]
Yield prediction scheme

- PDK (process, design, and kit)
- Building blocks + models
- Circuit netlist + layout
- Sensitivity of model parameters to fabrication parameters
  \[ \frac{\partial n_{\text{eff}}}{\partial w}, \ldots \]
- Wafer maps (or model) for fabrication parameters
- Place circuit on wafer and adjust model parameters
- Yield prediction
- Monte-Carlo on dies and wafers
- 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 ($\sigma = 1\,nm$)

Bogaerts, JSTQE 2019
BRINGING THE DEVICES CLOSER TOGETHER

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

peak separation

Transmission of a four-channel optical demultiplexer

Bogaerts, JSTQE 2019
Testing

Put the device on a measurement setup and characterize its performance.
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
• $n_{eff}(\lambda) \rightarrow$ polynomial?
• loss($\lambda$) $\rightarrow$ polynomial?
• nonlinearities?

How to measure $n_{eff}$?
Our Simple Design Flow

1. Idea/Concept
2. Design Function
3. Simulate Function
4. Design Layout
5. Check Design Rules
6. Verify Design Function
7. Fabricate Device
8. Test

Design Flow → Time
OUR SIMPLE DESIGN FLOW

design flow

Exchange of Information?

time
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

```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=i3.Layer(2),
                              begin_coord=(-r, -r-g),
                              end_coord=(r, -r-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

System design

Circuit design

Component design

Behavioral simulations

Physical simulations

design flow
time

idea/concept

design function

simulate function

check function

design layout

check rules

test

Behavioral simulations

Physical simulations
ABSTRACTIONS IN A CIRCUIT DESIGN FLOW

System design

System

Behavioral simulations

Circuit design

Circuit

Physical simulations

Component design

Component

handled by the fab

design flow

time
PDK: INTERFACE FROM FAB TO DESIGNER

PDK for photonics

component design simulation, measurement

technology and verification rules

circuit design and simulation

layout generation and verification

component libraries documentation support scripts

FAB

DESIGNER

PDK
ABSTRACTIONS IN A DESIGN FLOW

System design

Circuit design

Component design

Physical component design and electromagnetic simulations

Behavioral simulations

Physical simulations

design flow

time
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
PHOTONIC CIRCUIT DESIGN TOOLS
## Tool Capabilities

<table>
<thead>
<tr>
<th>Component sim</th>
<th>Circuit Sim</th>
<th>Component Layout</th>
<th>Circuit Layout</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Synopsys</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fullwave</td>
<td>OptSim</td>
<td>Optodesigner</td>
<td>Optodesigner</td>
<td>Optodesigner</td>
</tr>
<tr>
<td>Beamprop</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Cadence</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spectre AMS</td>
<td>Virtuoso</td>
<td>Virtuoso</td>
<td>Assura</td>
<td></td>
</tr>
<tr>
<td><strong>Siemens Mentor</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eldo</td>
<td>L-Edit</td>
<td>L-Edit LightSuite</td>
<td>Calibre</td>
<td></td>
</tr>
<tr>
<td><strong>Luceda</strong></td>
<td>Caphe</td>
<td>IPKISS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Camfr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>VP Photonics</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ModeDesigner</td>
<td>ComponentMaker</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Lumerical</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FDTD Solutions</td>
<td>Interconnect</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Ansys</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nazcadesign</td>
<td></td>
<td>Nazca</td>
<td>Nazca</td>
<td></td>
</tr>
<tr>
<td><strong>Nazca Design</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fimmwave</td>
<td>PICwave</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Photon Design</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Omnisim</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Simulia</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CST studio</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 …
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

Optical signals in and out
RF signals in
RF signals out

Photonic Processor
**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
- RF inputs: modulators
- RF outputs: balanced PDs
- Specialized high performance blocks
  - Connected by a programmable linear optical circuit
  - Optical input/output
  - RF inputs: modulators
  - RF outputs: balanced PDs
  - Specialized high performance blocks

- High-speed phase modulators
- Fiber interfaces
- Balanced photodetectors
- Waveguide mesh with tunable couplers and phase shifters
- 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

standard phase shifter

standard 2x2

standard 2x2

Customer 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)
NEW TYPES OF IP

Programming routines
Circuit synthesis
Control strategies
Pluggable design IP
• linear cores
• electronic controls

ASPIC
programmable core blocks for ASPICS
programmable PIC
programmable ASPICS
control strategy
circuit synthesis
extract functionality
programming algorithms
electronic control loops
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

COMPONENTS

- Directional Coupler
- Broad-band Directional Couplers
- Bragg grating
- Splitters
Automated die-level test for end-users:

1-2 min per device (50,000 points spectrum)

Hardware including nano-steppers, fibre array, microscope; laser & detectors

Software in Matlab & Python – Open Source

Lab Tour: http://bit.ly/SiPlab

Now commercially available from Maple Leaf Photonics
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)
**JUPYTER NOTEBOOKS**

interactive notebook

- text, figures
- formulas
- python code

simulation and design

- built-in IPKISS
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
Python script based

class RingResonator(i3.Cell):
    """A generic ring resonator class."""
    wg_template = i3.WaveguideTemplateProperty(default=TECH.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 __init__(self):
        return i3.Waveguide(name=self.name+"_ring", trace_template=self.wg_template)
    def __init__(self):
        return i3.Waveguide(name=self.name+"_bus", trace_template=self.wg_template)

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

    def __init__(self):
        ring_layout = self.cell.ring.get_default_view(13.LayoutView)
        ring_layout.set(trace_template=self.wg_template, shape=i3.ShapeCircle(center=(0, 0), radius=self.ring_radius))
        return ring_layout
    def __init__(self):
        bus_layout = self.cell.bus.get_default_view(13.LayoutView)
        bus_layout.set(trace_template=self.wg_template, shape=[(-r, -r), (r, r)])
        return bus_layout

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

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

- 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
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.
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
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,

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

In [3]:
print x+y, xy
7 Hey
```
Click here to go back to start

Folders with notebooks

Create blank notebook here
Navigating

Running Notebook

Notebook: click to start
PRESS H FOR ‘HELP’

Useful menu and toolbar

Keyboard shortcuts are extremely powerful
Interactive plots consume resources.
Close them when ready.
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
BUILDING YOUR FIRST PHOTONIC CIRCUITS
A SIMPLE PASSIVE CIRCUIT

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

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: $v_g(\lambda) = \frac{c}{n_g(\lambda)}$

$$S(\omega) = e^{j2\pi cn_{eff}(\omega) L/\omega}$$

$$S(\lambda) = e^{jn_{eff}(\omega) \lambda}$$

Guided mode: $n_{eff}(\lambda)$
Very common implementation of 2×2 coupler

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

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

\(\kappa_0\) and \(\kappa'\) are wavelength dependent

example: 50/50 for 1550nm

length sweep for 1550nm
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
Select functional blocks
Connect them together

Example: Mach Zehnder Interferometer

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

Interference with a delay: Periodic response
Ring Resonator: light circulates in the ring resonance when \( L \cdot n_{eff}(\lambda) = m \cdot \lambda \)
Circuit Effects: Components Can Interact

Example: weak reflections on two grating couplers:
A Fabry-Perot cavity is formed

Interference causes ripple on the transmission
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

Direct (logical) connection

Phase sensitive (delay in MZI) separate building block
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
Circuits with direct connections: no waveguide generation

```python
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
Generate waveguides for connections

```python
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)),
    }

manual_placement

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**

**auto-generate layout**
Use Hierarchy: You can use a circuit as a building block.

Circuits can be nested

Break up circuits into reusable parts
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 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\textsuperscript{1,2,*} and Lukas Chrostowski\textsuperscript{3}
PHOTONICS RESEARCH GROUP

Wim Bogaerts
Professor in Silicon Photonics

wim.bogaerts@ugent.be
+32 9 264 3324

@PhotonicsUGent
@WimBogaerts

www.photonics.intec.ugent.be