GHENT
UNIVERSITY

FACULTY OF ENGINEERING AND ARCHITECTURE

DEPARTMENT OF INFORMATION TECHNOLOGY
PHOTONICS RESEARCH GROUP

A SOFTWARE LANGUAGE APPROACH FOR DESCRIBING AND PROGRAMMING PHOTONICS HARDWARE

Master's thesis defence - Sébastien d'Herbais de Thun - 29th of June 2023

Promoters: Prof. dr. ir. Wim Bogaerts, Prof. dr. ir. Dirk Stroobandt

GHENT UNIVERSITY

About this presentation

- Introduction
- Elevator pitch
- Programmatic description: an overview
- Example: 16-QAM modulator
- Example: Lattice filter
- Conclusion
- Future work

To code, or not to code

- Not everybody is a programmer

To code, or not to code

- Not everybody is a programmer **and that's okay!**
  - Code sections will kept **short**
  - The language is **familiar**
  - Code will be **explained**
  - Code is shown in **boxes**

```python
print('Hello, world!')
```
Python

```php
fn main() {
    print("Hello, world!")
}
```
PHÔS

To code, or not to code

- Not everybody is a programmer **and that's okay!**
  - Code sections will kept **short**
  - The language is **familiar**
  - Code will be **explained**
  - Code is shown in **boxes**
- Code is **non-exhaustive**

```python
print('Hello, world!')
```
Python

```
fn main() {
    print("Hello, world!")
}
```
PHÔS

GHENT UNIVERSITY

To code, or not to code

- Not everybody is a programmer **and that's okay!**
  - Code sections will kept **short**
  - The language is **familiar**
  - Code will be **explained**
  - Code is shown in **boxes**
- Code is **non-exhaustive**
- Code is **not optimized**

Python

```python
print('Hello, world!')
```

PHÔS

```
fn main() {
    print("Hello, world!")
}
```

To code, or not to code

- Not everybody is a programmer **and that's okay!**
  - Code sections will kept **short**
  - The language is **familiar**
  - Code will be **explained**
  - Code is shown in **boxes**
- Code is **non-exhaustive**
- Code is **not optimized**
- Code is **illustrative**

```python
print('Hello, world!')
```
Python

```
fn main() {
    print("Hello, world!")
}
```
PHÔS

GHENT UNIVERSITY

The elevator pitch

GHENT
UNIVERSITY

Why programming?

- Scaling circuits is **really** hard
- Scaling code is **really** easy

Why programming?

- Scaling circuits is **really** hard
- Circuits are **inflexible**

- Scaling code is **really** easy
- Code is **flexible**

Why programming?

- Scaling circuits is **really** hard
- Circuits are **inflexible**
- Circuits are not **reusable**

- Scaling code is **really** easy
- Code is **flexible**
- Code is easily **reusable**

Why programming?

- Scaling circuits is **really** hard
- Circuits are **inflexible**
- Circuits are not **reusable**
- Circuits are not **expressive**

- Scaling code is **really** easy
- Code is **flexible**
- Code is easily **reusable**
- Code is **expressive**

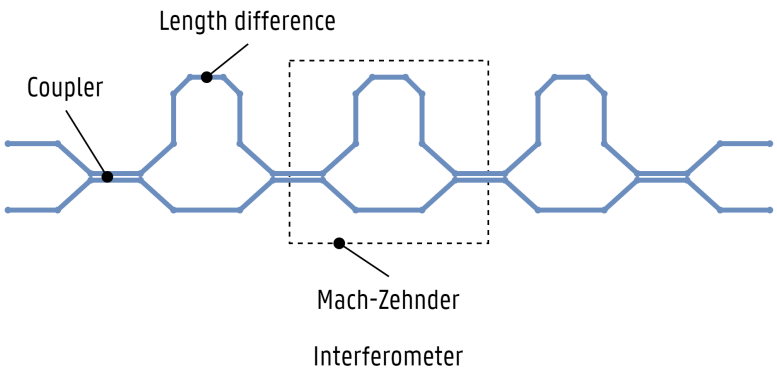

FIGURE 1 | A lattice filter circuit.

```
filter_kind_coefficients(filter_kind)
  |> fold((a, b), |acc, (coeff, phase)| {
    acc |> coupler(coeff)
        |> constrain(d_phase = phase)
  })
```

LISTING 1 | A lattice filter as code.

Levels of abstraction

- Currently low

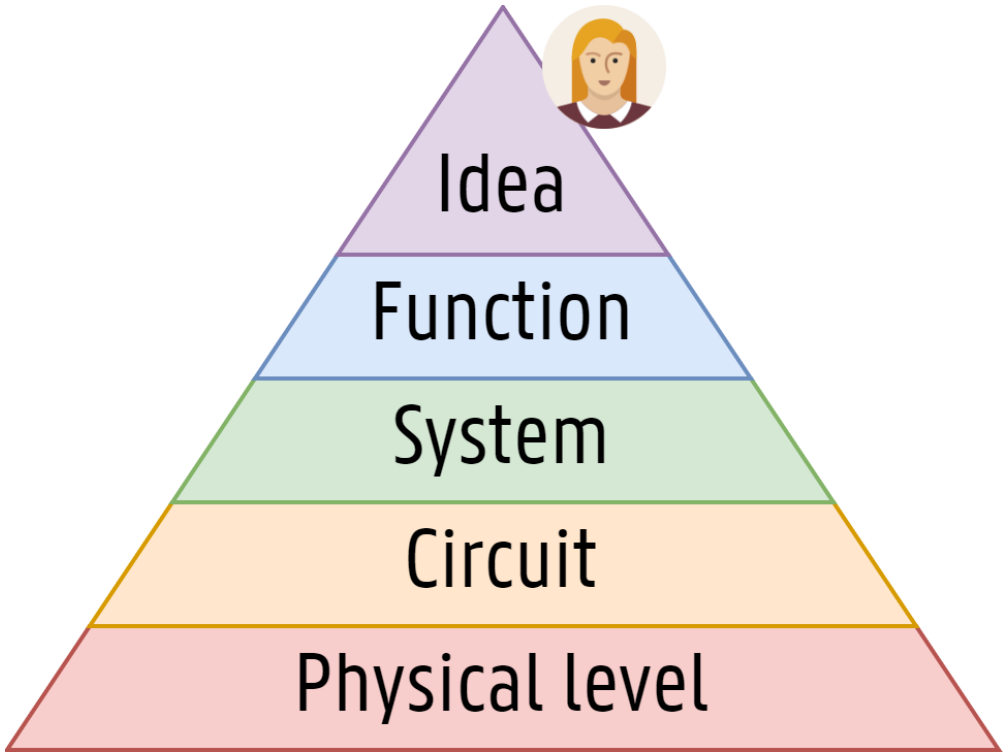

Figure 2 | Levels of abstraction in photonic circuit design.

Levels of abstraction

- Currently low
- We want to go higher

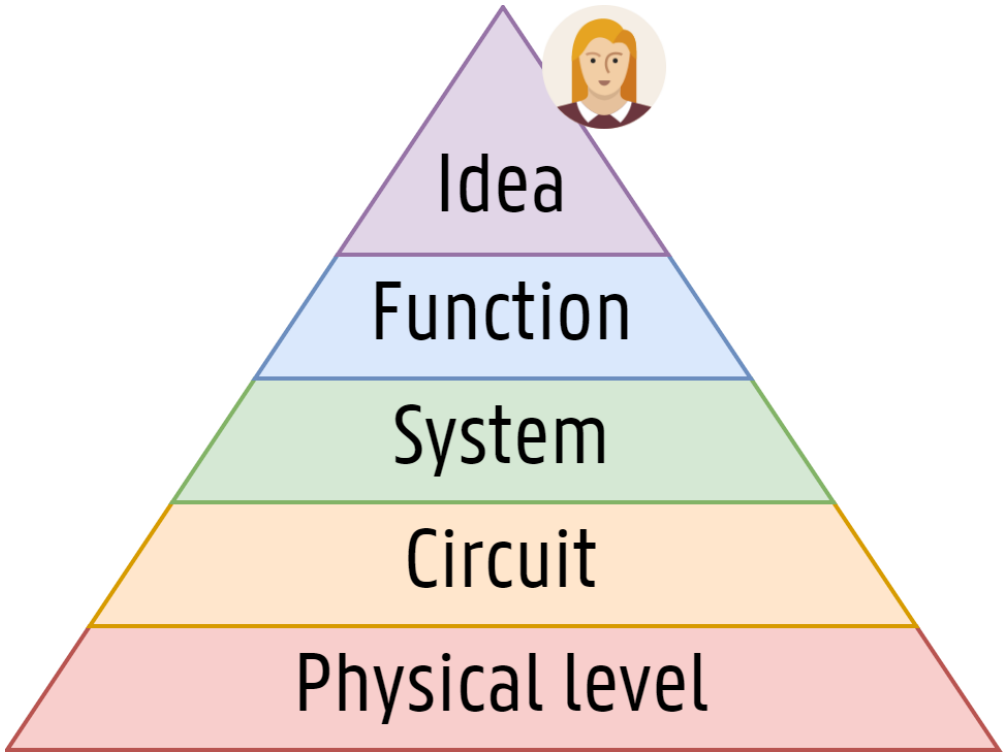

Figure 2 | Levels of abstraction in photonic circuit design.

Levels of abstraction

- Currently low
- We want to go higher
- We want to go **much** higher

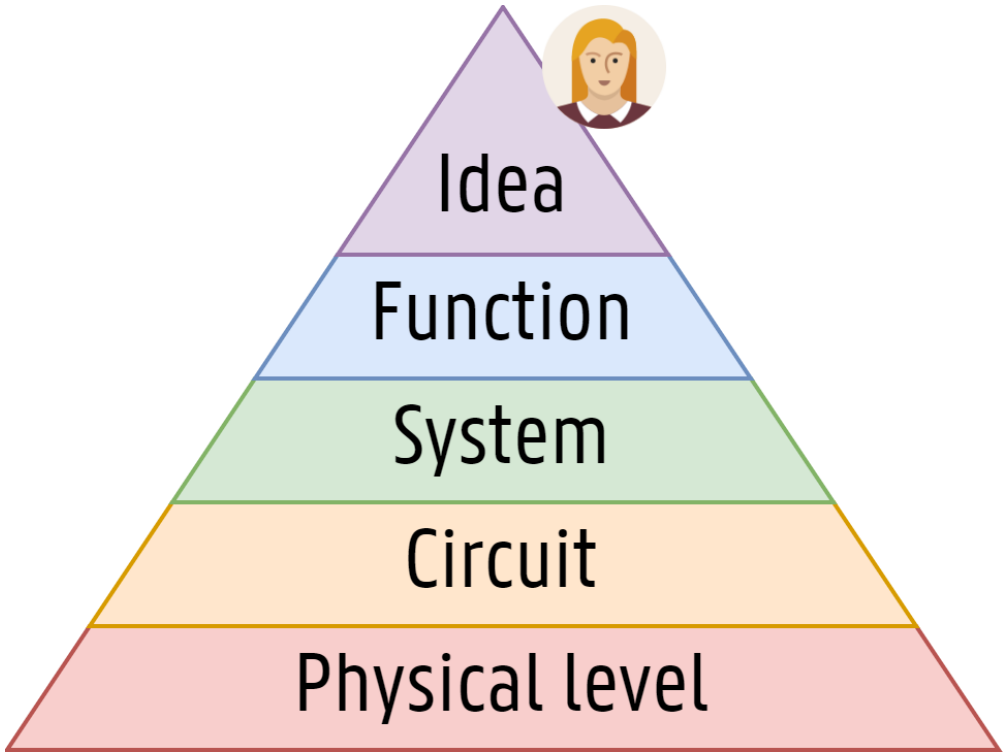

FIGURE 2 | Levels of abstraction in photonic circuit design.

Levels of abstraction

- Currently low
- We want to go higher
- We want to go **much** higher
- We need to build abstractions
  - Components (parametric)

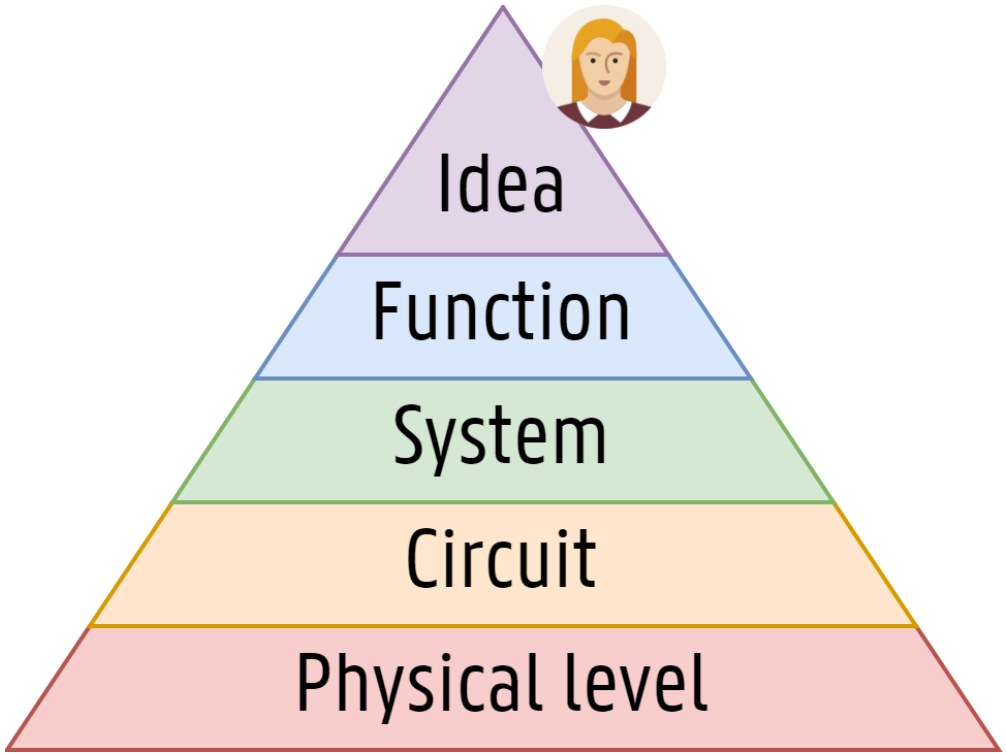

Figure 2 | Levels of abstraction in photonic circuit design.

Levels of abstraction

- Currently low
- We want to go higher
- We want to go **much** higher
- We need to build abstractions
  - Components (parametric)
  - Signal flow graphs

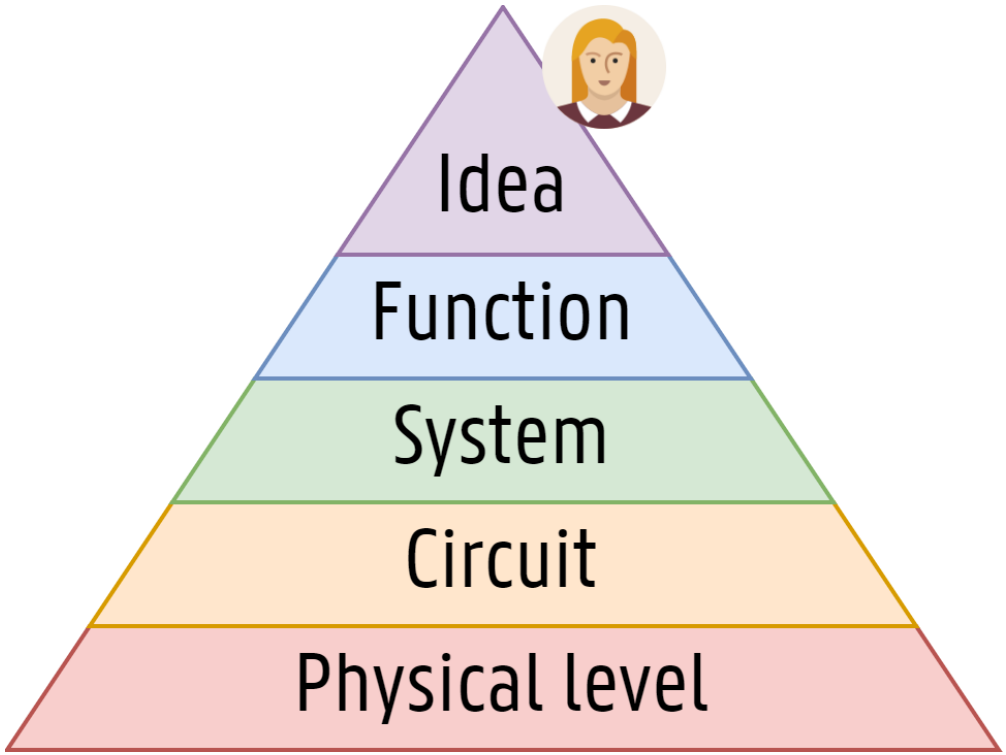

FIGURE 2 | Levels of abstraction in photonic circuit design.

Levels of abstraction

- Currently low
- We want to go higher
- We want to go **much** higher
- We need to build abstractions
  - Components (parametric)
  - Signal flow graphs
  - Black boxes

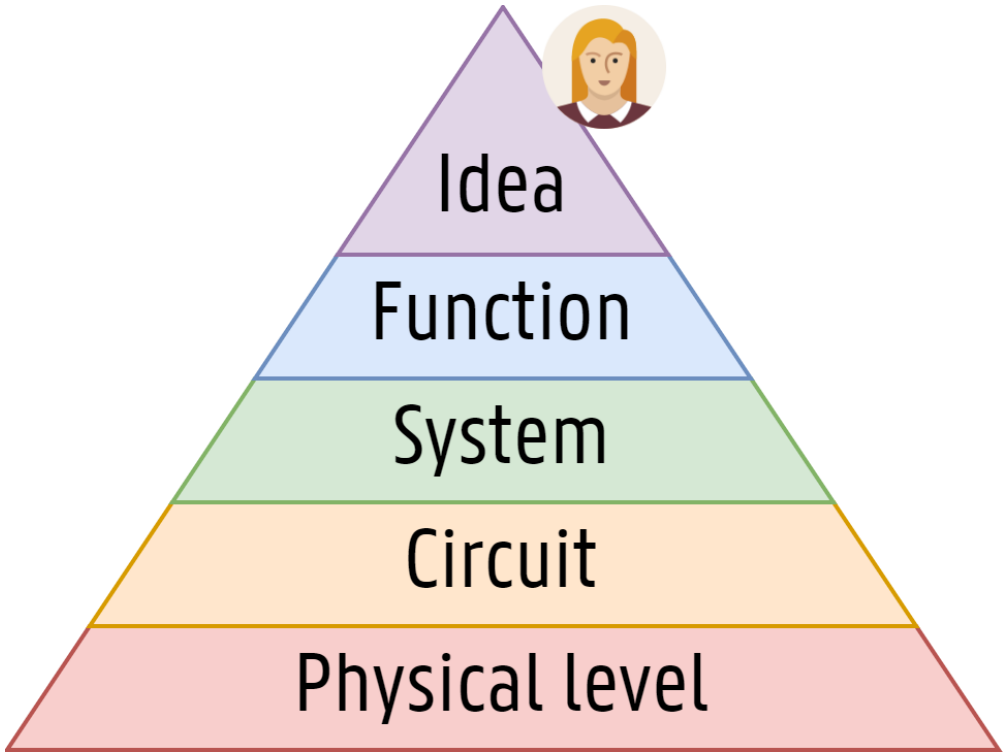

FIGURE 2 | Levels of abstraction in photonic circuit design.

Levels of abstraction

- Currently low
- We want to go higher
- We want to go **much** higher
- We need to build abstractions
  - Components (parametric)
  - Signal flow graphs
  - Black boxes
  - ???

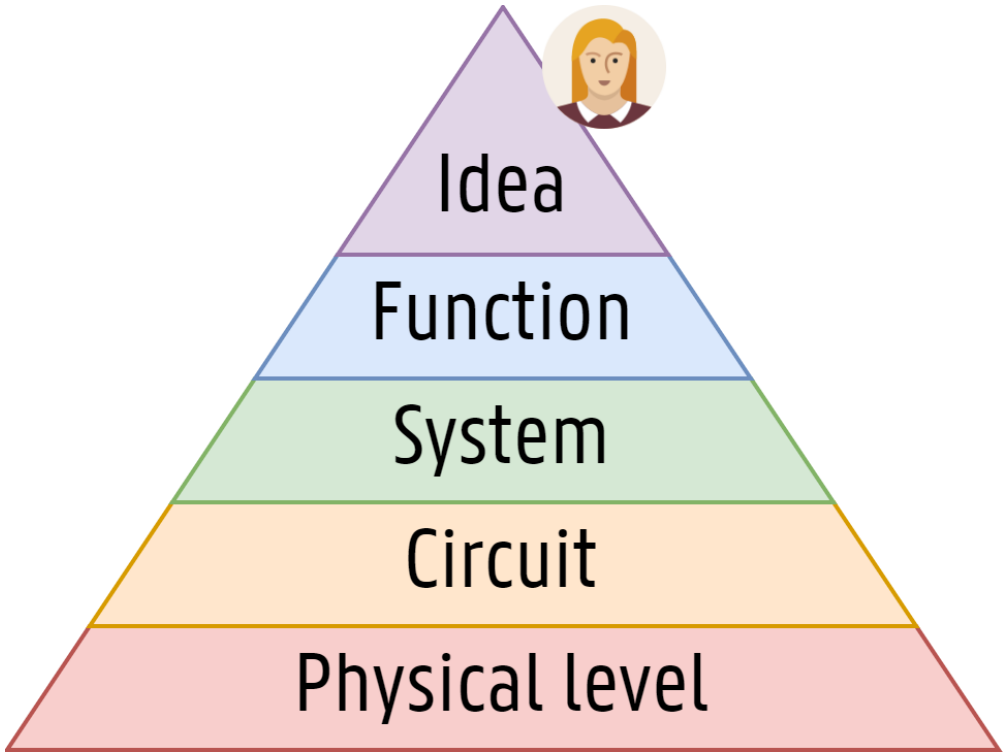

Figure 2 | Levels of abstraction in photonic circuit design.

Introducing PHÔS

- PHÔS is a **domain-specific language**

Introducing PHÔS

- PHÔS is a **domain-specific language**
- PHÔS describes **photonic circuits**

Introducing PHÔS

- PHÔS is a **domain-specific language**
- PHÔS describes **photonic circuits**
- PHÔS is **declarative**

Introducing PHÔS

- PHÔS is a **domain-specific language**
- PHÔS describes **photonic circuits**
- PHÔS is **declarative**
- PHÔS is **parametric**

Introducing PHÔS

- PHÔS is a **domain-specific language**
- PHÔS describes **photonic circuits**
- PHÔS is **declarative**
- PHÔS is **parametric**
- PHÔS is **expressive**

- PHÔS is the **function** and **system** levels
  - Filter synthesis
  - Signal flow graph generation
  - Component instantiation
  - Reconfigurability & tunability
  - Optimization

Introducing PHÔS

- PHÔS is a **domain-specific language**
- PHÔS describes **photonic circuits**
- PHÔS is **declarative**
- PHÔS is **parametric**
- PHÔS is **expressive**
- PHÔS is **extensible**

- PHÔS is the **function** and **system** levels
  - Filter synthesis
  - Signal flow graph generation
  - Component instantiation
  - Reconfigurability & tunability
  - Optimization
- PHÔS is **not** at the component level
  - ~~Component design~~
  - ~~Component simulation~~
  - ~~Component optimization~~

Programmatic description: an overview

GHENT
UNIVERSITY

Why a new language?

- Existing languages **do not** works for photonics
  - Hardware description languages: ~~VHDL~~, ~~MyHDL~~
  - High-level synthesis languages: ~~SystemC~~
  - Analog modeling languages: ~~Verilog-AMS~~, ~~SPICE~~
  - Traditional programming languages: ~~Python~~, ~~Rust~~

Why a new language?

- Existing languages **do not** works for photonics
  - Hardware description languages: ~~VHDL~~, ~~MyHDL~~
  - High-level synthesis languages: ~~SystemC~~
  - Analog modeling languages: ~~Verilog-AMS~~, ~~SPICE~~
  - Traditional programming languages: ~~Python~~, ~~Rust~~
- Libraries are **not expressive** enough

Why a new language?

- Existing languages **do not** works for photonics
  - Hardware description languages: ~~VHDL~~, ~~MyHDL~~
  - High-level synthesis languages: ~~SystemC~~
  - Analog modeling languages: ~~Verilog-AMS~~, ~~SPICE~~
  - Traditional programming languages: ~~Python~~, ~~Rust~~
- Libraries are **not expressive** enough
- Why? **Because photonics is different**

Why a new language?

- Existing languages **do not** works for photonics
  - Hardware description languages: ~~VHDL~~, ~~MyHDL~~
  - High-level synthesis languages: ~~SystemC~~
  - Analog modeling languages: ~~Verilog-AMS~~, ~~SPICE~~
  - Traditional programming languages: ~~Python~~, ~~Rust~~
- Libraries are **not expressive** enough
- Why? **Because photonics is different**
- We need a **domain-specific language**

Examples

Conclusion

GHENT
UNIVERSITY

Sources

Thank you for listening

GHENT
UNIVERSITY

FACULTY OF ENGINEERING
AND ARCHITECTURE

Sébastien d'Herbais de Thun

Student

Photonics Research Group

E sebastien.dherbaisdethun@ugent.be
M +32 (0) 473 12 86 57

www.ugent.be

Universiteit Gent
@ugent
@ugent
Ghent University

GHENT
UNIVERSITY