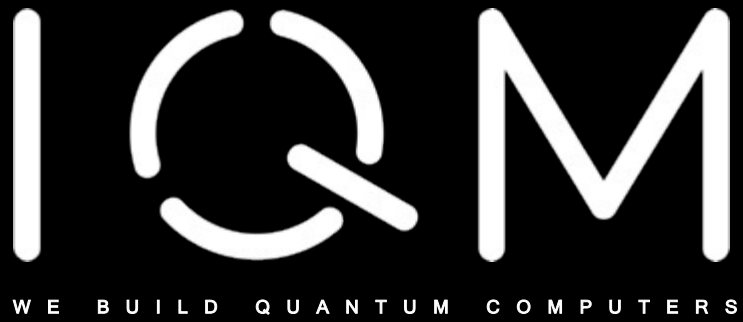
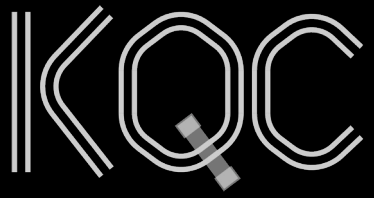
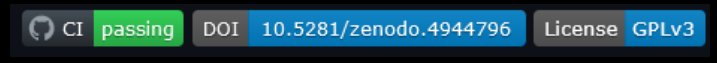


Winter School

Superconducting quantum processor design with KQCircuits®



github.com/iqm-finland/KQCircuits



Schedule

Monday	Tuesday	Wednesday	Thursday	Friday
Caspar & Pavel	Alessandro	Alessandro	Niko & Eelis	Caspar
Introduction to QPU design Installing KQCircuits First look around	Introduction to designing Create a custom qubit element	Design a custom chip	Finite element simulations	Mask export Composite waveguides GUI

Mask export

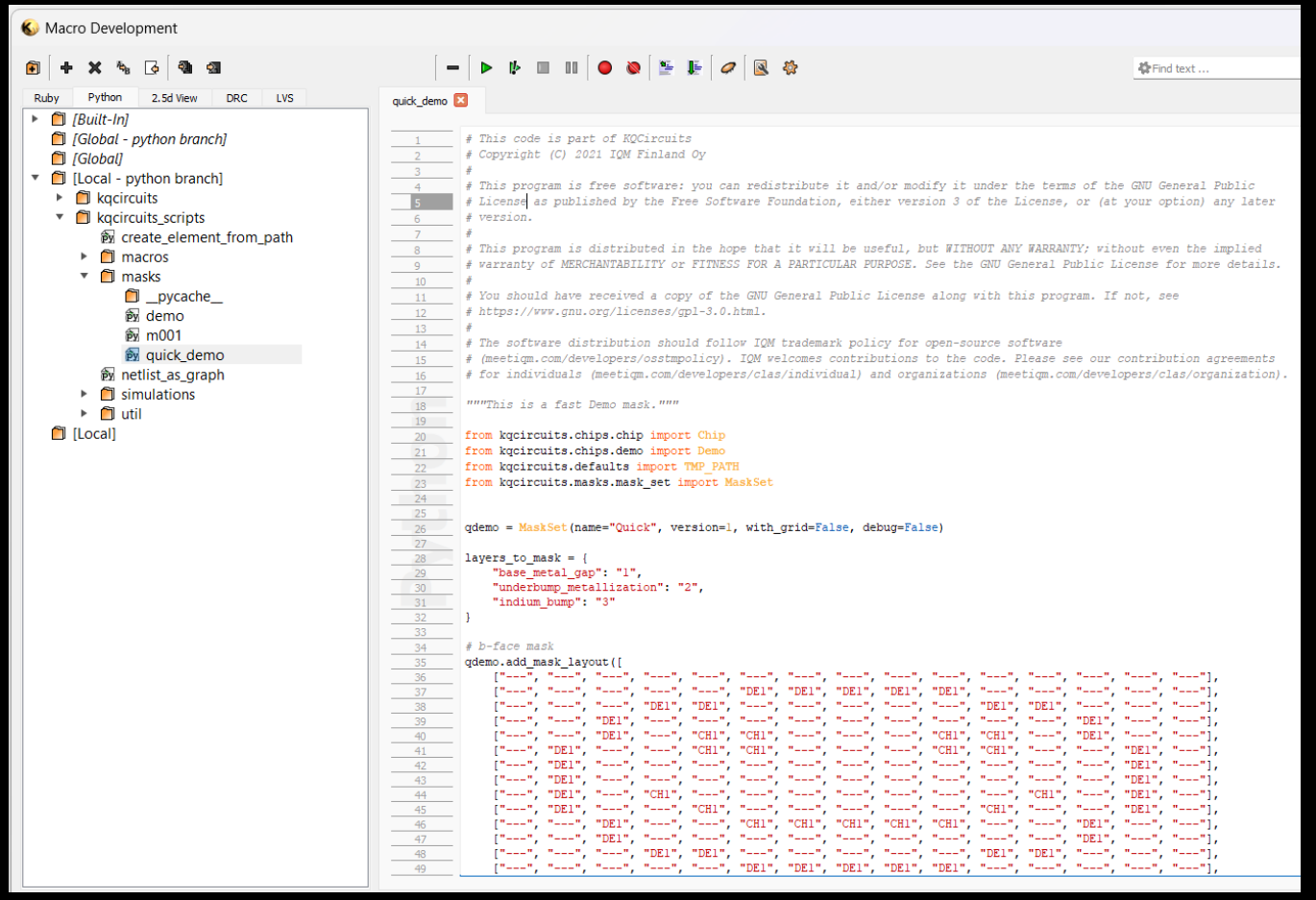
Purpose of KQCircuits mask export:

- Merge metallization layers
  - Turn `base_metal_gap_wo_grid`, `ground_grid` and `base_metal_added` into `base_metal_gap`
- Generate masks for wafer-scale lithography
  - Adds alignment markers
  - Configurable which layers are exported to a mask
  - Optionally invert layer polarity
- Export individual chips and related files

Mask export

In the KLayout Macro editor (F5):

- Open kqcircuits_scripts/masks/quick_demo.py
- Run with 
- Files are exported to KQCircuits/tmp folder



Mask export script

Create a `MaskSet` with the name, version, and optional ground grid setting

```
qdemo = MaskSet(name="Quick", version=1, with_grid=False, debug=False)
```

Define the chip map to place different chips or variants – give each variant a three-letter code

```
qdemo.add_mask_layout([
    ["---", "---", "---", "---", "---", "---", "---", "---", "---", "---", "---", "---", "---", "---", "---"],
    ["---", "---", "---", "---", "---", "DE1", "DE1", "DE1", "DE1", "DE1", "---", "---", "---", "---", "---"],
    ["---", "---", "---", "DE1", "DE1", "---", "---", "---", "---", "---", "DE1", "DE1", "---", "---", "---"],
    ["---", "---", "DE1", "---", "---", "---", "---", "---", "---", "---", "---", "---", "DE1", "---", "---"],
    ["---", "---", "DE1", "---", "CH1", "CH1", "---", "---", "---", "CH1", "CH1", "---", "DE1", "---", "---"],
    ["---", "DE1", "---", "---", "CH1", "CH1", "---", "---", "---", "CH1", "CH1", "---", "---", "DE1", "---"],
    ["---", "DE1", "---", "---", "---", "---", "---", "---", "---", "---", "---", "---", "---", "DE1", "---"],
    ["---", "DE1", "---", "---", "---", "---", "---", "---", "---", "---", "---", "---", "---", "DE1", "---"],
    ["---", "DE1", "---", "CH1", "---", "---", "---", "---", "---", "---", "---", "CH1", "---", "DE1", "---"],
    ["---", "DE1", "---", "---", "CH1", "---", "---", "---", "---", "---", "CH1", "---", "---", "DE1", "---"],
    ["---", "---", "DE1", "---", "---", "CH1", "CH1", "CH1", "CH1", "CH1", "---", "---", "DE1", "---", "---"],
    ["---", "---", "DE1", "---", "---", "---", "---", "---", "---", "---", "---", "---", "DE1", "---", "---"],
    ["---", "---", "---", "DE1", "DE1", "---", "---", "---", "---", "---", "DE1", "DE1", "---", "---", "---"],
    ["---", "---", "---", "---", "---", "DE1", "DE1", "DE1", "DE1", "DE1", "---", "---", "---", "---", "---"],
    ["---", "---", "---", "---", "---", "---", "---", "---", "---", "---", "---", "---", "---", "---", "---"],
], "ltl", layers_to_mask=layers_to_mask)
```

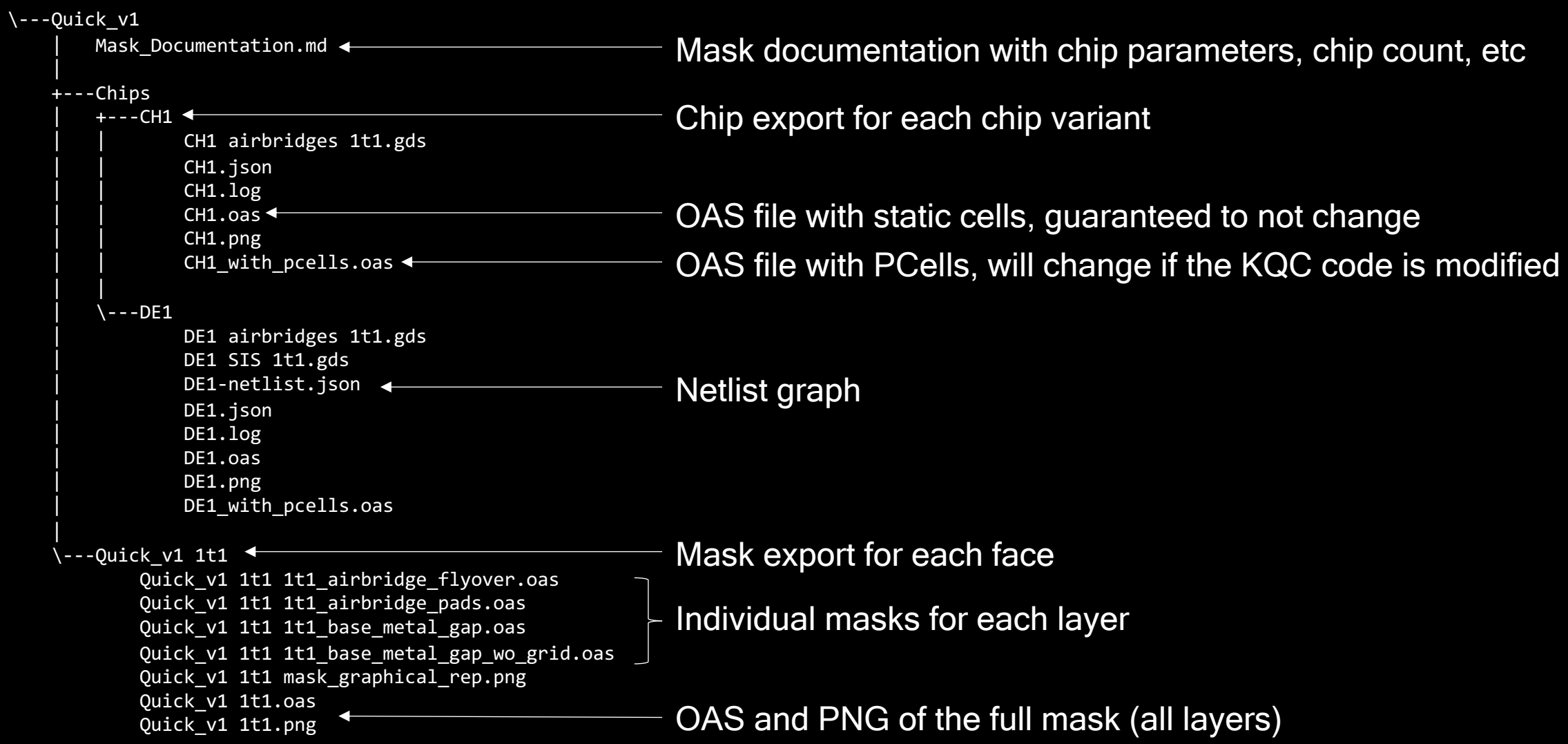
Add chip variants –specify the chip class and parameters

```
qdemo.add_chips([
    (Chip, "CH1"),
    (Demo, "DE1", parameter=value, ...),
], threads=2)
```

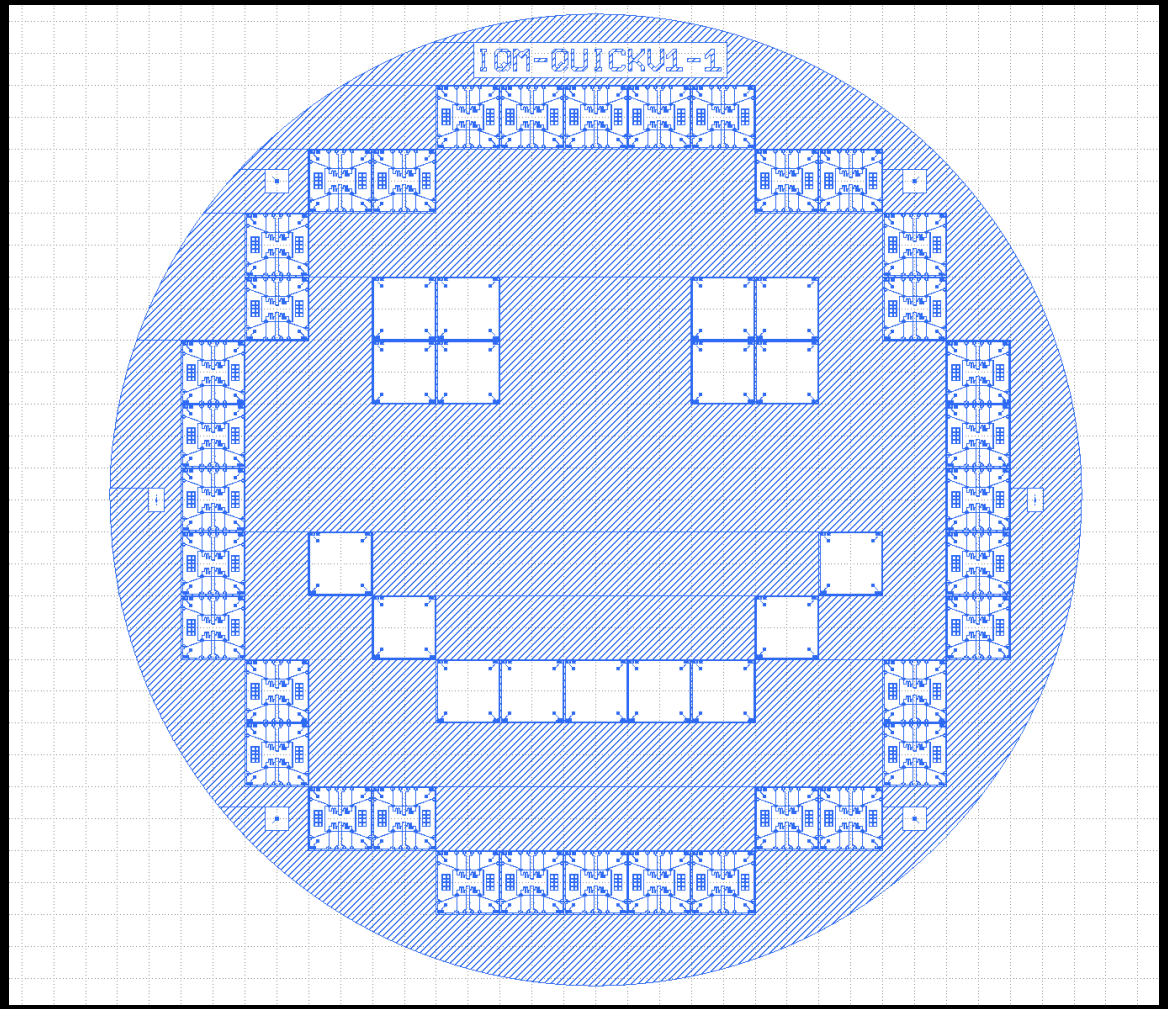
Build and export the mask

```
qdemo.build()
qdemo.export(TMP_PATH)
```

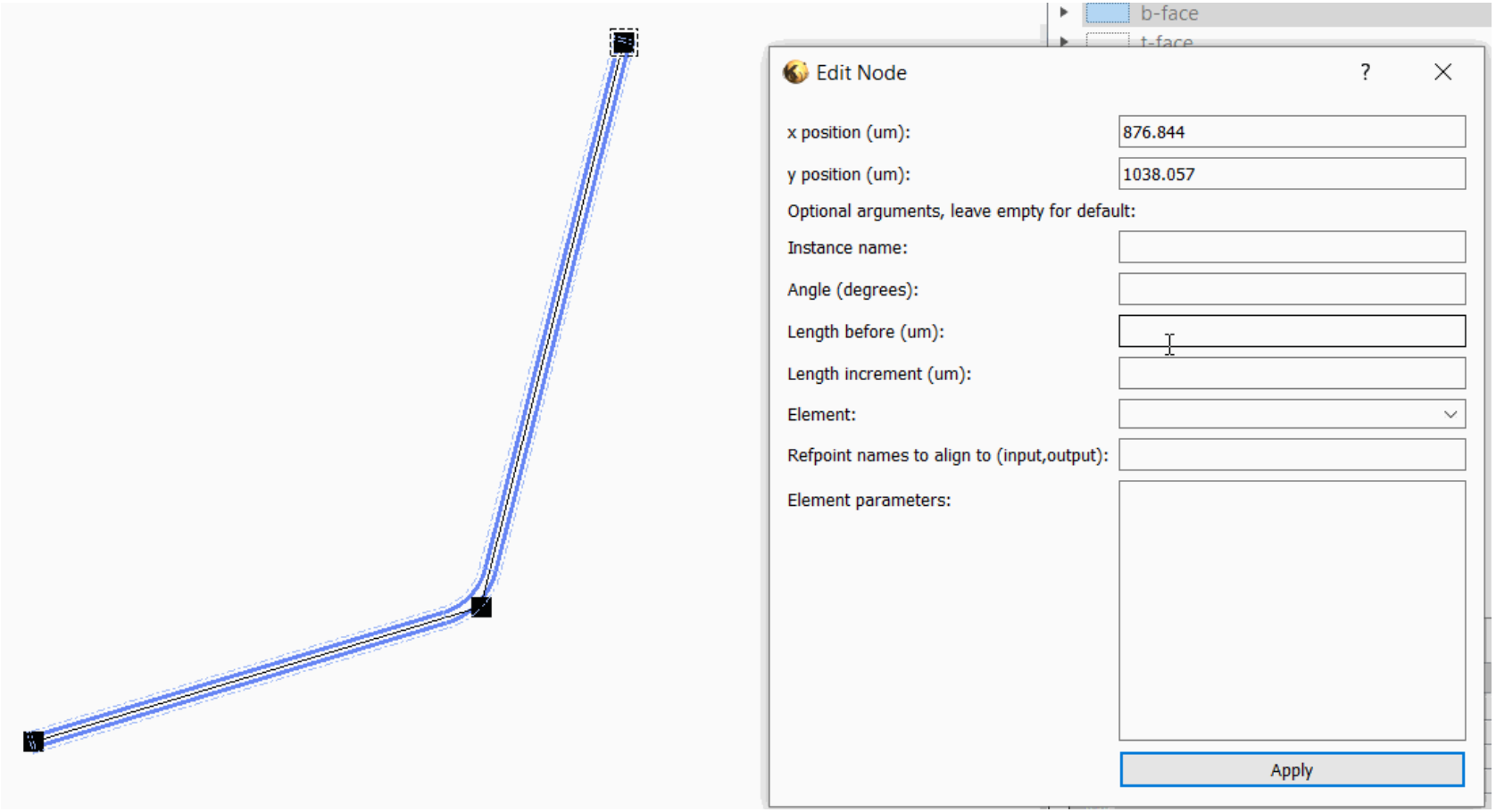
Exported files



Quick_v1 1t1 1t1_base_metal_gap.oas



GUI editing

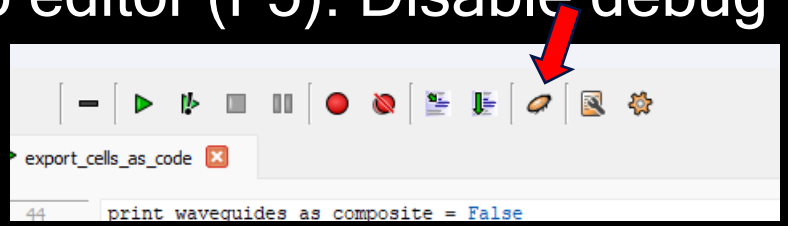

b-face
t-face
Edit Node
x position (um):
876.844
y position (um):
1038.057
Optional arguments, leave empty for default:
Instance name:
Angle (degrees):
Length before (um):
Length increment (um):
Element:
Refpoint names to align to (input,output):
Element parameters:
Apply

Configure KLayout for GUI editing

Under *File* → *Setup* :

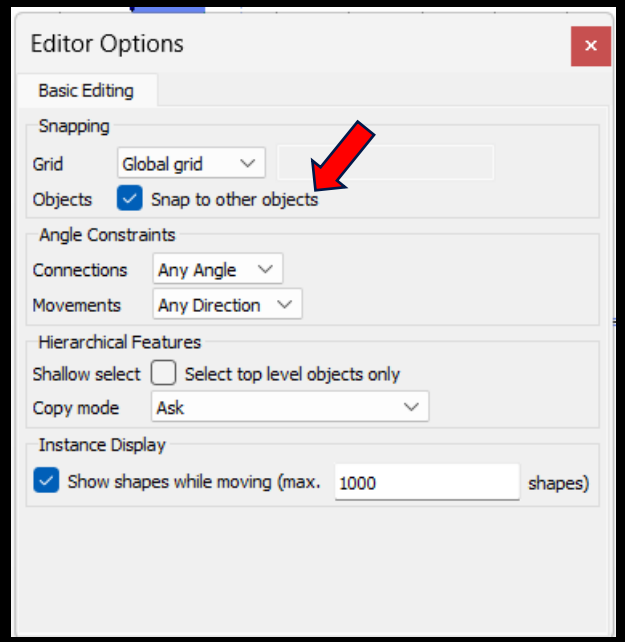
- *Application* → *Default Grids* : Add courser grid values, for example 10 and 100 um
  - While editing, select the grid you like under *View* → *Grid*.
- *Display* → *Cells* : Enable *Show PCell guiding and error shapes*.

Macro editor (F5): Disable debug mode

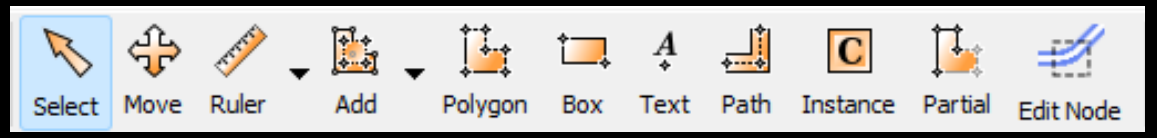


Editor Options (select *Partial* tool, then F3):

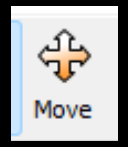
- *Snap to other objects* (optional)



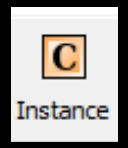
KLayout toolbar



Select: select cells, double click (Q) to change parameters

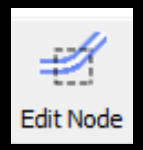


Move: move an entire PCell.



Instance: place a new PCell instance (or just drag from the library)

Partial: edit “guiding shapes”, includes routing waveguides

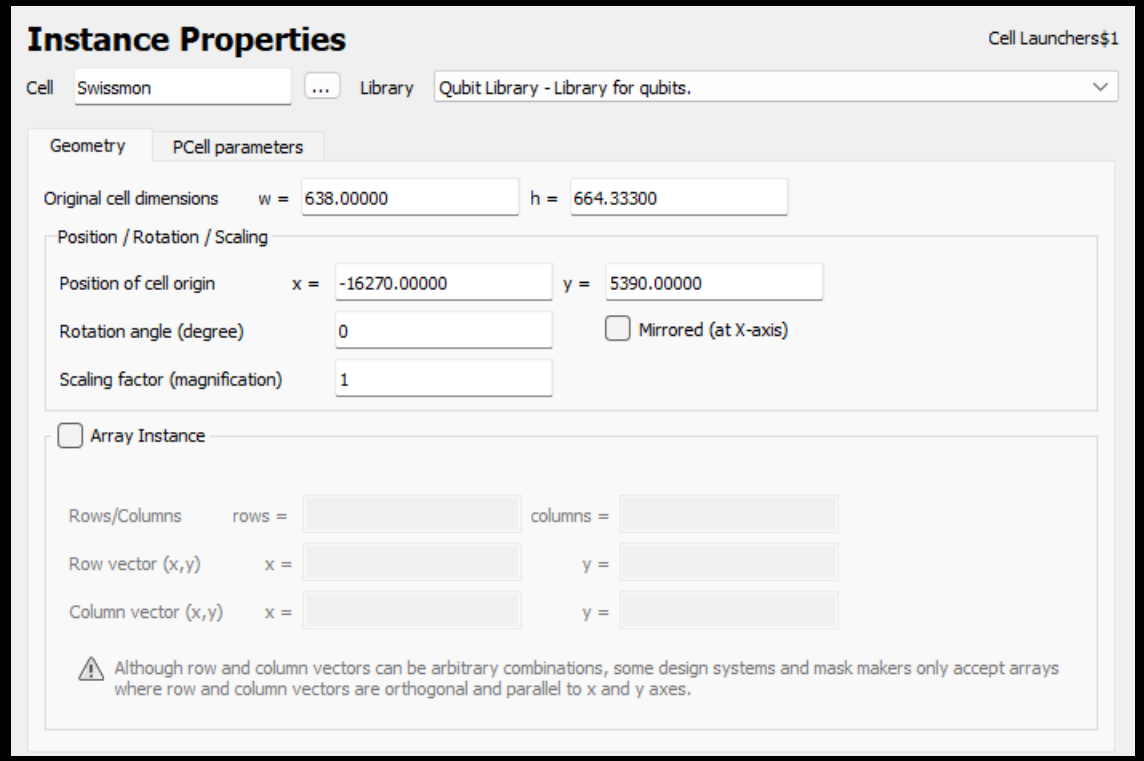


Edit Node: Edit advanced properties of a WaveguideComposite node

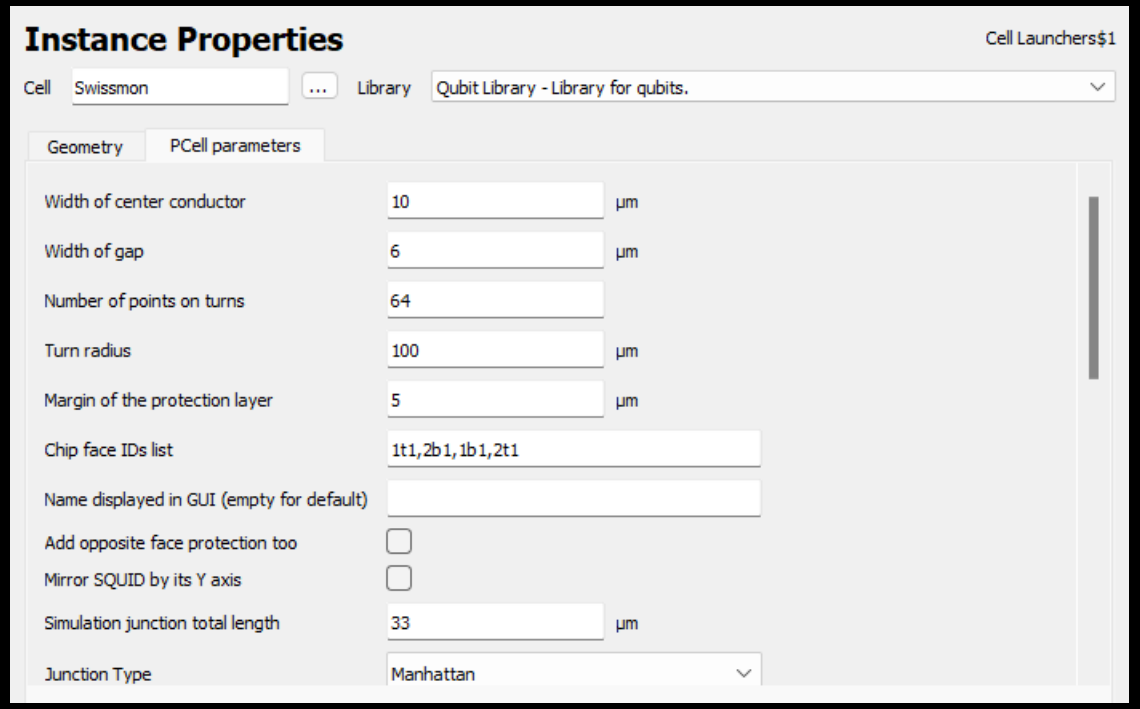
Editing PCell properties

Select cell, double click or press Q

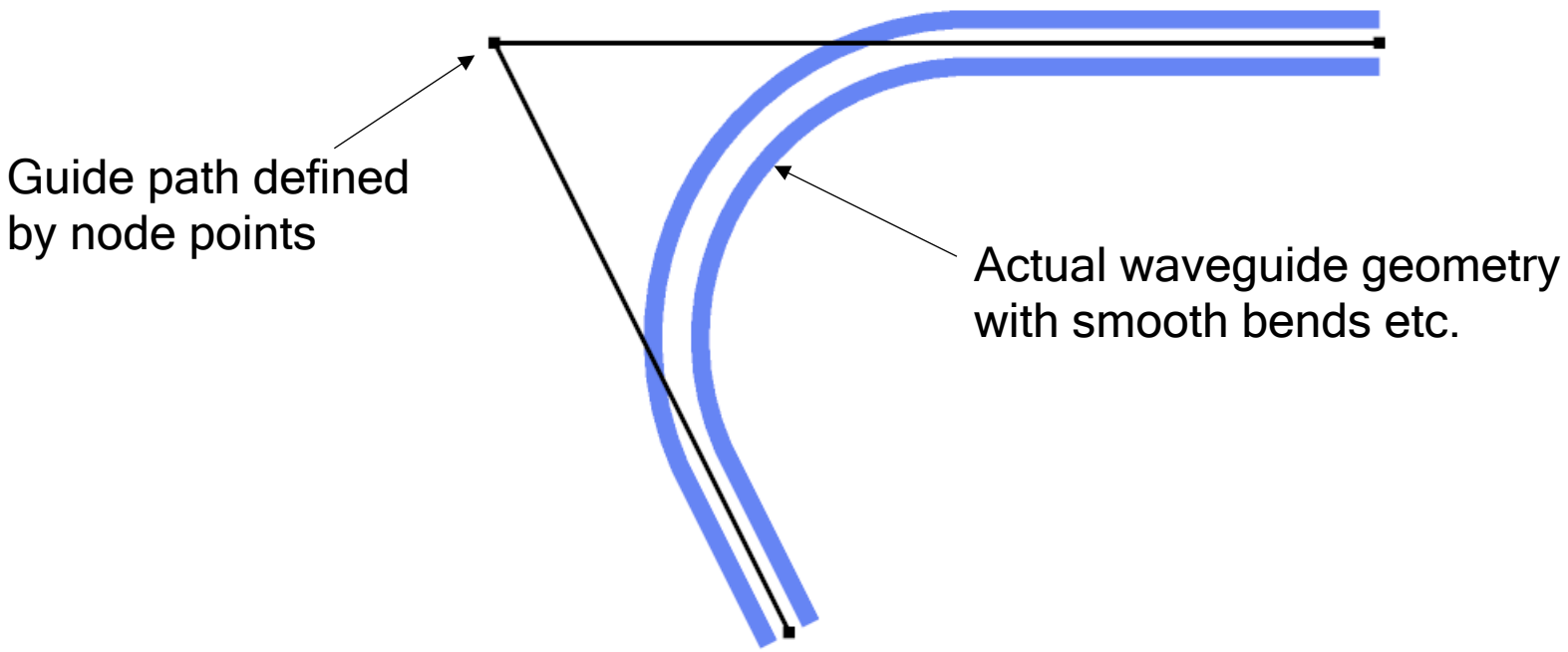
Geometry: position, scale and rotation



PCell parameters

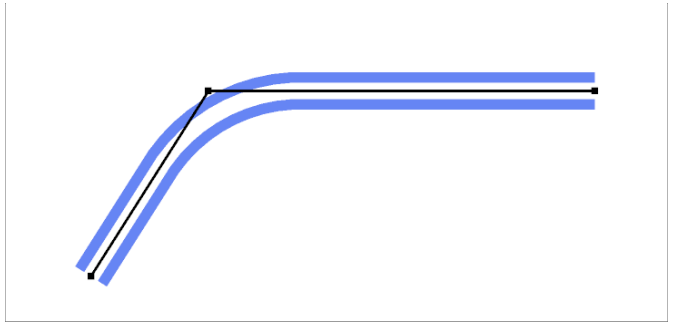


Editing waveguides

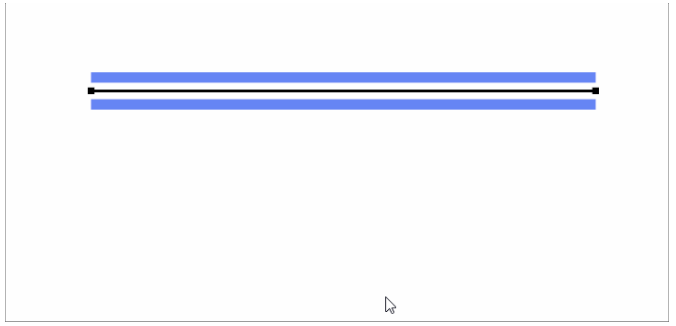


Use the Partial tool to edit the guide path

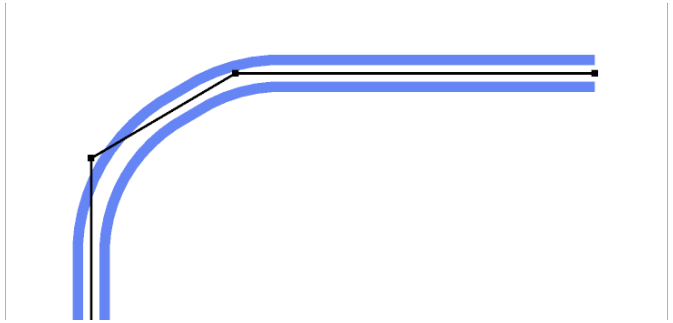
Editing with Partial tool



Click node, move, click
Move a single node

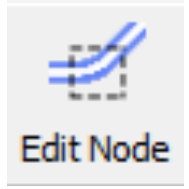


Double click: Add node
Click node, DEL key: Delete node



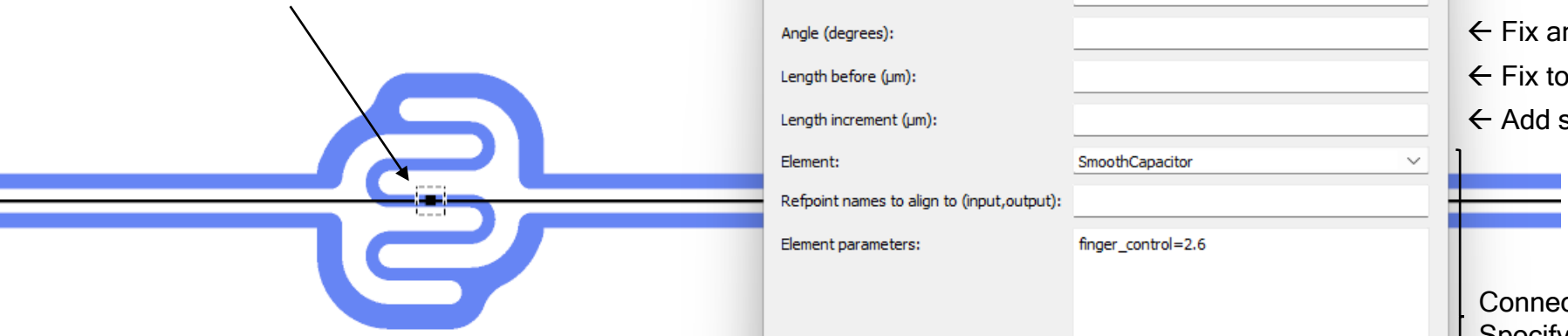
Click edge, move, click
Parallel move

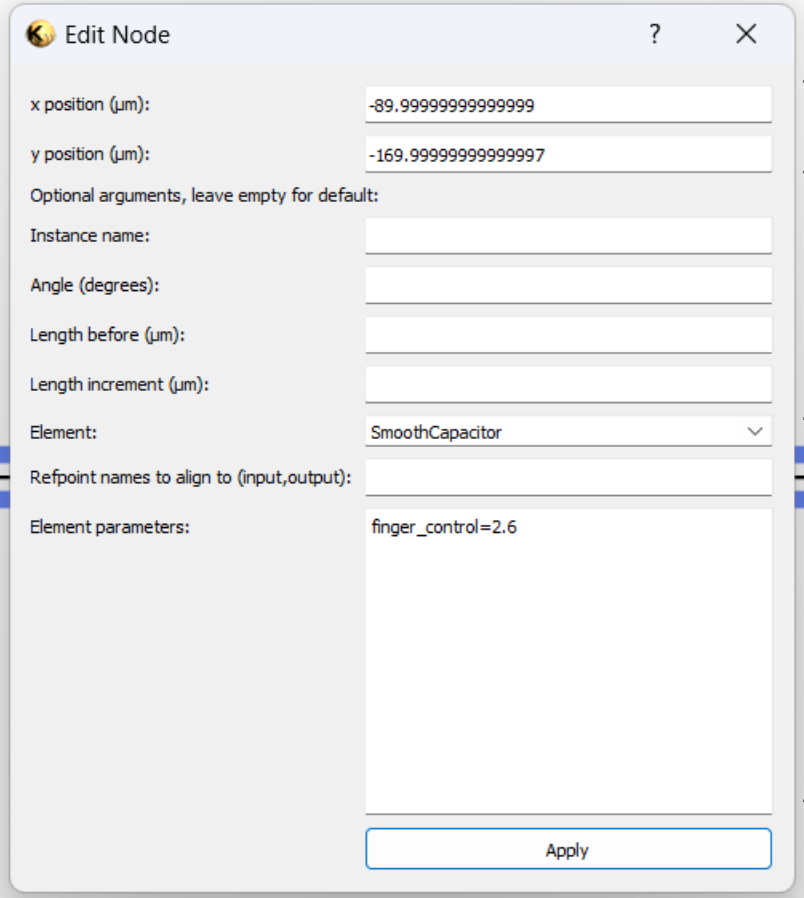
Edit Node tool



Only works on WaveguideComposite elements

Dashed box shows
Selected node





Node position

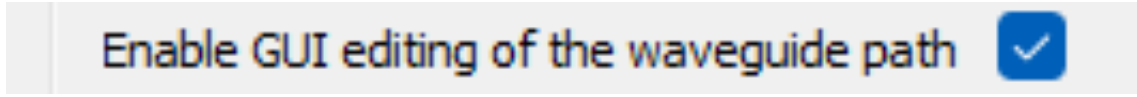
← Fix angle of the waveguide at this node

← Fix total length of segment before this node

← Add some amount of length w.r.t. straight waveguide

Connect inline element,
Specify inline element parameters as key=value pairs

WaveguideComposite parameters



Set to False for “code-only” waveguides to avoid guide path clutter

Defines how corners are made when connecting inline elements or using fixed node angles.

- `False` (default): turning point at one turn radius from the element - maximum corner angle 90 degrees.
- `True`: corners start turning as close to the element as possible – more flexible

GUI Editing with code generation

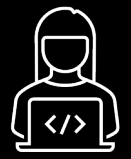
1. Place a chip in the layout
   - Use `Launchers` to start with an empty chip with only launchers
2. Convert the chip to a static cell and show as top
   - Select the cell, then *Edit → Cell → Convert Cell to Static*
   - Right click the cell, choose *Show As New Top*.
3. Edit or place elements
4. Export as code
   - Open the macro editor (F5)
   - Find `kqcircuits_scripts/macros/export/export_cells_as_code`
   - Adjust options (e.g. `output_format = “insert_cell+chip”`)
   - Copy the code to a new chip file, modify as needed.

Thank you for participating!

What to do next?

Switch to the main GitHub repository: https://github.com/iqm-finland/KQCircuits

Keep the discussion going in the Discord: https://discord.gg/4wP8WAeWuJ



Contribute with bug reports, feature requests and code!

We are hiring!

TECHNOLOGY · IQM GERMANY (MUNICH)

Quantum Engineer, Device Design

Apply for this job

What will I be doing?

As a member of our simulations team, you will design quantum processing units and contribute to the core technology of our quantum computers. This role has an emphasis on developing and integrating the tools to scale up quantum processor design. Your tasks will mainly focus on developing tools to integrate and automate the design workflow, while also significantly extending our open-source design framework KQCircuits. Depending on your background and interest, the work can also include:

- Quantum physics and design calculations.
- Electromagnetic field simulations, including finite element and lumped circuit modeling.
- Numerical modeling and optimization.

See all positions at https://meetiqm.com/careers/