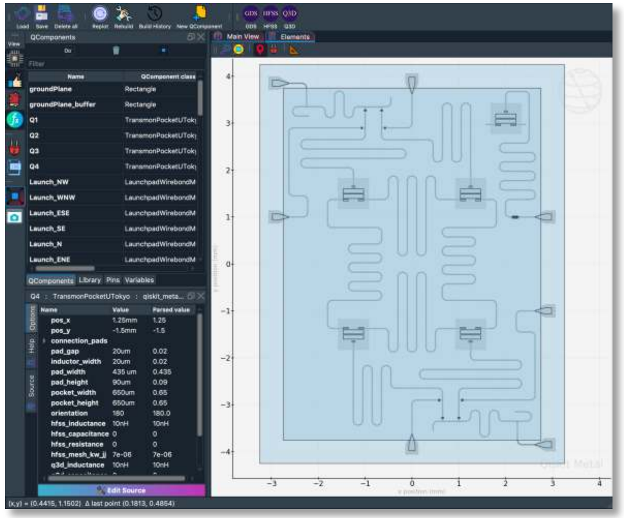
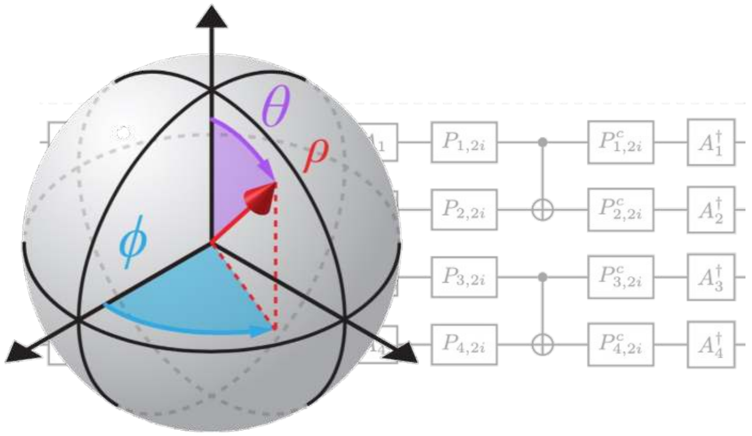
Overview of Quantum Hardware Design

From *Fab-it* to *Qubit* — Qiskit Metal





Zlatko K. Minev

IBM Quantum

@zlatko_minev

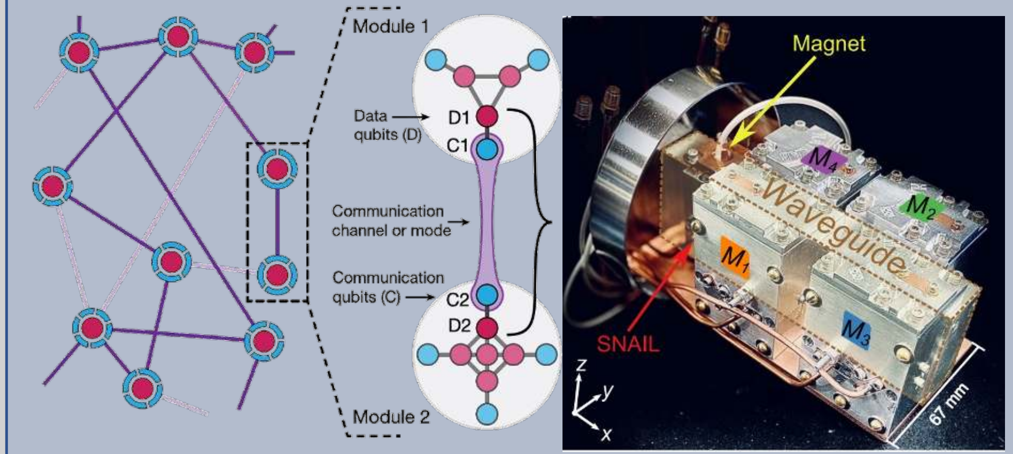
zlatko-minev.com

qiskit.org/metal

High-level modules?

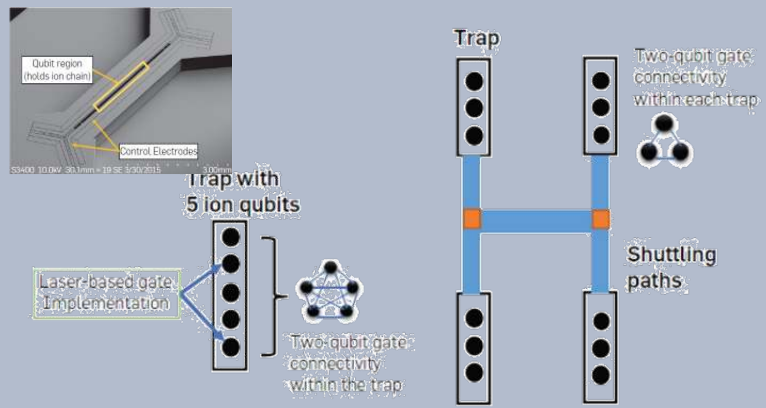
(A somewhat random sampling of modules)

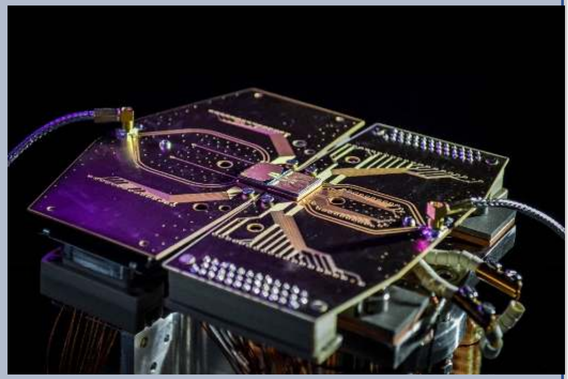
3D su-con



Chou *et al*. Nature 561, 368 … Zhou *et al*. arXiv:2109.06848 …

Ions





Murali *et al.* Comm. of ACM v65-3, 101 …

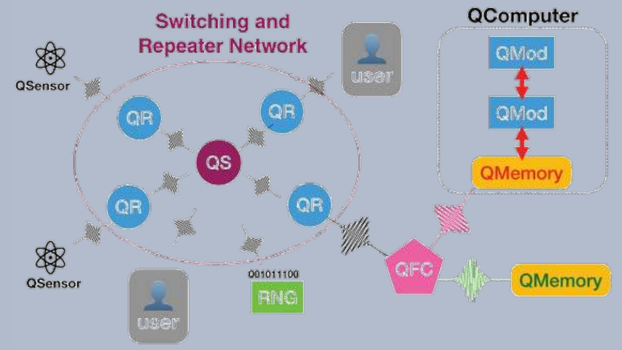
U Sussex …

2D / 2.5D su-con

Gold *et al*., npj QI 7, 142

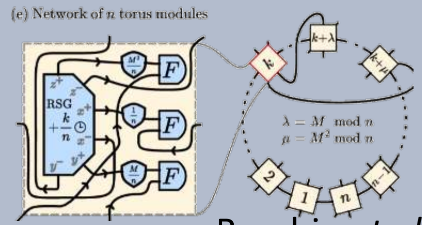
IBM …

"Quantum interconnects"*

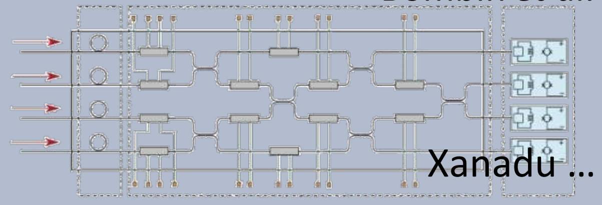


Awschalom *et al*. PRX Quantum 2, 017002 …

Photonic



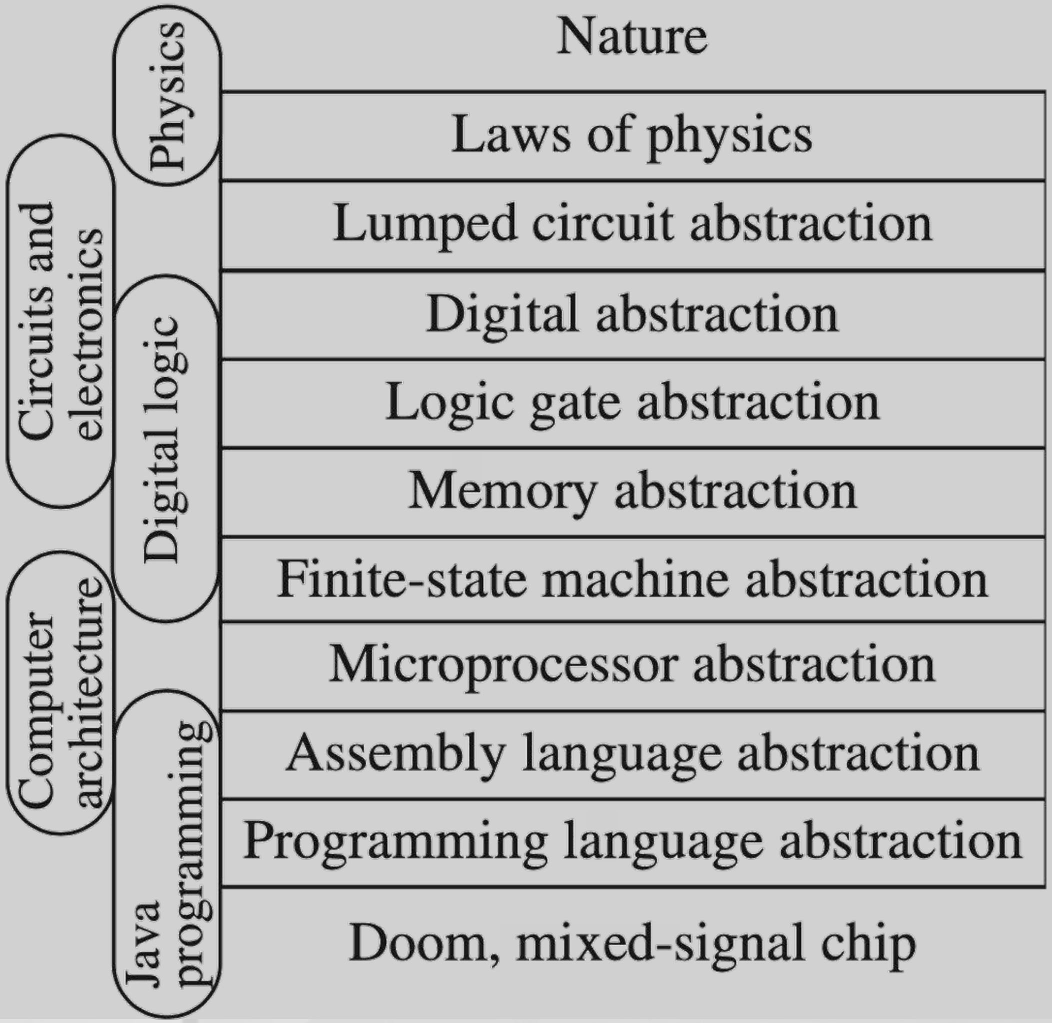
Bombin *et al.* …



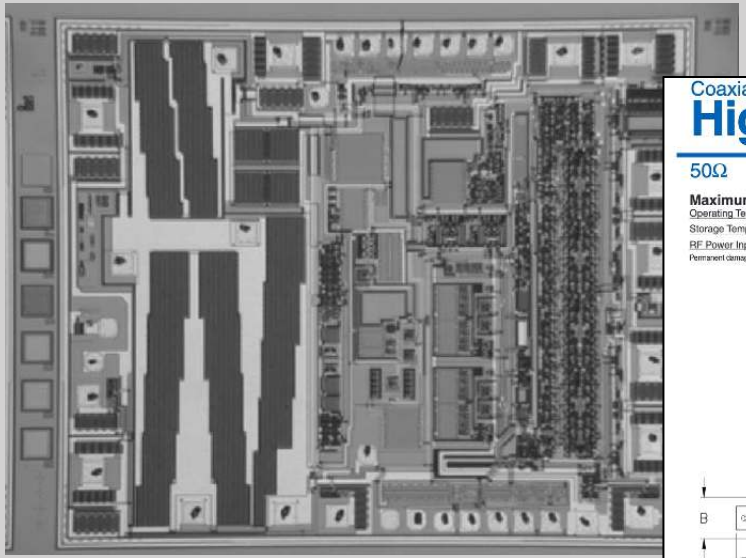
Xanadu …

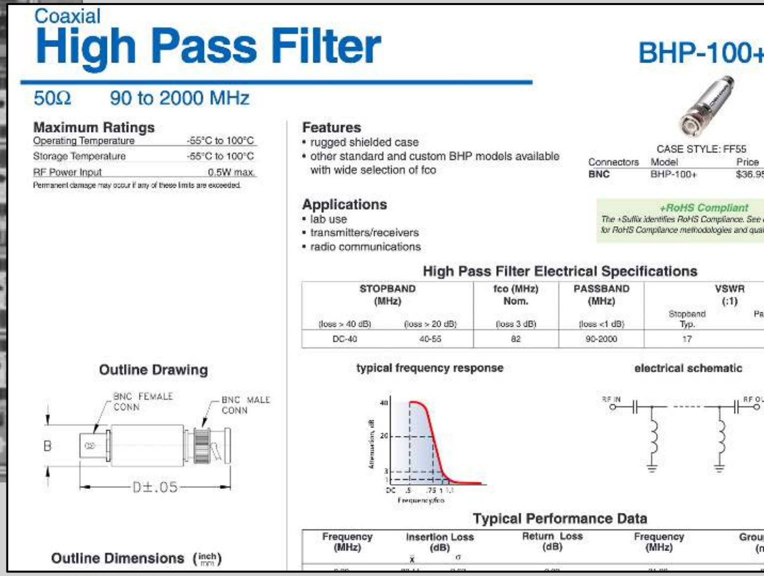
Classical hierarchy of abstractions

Quantum?

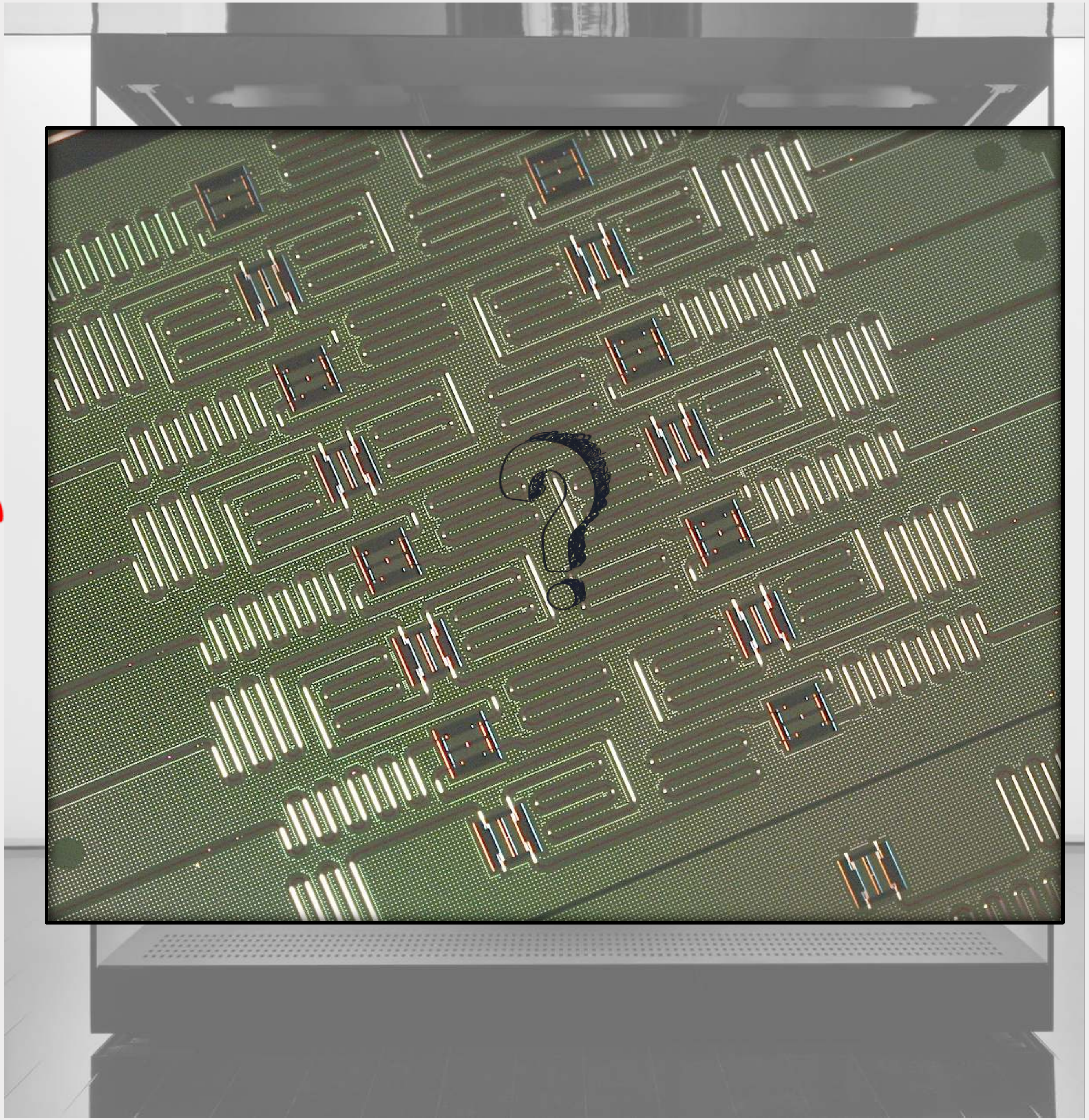


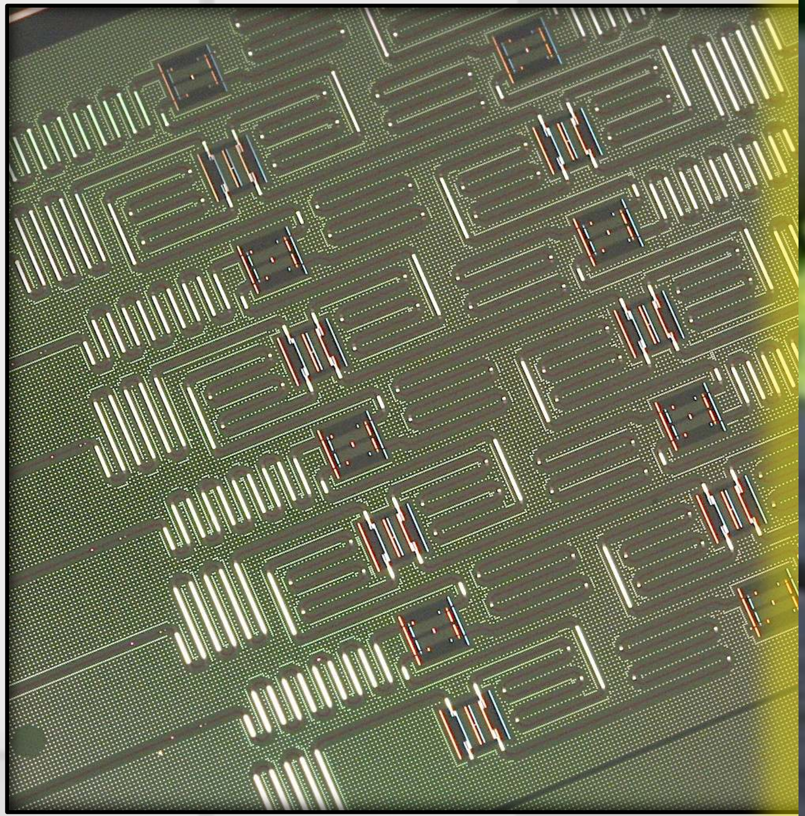
Abstraction layers
Effective models
Modularity
General simulators
…



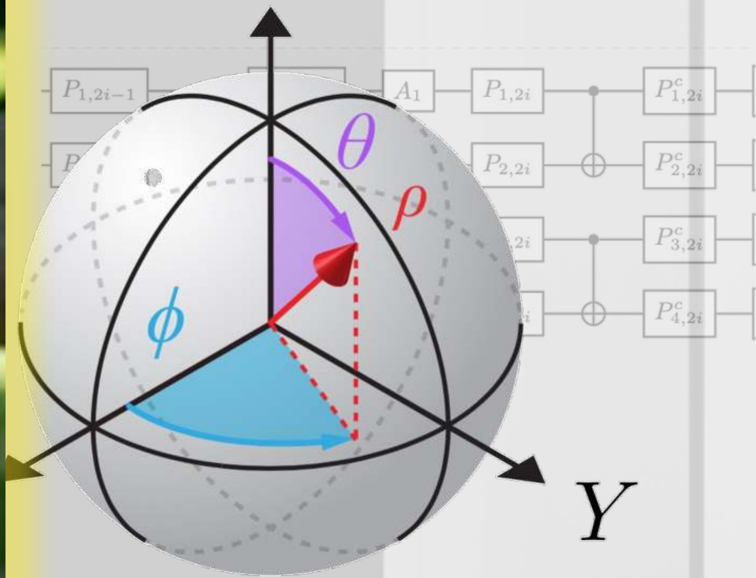


How to design a quantum processor?

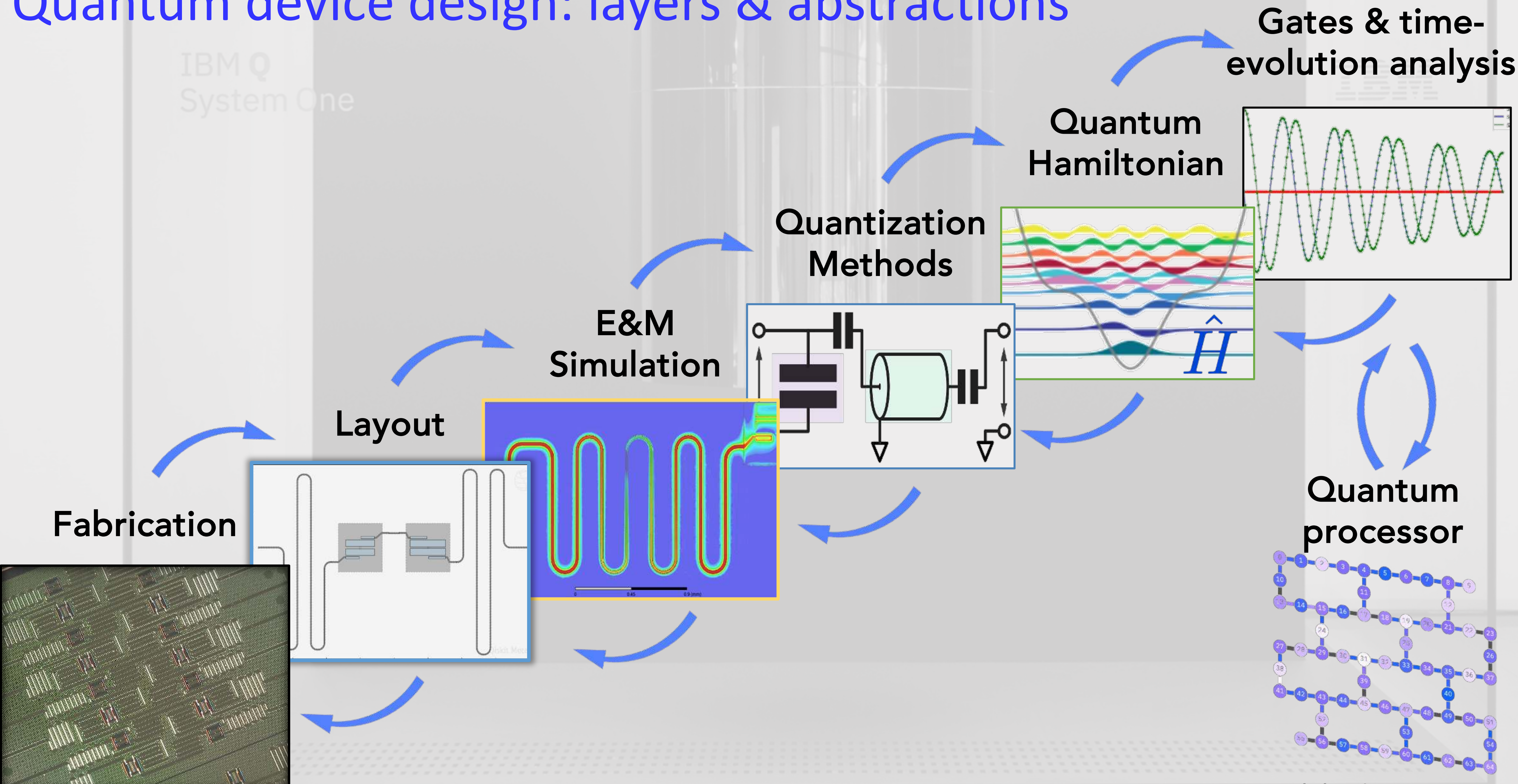




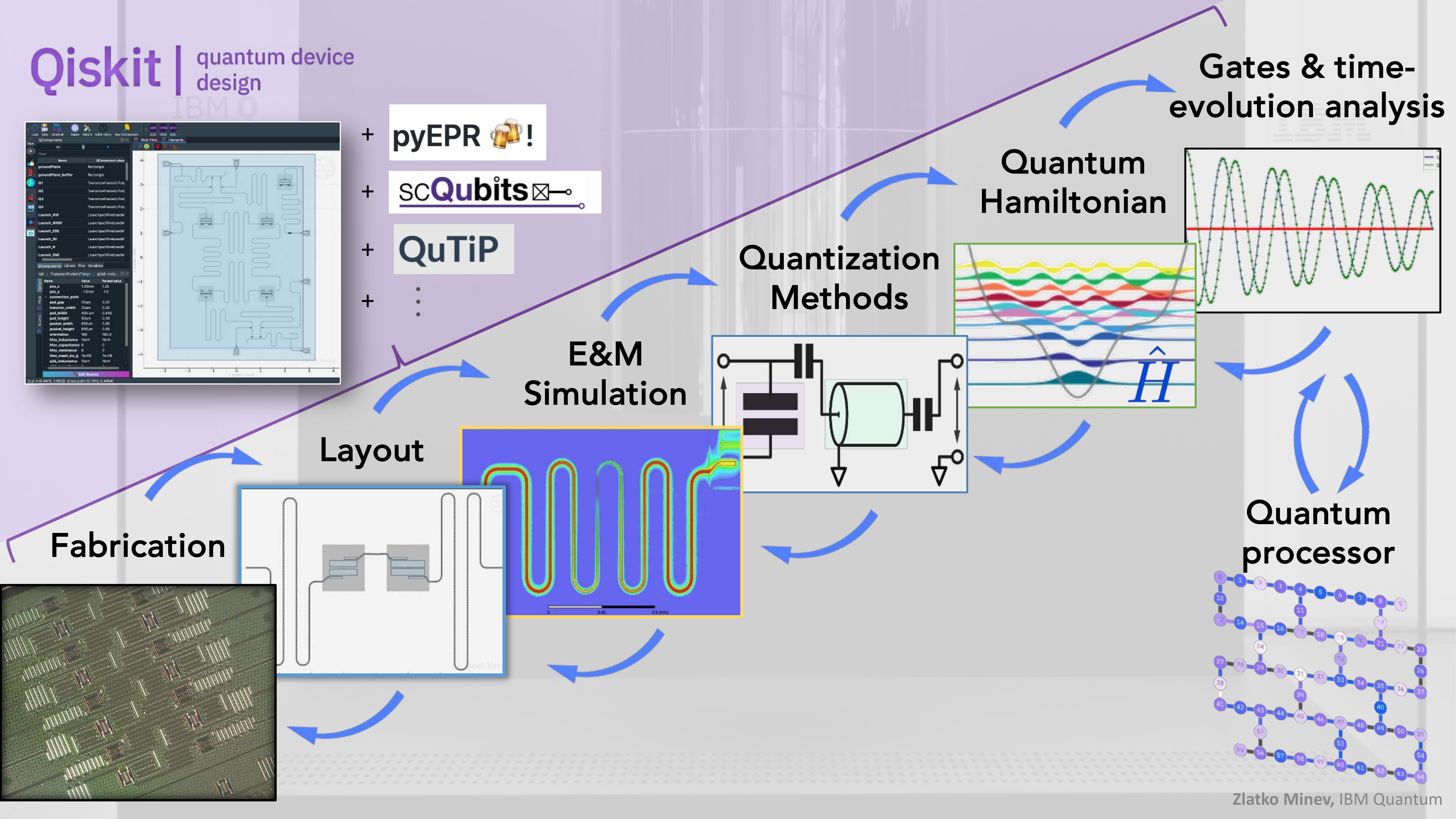

IT'S A MIRACLE


θ
ρ
ϕ
Y

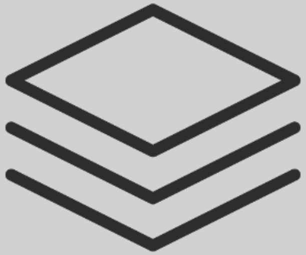
Quantum device design: layers & abstractions



Zlatko Minev, IBM Quantum


Qiskit | quantum device design
+ pyEPR !
+ scQubits
+ QuTiP
+ ⋮
Gates & time-evolution analysis
Quantum Hamiltonian
Quantization Methods
E&M Simulation
Layout
Fabrication
Quantum processor
$\hat{H}$
Zlatko Minev, IBM Quantum

A hardware designer’s wish list

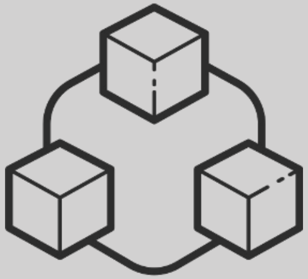


Full Stack

A streamlined framework that allows designers to go from hardware design to gates simulation

Speed

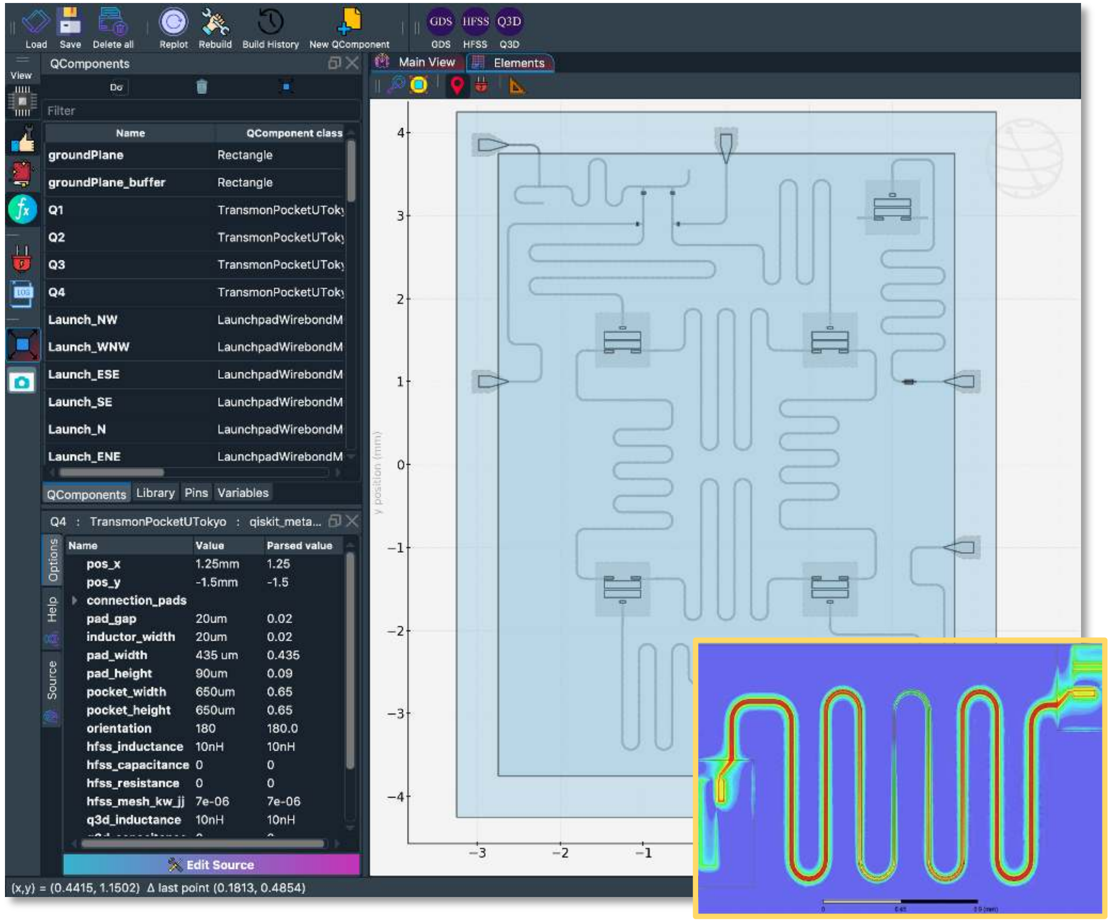
Good trade-off between speed and accuracy. Not every device requires hours of full wave simulations

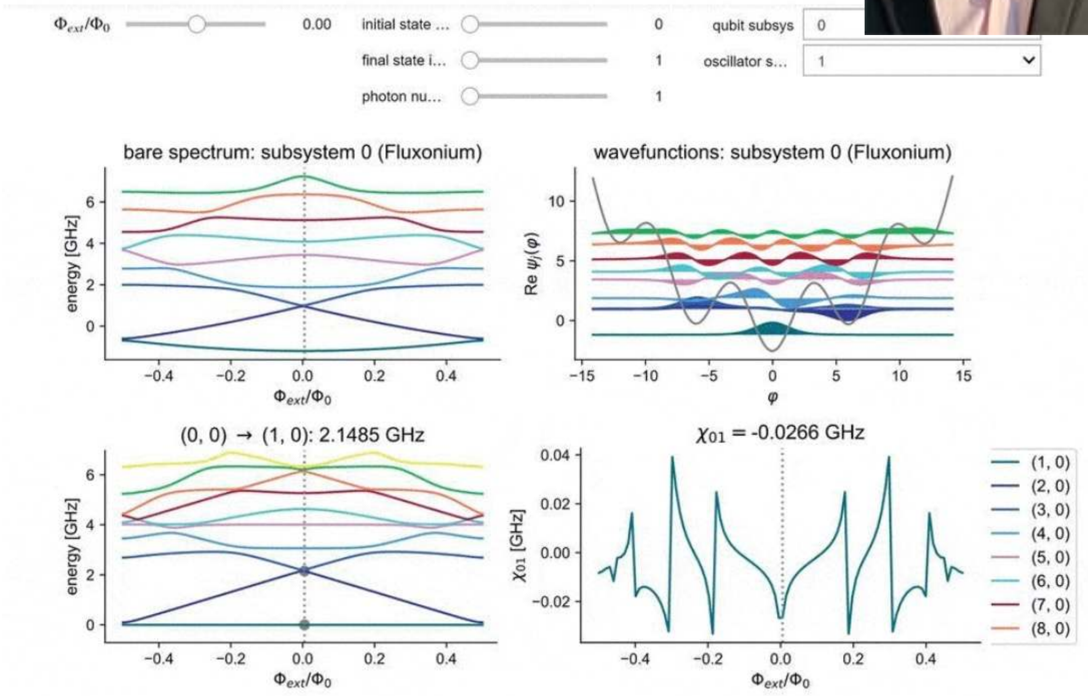


Modular

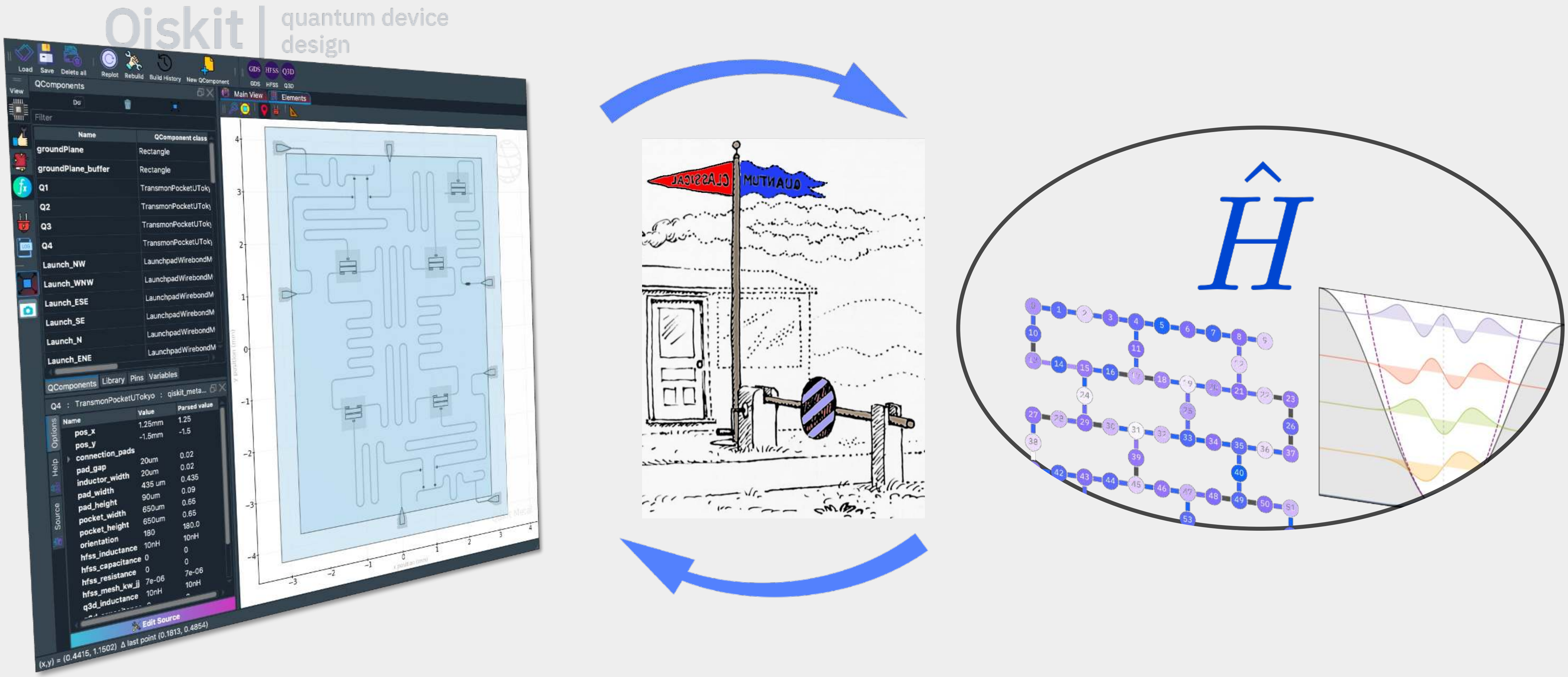
Divide and conquer: take advantage of known building blocks; modular software use case

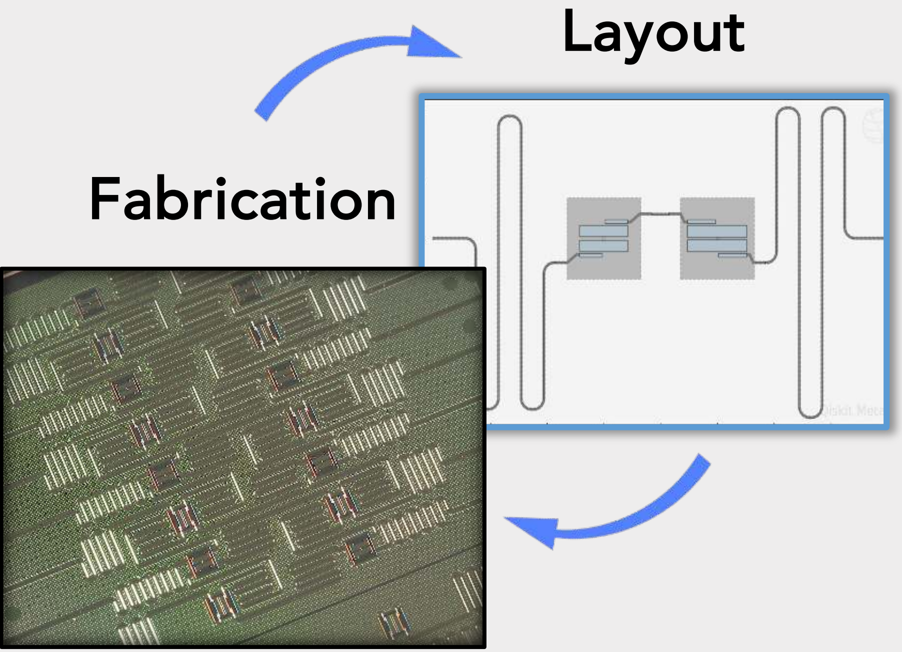
Qiskit Metal → scQubits





Example with Qiskit Metal




Layout
Fabrication

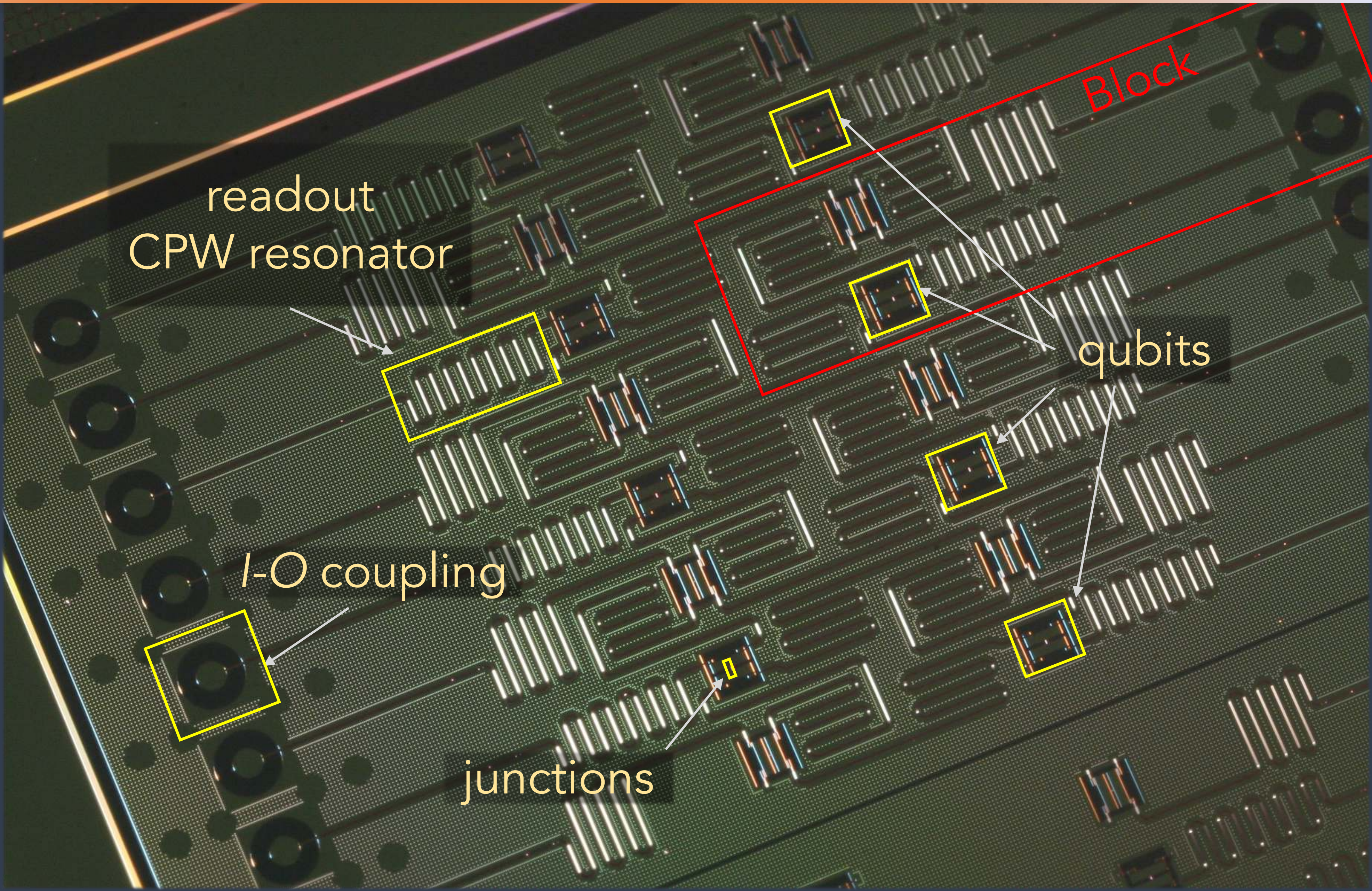
Regularity is our greatest ally in chip design

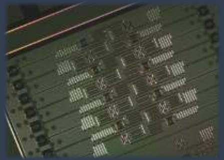
Reuse

Fine-tune

Automate

Extend

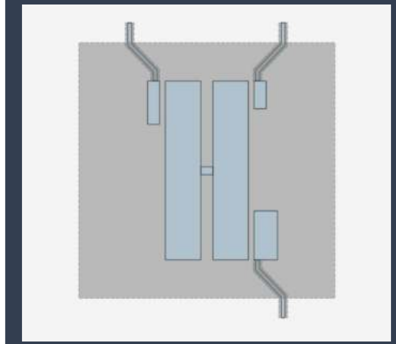




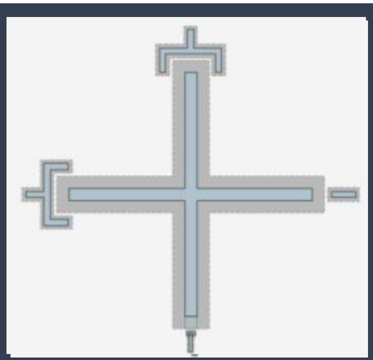
Device Library

Qiskit | quantum device design

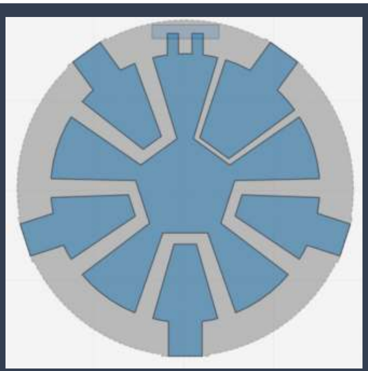
Qubits



Single transmon-floating

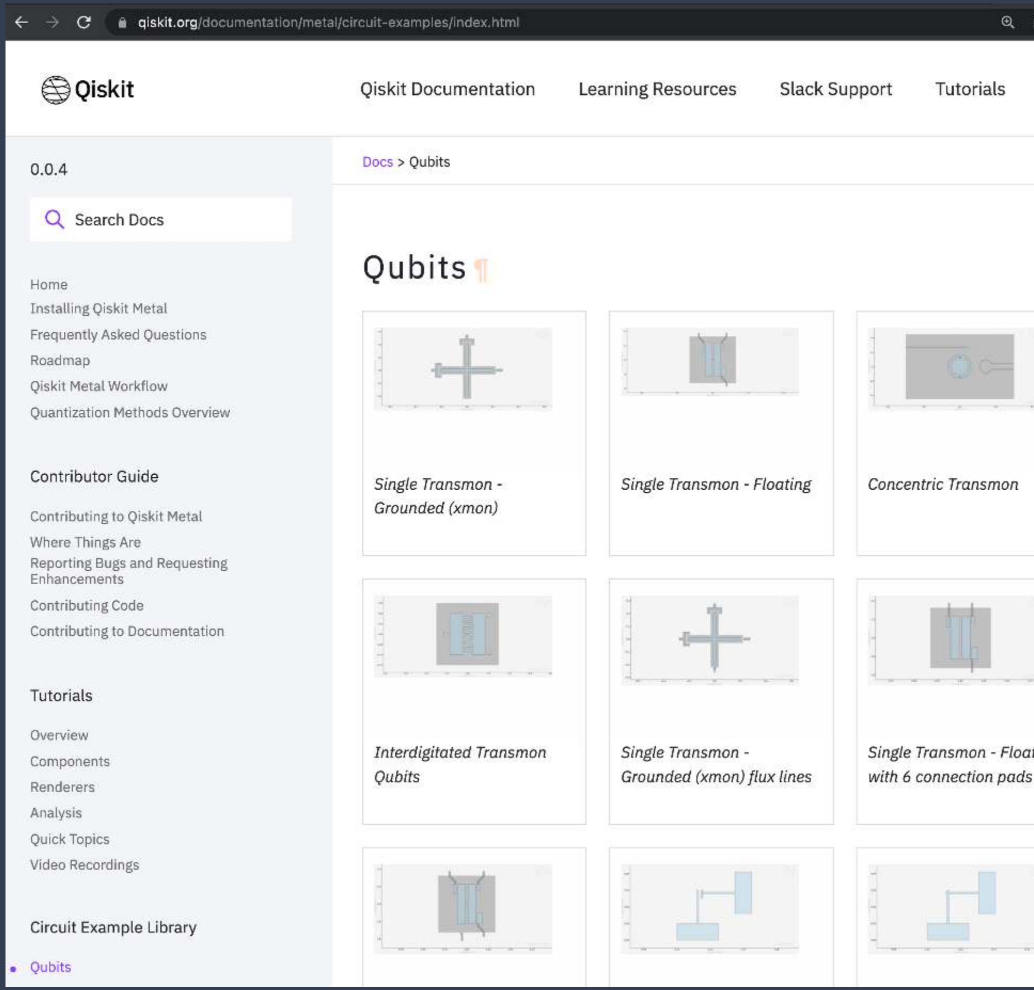


Single transmon-grounded flux lines



Star-shaped qubit

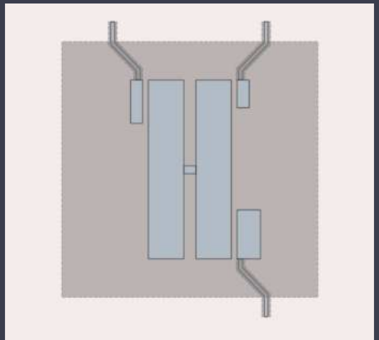
qiskit.org/documentation/metal



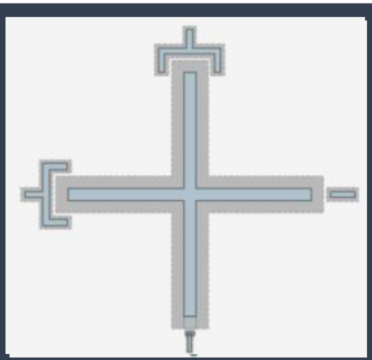
Device Library

Qiskit | quantum device design

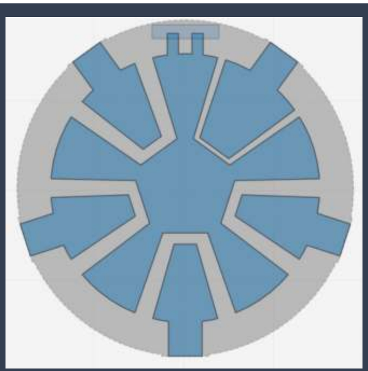
Qubits



Single transmon-
floating

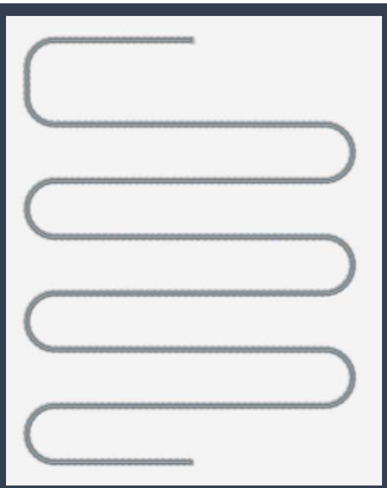


Single transmon-
grounded flux lines



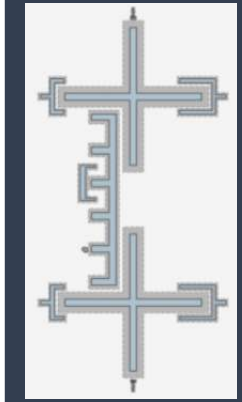
Star-shaped qubit

Resonators

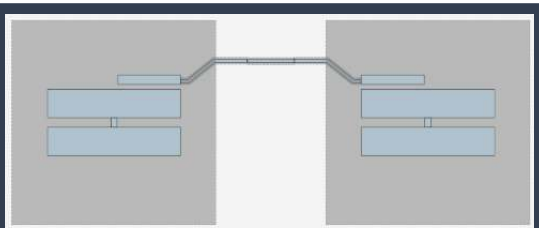


Simple coplanar
waveguide (CPW)
meander

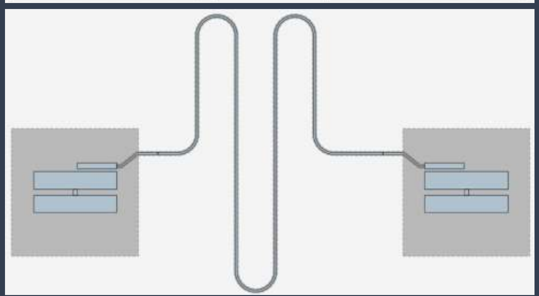
Qubit Couplers



Tunable
coupler (MIT)

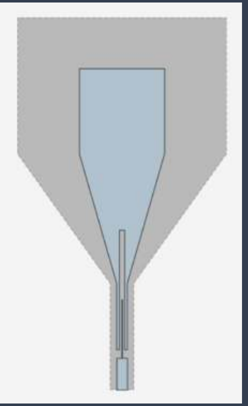


Direct
coupler

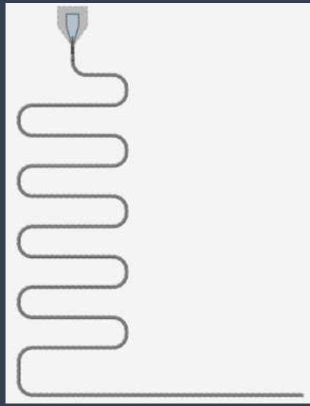


Bus resonator
coupler

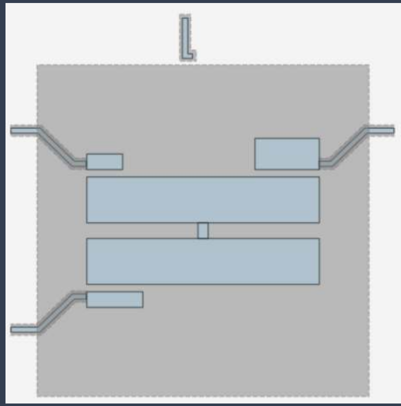
Input – Output Coupling



CPW
launch pad

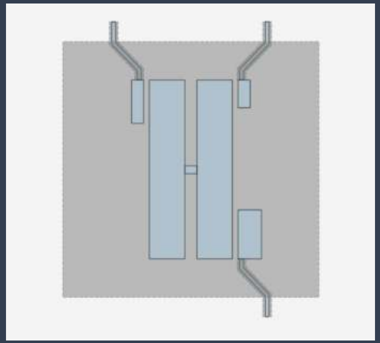


Readout line



Charge line

Device Library



Single transmon-floating

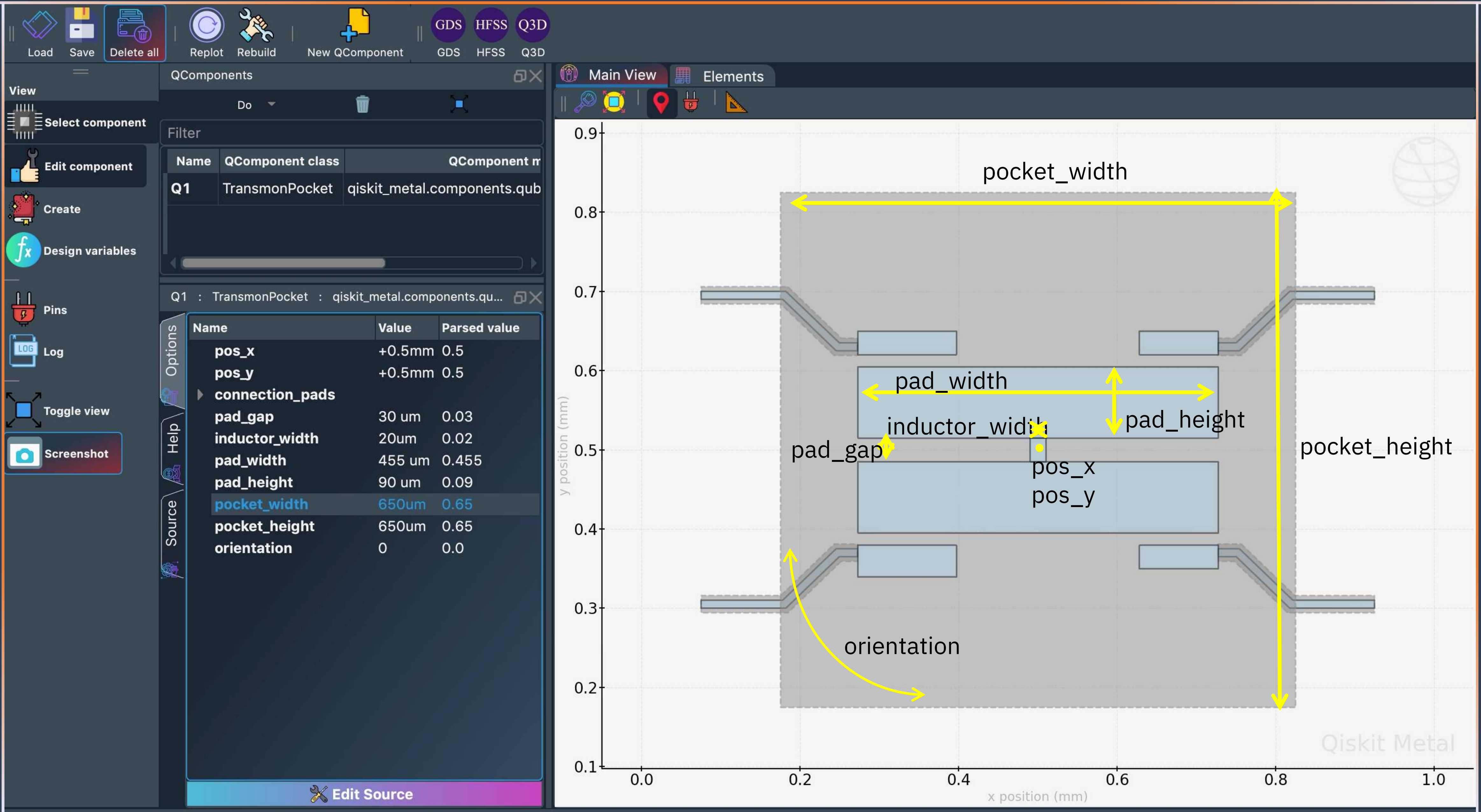
Code

```
from qiskit_metal.qlibrary import qubits
q1 = qubits.TransmonPocket(’Q1’, options=dict(...))
```

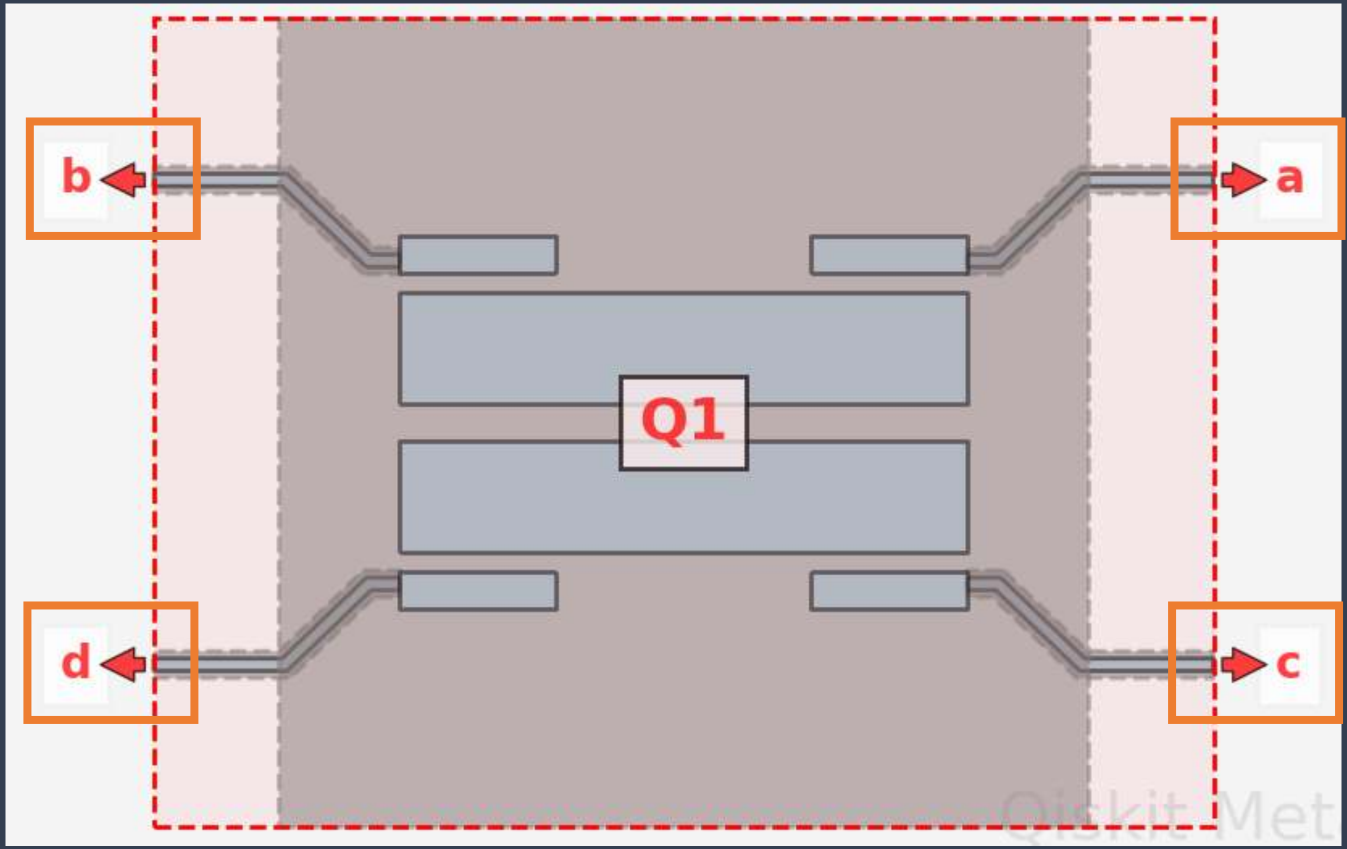
Synced GUI

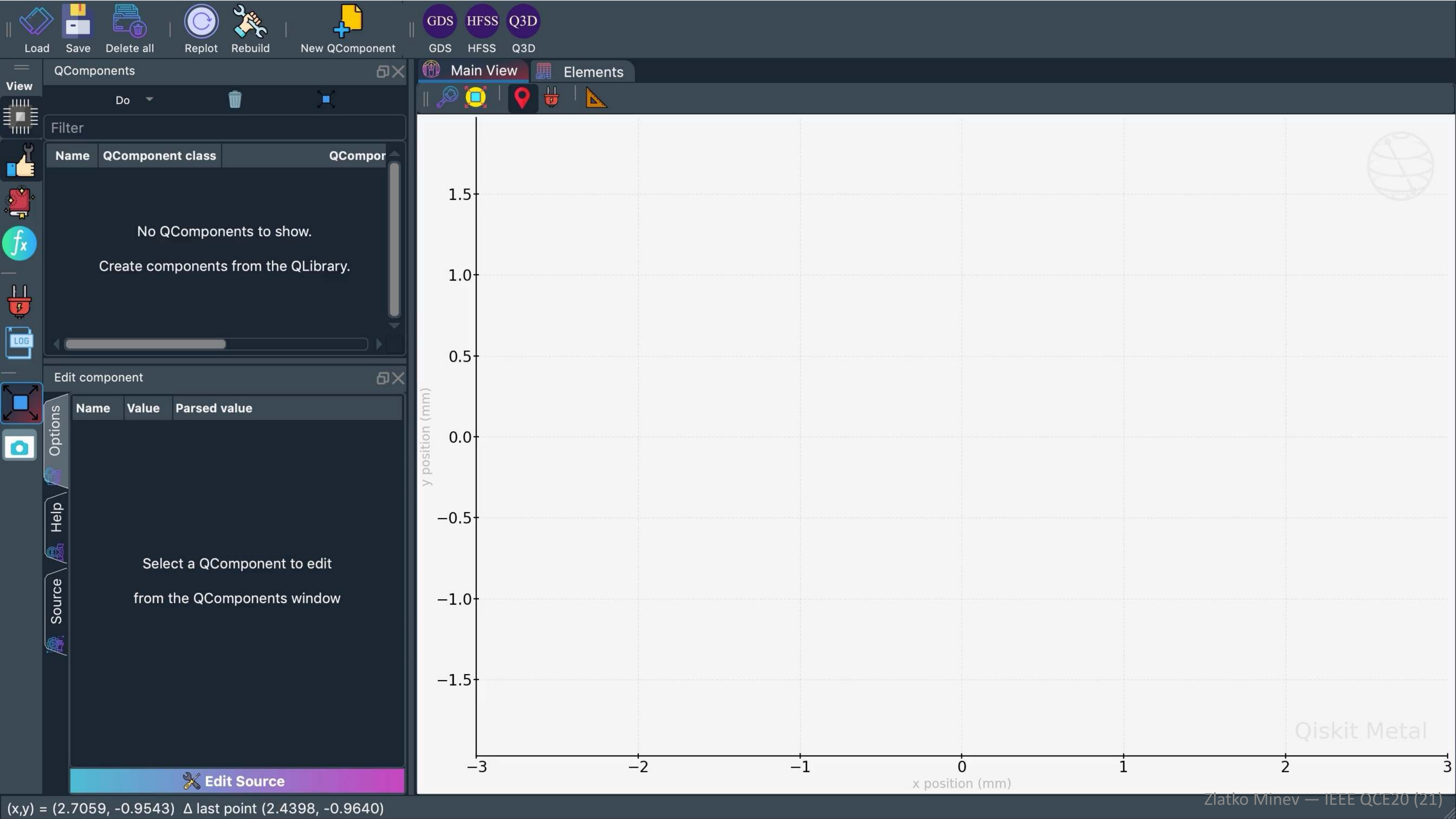
```
from qiskit_metal import MetalGUI
MetalGUI()
```

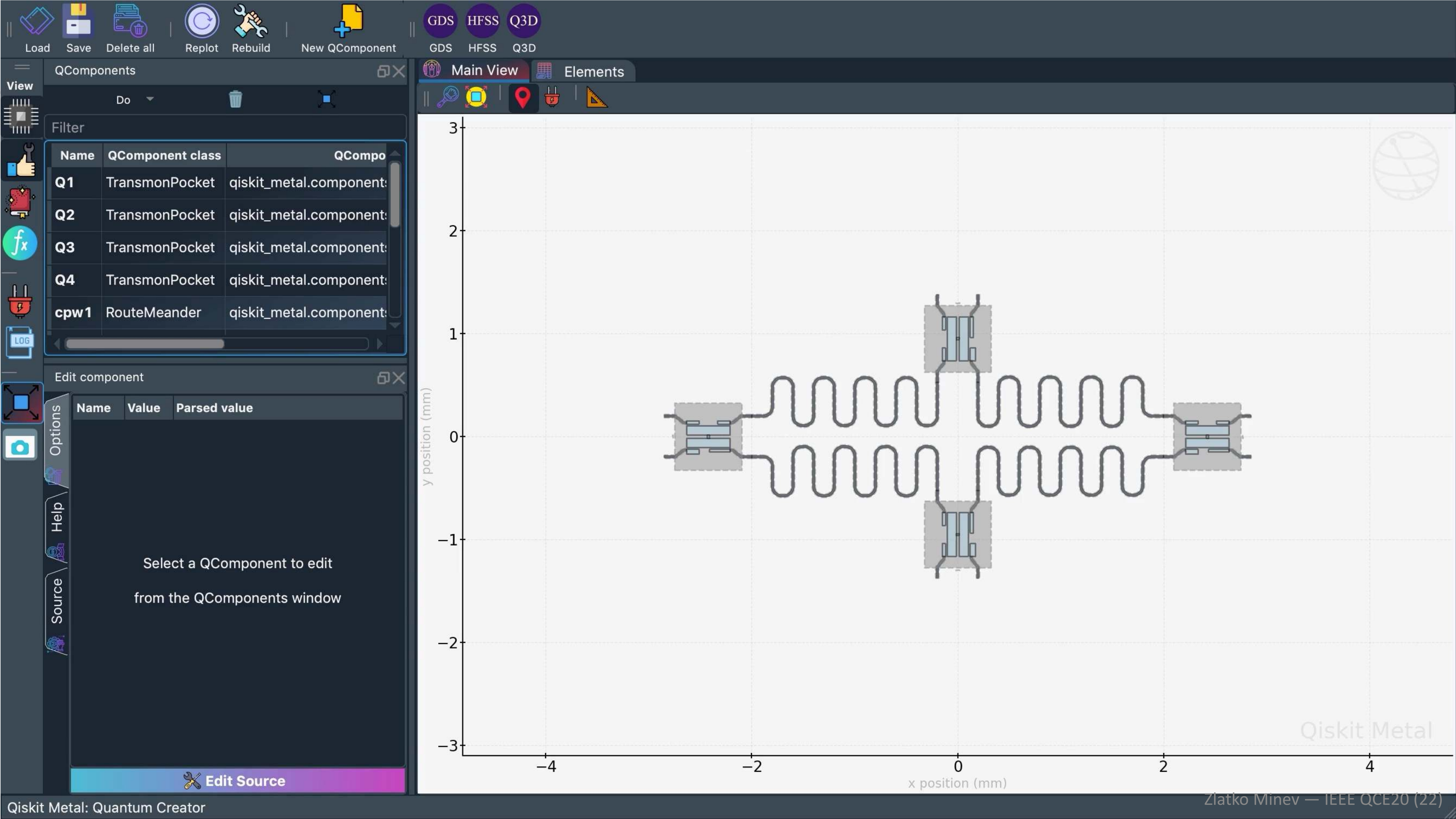
Fine-tune and automate parameters



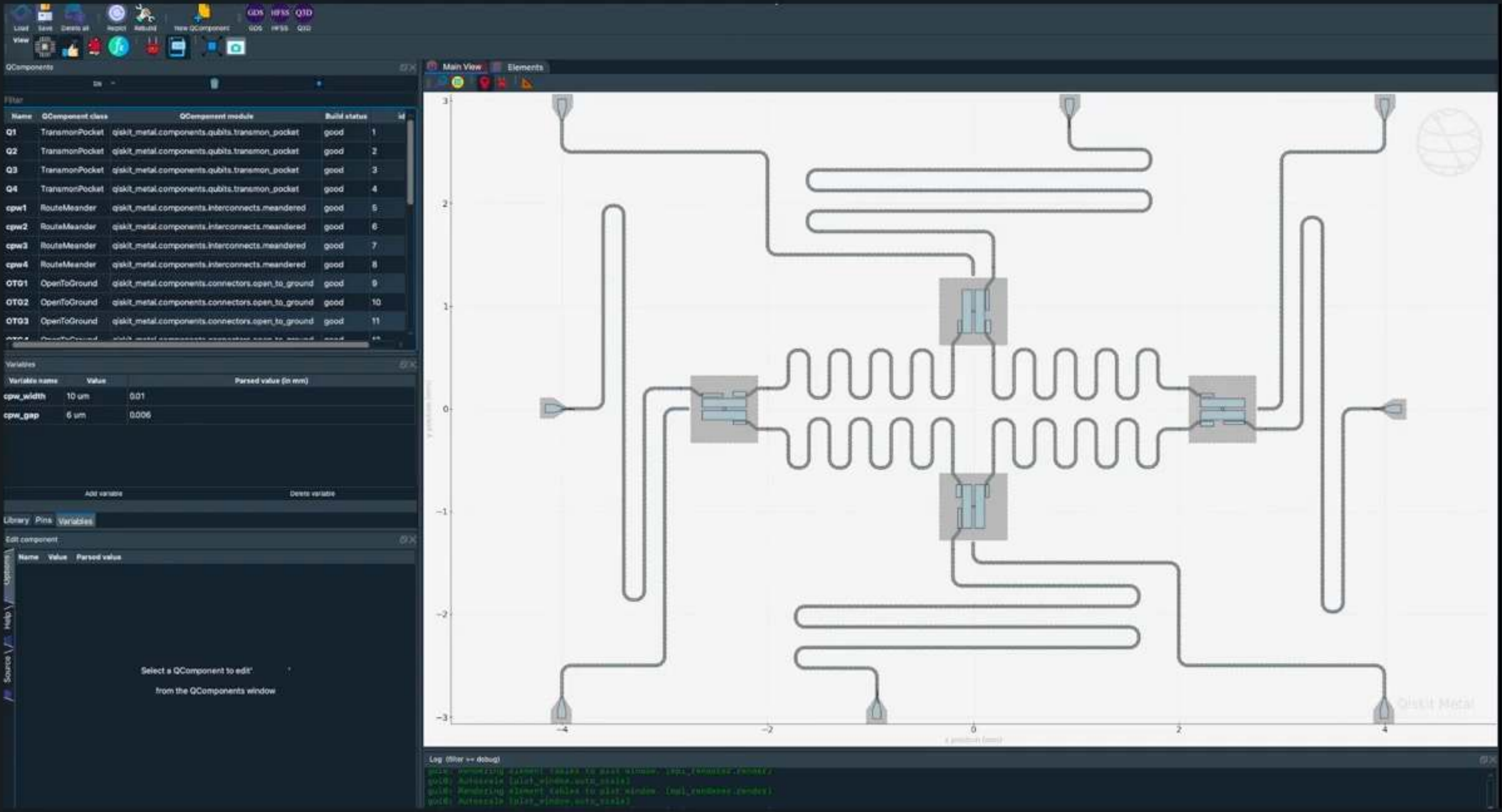
Dynamically –connected quantum devices




Load
Save
Delete all
Replot
Rebuild
New QComponent
GDS
HFSS
Q3D
View
QComponents
Do
Filter
Name
QComponent class
QCompor
No QComponents to show.
Create components from the QLibrary.
Edit component
Options
Help
Source
Name
Value
Parsed value
Select a QComponent to edit
from the QComponents window
Edit Source
Main View
Elements
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
y position (mm)
−3
−2
−1
0
1
2
3
x position (mm)
Qiskit Metal
(x,y) = (2.7059, -0.9543) Δ last point (2.4398, -0.9640)


Load
Save
Delete all
Replot
Rebuild
New QComponent
GDS
HFSS
Q3D
View
QComponents
Do
Filter
Name
QComponent class
QCompo
Q1 TransmonPocket qiskit_metal.component
Q2 TransmonPocket qiskit_metal.component
Q3 TransmonPocket qiskit_metal.component
Q4 TransmonPocket qiskit_metal.component
cpw1 RouteMeander qiskit_metal.component
Edit component
Name
Value
Parsed value
Options
Help
Source
Select a QComponent to edit
from the QComponents window
Edit Source
Main View
Elements
y position (mm)
x position (mm)
Qiskit Metal
Qiskit Metal: Quantum Creator

Render your design as a GDS file

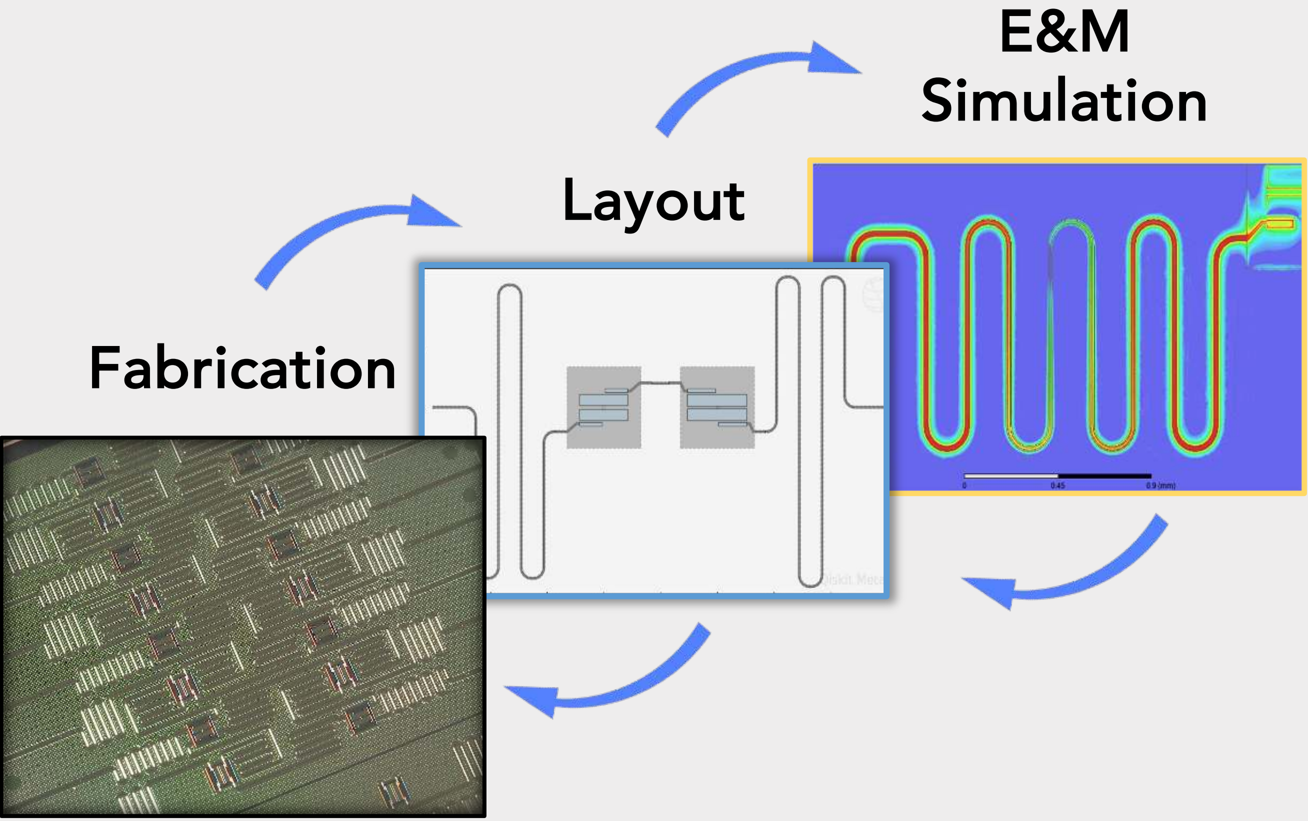


Load
Save
Delete all
Replot
Rebuild
New QComponent
GDS
HFSS
Q3D
View
QComponents
Filter
Name
QComponent class
QComponent module
Build status
id
Q1 TransmonPocket qiskit_metal.components.qubits.transmon_pocket good 1
Q2 TransmonPocket qiskit_metal.components.qubits.transmon_pocket good 2
Q3 TransmonPocket qiskit_metal.components.qubits.transmon_pocket good 3
Q4 TransmonPocket qiskit_metal.components.qubits.transmon_pocket good 4
cpw1 RouteMeander qiskit_metal.components.interconnects.meandered good 5
cpw2 RouteMeander qiskit_metal.components.interconnects.meandered good 6
cpw3 RouteMeander qiskit_metal.components.interconnects.meandered good 7
cpw4 RouteMeander qiskit_metal.components.interconnects.meandered good 8
OTG1 OpenToGround qiskit_metal.components.connectors.open_to_ground good 9
OTG2 OpenToGround qiskit_metal.components.connectors.open_to_ground good 10
OTG3 OpenToGround qiskit_metal.components.connectors.open_to_ground good 11
Variables
Variable name
Value
Parsed value (in mm)
cpw_width 10 um 0.01
cpw_gap 6 um 0.006
Add variable
Delete variable
Library
Pins
Variables
Edit component
Name
Value
Parsed value
Options
Help
Source
Select a QComponent to edit
from the QComponents window
Main View
Elements
Log (filter >= debug)

Can I make my own qubit? ☺

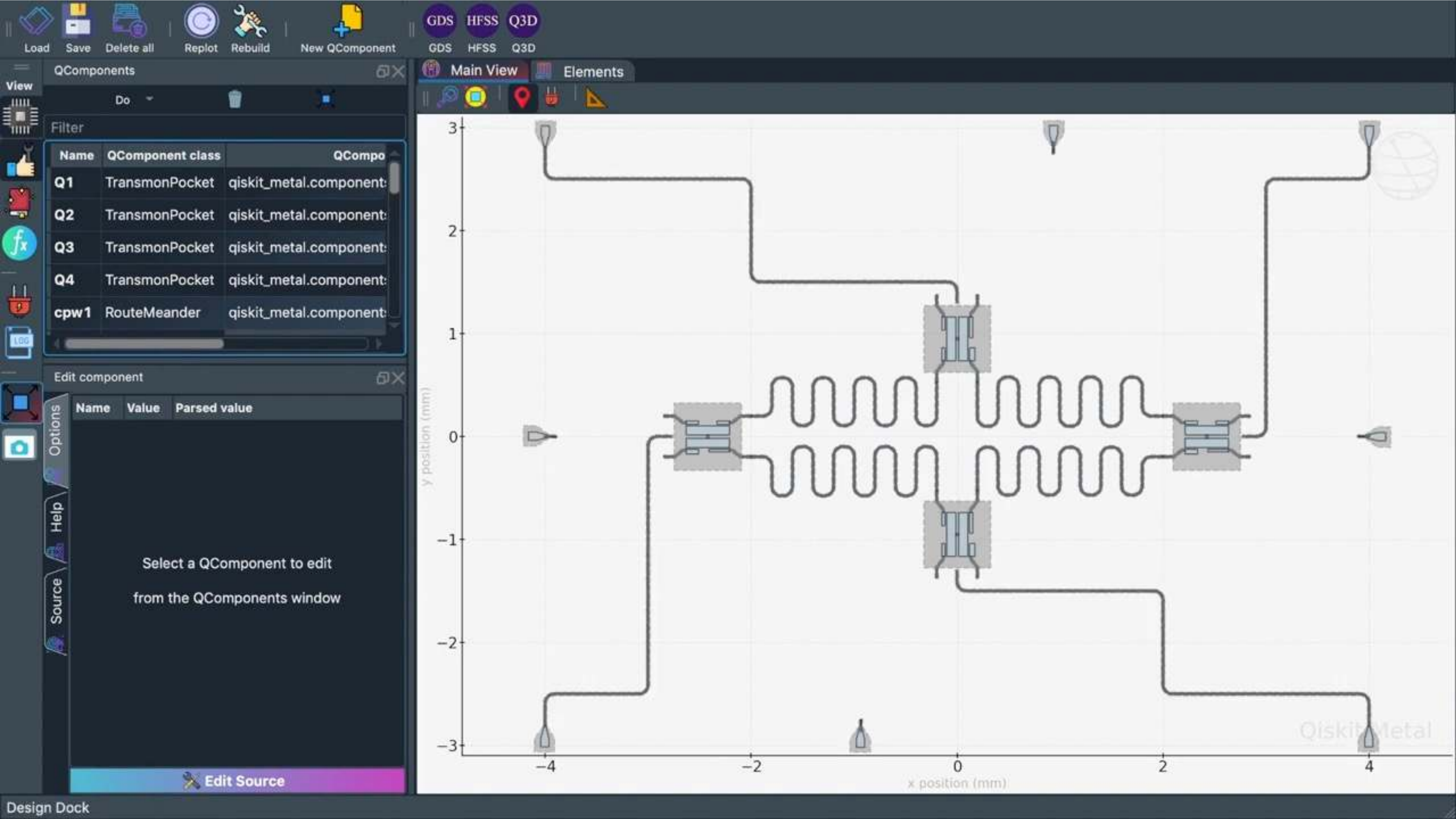
```
smile = draw.shapely.geometry.Point(0, 0).buffer(0.8)
cut_sq = draw.shapely.geometry.box(-1, -0.3, 1, 1)
smile = draw.subtract(smile, cut_sq)
face = draw.subtract(face, smile)
face = draw.subtract(face, eye_r)
face = draw.subtract(face, eye_l)
face
```

[7]:

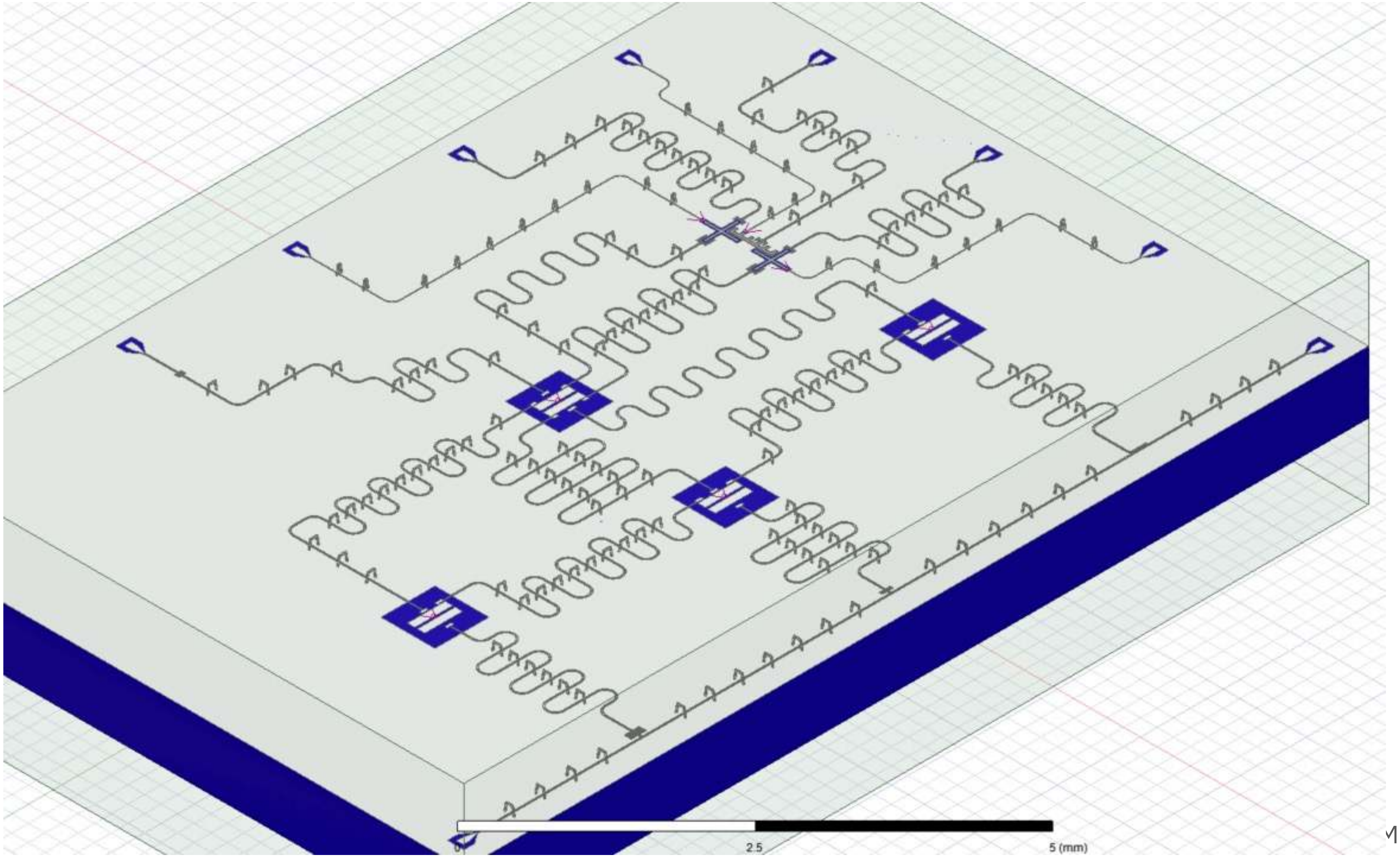

E&M
Simulation
Layout
Fabrication

Render your design in an EM Solver

Example: Ansys HFSS

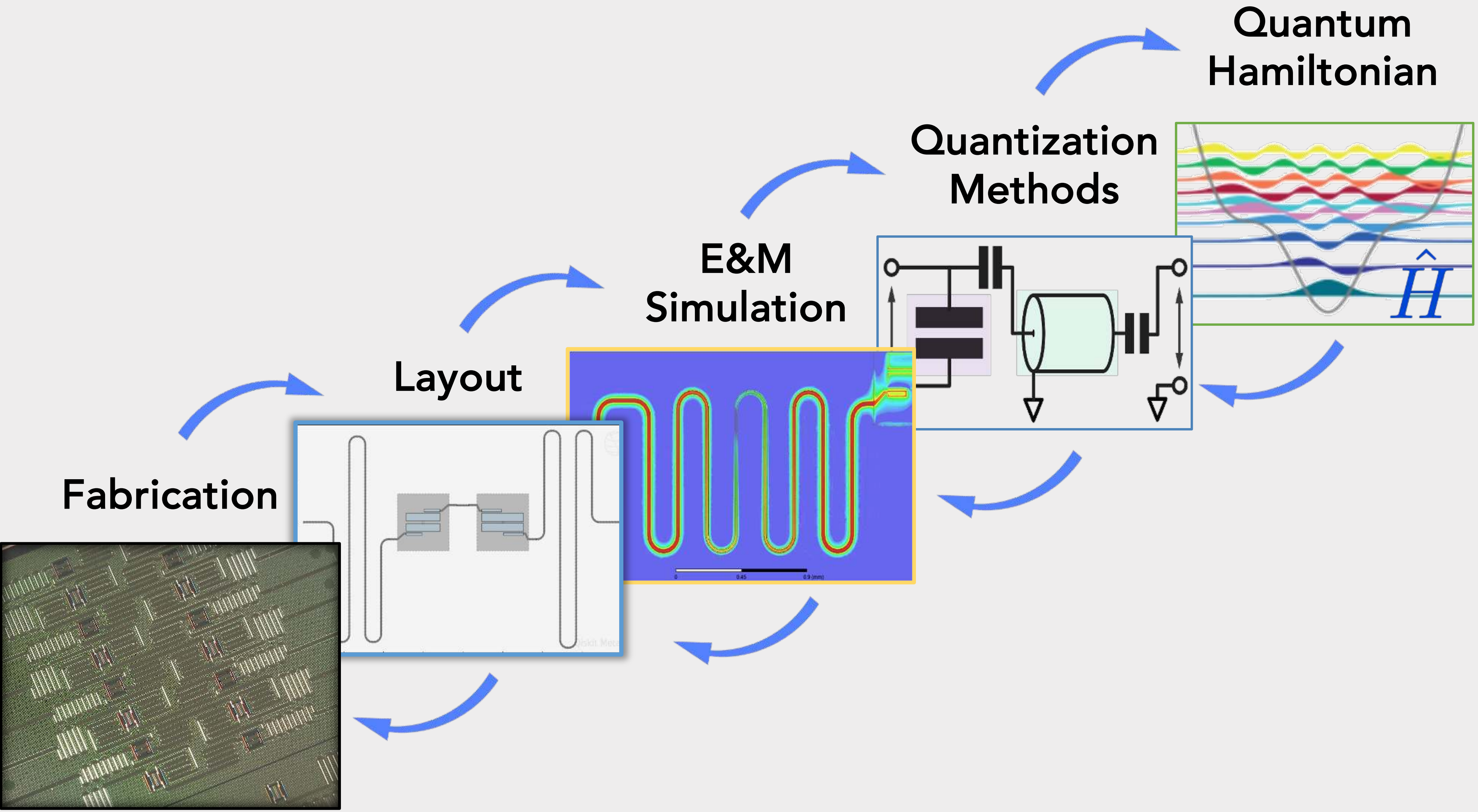

Load
Save
Delete all
Replot
Rebuild
New QComponent
GDS
HFSS
Q3D
View
QComponents
Do
Filter
Name
QComponent class
QCompo
Q1
TransmonPocket
qiskit_metal.component
Q2
TransmonPocket
qiskit_metal.component
Q3
TransmonPocket
qiskit_metal.component
Q4
TransmonPocket
qiskit_metal.component
cpw1
RouteMeander
qiskit_metal.component
Edit component
Name
Value
Parsed value
Options
Help
Source
Select a QComponent to edit
from the QComponents window
Edit Source
Main View
Elements
y position (mm)
x position (mm)
Qiskit Metal
Design Dock

More complex quantum chips – see next tutorial



M Quantum (29

Qiskit | quantum device design



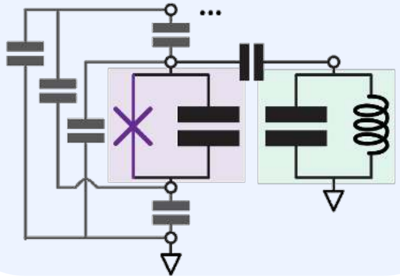
Quantization methods landscape

IBM **Quantum**

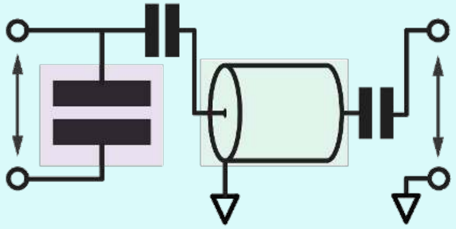
quasi-static

full-wave

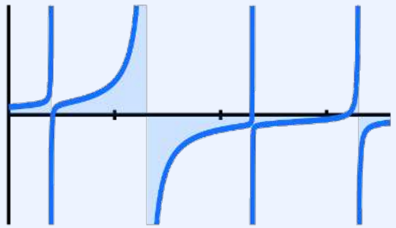
lumped



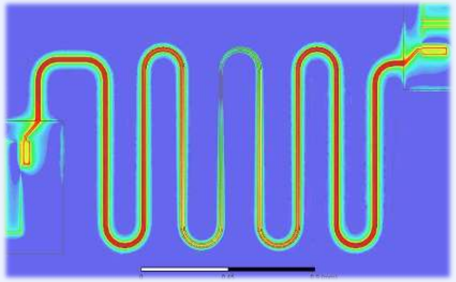
quasi-lumped



impedance



energy



Yurke & Denker (1984), Devoret (1997), Burkard *et al.* (2004), Koch *et al…*

Malekakhlagh *et al.* (2017, 2019), Gely *et al.* (2019), Parra-Rodriguez *et al.* (2019), Minev *et al.* (2021), ...

Nigg et al. (2012), Bourassa et al. (2012), Solgun et al. (2014, 2015, 2017) …

Minev (2018)
Minev *et al.* (2020)

fast

more information, complexity, accuracy

Tutorials: Quantum analysis library

https://qiskit.org/documentation/metal/tut/index.html

Qiskit

https://qiskit.org/documentation/metal/tut/index.html

Tutorials: Quantum analysis library

Resonators
Composite Bi-Partite Systems
Qubit Couplers
Input-Ouput Coupling
Small Quantum Chips
Design Flow

Libraries

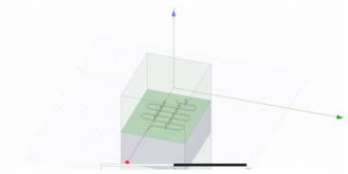
All Quantum Devices

API References

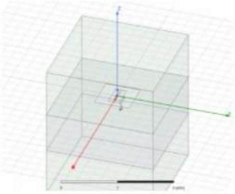
Overview
QDesigns
QComponents
Analyses
QRenderes
Toolbox
QGeometry
GUI

Code of Conduct

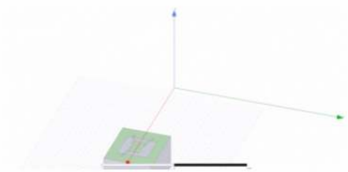
Qiskit



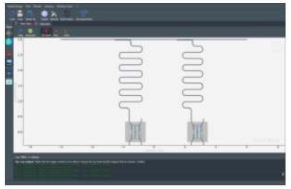
Eigenmode and EPR



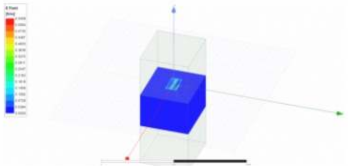
Impedance Analysis



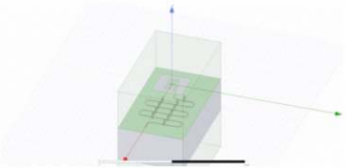
Capacitance Matrix and LOM Analysis



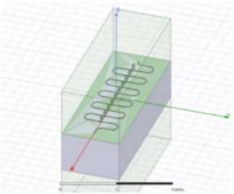
Sweep a option in QComponent, get Ansys HFSS driven-modal scatter matrix, impedance matrix, and admittance matrix.



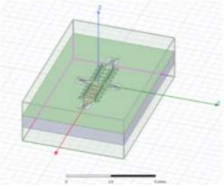
Analyizing and Tuning a Transmon Qubit



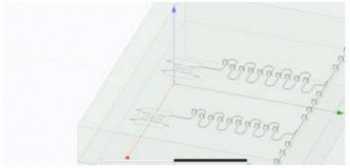
Analyzing a transmon and a resonator



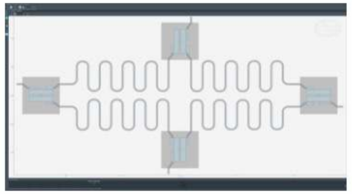
Analyzing a resonator



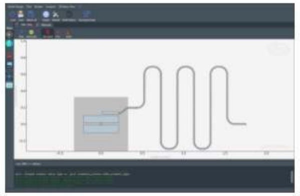
How to Render a Metal Design into Ansys



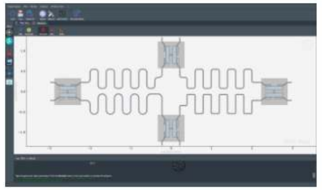
Driven Modal Simulation and S-Parameters



Sweep an option in QComponent, get capacitance matrix from Ansys Q3D and pyEPR



Sweep an option in QComponent, get Ansys HFSS eigenmode solution matrix.



How to Render a Metal Design into GDS

https://qiskit.org/documentation/metal/tut/index.html

Resonators
Composite Bi-Partite Systems
Qubit Couplers
Input-Ouput Coupling
Small Quantum Chips
Design Flow

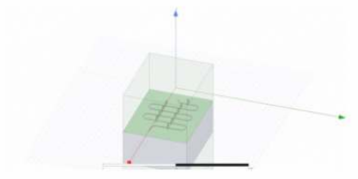
Libraries

All Quantum Devices

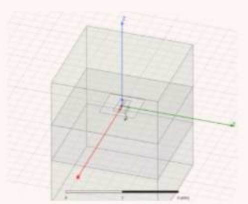
API References

Overview
QDesigns
QComponents

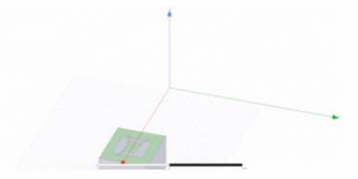
Tutorials: Quantum analysis library



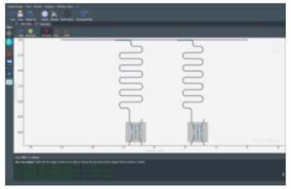
Eigenmode and EPR



Impedance Analysis



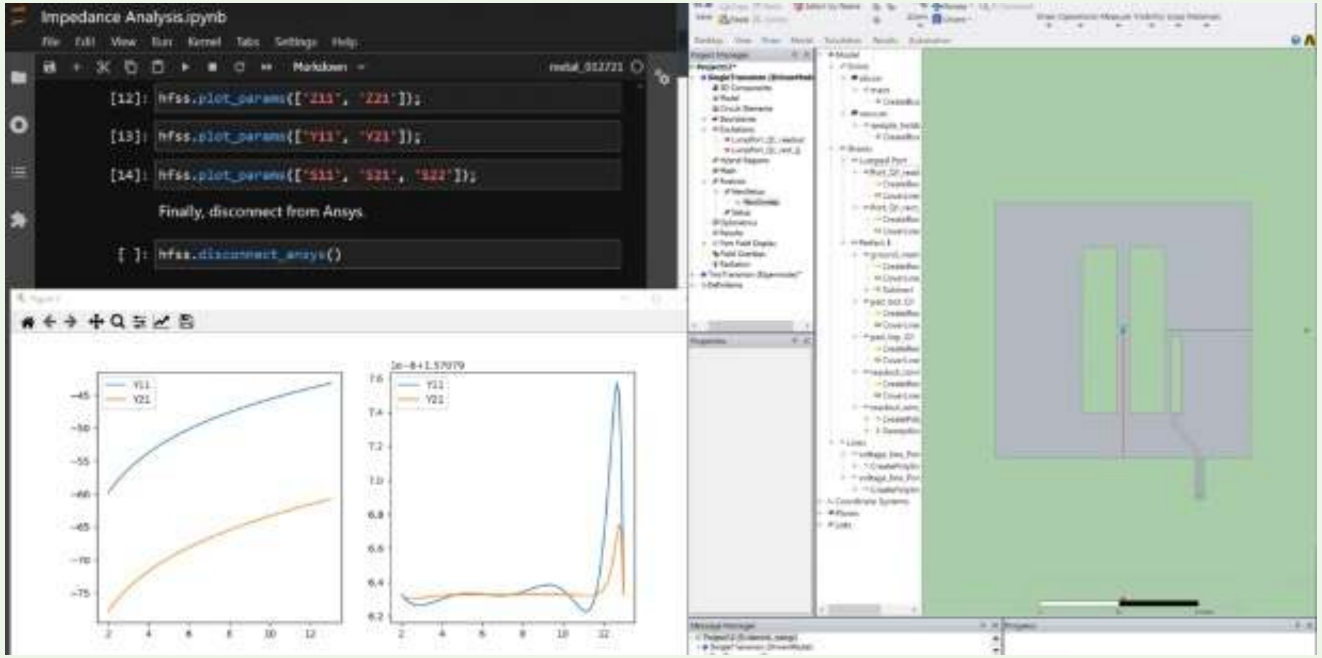
Capacitance Matrix and LOM Analysis



Sweep a option in QComponent, get Ansys HFSS driven-modal scatter matrix, impedance matrix, and admittance matrix.

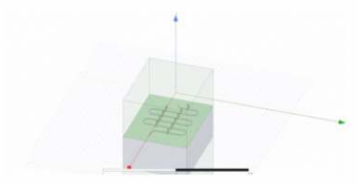
S, Z, Y
Impedance
Scattering

arXiv:1204.0587 …
quanutm analysis part WIP

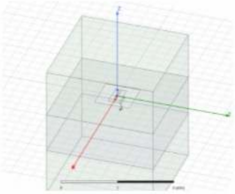


Tutorials: Quantum analysis library

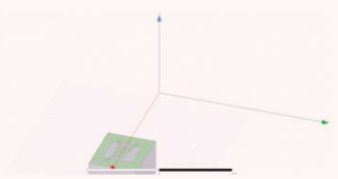
Qisk



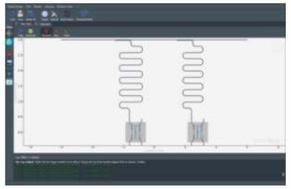
Eigenmode and EPR



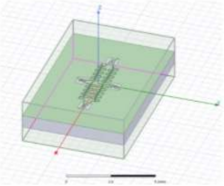
Impedance Analysis



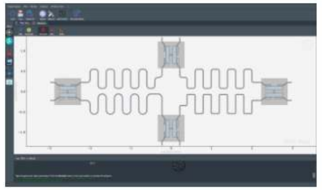
Capacitance Matrix and LOM Analysis



Sweep a option in QComponent, get Ansys HFSS driven-modal scatter matrix, impedance matrix, and admittance matrix.



How to Render a Metal Design into Ansys

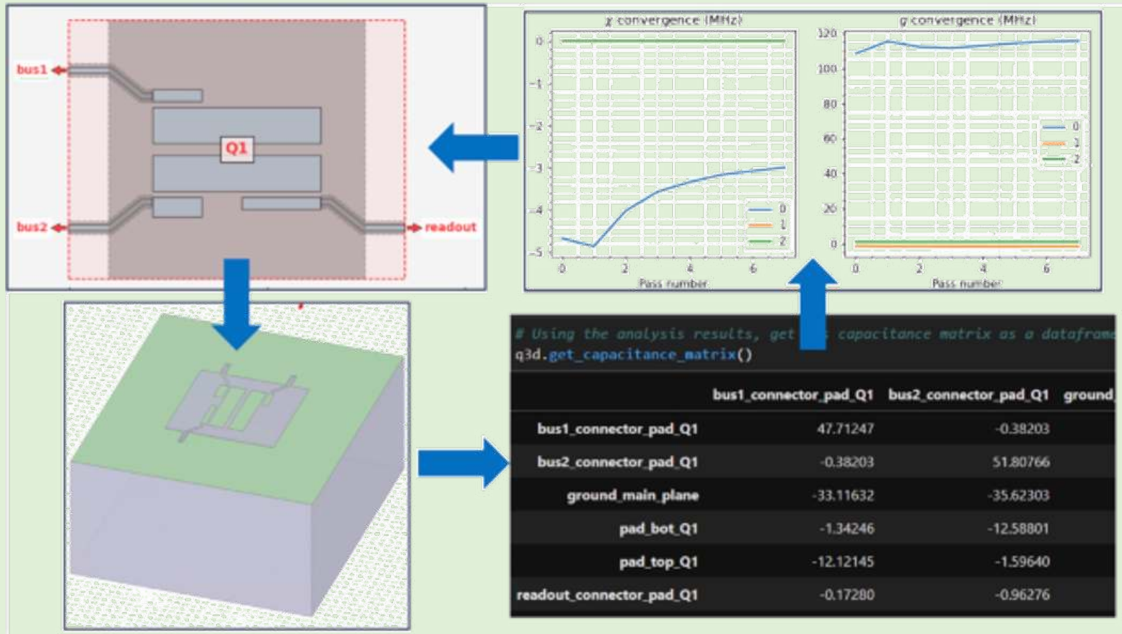


How to Render a Metal Design into GDS

Capacitive

Jay's code

arXiv:2103.10344 ...




```
# Using the analysis results, get   capacitance matrix as a datafram
q3d.get_capacitance_matrix()
```

	bus1_connector_pad_Q1	bus2_connector_pad_Q1	ground
bus1_connector_pad_Q1	47.71247	-0.38203	
bus2_connector_pad_Q1	-0.38203	51.80766	
ground_main_plane	-33.11632	-35.62303	
pad_bot_Q1	-1.34246	-12.58801	
pad_top_Q1	-12.12145	-1.59640	
readout_connector_pad_Q1	-0.17280	-0.96276	

https://qiskit.org/documentation/metal/tut/index.html

Tutorials: Quantum analysis library

- Resonators
- Composite Bi-Partite Systems
- Qubit Couplers
- Input-Ouput Coupling
- Small Quantum Chips
- Design Flow

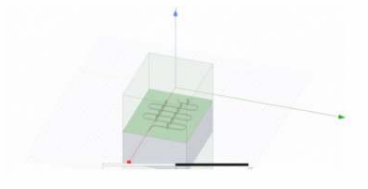
Libraries

- All Quantum Devices

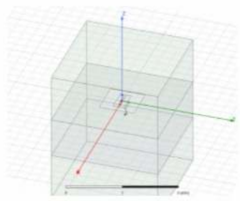
API References

- Overview
- QDesigns
- QComponents
- Analyses
- QRenderes
- Toolbox
- QGeometry
- GUI

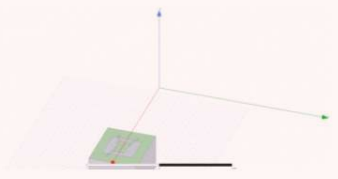
Code of Conduc



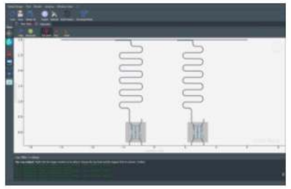
Eigenmode and EPR



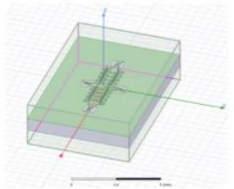
Impedance Analysis



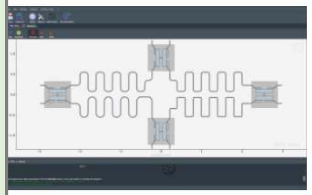
Capacitance Matrix and LOM Analysis



Sweep a option in QComponent, get Ansys HFSS driven-modal scatter matrix, impedance matrix, and admittance matrix.



ow to Render a Metal Design to Ansys



ow to Render a Metal Design to GDS

Cornell University

arXiv.org > quant-ph > arXiv:2103.10344

Quantum Physics

Circuit quantum electrodynamics (cQED) with modular quasi-lumped models

Zlatko K. Minev,* Thomas G. McConkey, Maika Takita, Antonio Corcoles, and Jay M. Gambetta
IBM Quantum, IBM T.J. Watson Research Center, Yorktown Heights, US

WIP: General capacitive analysis code map to known building blocks: e.g., transmon, fluxonium, zero-pi, etc.

https://qiskit.org/documentation/metal/tut/index.html

Resonators
Composite Bi-Partite Systems
Qubit Couplers
Input-Ouput Coupling
Small Quantum Chips
Design Flow

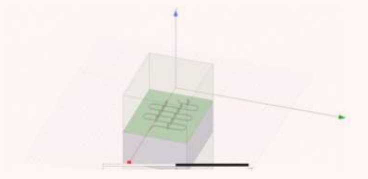
Libraries

All Quantum Devices

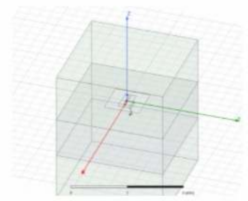
API References

Overview

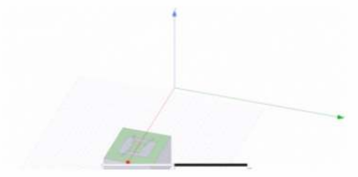
Tutorials: Quantum analysis library



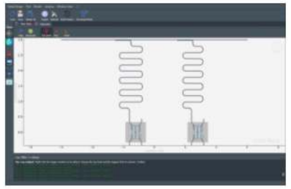
Eigenmode and EPR



Impedance Analysis



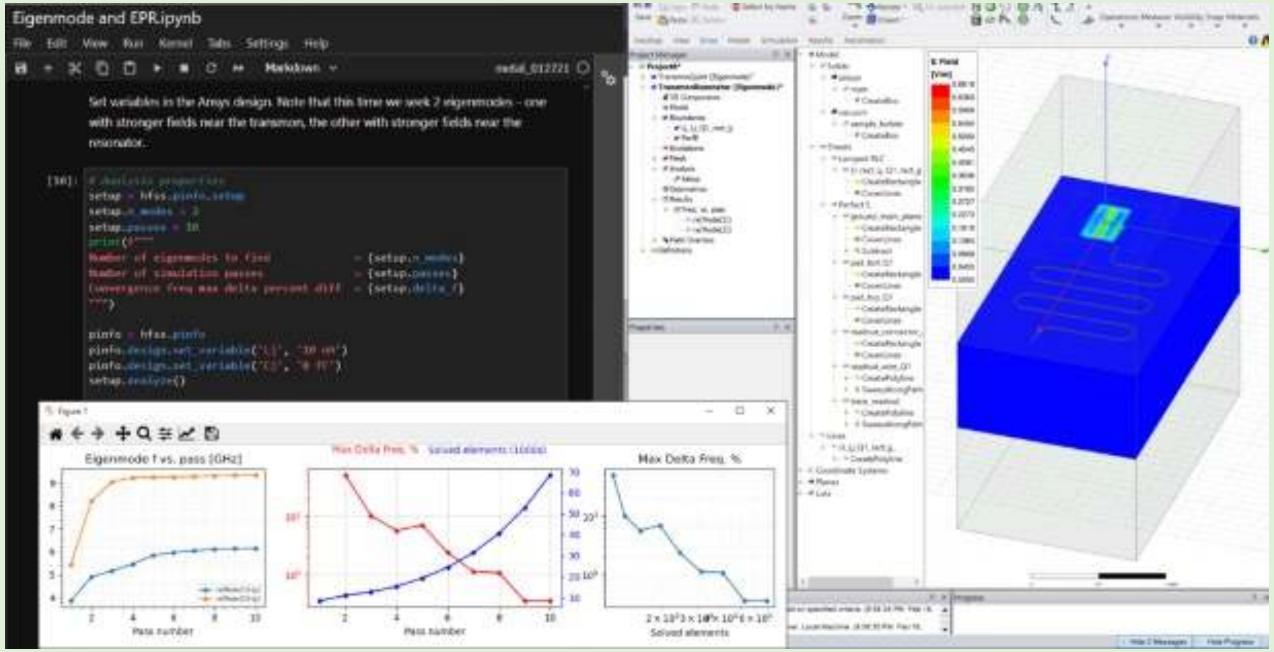
Capacitance Matrix and LOM Analysis



Sweep a option in QComponent, get Ansys HFSS driven-modal scatter matrix, impedance matrix, and admittance matrix.

Eigenmode

arXiv: 2010.00620
arXiv: 1902.10355



https://qiskit.org/documentation/metal/tut/index.html

Tutorials: Quantum analysis library

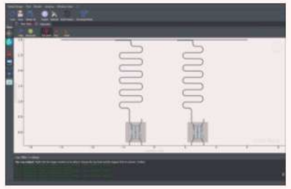
Resonators
Composite Bi-Partite Systems
Qubit Couplers
Inp
S
G

Code of Conduct

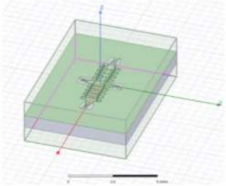
Qiskit

Sweeps

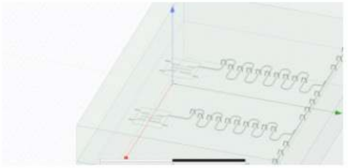
with
Z S Y impedance, scattering params
eigenmode / EPR
capacititive / lumped model



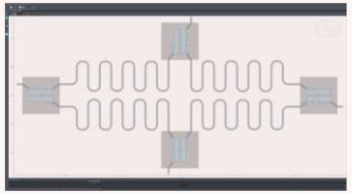
Sweep a option in QComponent, get Ansys HFSS driven-modal scatter matrix, impedance matrix, and admittance matrix.



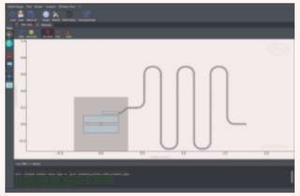
How to Render a Metal Design into Ansys



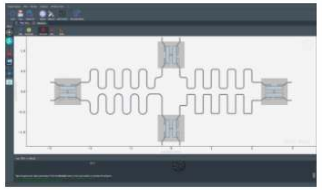
Driven Modal Simulation and S-Parameters



Sweep an option in QComponent, get capacitance matrix from Ansys Q3D and pyEPR



Sweep an option in QComponent, get Ansys HFSS eigenmode solution matrix.



How to Render a Metal Design into GDS

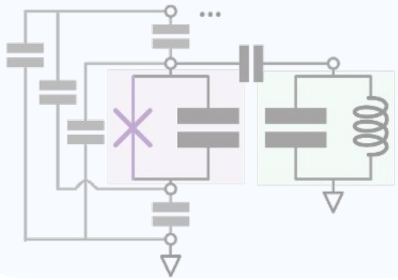
Quantization methods landscape

IBM **Quantum**

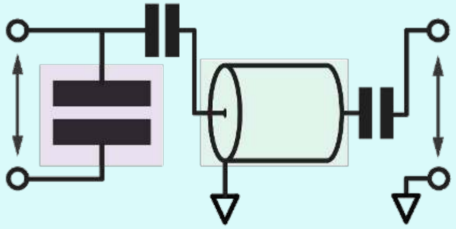
quasi-static

full-wave

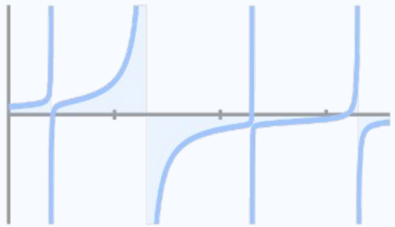
lumped



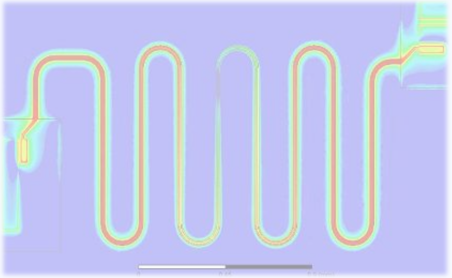
quasi-lumped



impedance



energy



Yurke & Denker (1984), Devoret (1997), Burkard *et al.* (2004), Koch *et al*…

Malekakhlagh *et al.* (2017, 2019), Gely *et al.* (2019), Parra-Rodriguez *et al.* (2019), Minev *et al.* (2021), …

Nigg et al. (2012), Bourassa et al. (2012), Solgun et al. (2014, 2015, 2017) …

Minev (2018)
Minev *et al.* (2020)

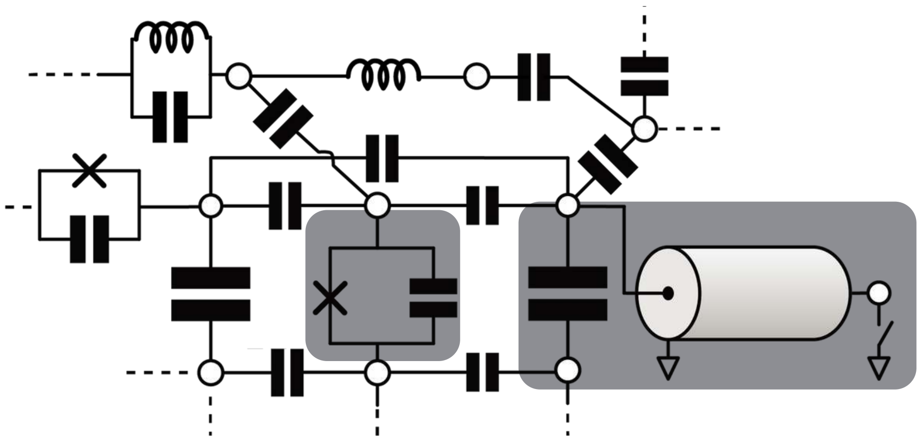
fast

more information, complexity, accuracy

Quasi-lumped circuit model:

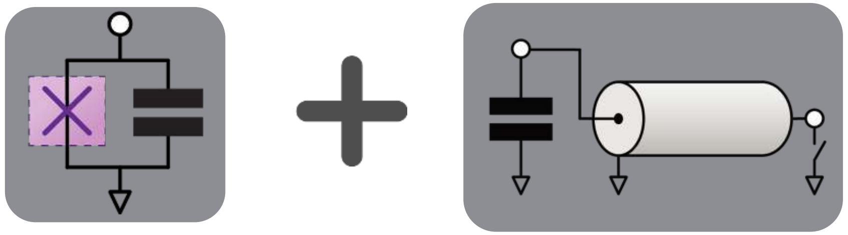
Composite system

Partition device
to subsystem
building blocks



Qubit subsystem

transmon
fluxonium
3-junction flux
qubit
0-π qubit
...

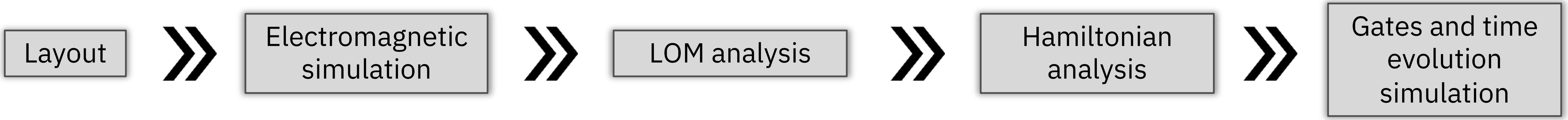


Resonator subsystem

Single-side loaded
transmission line
resonator

Minev et al. arXiv:2103.10344

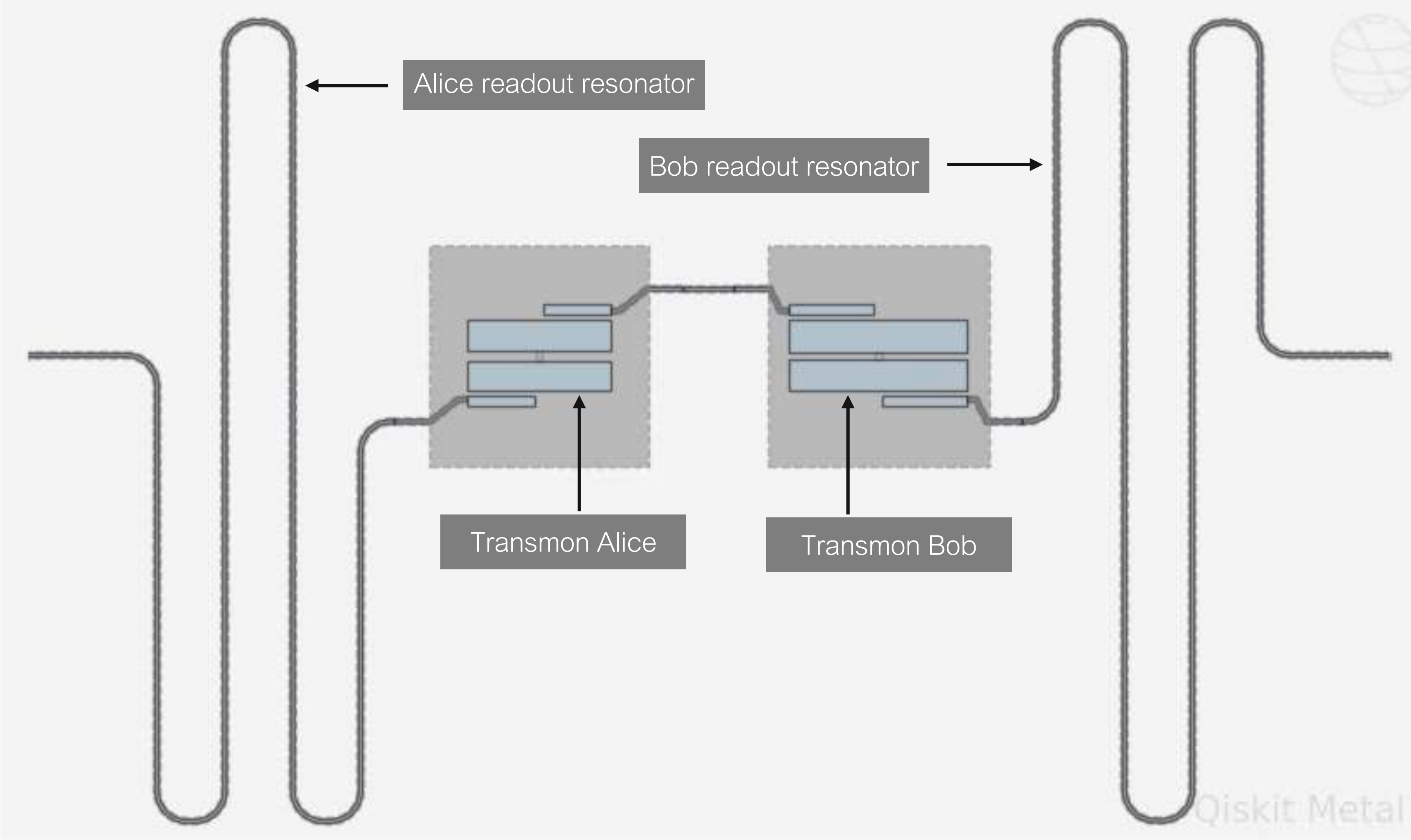
Qiskit Metal lumped-oscillator model framework



Physical layout

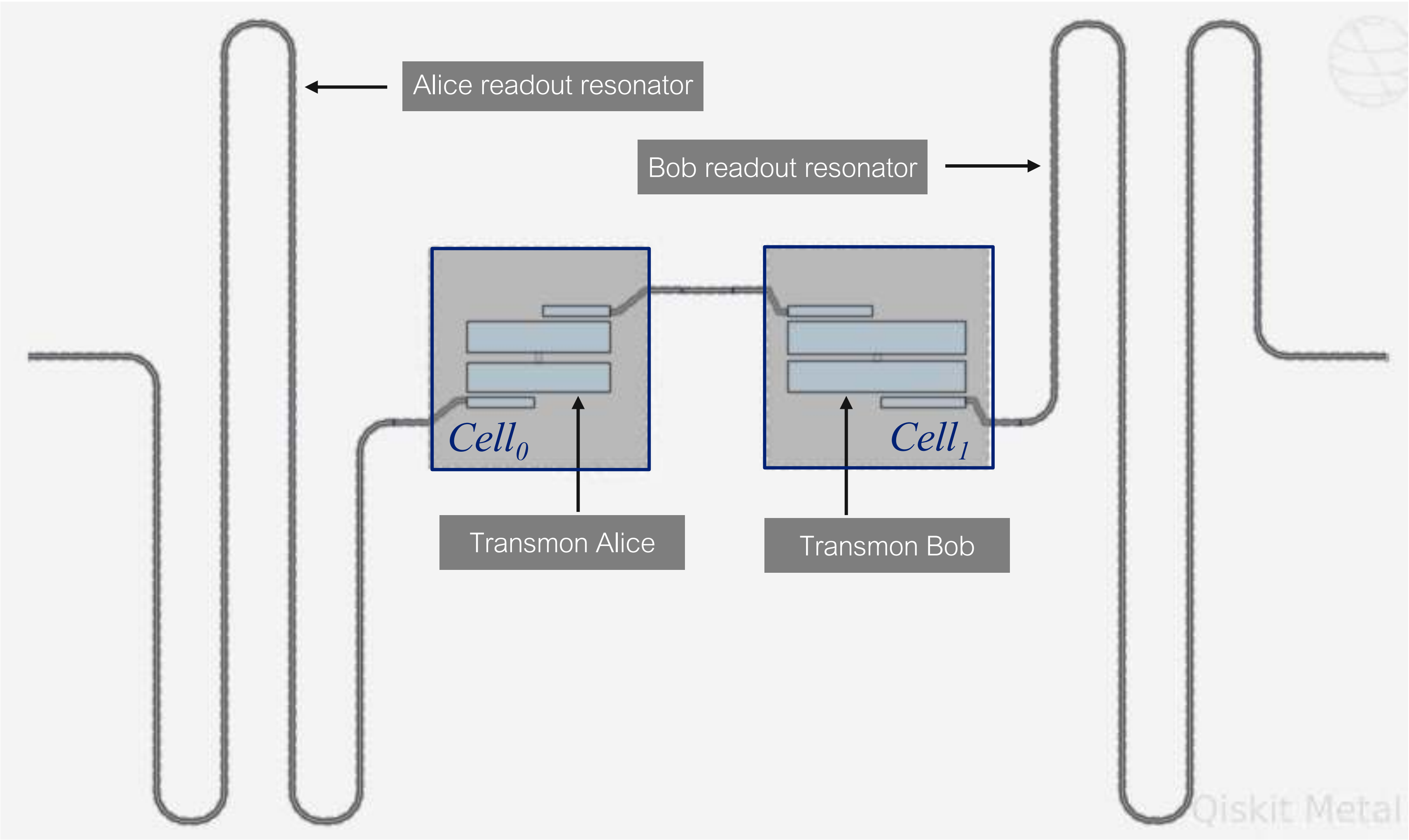
IBM **Quantum**

Layout » Electromagnetic simulation » LOM analysis » Hamiltonian analysis » Time evolution simulation



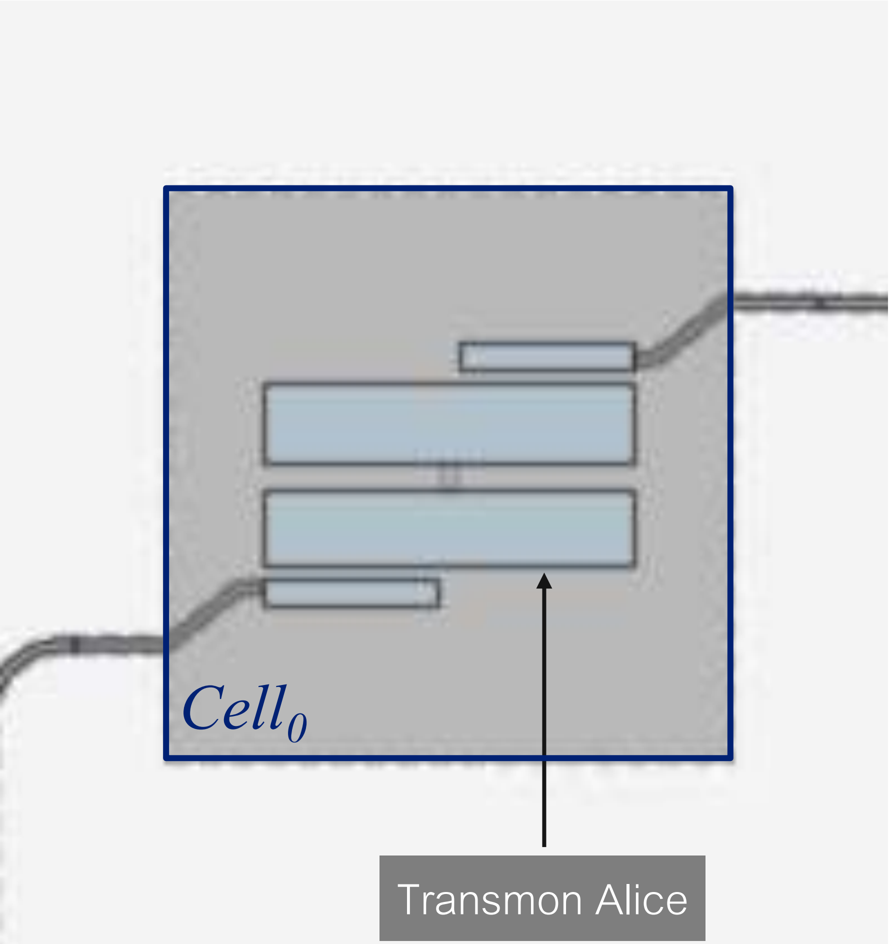
Physical layout

Layout » Electromagnetic simulation » LOM analysis » Hamiltonian analysis » Time evolution simulation



Simulating a cell

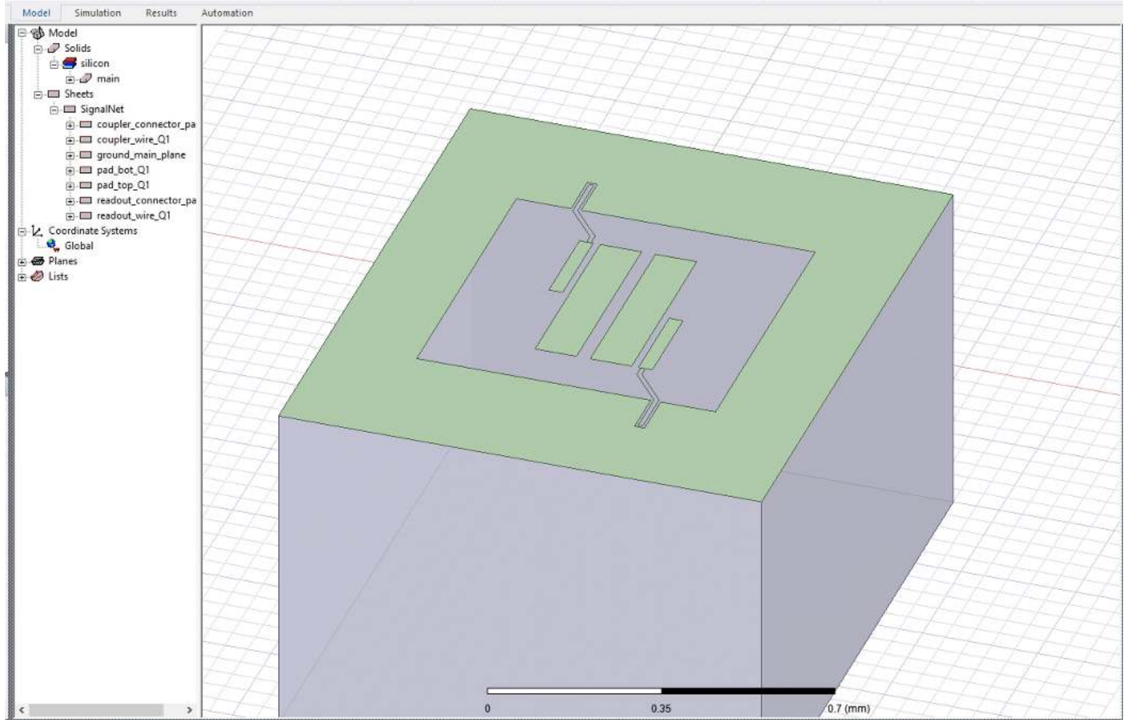
Layout » **Electromagnetic simulation** » LOM analysis » Hamiltonian analysis » Time evolution simulation



Automated with

Qiskit | quantum device design

extractor model



Extracting a cell parameters

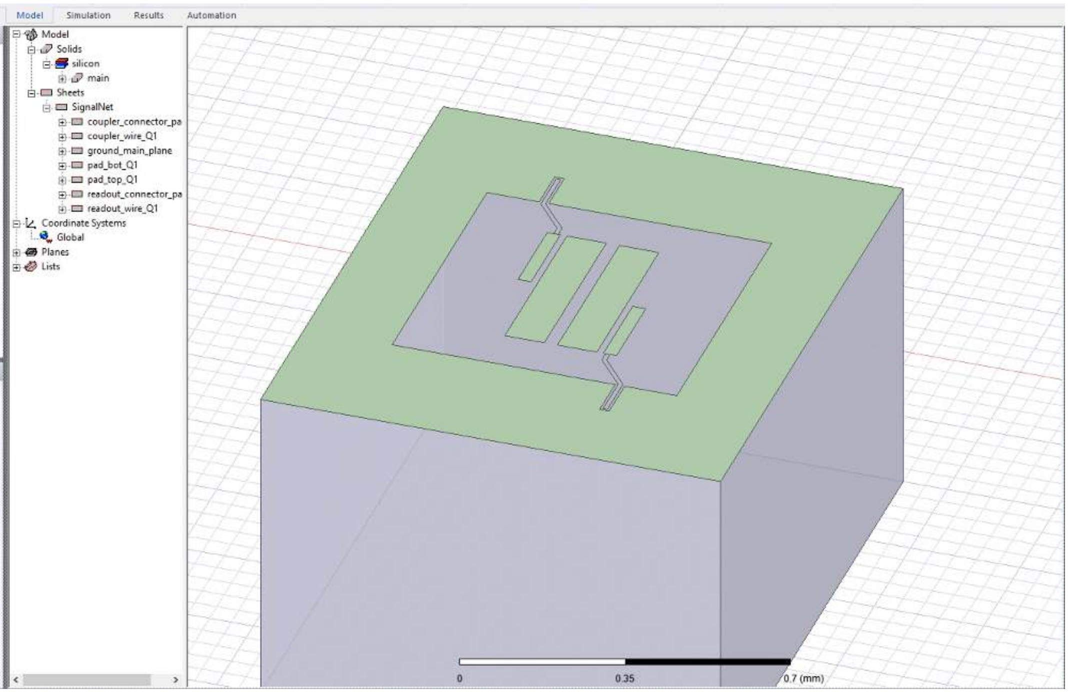
IBM **Quantum**

Layout » **Electromagnetic simulation** » LOM analysis » Hamiltonian analysis » Time evolution simulation

Automated with

Qiskit | quantum device design

extractor model



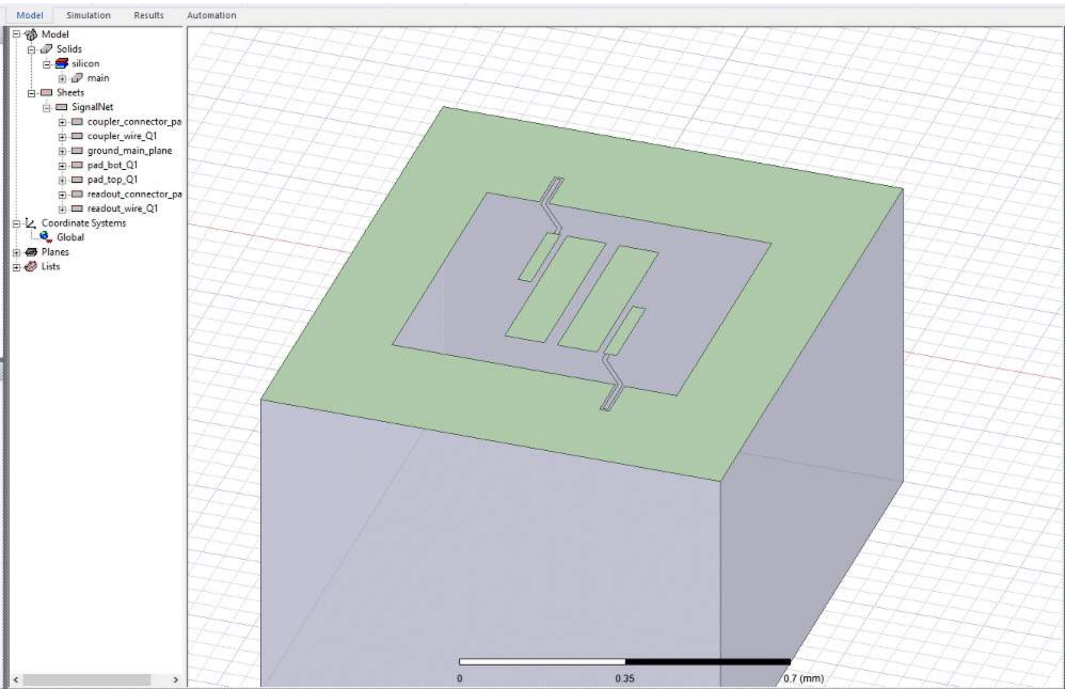
Maxwell capacitance matrix

$$
\begin{array}{c}
 \\ n_0 \\ n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5
\end{array}
\begin{array}{c}
\begin{array}{cccccc} n_0 & n_1 & n_2 & n_3 & n_4 & n_5 \end{array} \\
\begin{pmatrix}
C_{0\Sigma} & -C_{01} & -C_{02} & -C_{03} & -C_{04} & -C_{05} \\
 & C_{1\Sigma} & -C_{12} & -C_{13} & -C_{14} & -C_{15} \\
 & & C_{2\Sigma} & -C_{23} & -C_{24} & -C_{25} \\
 & & & C_{3\Sigma} & -C_{34} & -C_{35} \\
 & & & & C_{4\Sigma} & -C_{45} \\
 & & & & & C_{5\Sigma}
\end{pmatrix}
\end{array}
$$

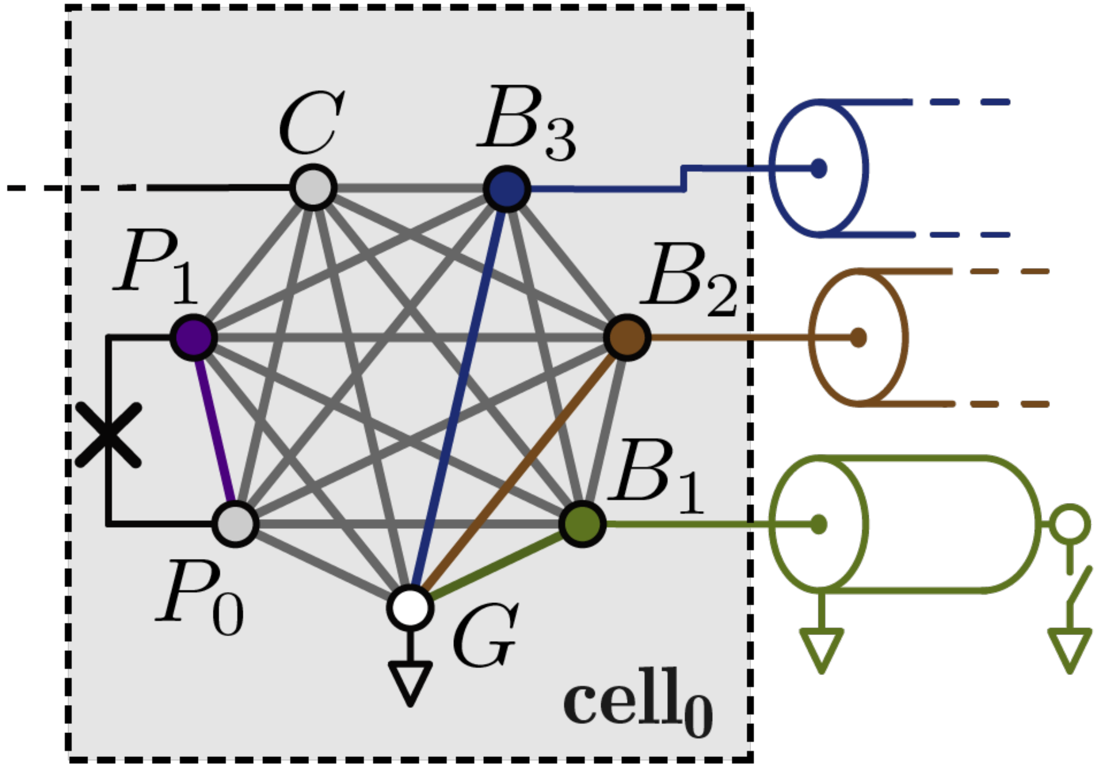
Extracting a cell parameters

Layout » **Electromagnetic simulation** » LOM analysis » Hamiltonian analysis » Time evolution simulation

extractor model



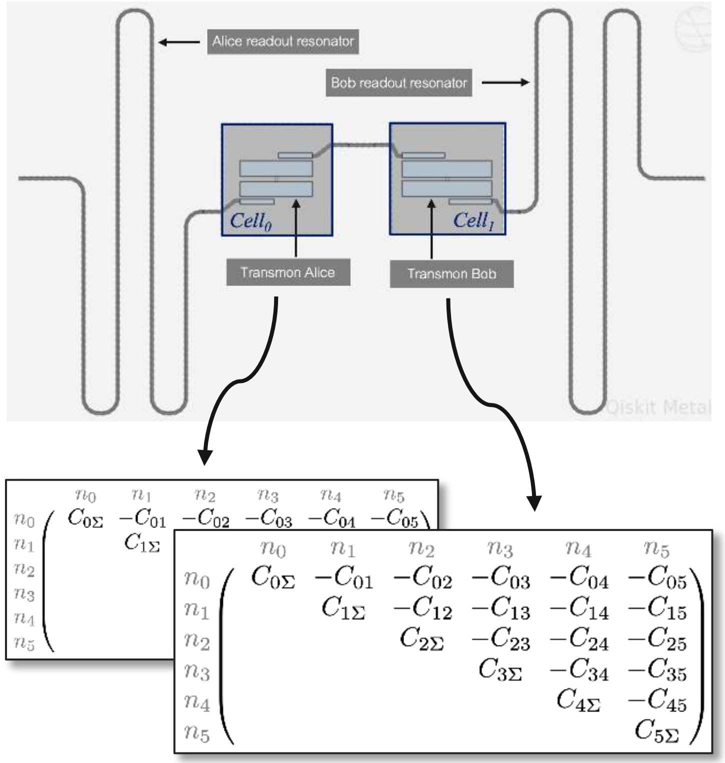
Matrices and graph theory!!

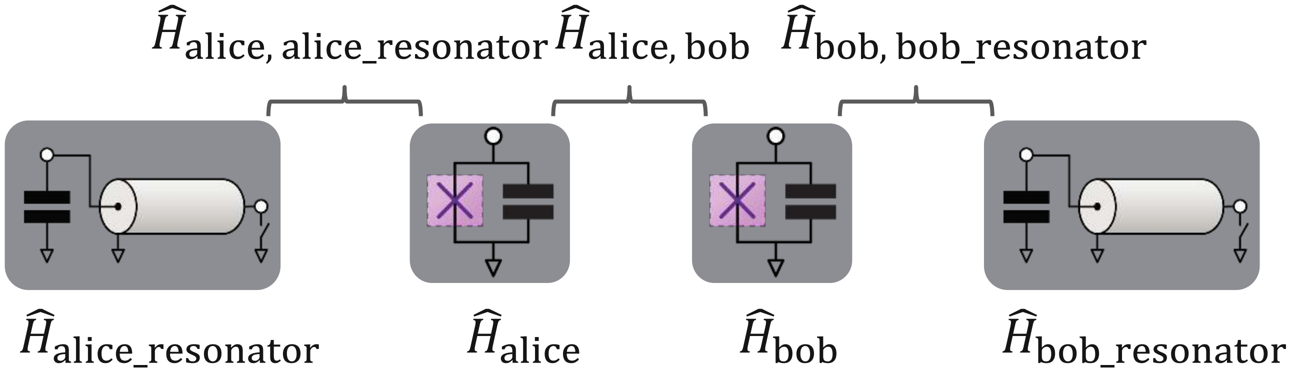


Apply quasi-lumped model to composite system

IBM **Quantum**

Layout » Electromagnetic simulation » **LOM Analysis** » Hamiltonian analysis » Time evolution simulation





$$\hat{H}_{\text{full}} = \hat{H}_0 + \sum_{n=1}^{K} \hat{H}_n + \sum_{n=0}^{K} \sum_{m=n+1}^{K} \hat{H}_{nm}$$

Minev et al. arXiv:2103.10344

Diagonalizing the composite Hamiltonian

IBM **Quantum**

Layout » Electromagnetic simulation » LOM analysis » **Hamiltonian analysis** » Time evolution simulation

$$\hat{H}_{\mathrm{full}} = \hat{H}_0 + \sum_{n=1}^{K} \hat{H}_n + \sum_{n=0}^{K} \sum_{m=n+1}^{K} \hat{H}_{nm}$$

scQubits

Generate the hilberspace from the composite system, leveraging the scqubits package

```
hilberspace = composite_sys.create_hilberspace()
hilberspace = composite_sys.add_interaction()
```

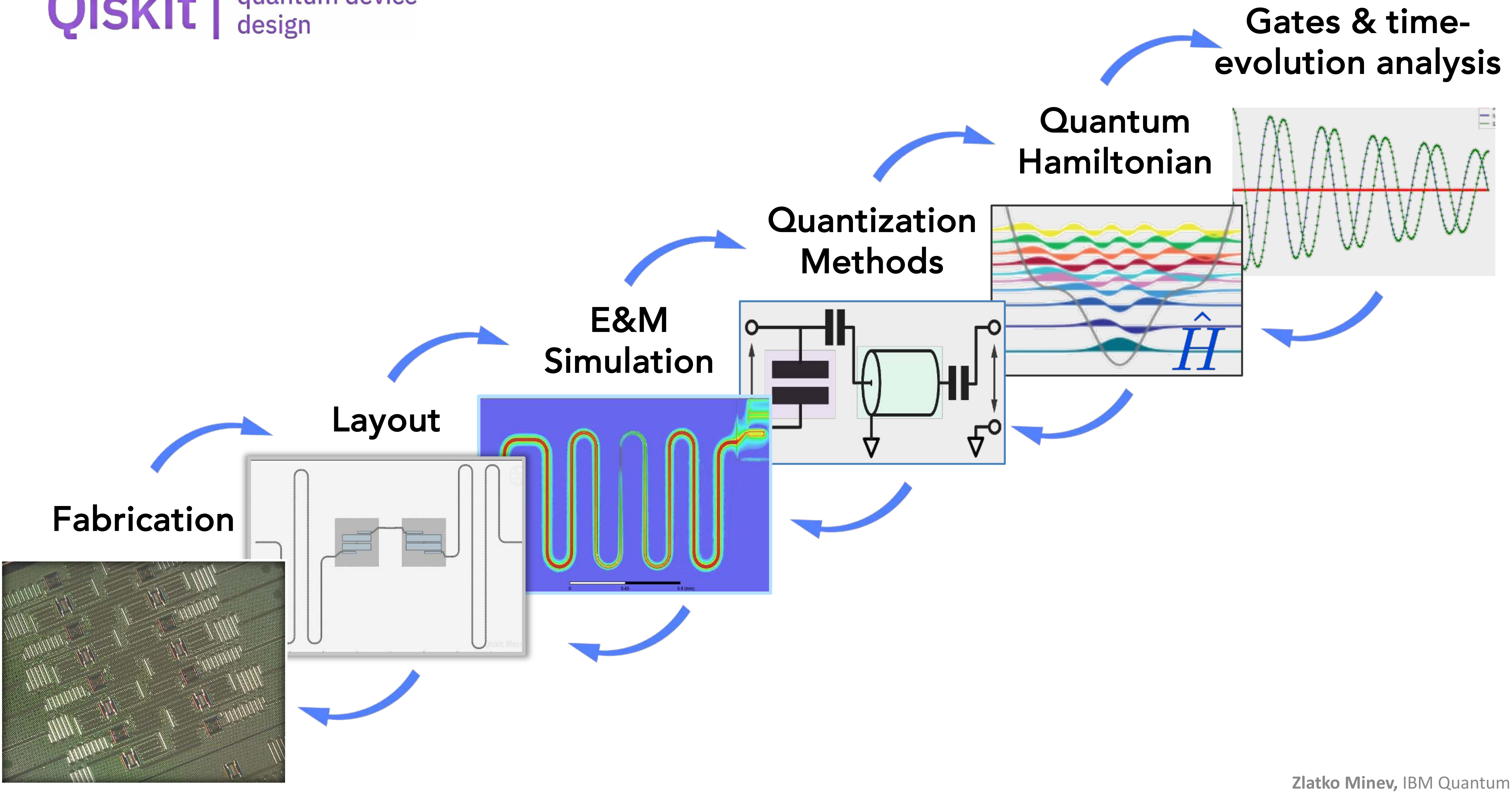
Print the results

```
hamiltonian_results = composite_sys.hamiltonian_results(hilberspace, evals_count=30)
```

```
Finished eigensystem.

system frequencies in GHz:
--------------------------
{'transmon_alice': 6.053360688806868, 'transmon_bob': 4.7989883222888094, 'readout_alice': 8.00905482071

Chi matrices in MHz
--------------------------
                transmon_alice  transmon_bob  readout_alice  readout_bob
transmon_alice     -353.239816     -0.542895      -4.132854    -0.003120
transmon_bob         -0.542895   -263.940098      -0.001154    -1.460416
readout_alice        -4.132854     -0.001154       4.283111    -0.000017
readout_bob          -0.003120     -1.460416      -0.000017     3.829744
```

Groszkowski et al. arXiv:2107.08552

Qiskit | quantum device design



Zlatko Minev, IBM Quantum

From Hamiltonian to experiment simulation

IBM **Quantum**

Layout » Electromagnetic simulation » LOM analysis » Hamiltonian analysis » **Time evolution simulation**

```
system frequencies in GHz:
--------------------------
{'transmon_alice': 6.053360688806868, 'transmon_bob': 4.7989883222888094, 'readout_alice': 8.00905482071

Chi matrices in MHz
--------------------------
                 transmon_alice  transmon_bob  readout_alice  readout_bob
transmon_alice      -353.239816     -0.542895      -4.132854    -0.003120
transmon_bob          -0.542895   -263.940098      -0.001154    -1.460416
readout_alice         -4.132854     -0.001154       4.283111    -0.000017
readout_bob           -0.003120     -1.460416      -0.000017     3.829744
```

```
hilbertspace.hamiltonian()
```

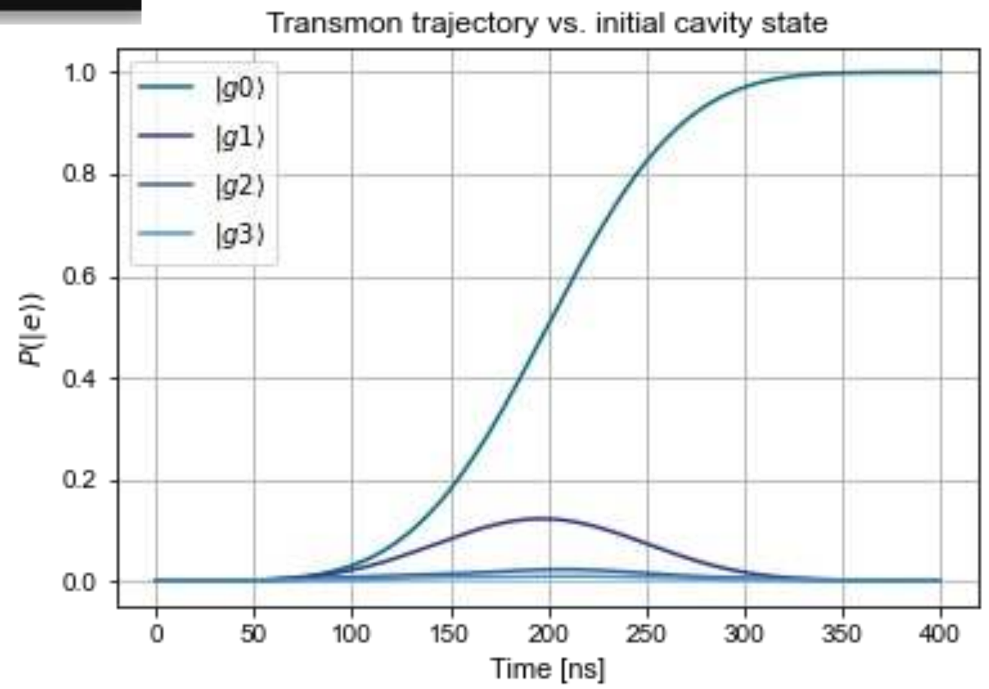
```
[11]:  Quantum object: dims = [[10, 10, 3, 3], [10, 10, 3, 3]], s
```

Qiskit Pulse/Dynamics

QuTiP

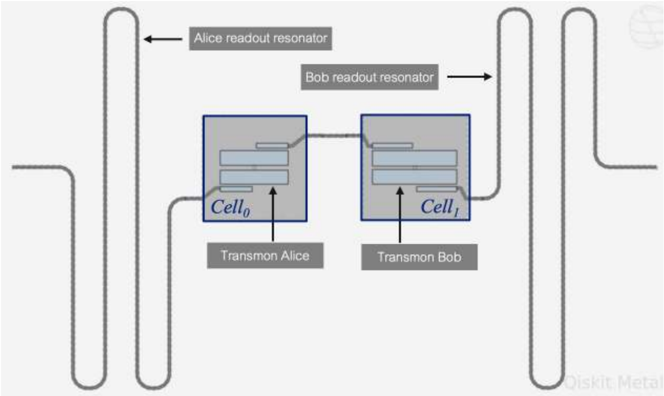
seQuencing

Simulate realistic quantum control sequences using QuTiP

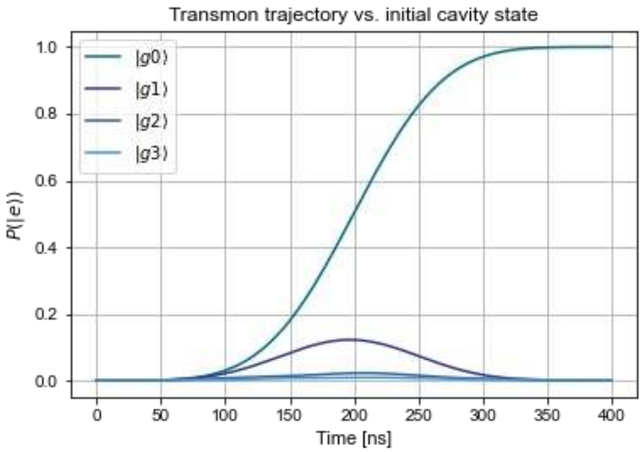


Device layout to gate simulation in one integrated workflow

Layout » **Electromagnetic simulation** » **LOM analysis** » **Hamiltonian analysis** » **Gates and time evolution simulation**



$$\hat{H}_{\text{full}} = \hat{H}_0 + \sum_{n=1}^{K} \hat{H}_n + \sum_{n=0}^{K} \sum_{m=n+1}^{K} \hat{H}_{nm}$$



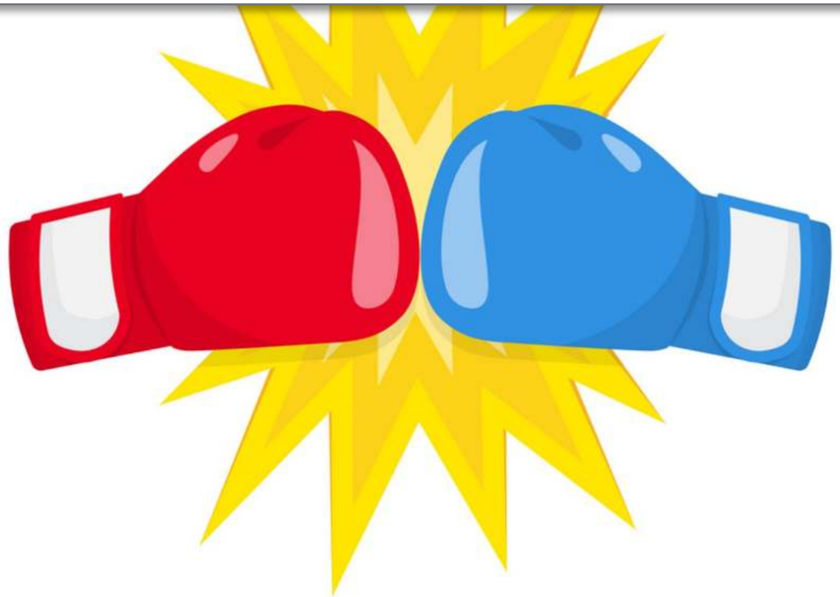
Planar devices & comparison to other methods

Quantum Physics

[Submitted on 2 Feb 2021]

Exploiting dynamic quantum circuits in a quantum algorithm with superconducting qubits

Antonio D. Corcoles, Maika Takita, Ken Inoue, Scott Lekuch, Zlatko K. Minev, Jerry M. Chow, Jay M. Gambetta

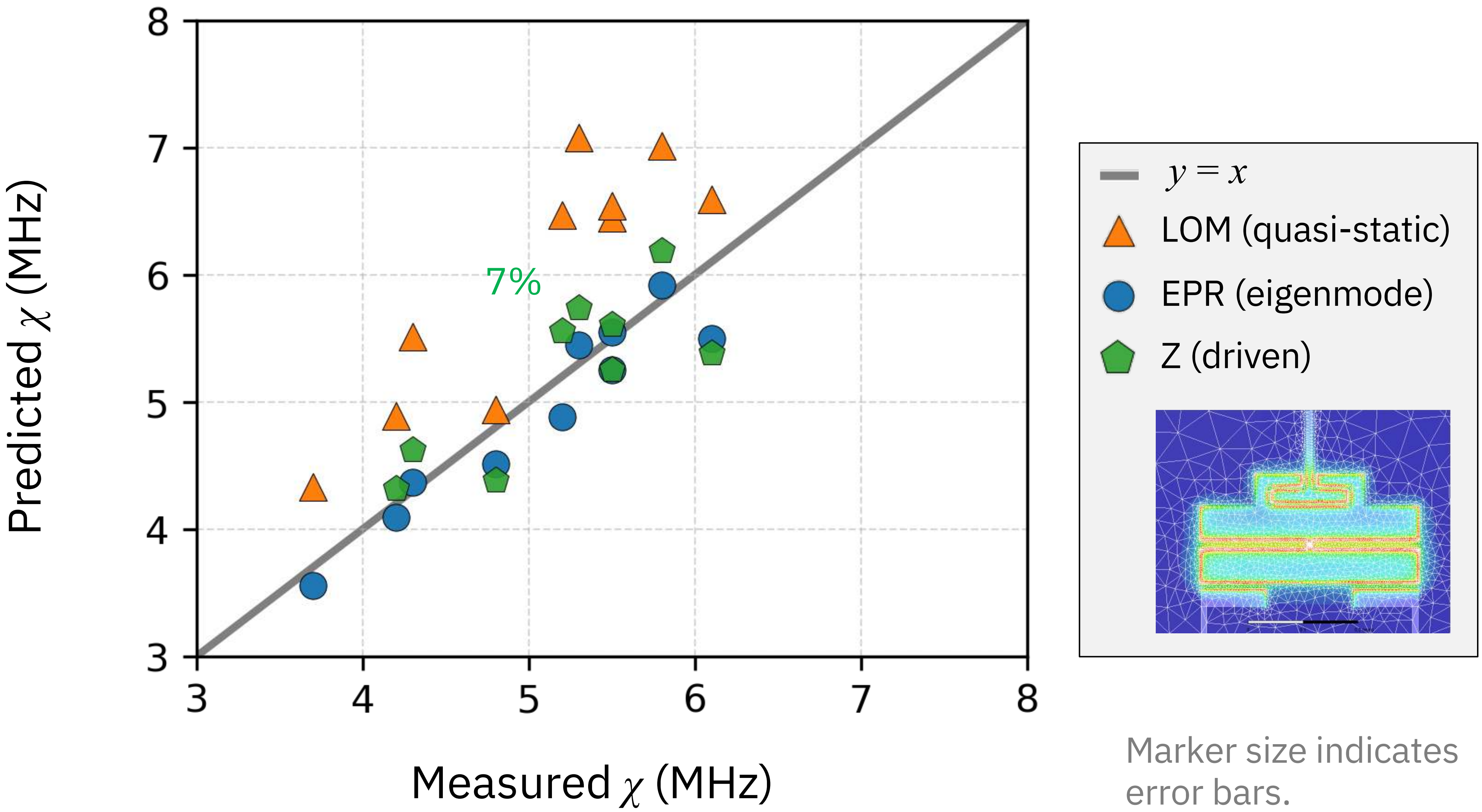


Automated with

Qiskit | quantum device design

image: makc76

Measured vs. predicted: qubit-readout cross-Kerr



Marker size indicates error bars.

Summary

Qiskit Metal quasi-lumped model design framework

Full Stack ✓

Speed ✓

Modular ✓

Open Source ✓

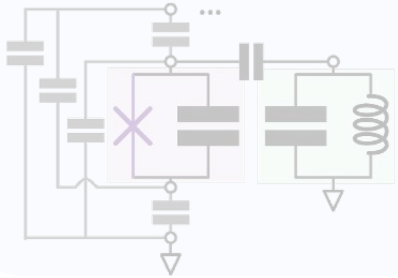
Quantization methods landscape

IBM **Quantum**

quasi-static

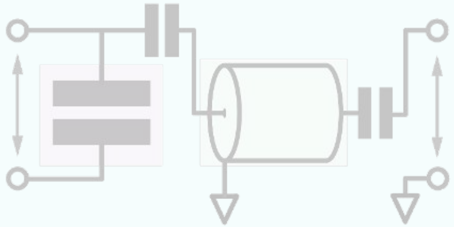
full-wave

lumped



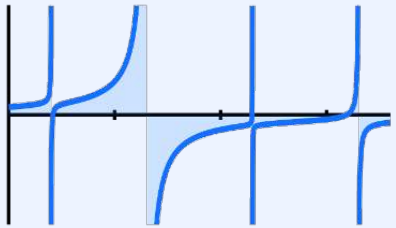
Yurke & Denker (1984), Devoret (1997), Burkard *et al.* (2004), Koch *et al*…

quasi-lumped



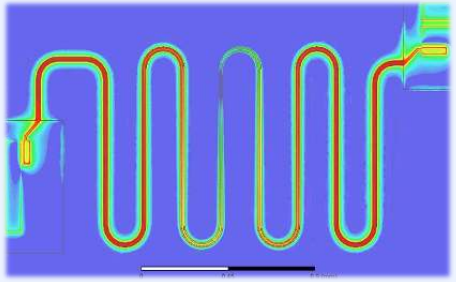
Malekakhlagh *et al.* (2017, 2019), Gely *et al.* (2019), Parra-Rodriguez *et al.* (2019), Minev *et al.* (2021), …

impedance



Nigg et al. (2012), Bourassa et al. (2012), Solgun et al. (2014, 2015, 2017) …

energy

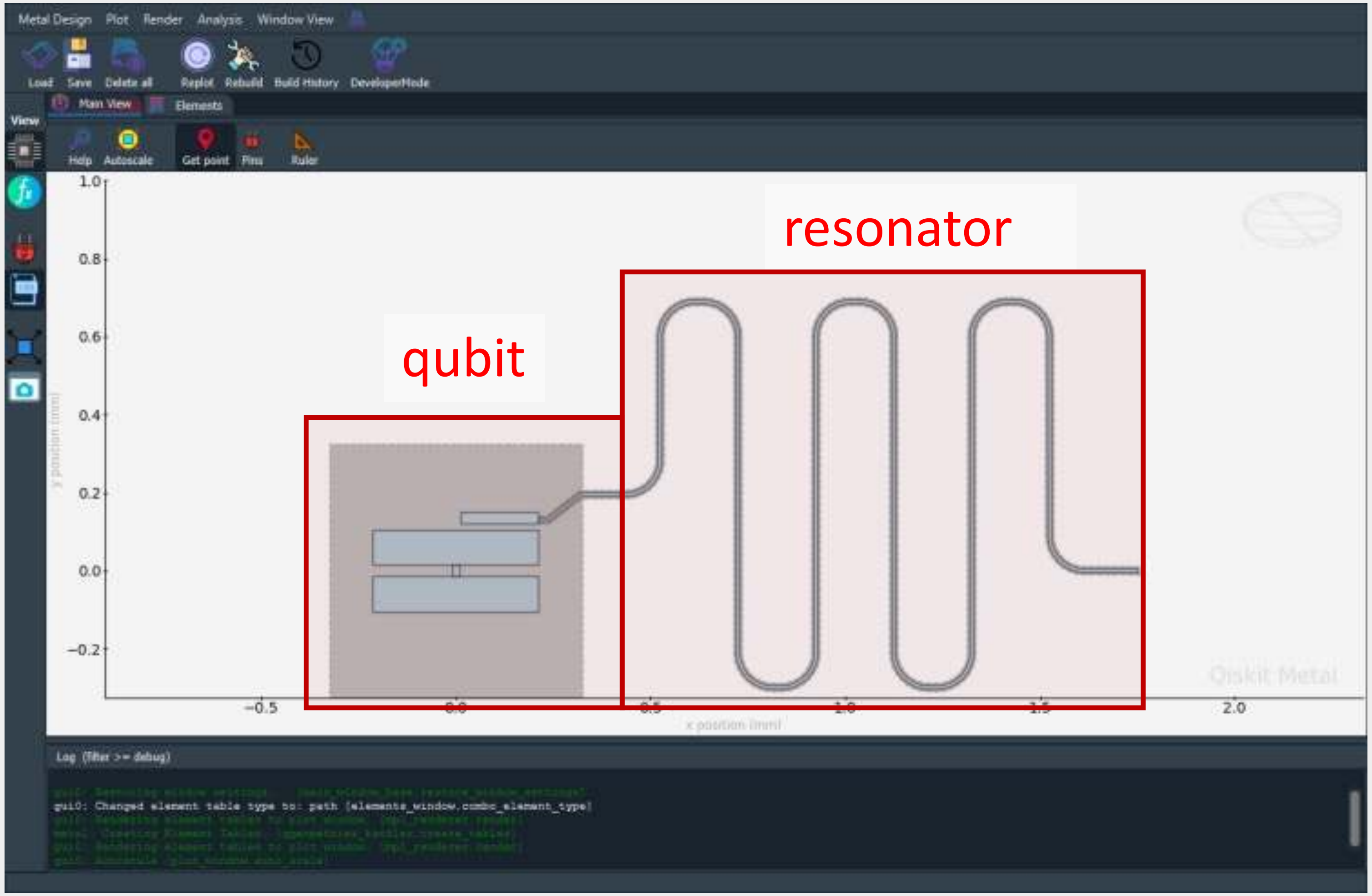


Minev (2018)
Minev *et al.* (2020)

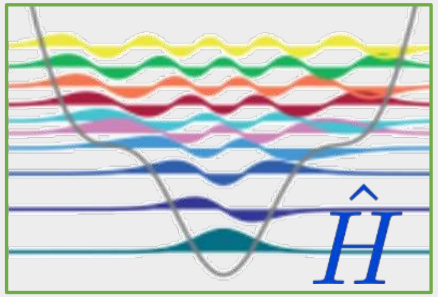
fast

more information, complexity, accuracy

Simple example

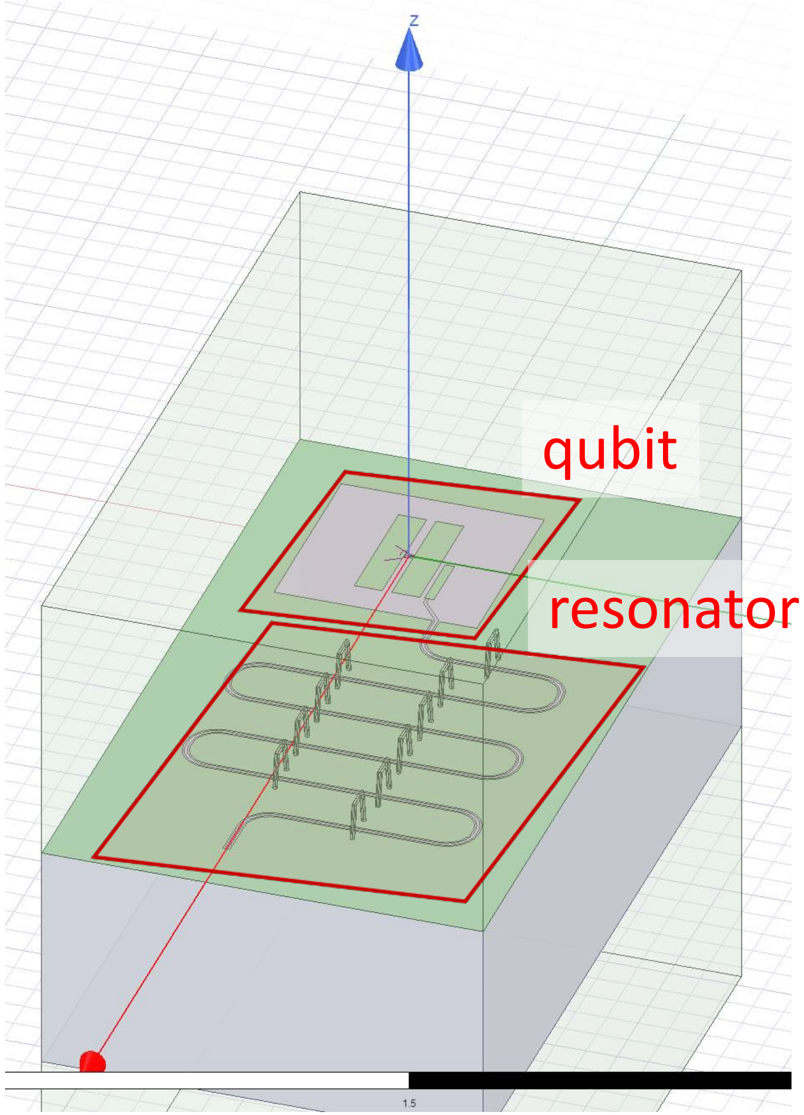


?

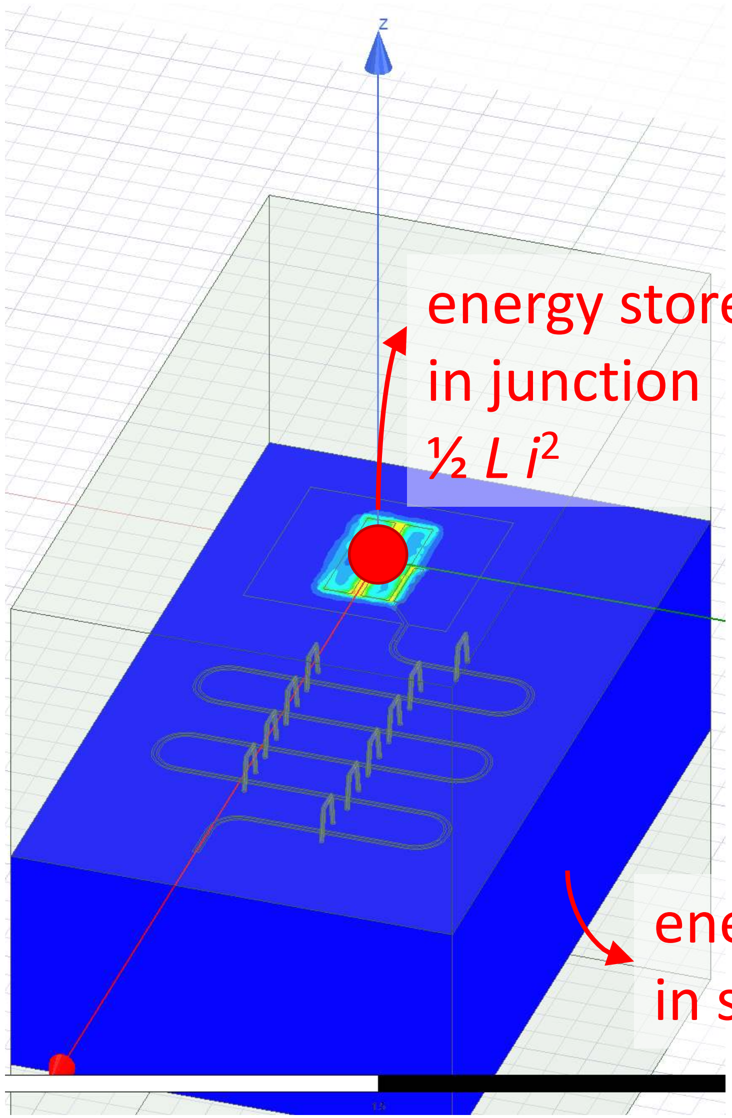


Example of step 2 with EPR

Physical (linear) model



Eigenmodes



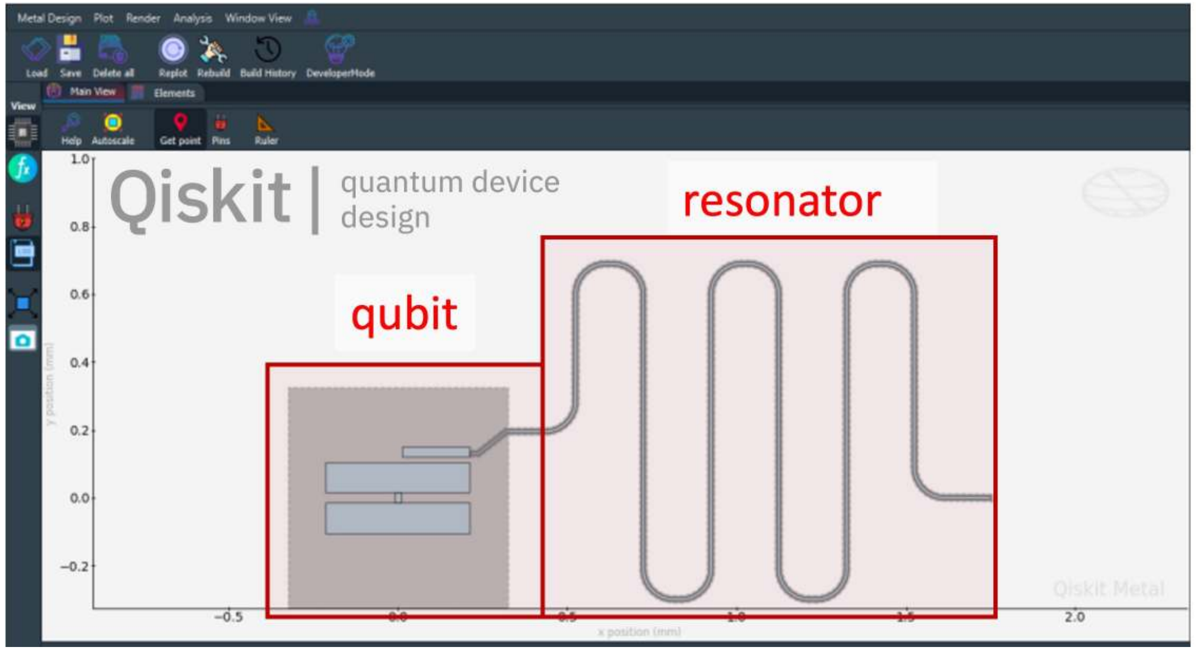
Non-linear device params

$$p_j \longrightarrow \hat{H}_{\mathrm{full}}$$

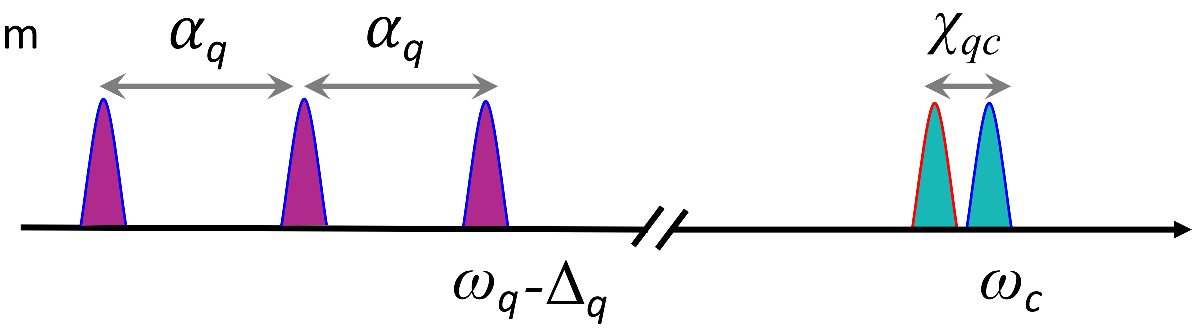
Dissipation budget

$$p_l \longrightarrow \mathcal{D}[\sqrt{\kappa}\hat{a}]$$

Step 3: Hamiltonian



Transition spectrum



Qubit/cavity anharmonicity

$$\alpha_{q/c} = p_{q/c}^2 \frac{\hbar\omega_{q/c}^2}{8E_J}$$

Qubit-cavity dispersive shifty

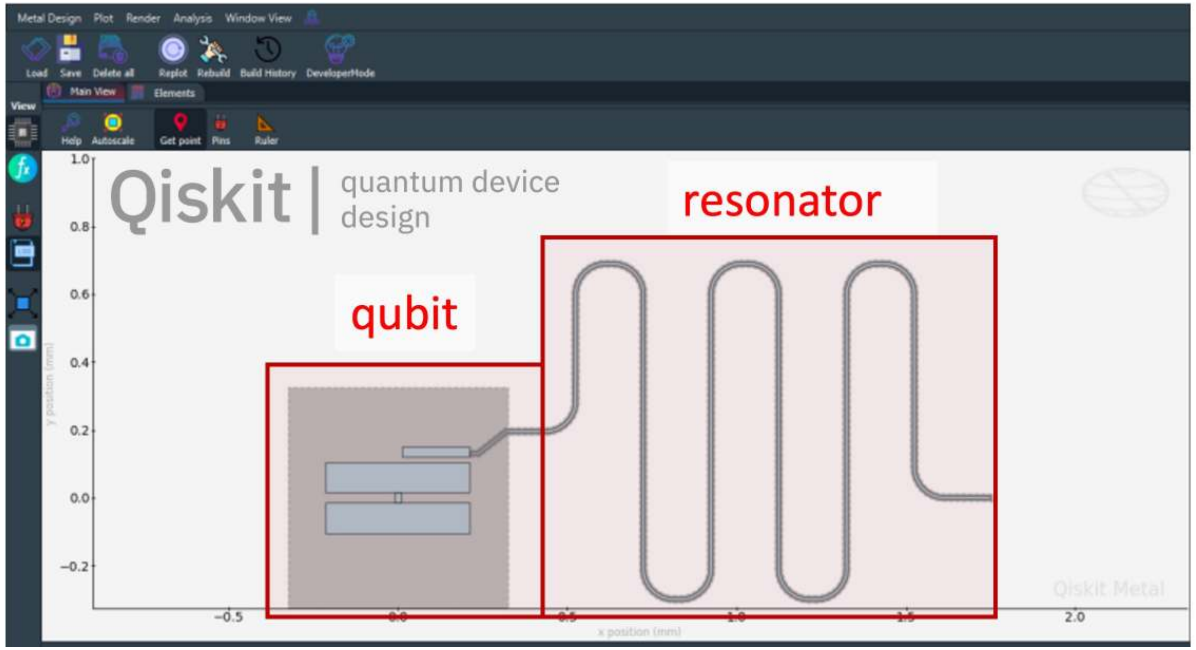
$$\chi_{qc} = p_q p_c \frac{\hbar\omega_q\omega_c}{4E_J}$$

Qubit Lamb shift

$$\Delta_q = \alpha_q - \frac{1}{2}\chi_{qc}$$

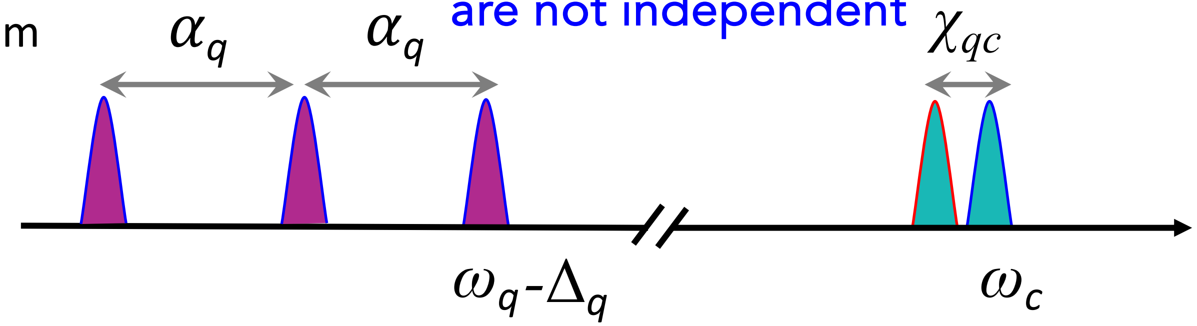
H_{eff}: for simplicity, showing up to $\mathcal{O}(\varphi^6)$ in RWA

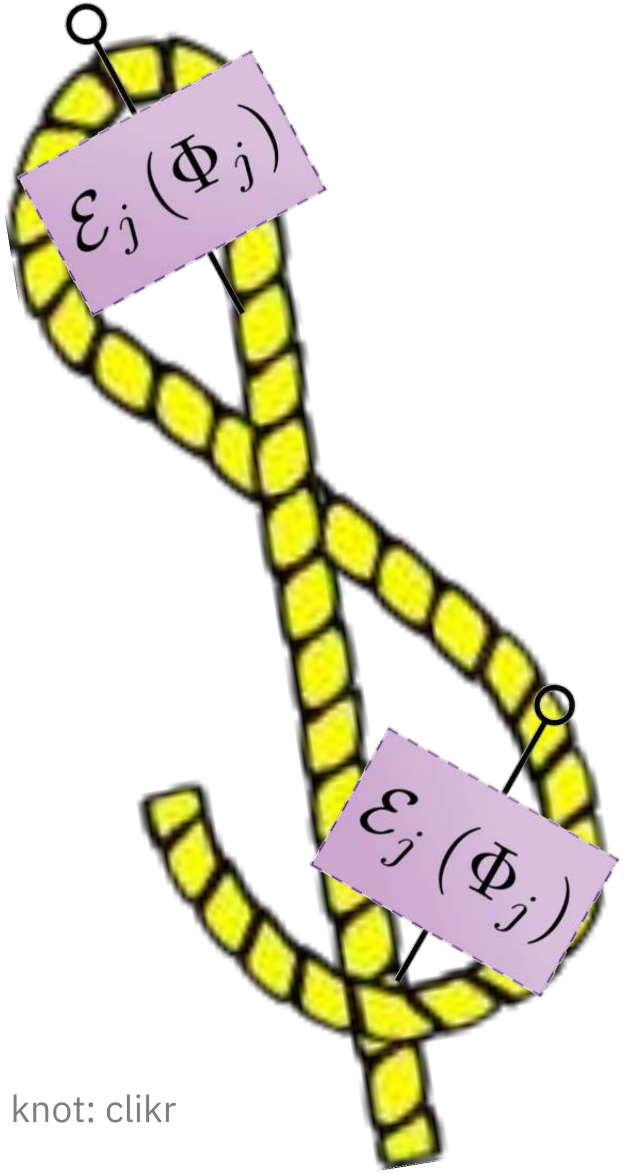
Step 3: Hamiltonian: Universal constraints & EPR monogamy



Transition spectrum

Quantum parameters are not independent



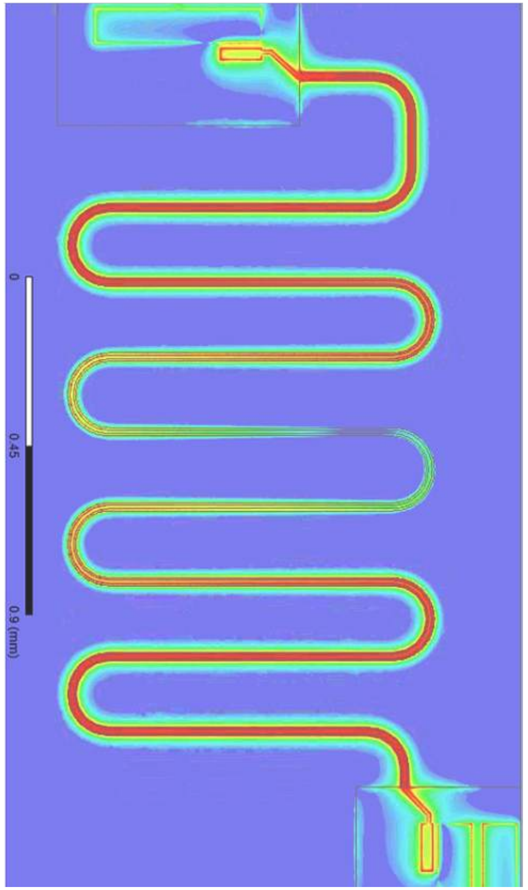


knot: clikr

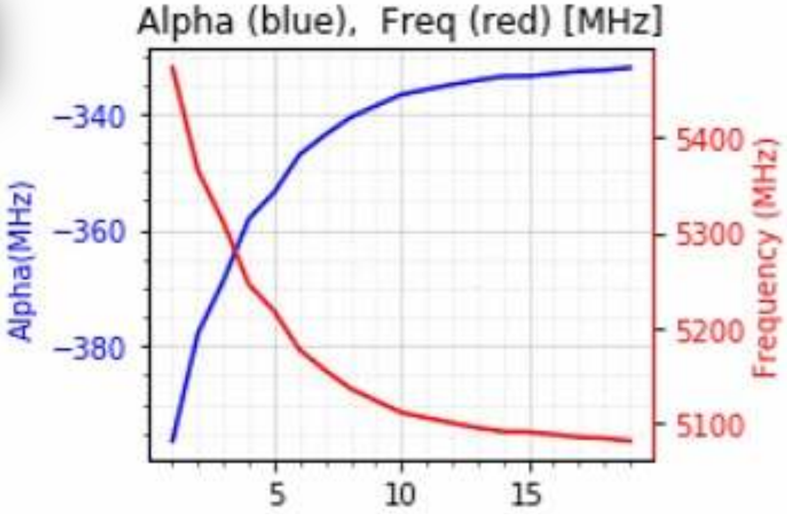
H_{eff}: for simplicity, showing up to $\mathcal{O}(\varphi^6)$ in RWA

Automated analysis and reports

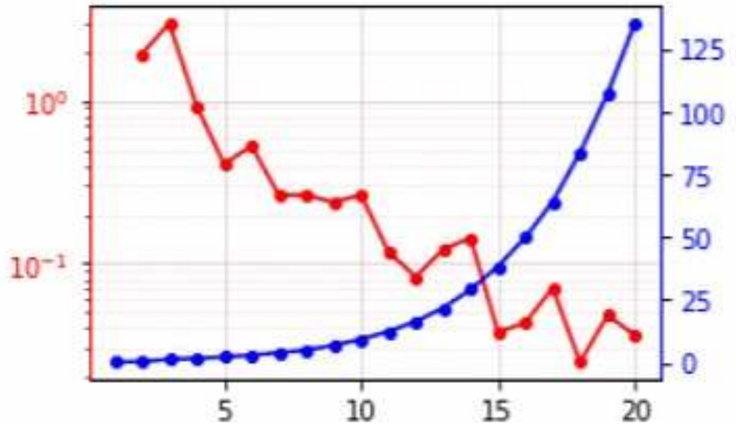
```
metal.analysis.lumped_model.analyze('Q1')
```



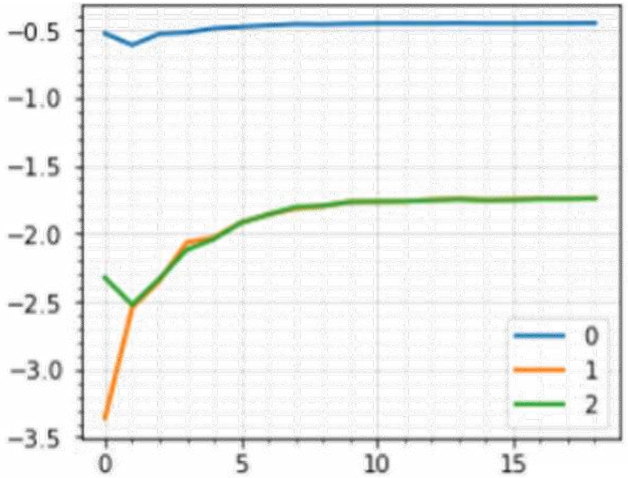
Qubit frequency & anharmonicity



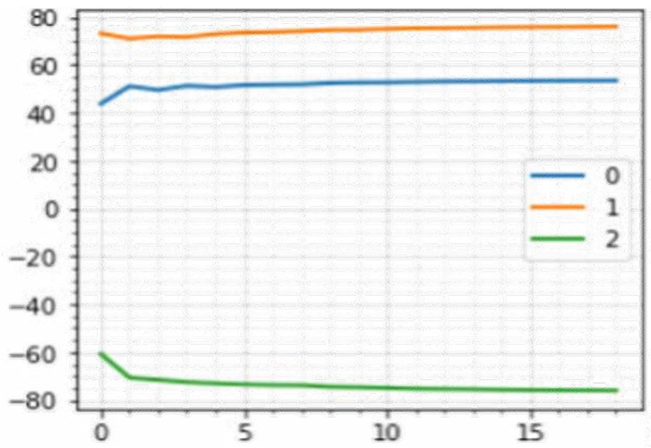
FE simulation convergence



cross-Kerr χ coupling (MHz)



Linear g coupling (MHz)



Analysis pass number

EPR theory vs. experiment

arXiv.org > quant-ph > arXiv:2010.00620

Quantum Physics

[Submitted on 1 Oct 2020]

Energy-participation quantization of Josephson circuits

Zlatko K. Minev, Zaki Leghtas, Shantanu O. Mundhada, Lysander Christakis, Ioan M. Pop, Michel H. Devoret

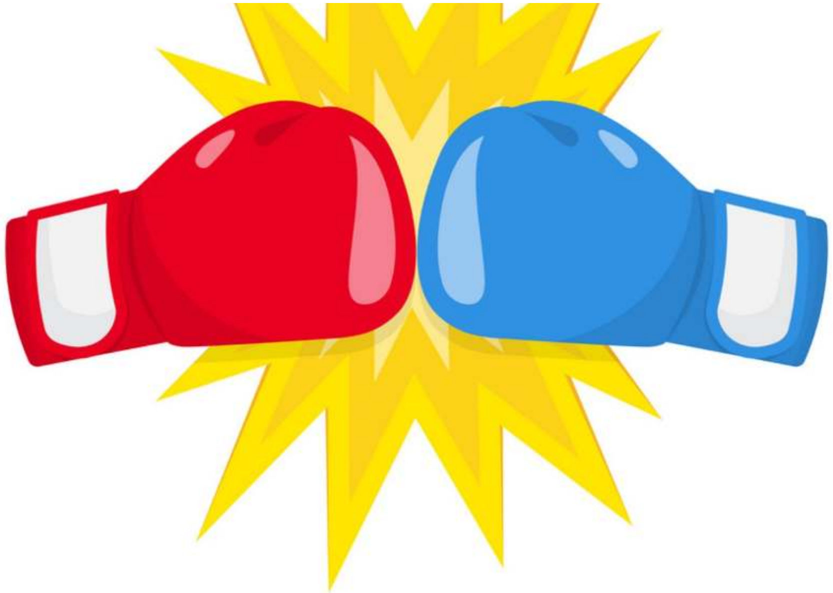
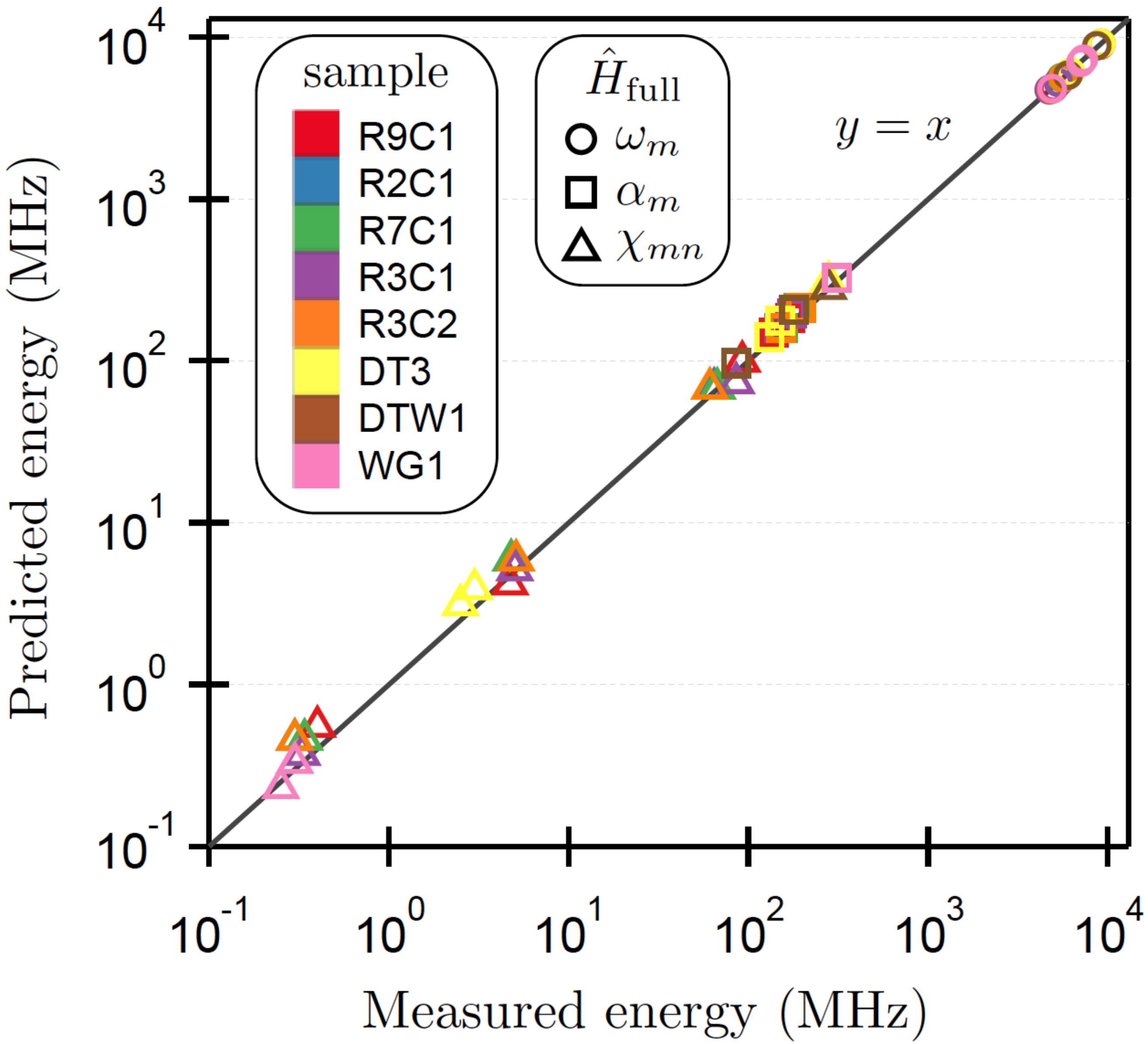
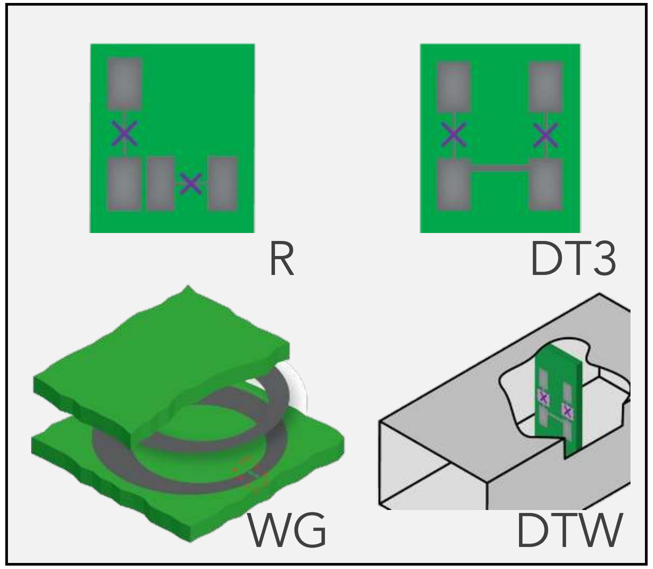


image: makc76

Theory vs. experiment: agreement over 5 orders of magnitude





R: Minev *et al.* (2018)
WG: Minev *et al.* (2013, 2016)
DT3, DTW: Minev *et al.* (2019)

Use in the community

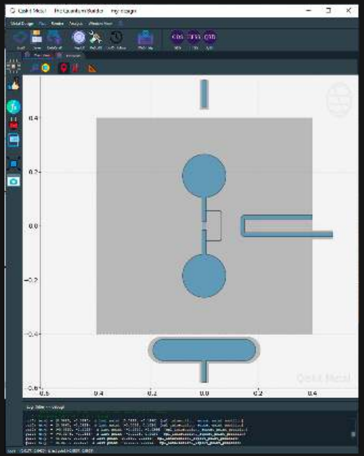
Qiskit -Metal is being used by researchers, cont'd.

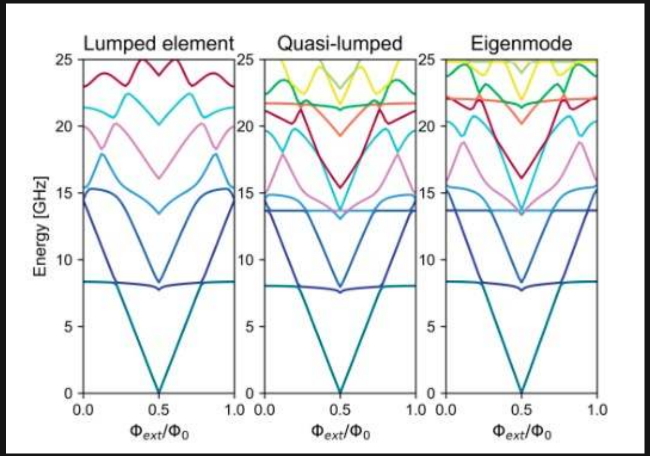
QuTech / **Technische Universiteit Delft**

Figen Yilmaz-Andersen Lab: Fluxonium chip design

Roald van den Boogaart-Andersen Lab: Fluxonium spectrum (eigenmode = pyEPR) to show how they have extended the analysis methods to be able to handle fluxonium qubits

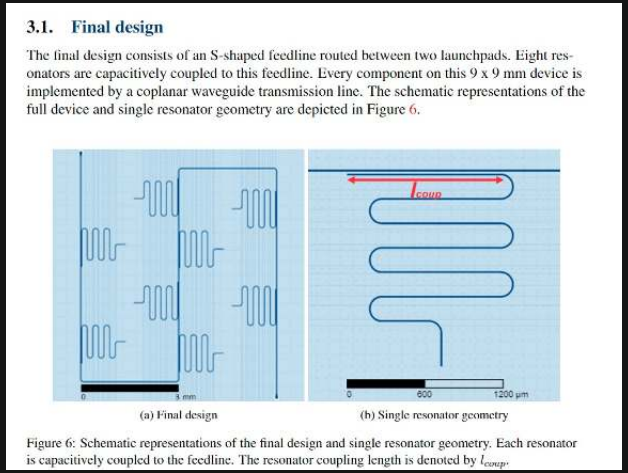
Sara Buhktari-Andersen Lab: Fabricated resonator design being measured





"Qiskit-Metal is the best tool if one studies superconducting qubits"

-- Figen Yilmaz



3.1. Final design

The final design consists of an S-shaped feedline routed between two launchpads. Eight resonators are capacitively coupled to this feedline. Every component on this 9 x 9 mm device is implemented by a coplanar waveguide transmission line. The schematic representations of the full device and single resonator geometry are depicted in Figure 6.

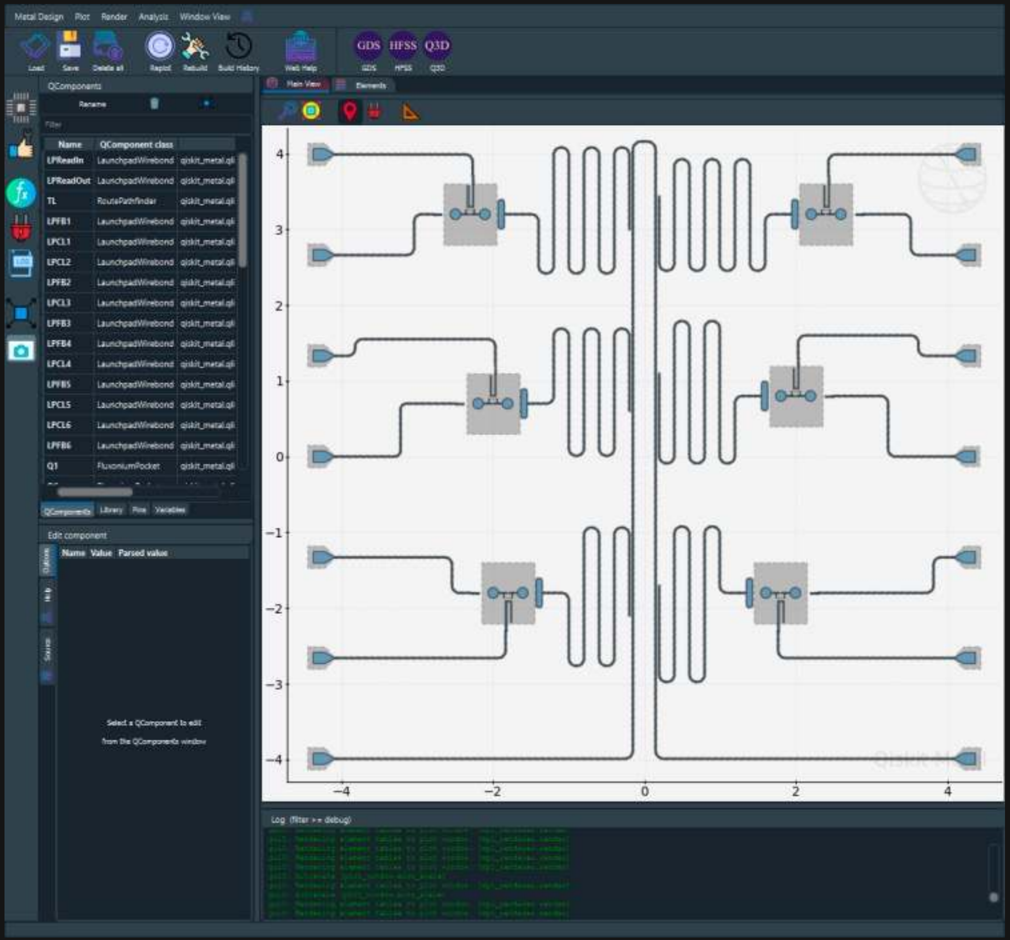
(a) Final design

(b) Single resonator geometry

Figure 6: Schematic representations of the final design and single resonator geometry. Each resonator is capacitively coupled to the feedline. The resonator coupling length is denoted by l_{coup}.

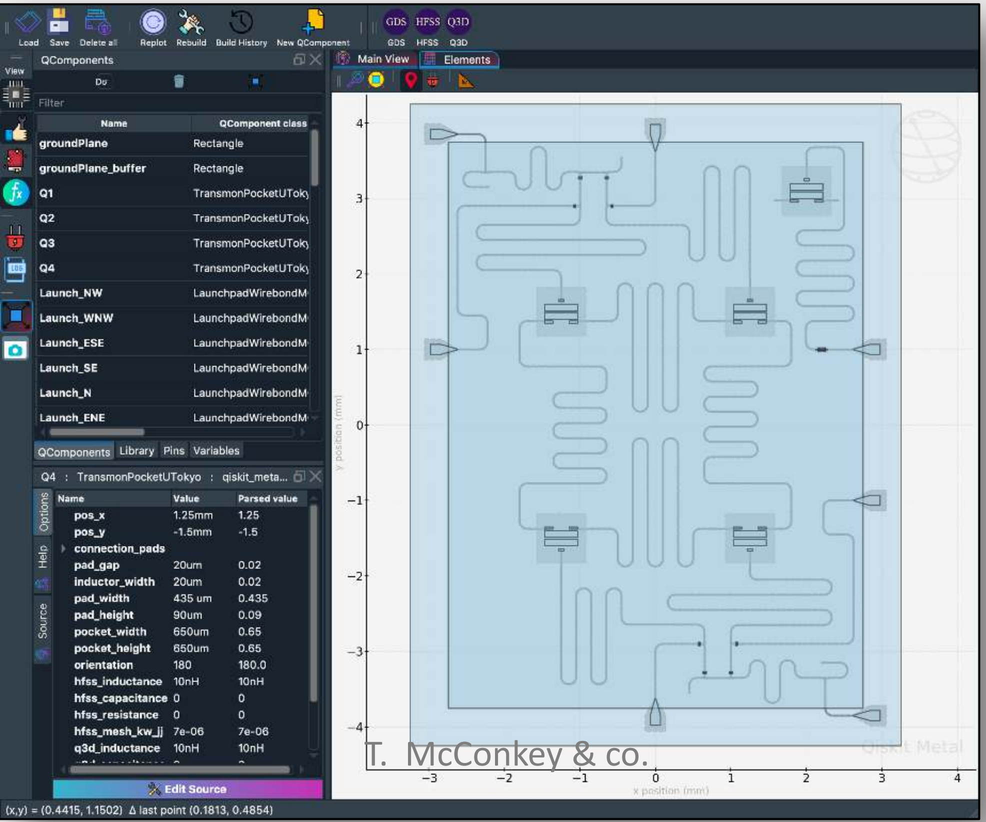
QuTech / **Technische Universiteit Delft**

Intermediate chip design.

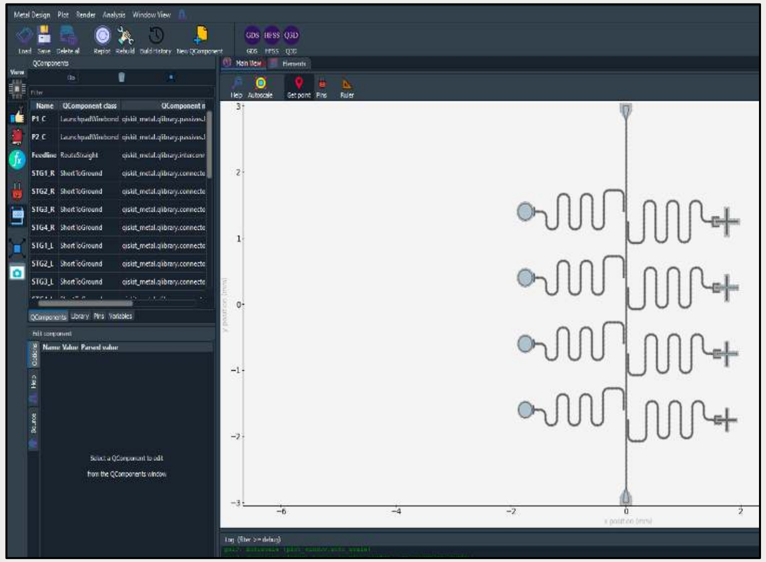


Qiskit | quantum device design

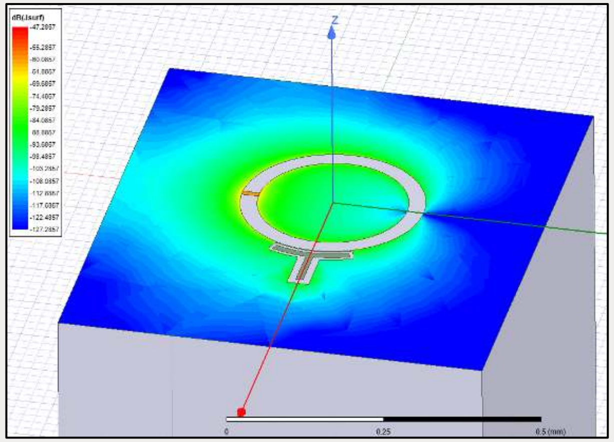
IBM 5Q Tsuru U Tokyo

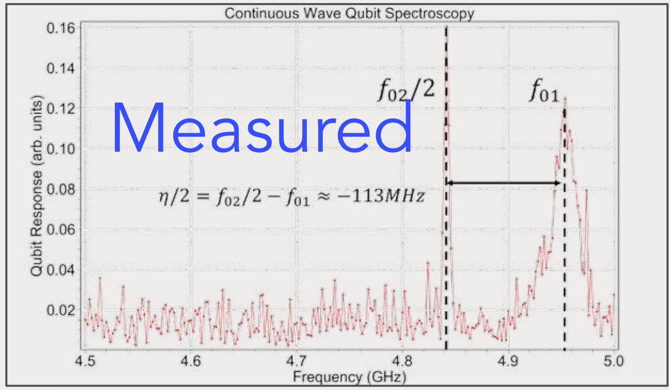


Chalmers 8Q

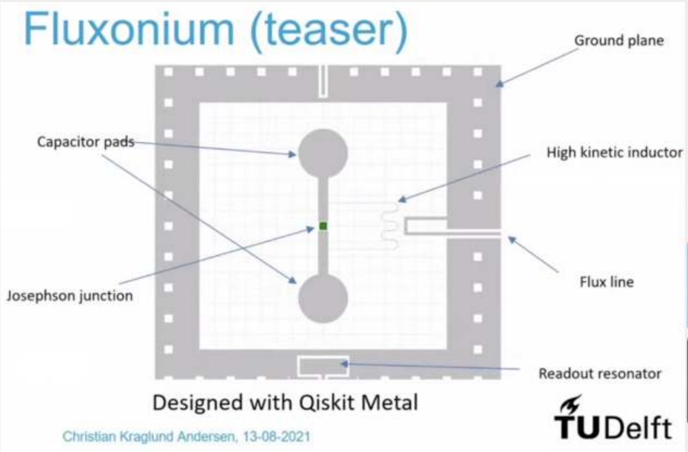


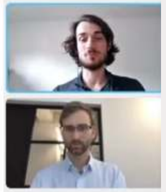
Simulated

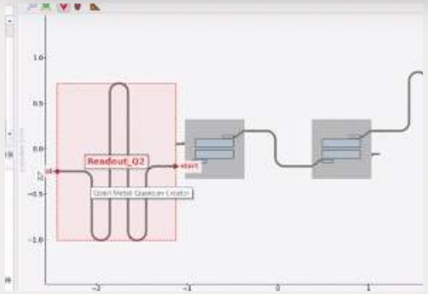




TU Delft Fluxonium







APS physics MARCH MEETING 2021 MARCH 15-19 ONLINE

CEC/ICMC 21 VIRTUAL CONFERENCE JULY 19-23 Cryogenics·Through·Collaboration

...

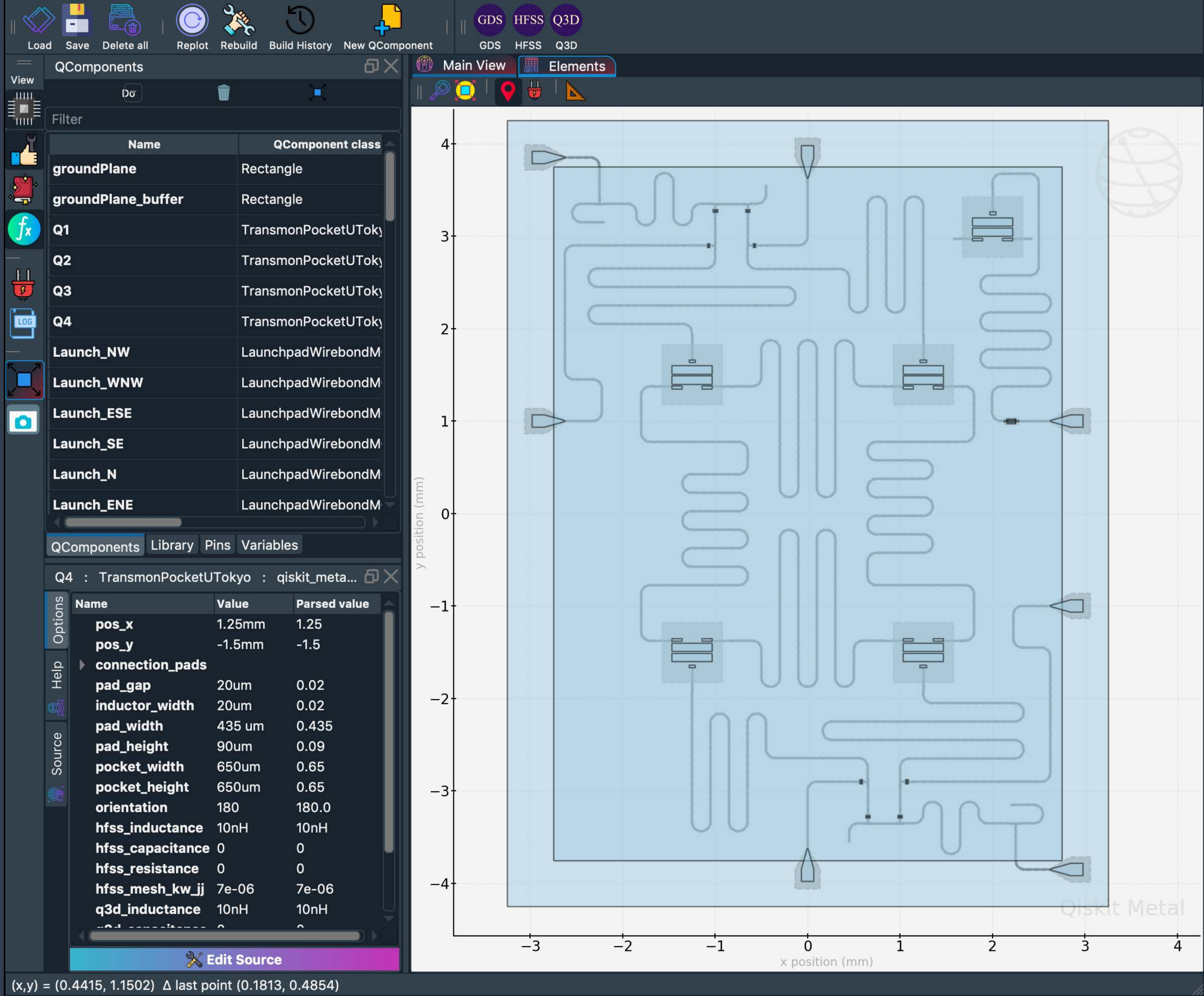
IEEE QUANTUM WEEK — IEEE International Conference on Quantum Computing and Engineering — QCE21

Zlatko Minev, IBM Quantum

Qiskit | quantum device design

Devices made with Metal

IBM 5Q Tsuru U Tokyo



Devices made with Metal - IBM 5Q Tsuru U Tokyo

GDS Export

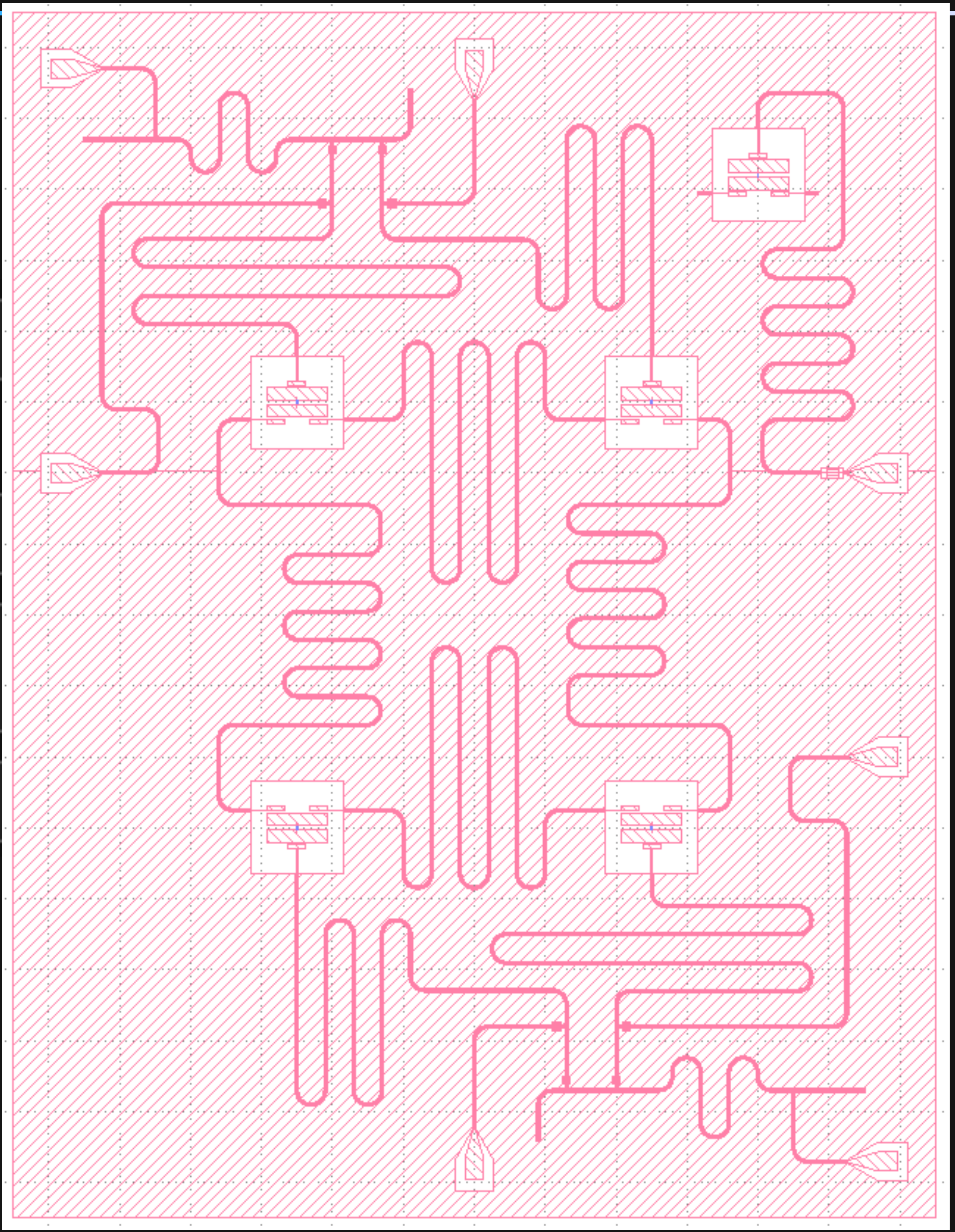
```
full_chip_gds = design.renderers.gds
```

```
...
```

```
full_chip_gds.options
```

```
full_chip_gds.options['path_filename'] ='../resources/Fake_Junctions.GDS'
full_chip_gds.options['no_cheese']['buffer']='50um'
```

```
full_chip_gds.export_to_gds('Full_Chip_01.gds')
```



Recent work at APS MM

APS March Meeting 2022

Session A36: Superconducting Qubits: cQED Design Tools

APS March Meeting 2022

Monday–Friday, March 14–18, 2022; Chicago

Session A36: Superconducting Qubits: cQED Design Tools
8:00 AM–11:00 AM, Monday, March 14, 2022
Room: McCormick Place W-194A

Sponsoring Unit: DQI
Chair: Alexander McDonald, University of Chicago

Abstract: A36.00003 : Simulation of Parametrically Coupled Qubits with Qiskit Metal and pyEPR
8:24 AM–8:36 AM

← *Abstract* →

Presenter:
Zachary L Parrott
(University of Colorado, Boulder)

Authors:
Zachary L Parrott
(University of Colorado, Boulder)

Xiaoyue Jin
(National Institute of Standards and Technology Boulder)

Taewan Noh
(National Institute of Standards and Technology Boulder)

Raymond W Simmonds
(National Institute of Standards and Technology Boulder)

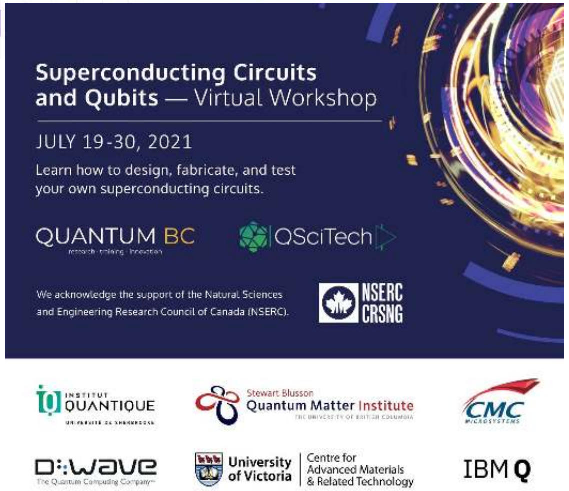
Various forms of circuit simulation can be an indispensable tool in designing superconducting qubit experiments. Before committing to a particular device, fixed after fabrication, simulations can be used to explore design trade-offs in order to optimize the desired circuit behavior. In addition, simulations can be used to help diagnose unexplained results in existing experiments. In this talk, we discuss the open source Qiskit Metal toolkit and the energy participation ratio (EPR) method for simulating various ongoing experiments of parametrically coupled circuits. By sweeping various design parameters, we can predict how these parameters influence the circuit's mode frequencies and relevant features such as static cross-couplings between these modes. Through comparisons with existing measured devices we are able to identify what design parameters strongly influence the device characteristics we wish to adjust and how to optimize them. Additionally we can assess what dynamics or behaviors are not being properly modelled or accounted for, as well as identify key features that should not be omitted in traditional lumped element design approaches. Finally we will discuss how we can use this simulation framework to inform the design of future devices and experiments within our

Qiskit | quantum device design

Conferences

IEEE QUANTUM

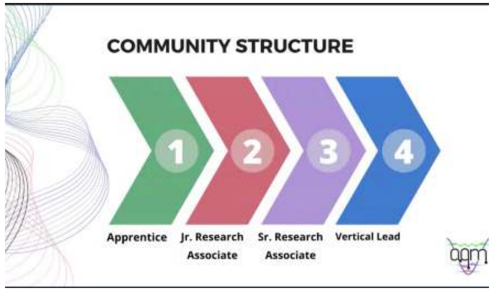
Education



Undergraduate School on Experimental Quantum Information Processing

Qiskit Metal Communities

AiQyaM
Quantum & Beyond



Lab Sessions:-

In Lab sessions, after the demonstration of the task, exercises as checkpoints will be given.

Session 1- Introduction to Qiskit Metal and ANSYS Electronics
Topics- Installation and Hands on introduction to using interface of Qiskit Metal and ANSYS Electronics

Session 2- Hands on session with Metal- 1
Topics- Introduction to workflow of Qiskit Metal, Elements of Qiskit Metal- QDesign, QComponent QRenderer, QAnalysis

Session 3- Hands on session with Metal-2
Topics- Demonstration of complete design flow, Using Components Library, Designing your own Components

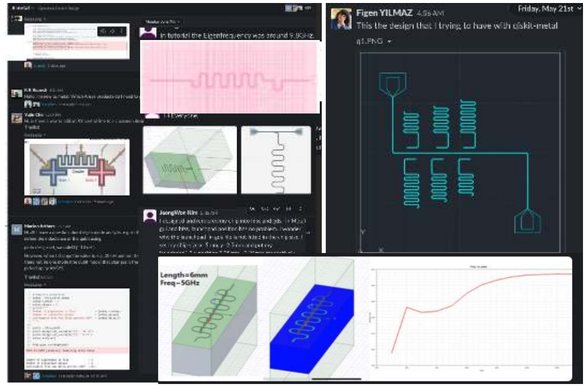
Session 4- Implementing One Qubit chip on Metal
Topics- Complete demonstration of One Qubit chip from designing to simulation on ANSYS, Interpretation of Simulation Data from ANSYS

Session 5- Implementing Two Qubit Chip on Metal
Topics- Complete demonstration of Two Qubit chip from designing to simulation on ANSYS, Interpretation of

Last 6 month: public launch & adoption

Qiskit | quantum device design

800+ users
4,000+ Q&A
From 250+ unique usrs

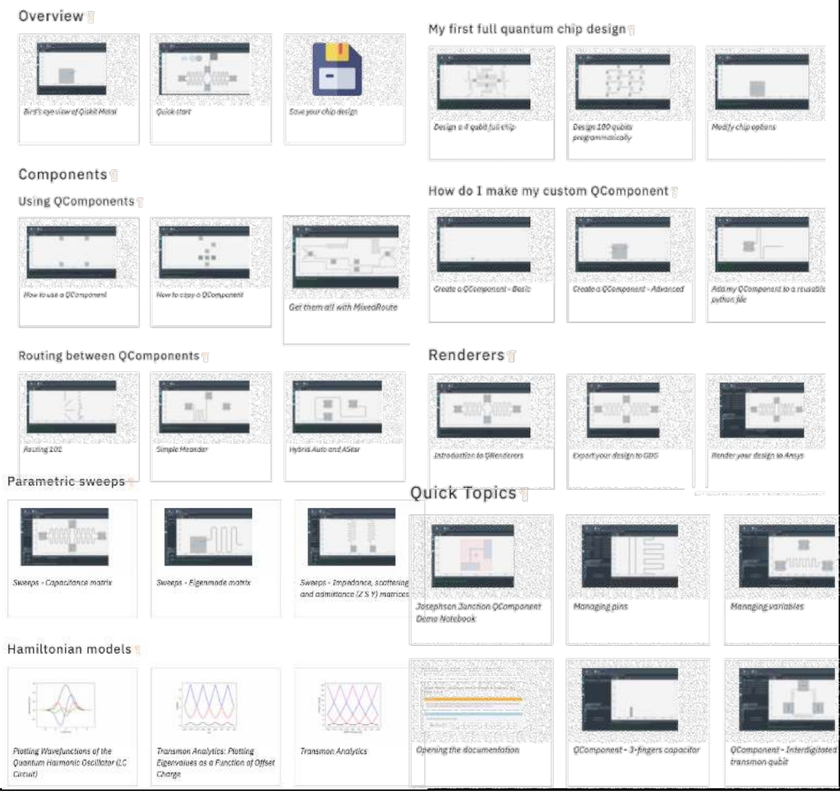


GitHub

	Star	Fork
/[qiskit-metal]()	133	81
/[qiskit-nature]()	84	72
/[qiskit-optimization]()	70	52
/[qiskit-fianance]()	57	41

47 tutorials
+ 21 demos

jupyter

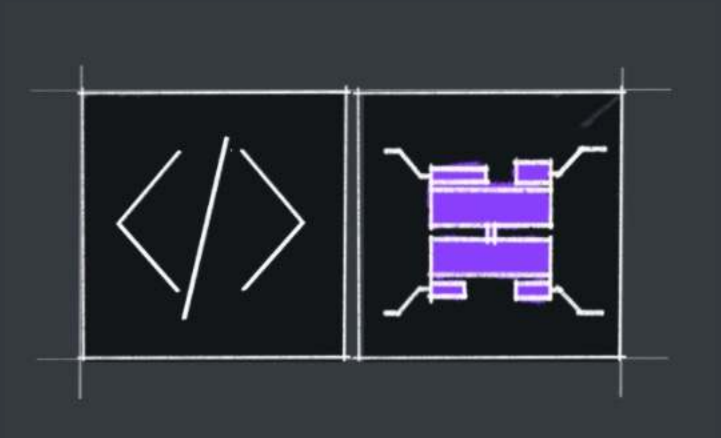


26 live tutorials
19+ hours

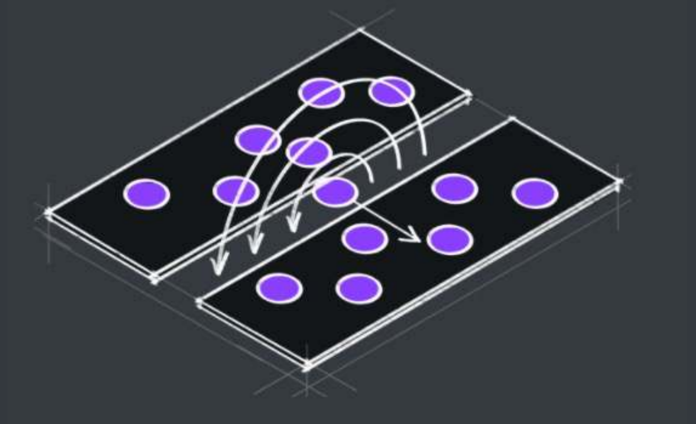
Why the Vision of Qiskit Metal

End-to-end automation

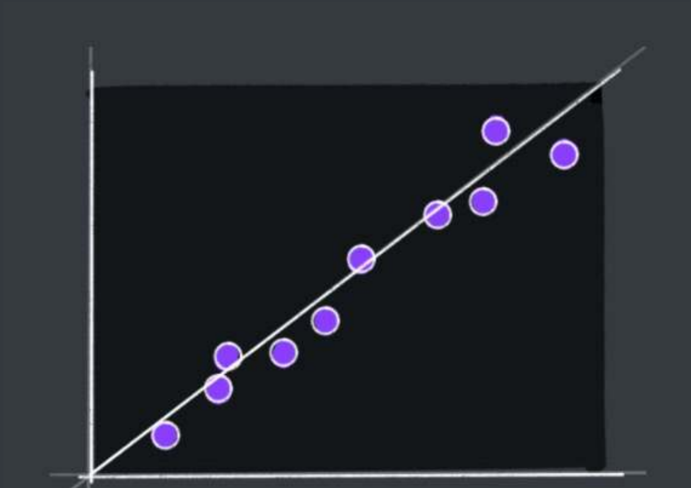
Flexible & extensible



Light-weight interoperability

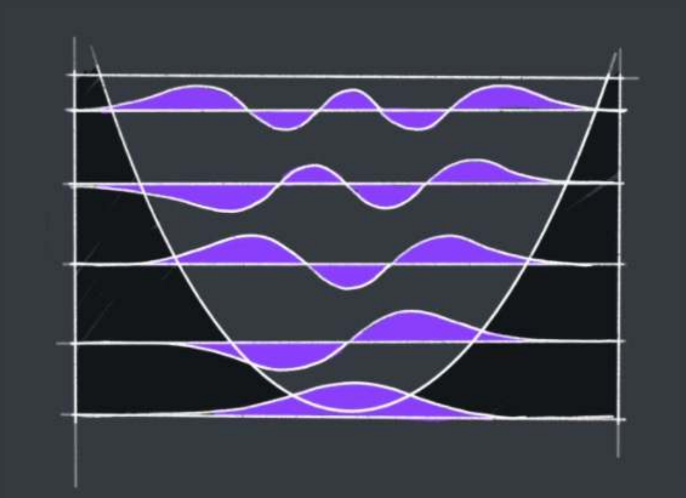


Experimentally tested



Library of components

Cutting edge resources



qiskit.org/metal

Thank you!

Building together

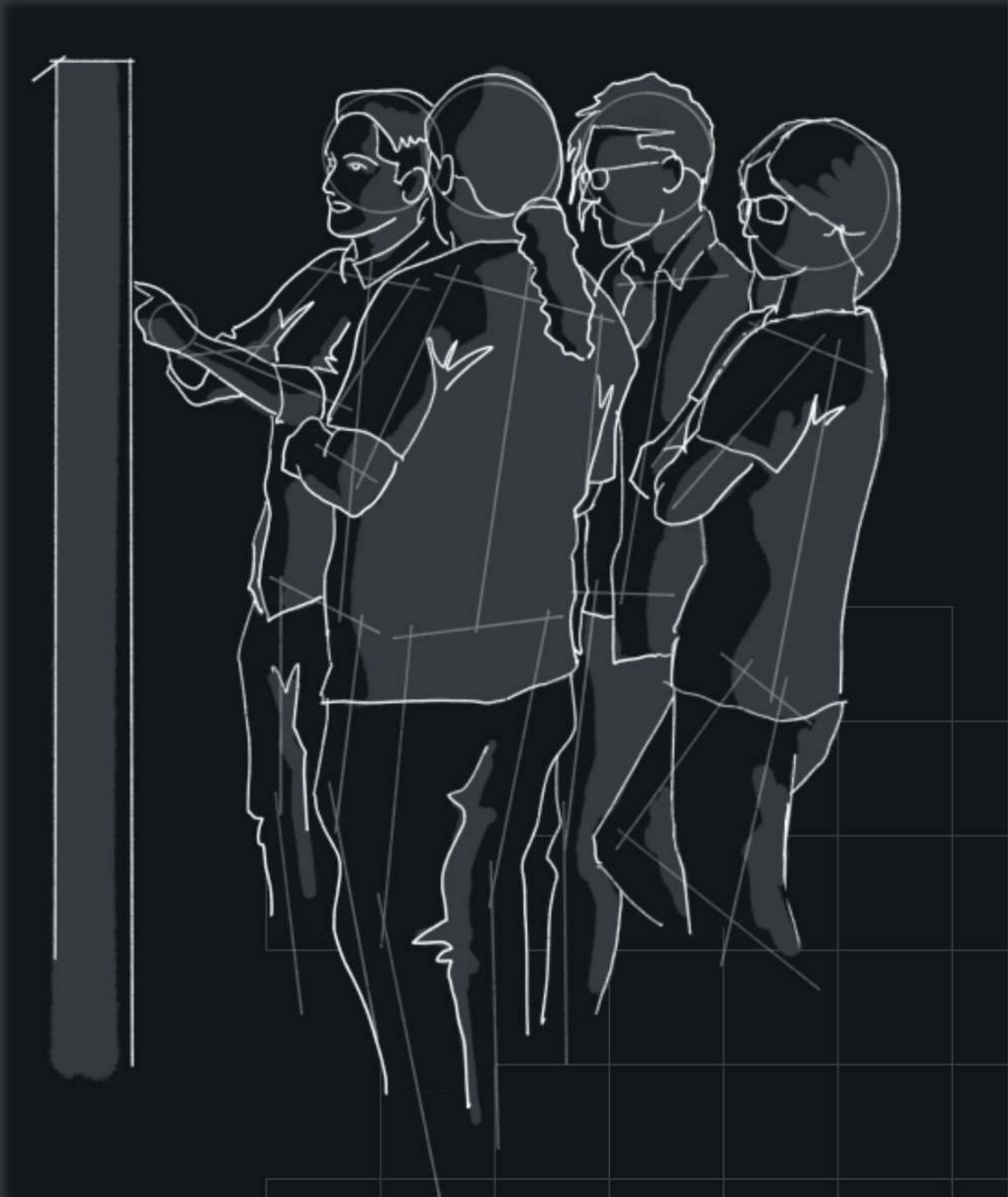
Call for community participation

Open source

Education

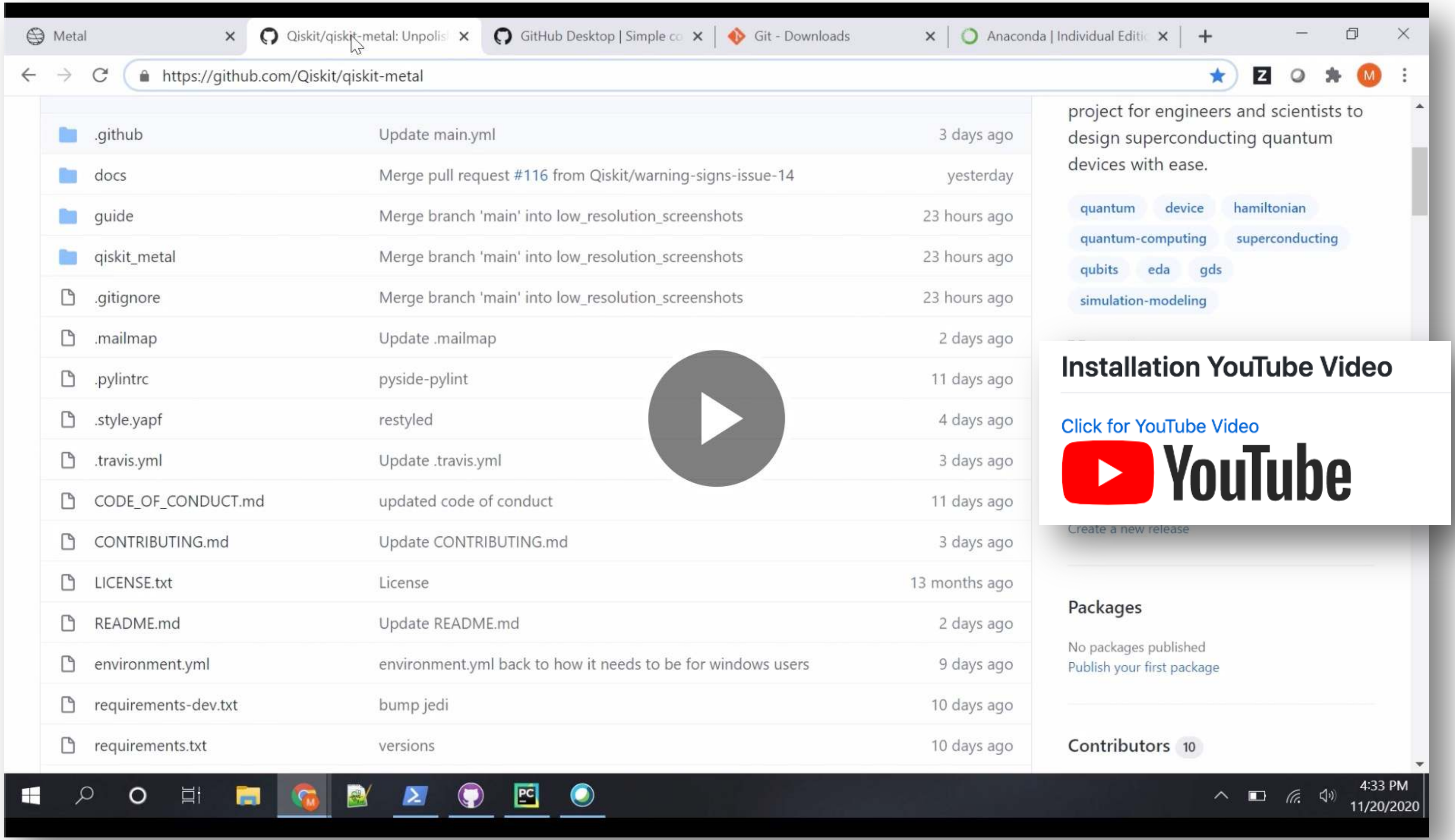
See summer school lectures 16-21 by Z. Minev from *Introduction to Quantum Computing and Quantum Hardware* and the *Qiskit Textbook*

qiskit.org/metal



@zlatko_minev

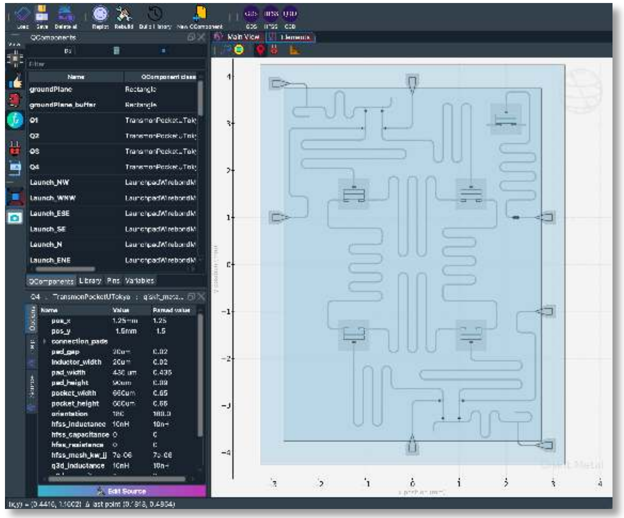
So, how do I get started?

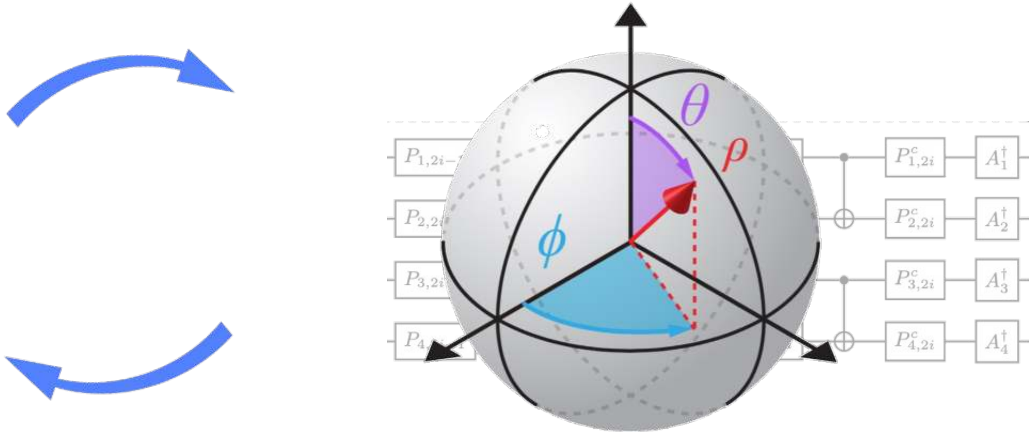


See github readme, or slack channel for link

Overview of Quantum Hardware Design

From *Fab-it* to *Qubit* — Qiskit Metal





Zlatko K. Minev

IBM Quantum

@zlatko_minev

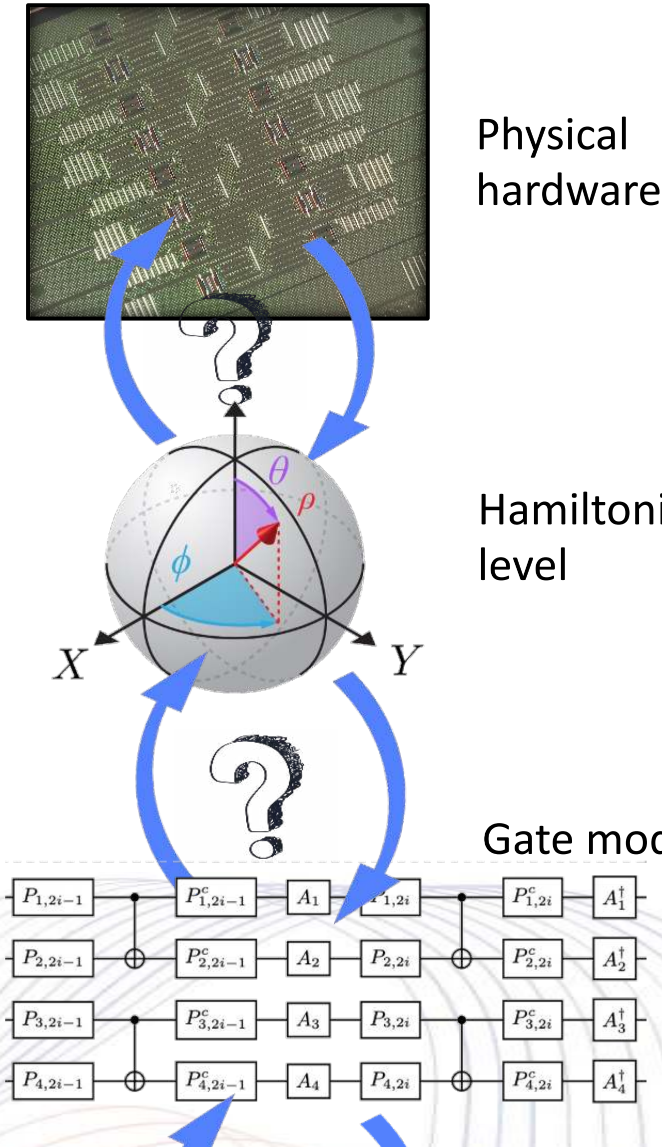
zlatko-minev.com

qiskit.org/metal

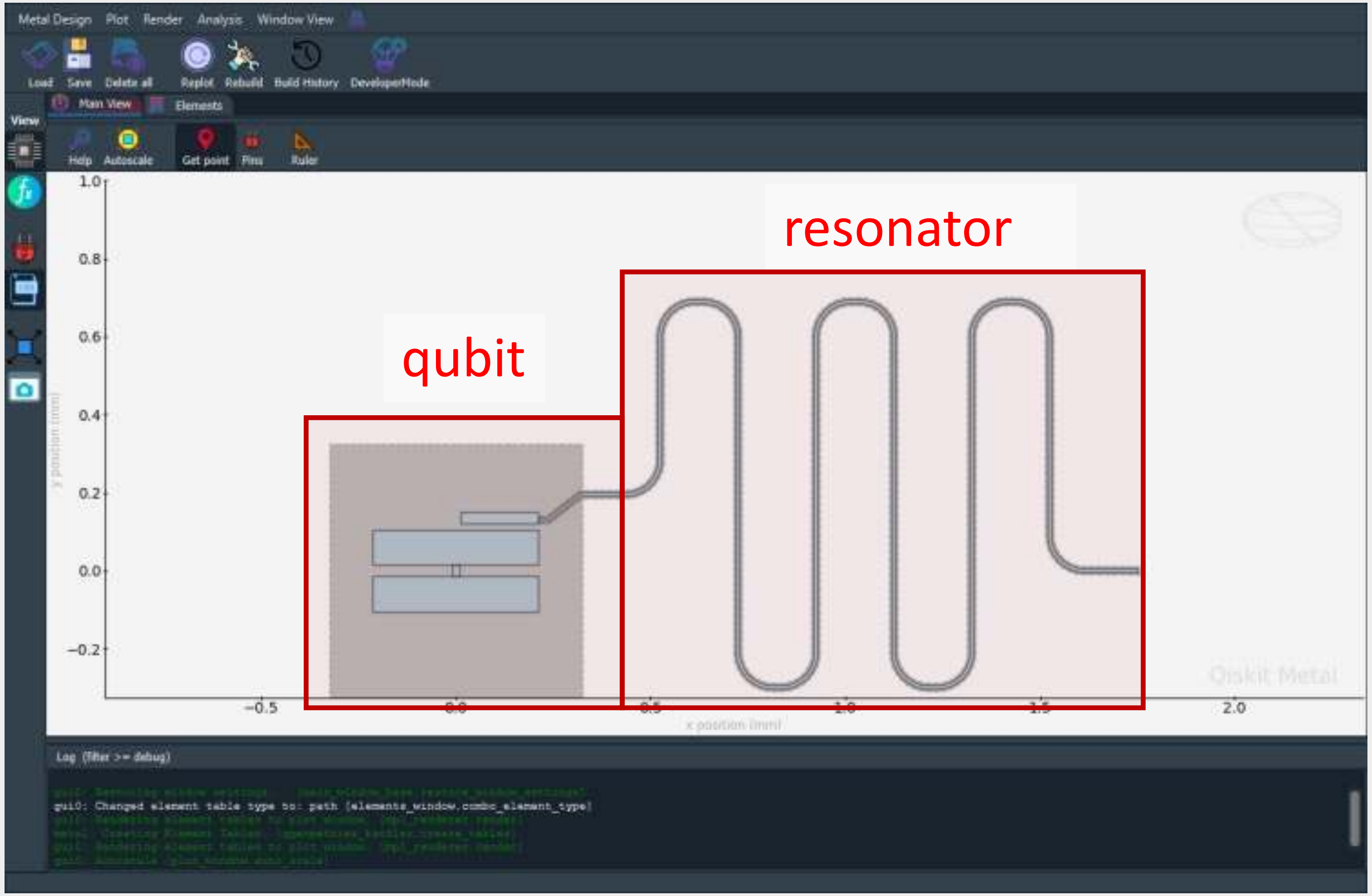
“Engineering is the purposeful use of science.”

STEVE SENTURIA

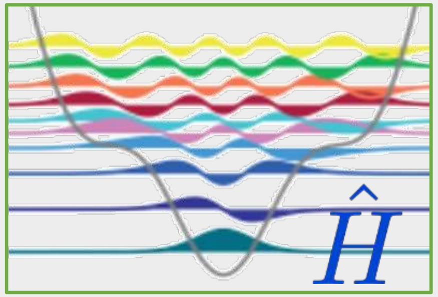
Quantum

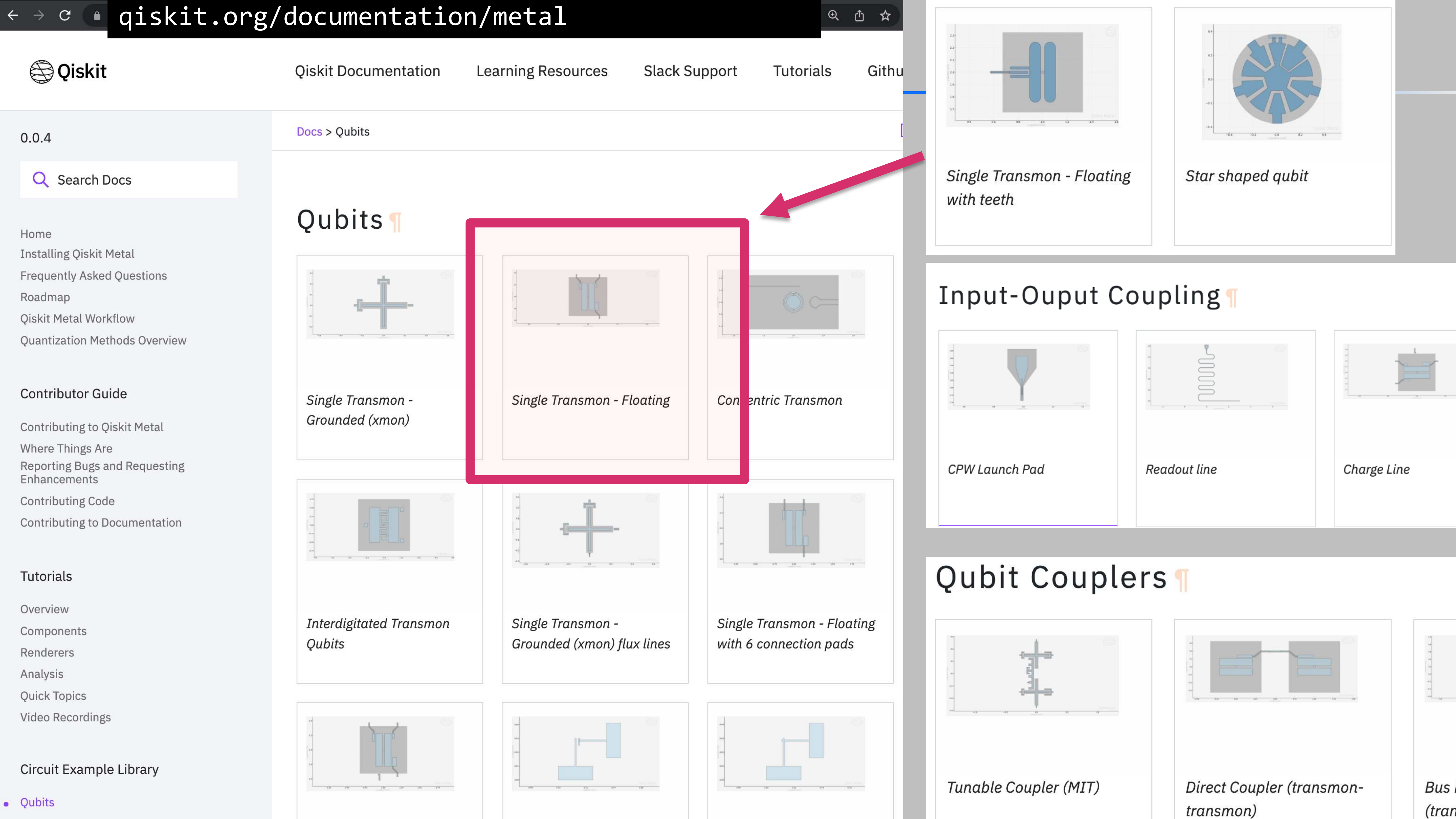


Simple overview example



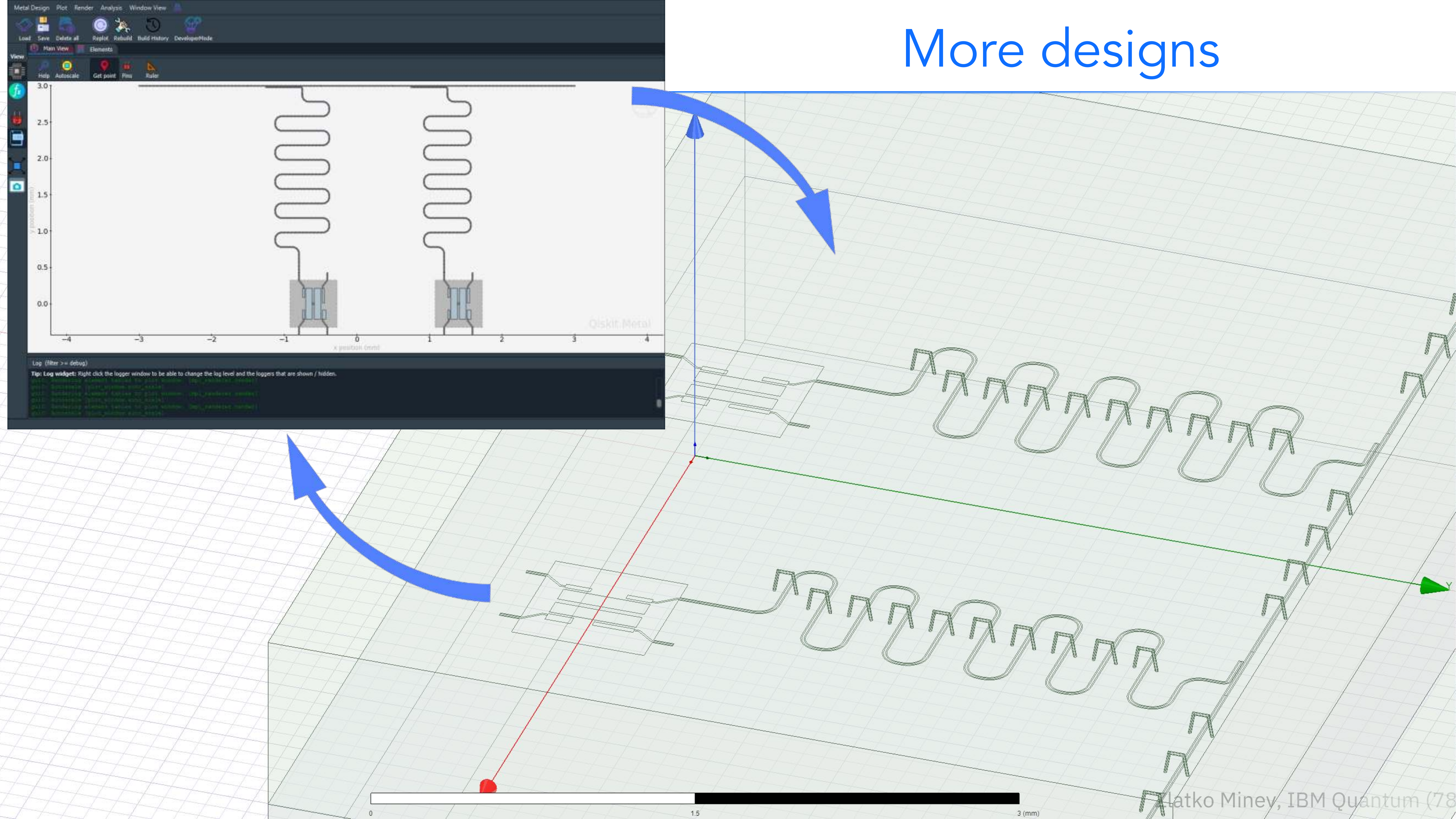
?





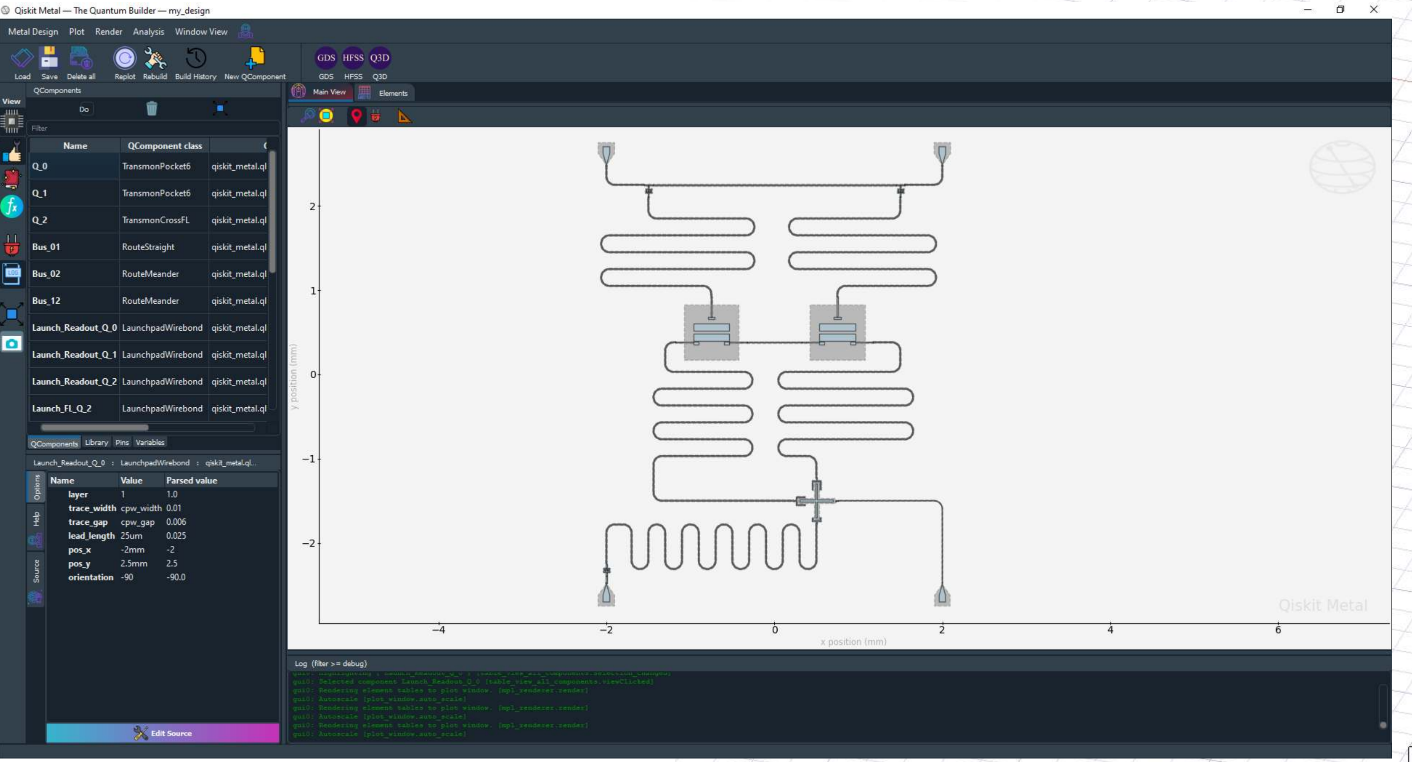
qiskit.org/documentation/metal
Qiskit
Qiskit Documentation
Learning Resources
Slack Support
Tutorials
0.0.4
Search Docs
Home
Installing Qiskit Metal
Frequently Asked Questions
Roadmap
Qiskit Metal Workflow
Quantization Methods Overview
Contributor Guide
Contributing to Qiskit Metal
Where Things Are
Reporting Bugs and Requesting Enhancements
Contributing Code
Contributing to Documentation
Tutorials
Overview
Components
Renderers
Analysis
Quick Topics
Video Recordings
Circuit Example Library
Qubits
Docs > Qubits
Qubits
Single Transmon - Grounded (xmon)
Single Transmon - Floating
Interdigitated Transmon Qubits
Single Transmon - Grounded (xmon) flux lines
Single Transmon - Floating with 6 connection pads
Single Transmon - Floating with teeth
Star shaped qubit
Input-Ouput Coupling
CPW Launch Pad
Readout line
Charge Line
Qubit Couplers
Tunable Coupler (MIT)
Direct Coupler (transmon-transmon)

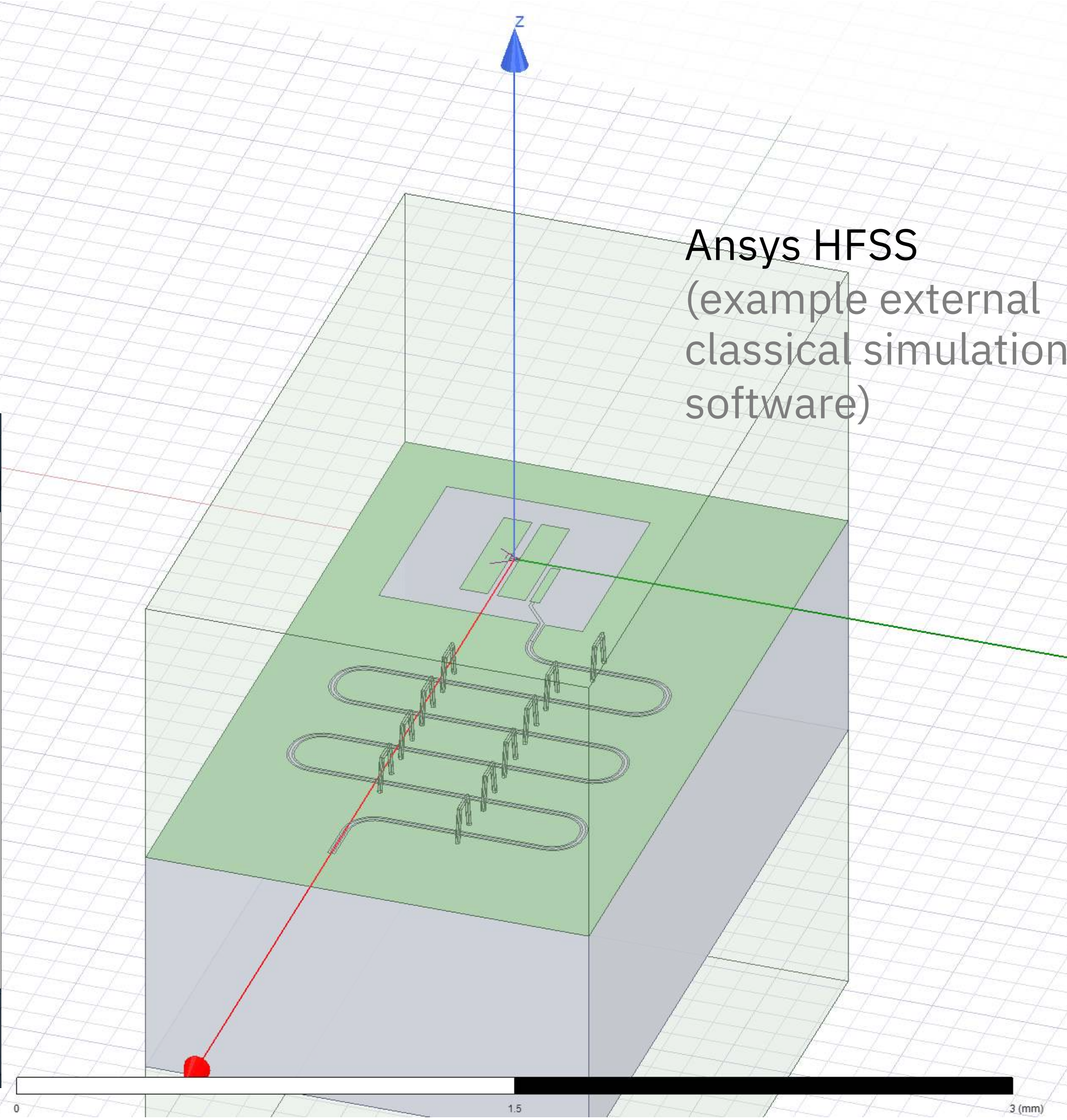
More designs



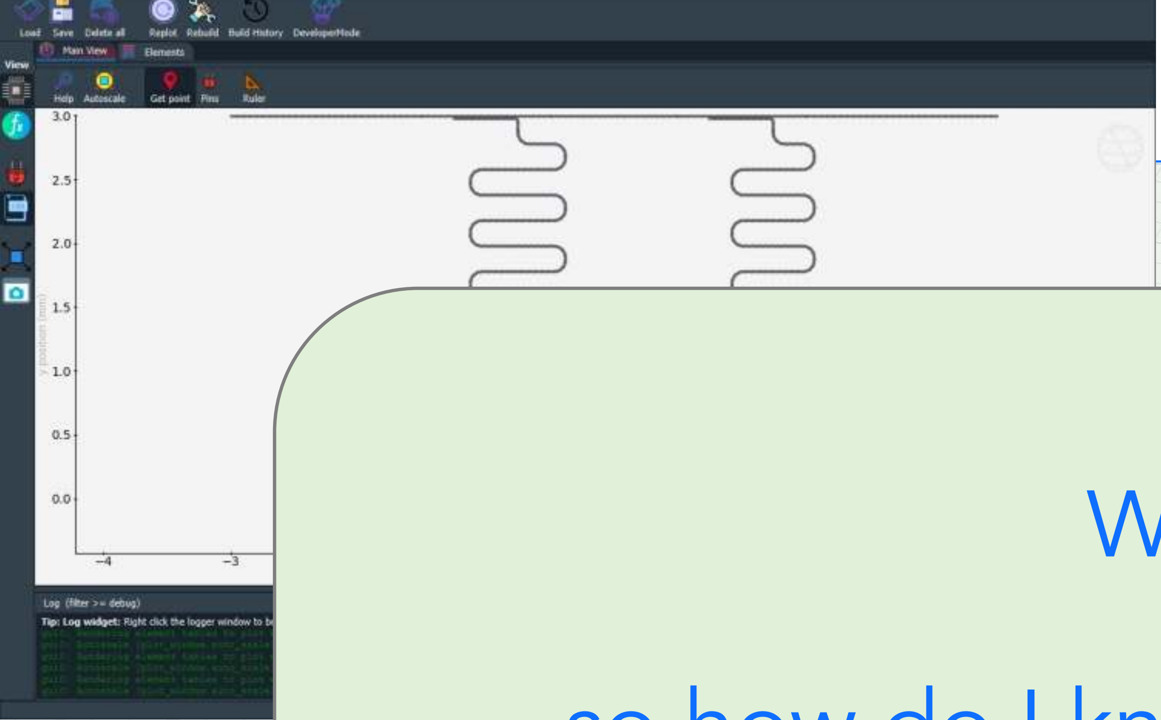
Automated with

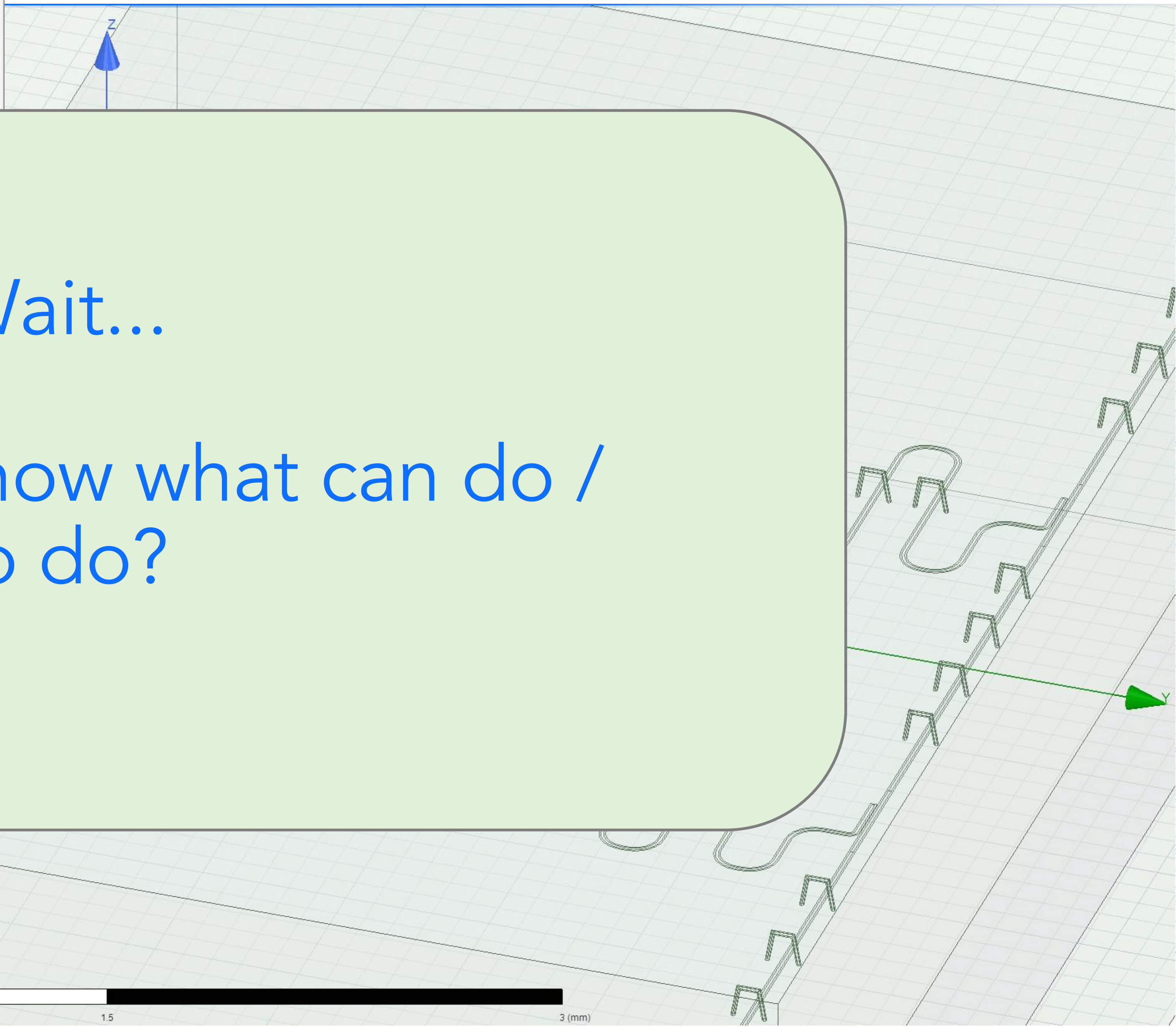
Qiskit | quantum device design





Ansys HFSS
(example external classical simulation software)

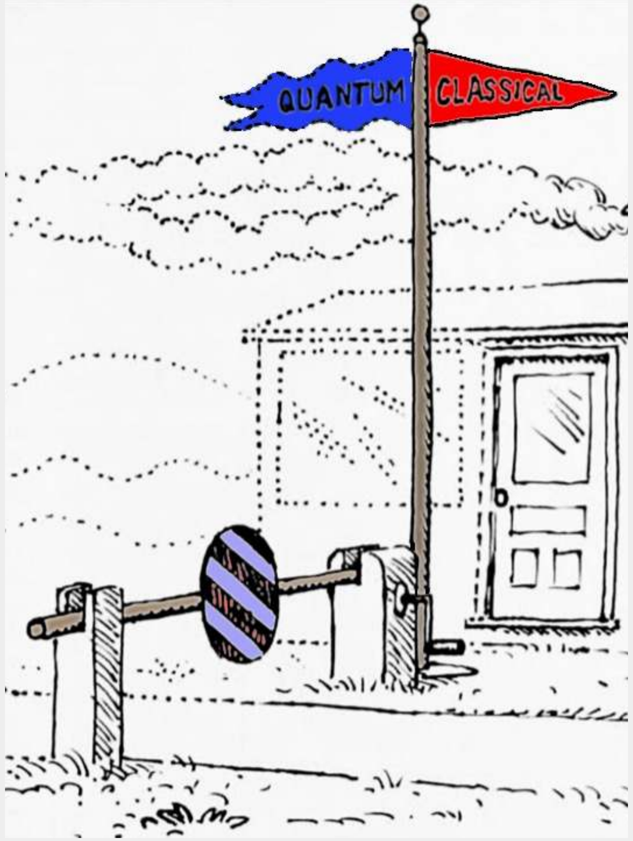




Wait...

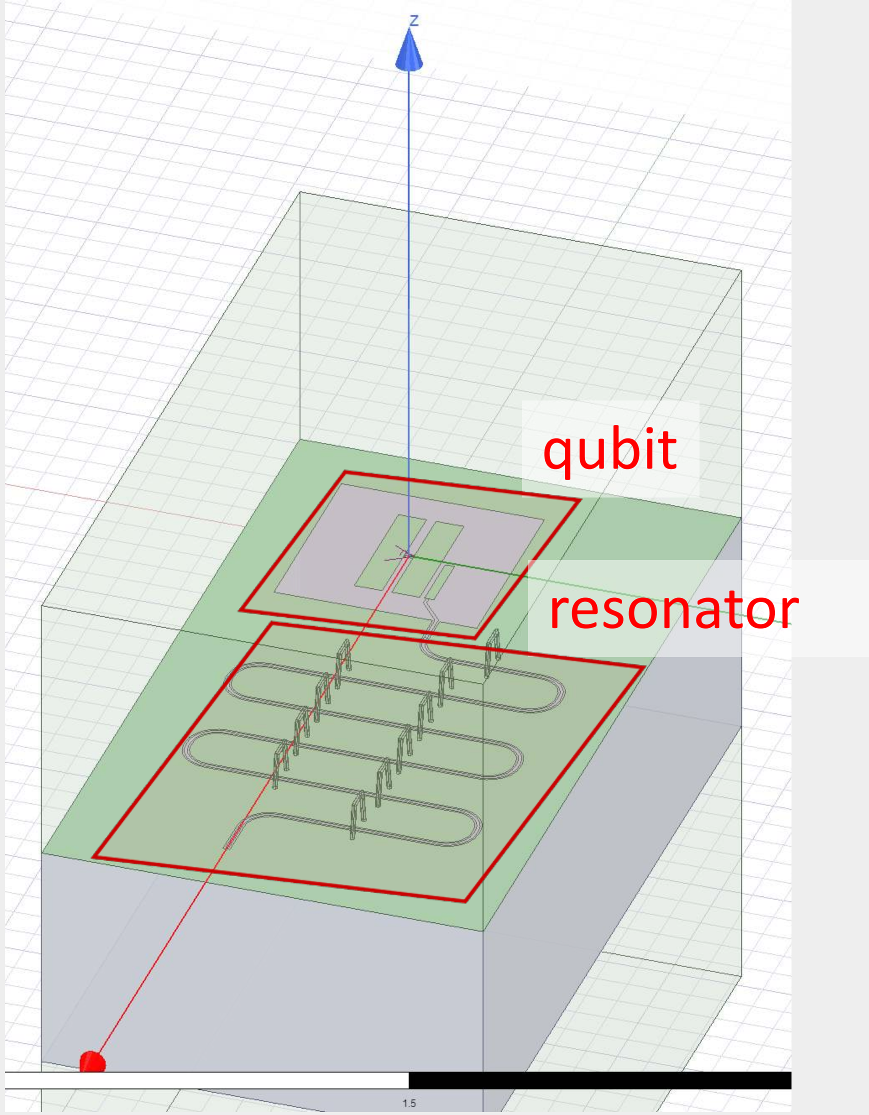
so how do I know what can do / to do?

Higher abstraction layers & Analysis

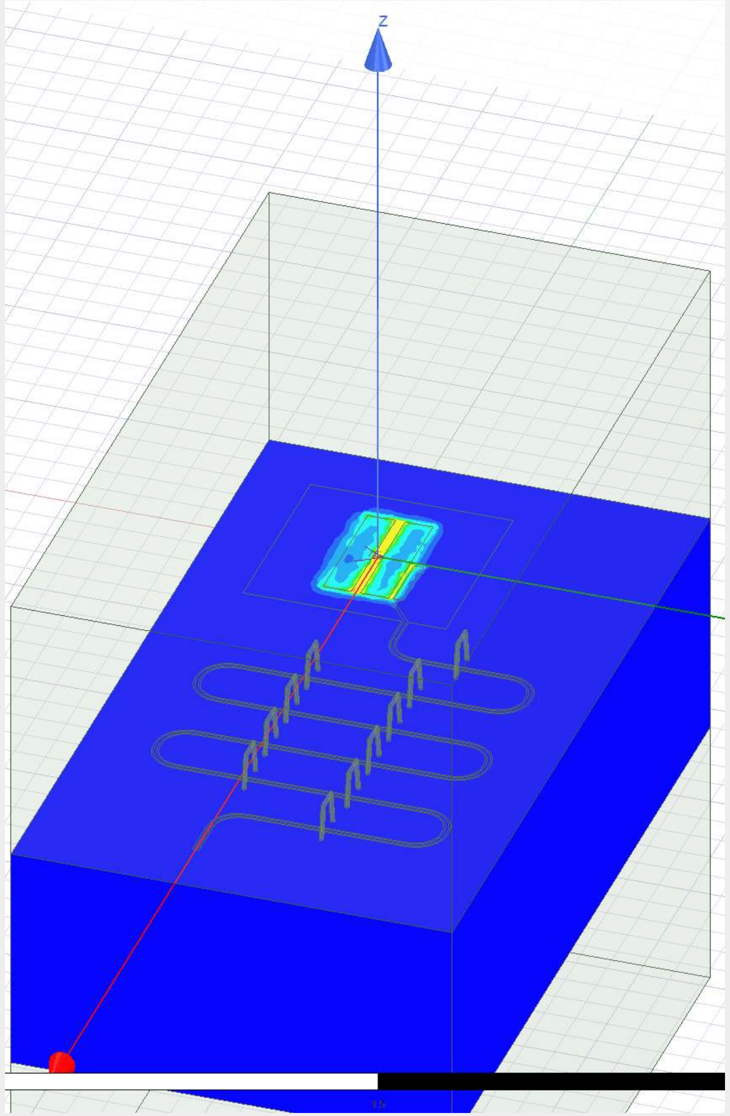


Modeling step 1: Classical E&M approximations

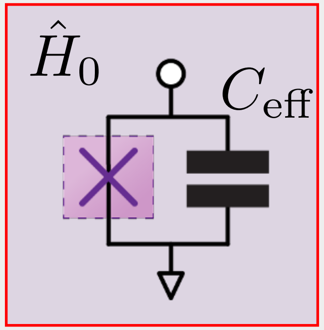
Physical (linear) model



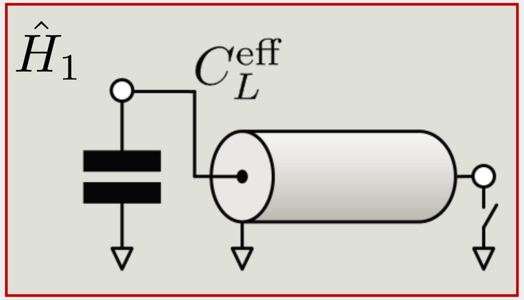
Eigenmodes



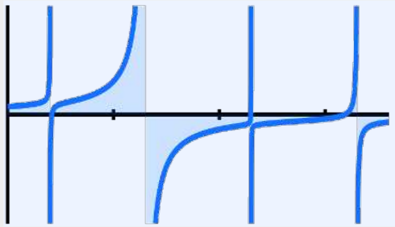
Quasi lumped models



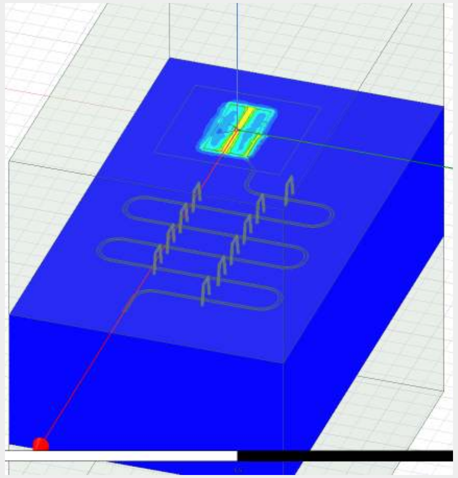
$$
\begin{array}{c}
 \\ n_0 \\ n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5
\end{array}
\begin{array}{c}
\begin{array}{cccccc} n_0 & n_1 & n_2 & n_3 & n_4 & n_5 \end{array} \\
\begin{pmatrix}
C_{0\Sigma} & -C_{01} & -C_{02} & -C_{03} & -C_{04} & -C_{05} \\
 & C_{1\Sigma} & -C_{12} & -C_{13} & -C_{14} & -C_{15} \\
 & & C_{2\Sigma} & -C_{23} & -C_{24} & -C_{25} \\
 & & & C_{3\Sigma} & -C_{34} & -C_{35} \\
 & & & & C_{4\Sigma} & -C_{45} \\
 & & & & & C_{5\Sigma}
\end{pmatrix}
\end{array}
$$

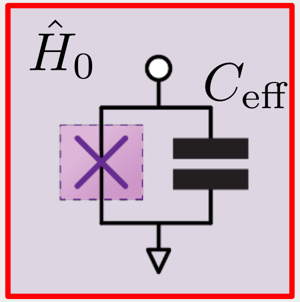


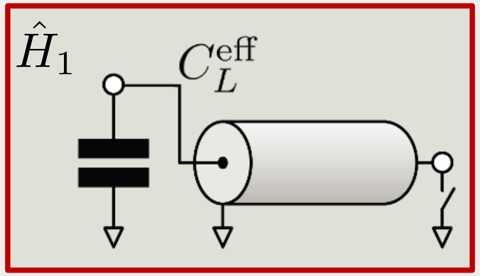
Impedance

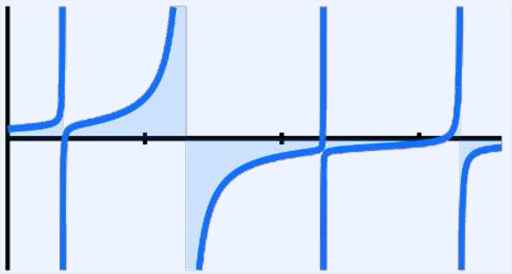


Modeling step 2: Quantize









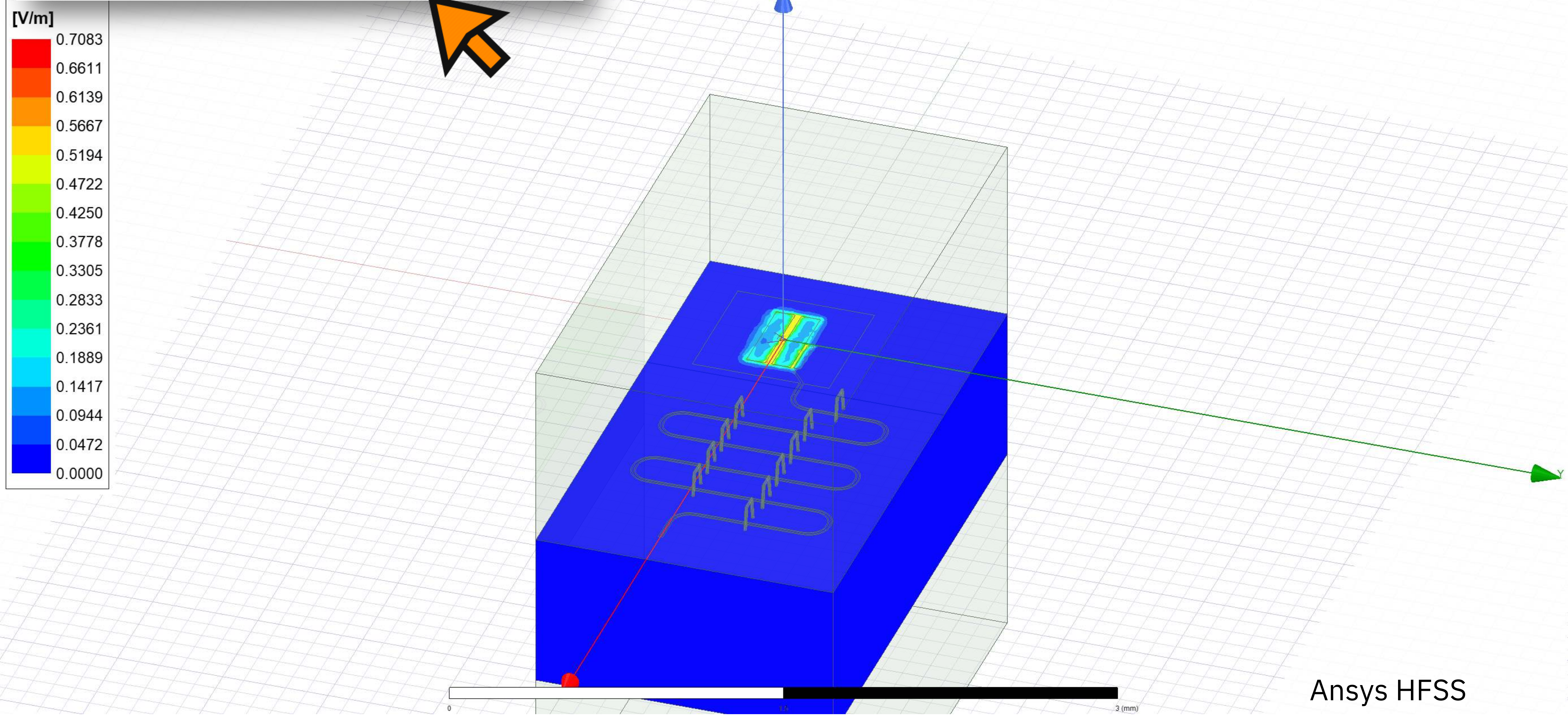
$$\hat{H}_{\mathrm{tot}}$$

$$\hat{H}_{\mathrm{tot}} = \hat{H}_{\mathrm{sys}} + \hat{H}_{\mathrm{int}}$$

$$\hat{H}_{\mathrm{tot}} = \hat{H}_{\mathrm{lin}} + \hat{H}_{\mathrm{nl}}$$

There's a description for every job!

```python
metal.analysis.lumped_model.analyze('Q1')
```

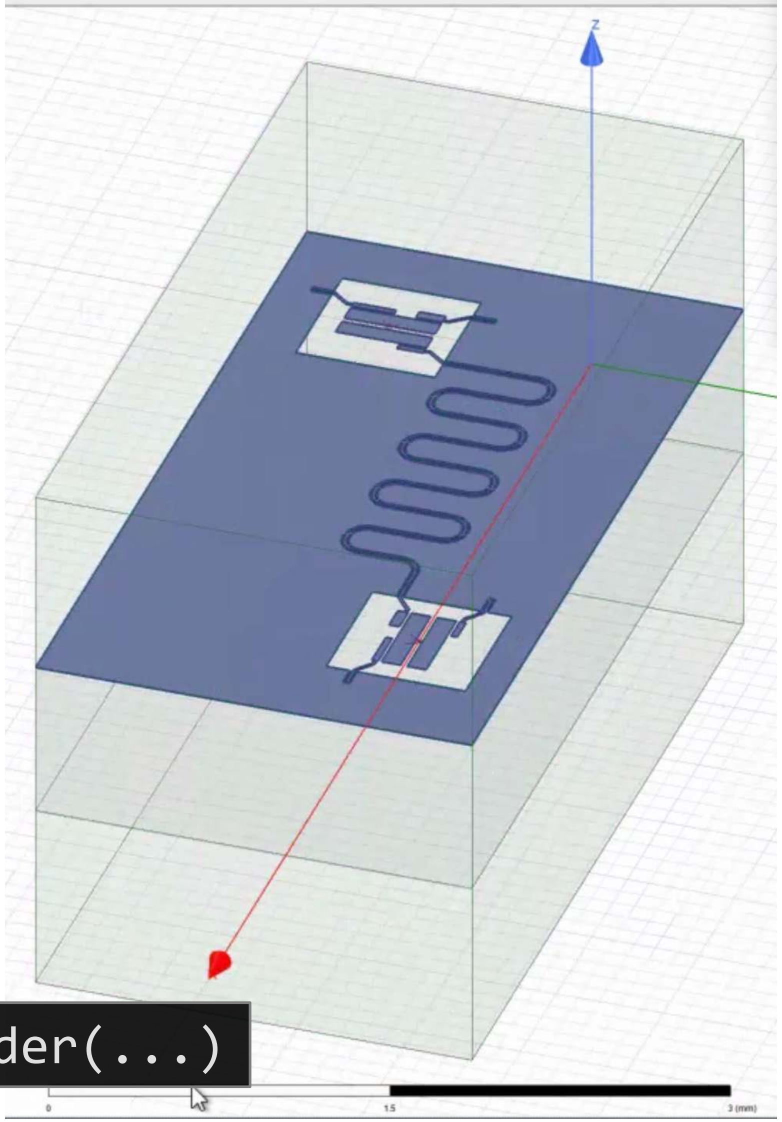


Ansys HFSS

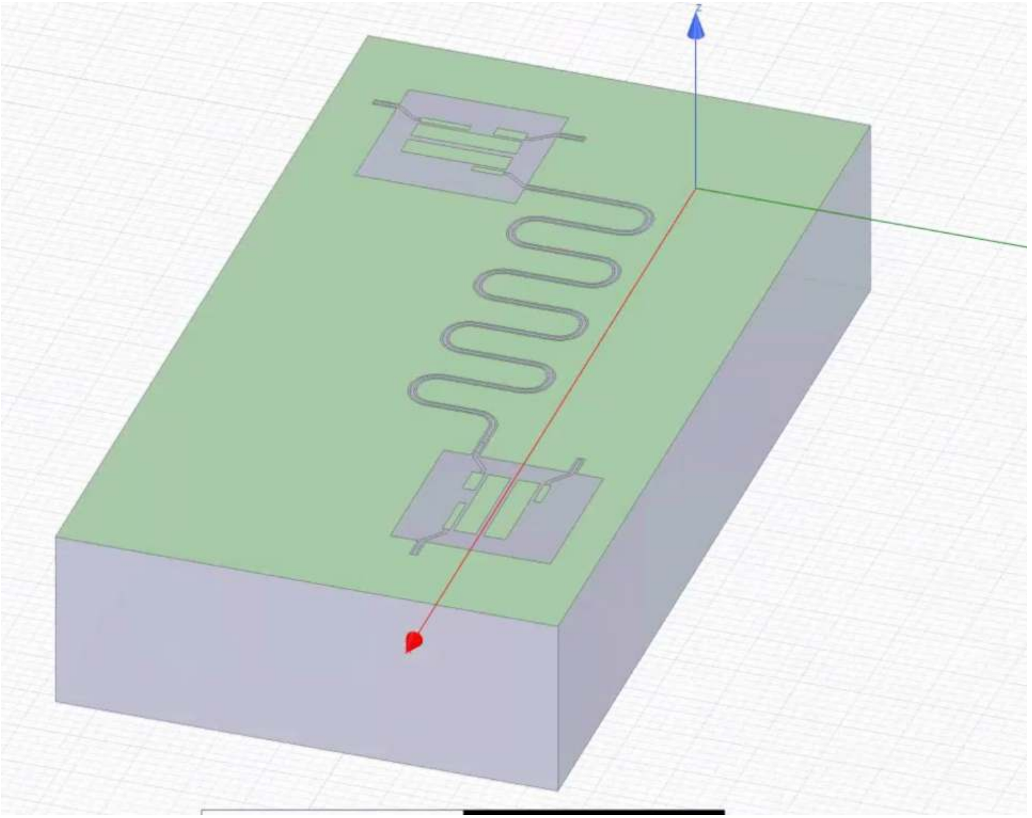
Qiskit Metal render & electromagnetic analysis

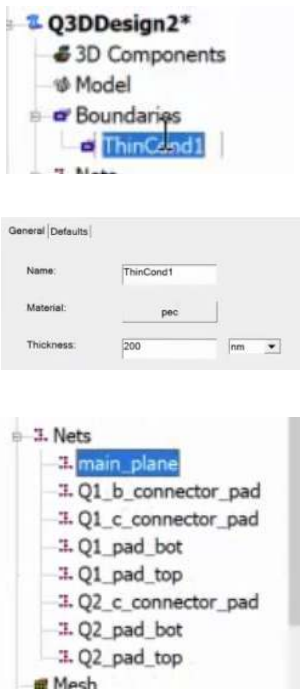
For example: Ansys HFSS

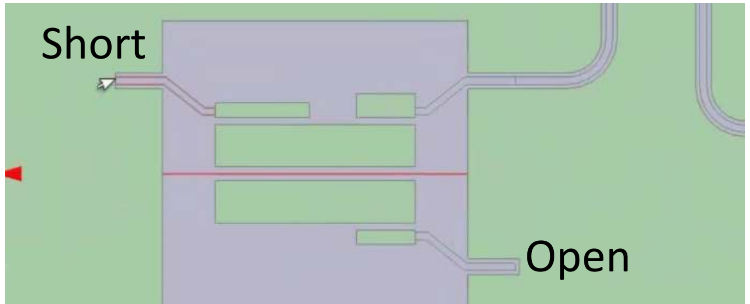
Q3D



```
design.render(...)
```

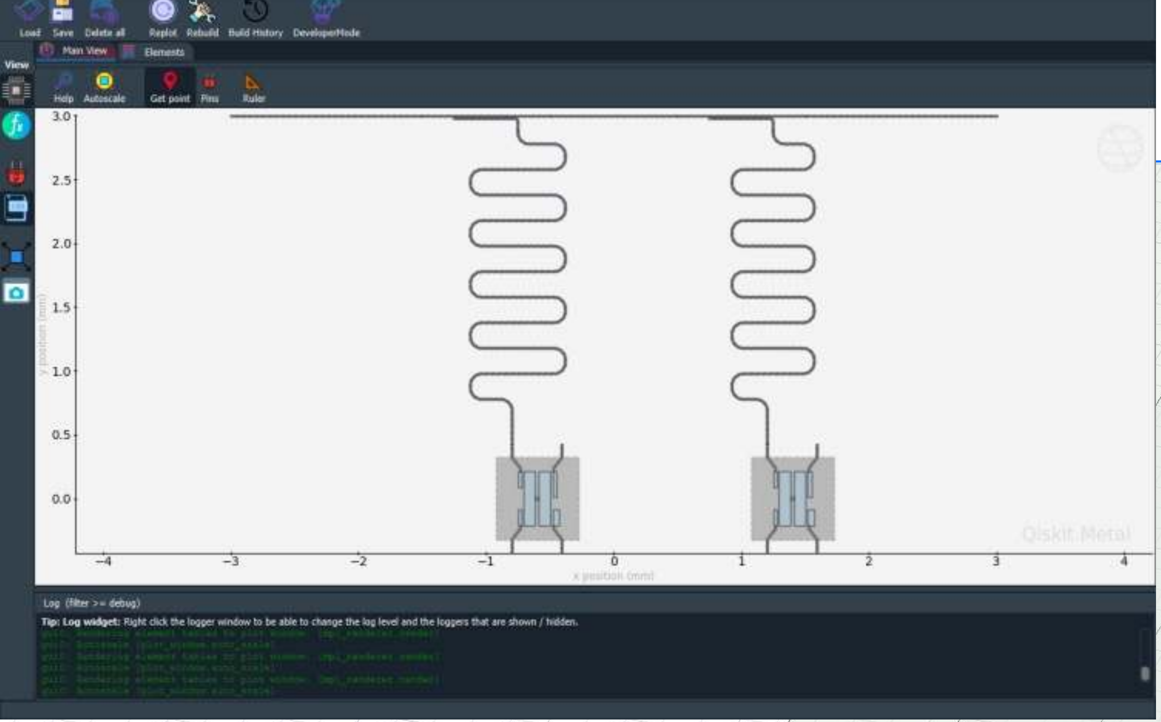


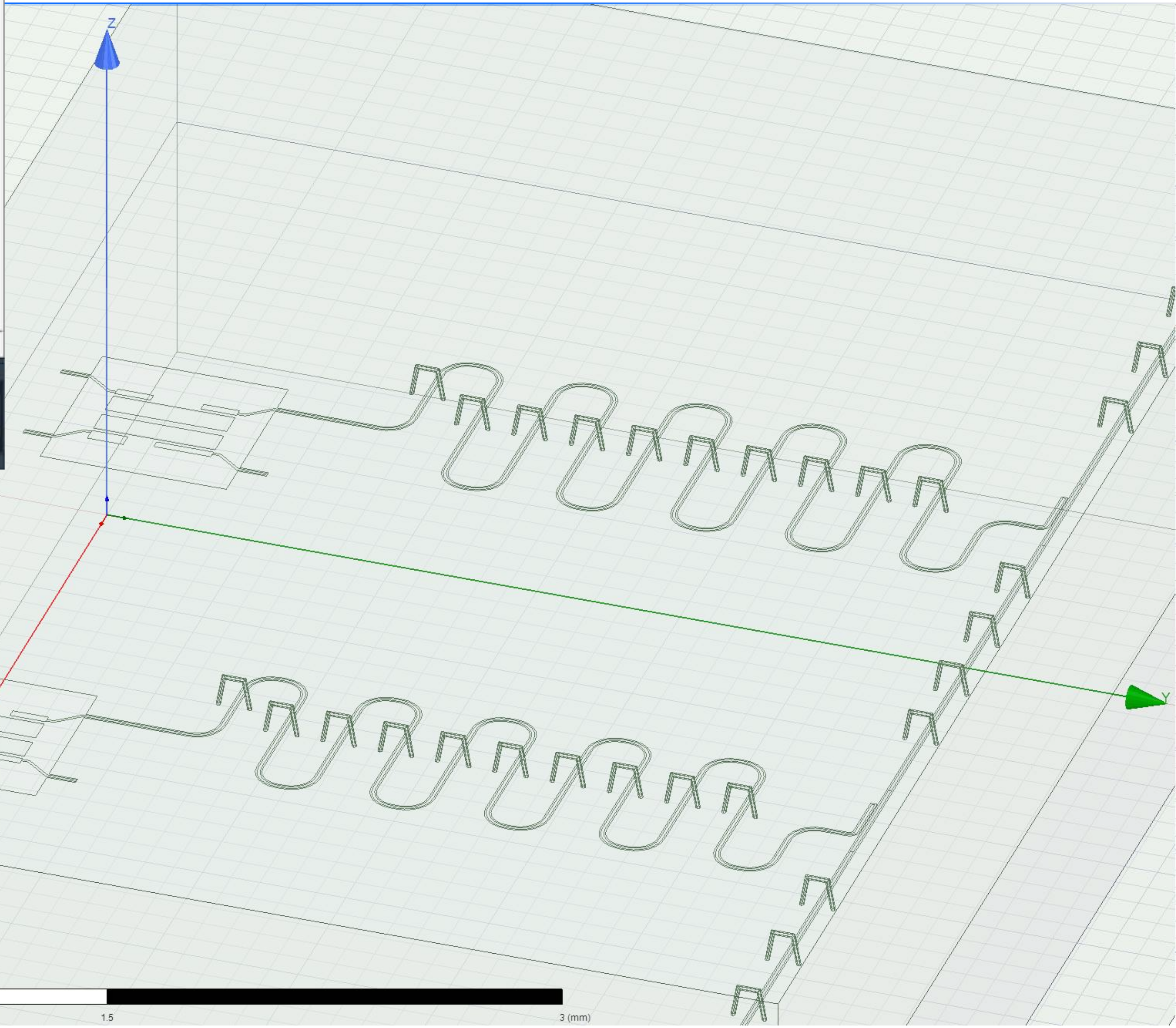




IBM **Quantum**

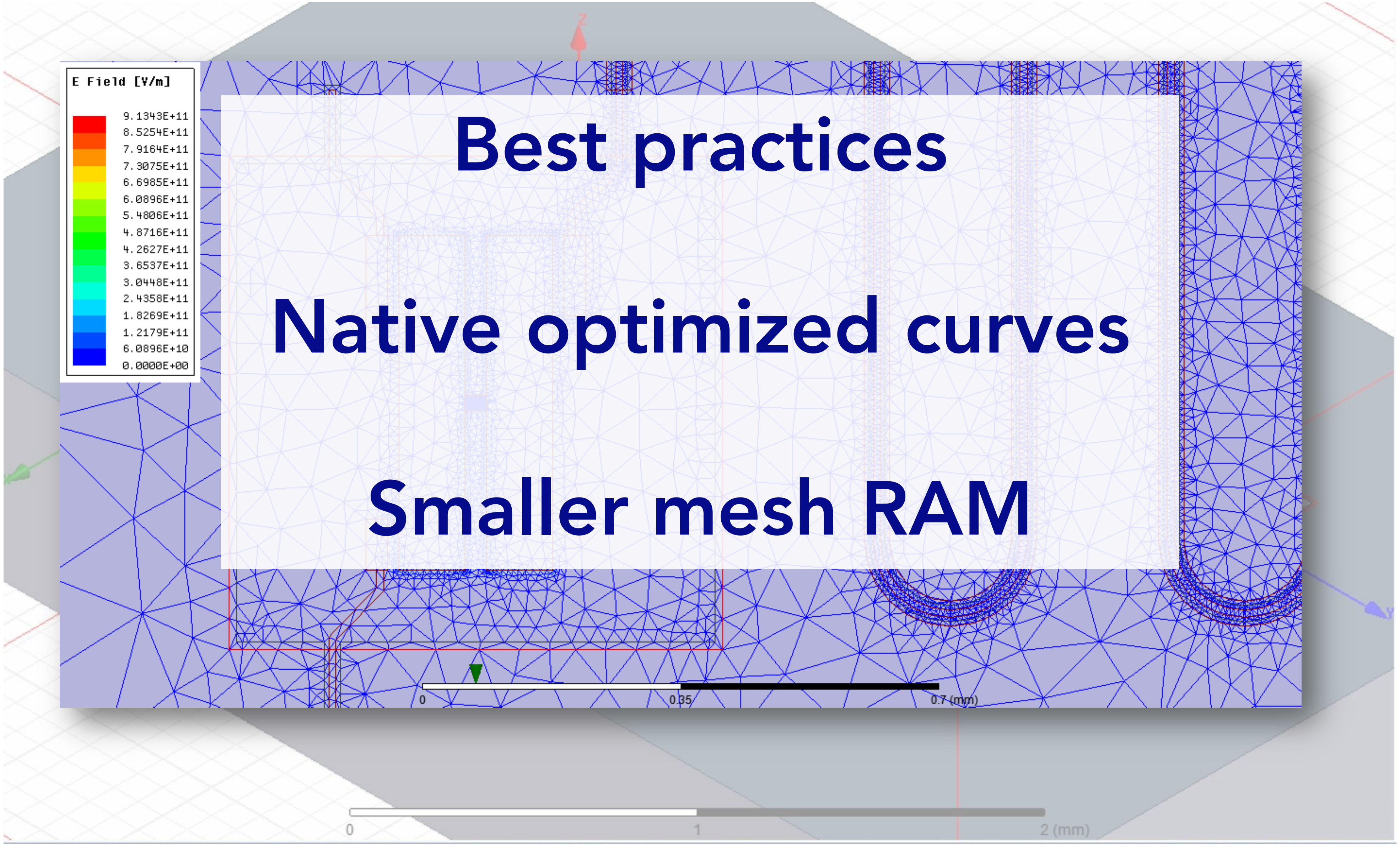
More complex designs





Qiskit Metal

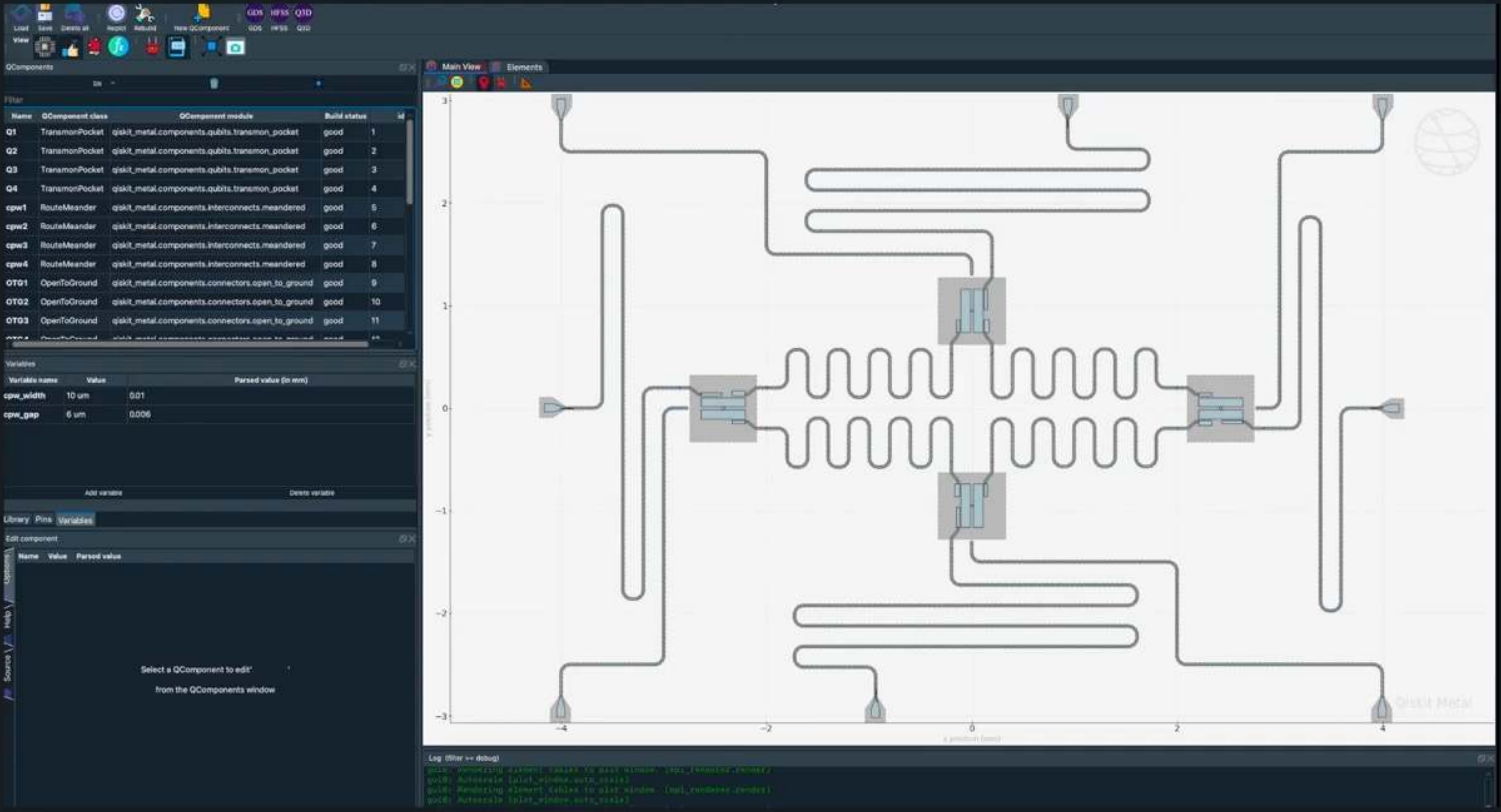
Sub-Circuit Analysis



Best practices

Native optimized curves

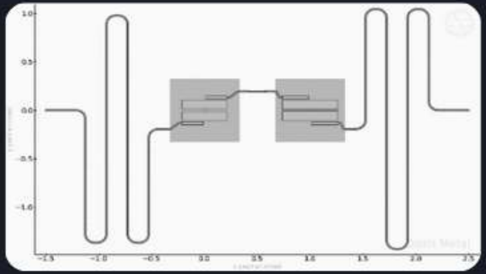
Smaller mesh RAM


Load
Save
Delete all
Replot
Rebuild
New QComponent
GDS
HFSS
Q3D
View
QComponents
Filter
Name
QComponent class
QComponent module
Build status
id
Q1 TransmonPocket qiskit_metal.components.qubits.transmon_pocket good 1
Q2 TransmonPocket qiskit_metal.components.qubits.transmon_pocket good 2
Q3 TransmonPocket qiskit_metal.components.qubits.transmon_pocket good 3
Q4 TransmonPocket qiskit_metal.components.qubits.transmon_pocket good 4
cpw1 RouteMeander qiskit_metal.components.interconnects.meandered good 5
cpw2 RouteMeander qiskit_metal.components.interconnects.meandered good 6
cpw3 RouteMeander qiskit_metal.components.interconnects.meandered good 7
cpw4 RouteMeander qiskit_metal.components.interconnects.meandered good 8
OTG1 OpenToGround qiskit_metal.components.connectors.open_to_ground good 9
OTG2 OpenToGround qiskit_metal.components.connectors.open_to_ground good 10
OTG3 OpenToGround qiskit_metal.components.connectors.open_to_ground good 11
Variables
Variable name
Value
Parsed value (in mm)
cpw_width 10 um 0.01
cpw_gap 6 um 0.006
Add variable
Delete variable
Library
Pins
Variables
Edit component
Name
Value
Parsed value
Select a QComponent to edit
from the QComponents window
Main View
Elements
Log (filter >= debug)

Qiskit Metal lumped model framework

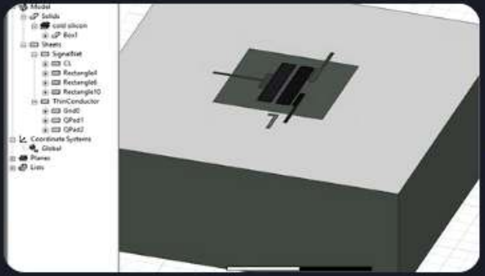
Qiskit | quantum device design

From layout to gates simulation in one notebook



Layout

Layout with Qiskit Metal's built-in EDA with either GUI or jupyter lab programitically



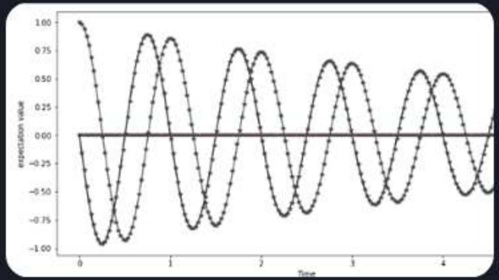
Electromagnetic simulation

Convenient interface with third-party EM simulation suites

$$\hat{H}_{\mathrm{full}} = \hat{H}_0 + \sum_{n=1}^{K} \hat{H}_n + \sum_{n=0}^{K} \sum_{m=n+1}^{K} \hat{H}_{nm}$$

Quantum analysis

Lumped model analysis and Hamiltonian analysis



Pulse simulation

Convenient Hamiltonian interface with Qutip, Qiskit Dynamics, seQuencing for time evolution simulation load transmon cell Q3d simulation results ¶

```
path1 = './Q1_TwoTransmon_CapMatrix.txt'
ta_mat, _, _, _ = load_q3d_capacitance_matrix(path1, _disp=False)

path2 = './Q2_TwoTransmon_CapMatrix.txt'
tb_mat, _, _, _ = load_q3d_capacitance_matrix(path2, _disp=False)
```

Create LOM cells from capacitance matrices

```
# cell 1: transmon Alice cell

opt1 = dict(
    node_rename = {'coupler_connector_pad_Q1': 'coupling', 'readout_connector_pad_Q1': 'readout_alice'},
    cap_mat = ta_mat,
    ind_dict = {('pad_top_Q1', 'pad_bot_Q1'):10},
    jj_dict = {('pad_top_Q1', 'pad_bot_Q1'):'j1'},
    cj_dict = {('pad_top_Q1', 'pad_bot_Q1'):2}, # 
)
cell_1 = Cell(opt1)


# cell 2: transmon Bob cell
opt2 = dict(
    node_rename = {'coupler_connector_pad_Q2': 'cou
    cap_mat = tb_mat,
    ind_dict = {('pad_top_Q2', 'pad_bot_Q2'): 12},
    jj_dict = {('pad_top_Q2', 'pad_bot_Q2'):'j2'},
    cj_dict = {('pad_top_Q2', 'pad_bot_Q2'):2}, # 
)
cell_2 = Cell(opt2)
```

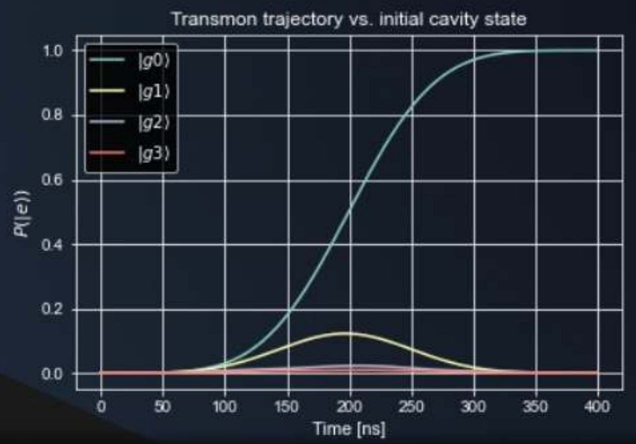
Make subsystems

```
# subsystem 1: transmon Alice
transmon_alice = Subsystem(name='transmon_alice', sys_type='TRANSMON', nodes=['j1'])


# subsystem 2: transmon Bob
transmon_bob = Subsystem(name='transmon_bob', sys_type='TRANSMON', nodes=['j2'])


# subsystem 3: Alice readout resonator
q_opts = dict(
    f_res = 8, # resonator dressed frequency in GHz
    Z0 = 50, # characteristic impedance in Ohm
    vp = 0.404314 * c_light # phase velocity
)
res_alice = Subsystem(name='readout_alice', sys_type='TL_RESONATOR', nodes=['readout_alice'], q_opts=q_opts)


# subsystem 4: Bob readout resonator
q_opts = dict(
    f_res = 7.6, # resonator dressed frequency in GHz
    Z0 = 50, # characteristic impedance in Ohm
    vp = 0.404314 * c_light # phase velocity
)
res_bob = Subsystem(name='readout_bob', sys_type='TL_RESONATOR', nodes=['readout_bob'], q_opts=q_opts)
```

Creat the composite system from the cells and the subsystems

```
composite_sys = CompositeSystem(
    subsystems=[transmon_alice, transmon_bob, res_alice, res_bob],
    cells=[cell_1, cell_2],
    grd_node='ground_main_plane',
    nodes_force_keep=['readout_alice', 'readout_bob']
)
```

	coupler_connector_pad_Q1	ground_main_plane	pad_bot_Q1	pad_top_Q1	readout_connector_pad_Q1
coupler_connector_pad_Q1	59.19879	-37.28461	-2.00818	-19.10774	-0.22976
ground_main_plane	-37.28461	246.32877	-39.78914	-39.86444	-37.29686
pad_bot_Q1	-2.00818	-39.78914	93.05074	-30.61038	-19.21994
pad_top_Q1	-19.10774	-39.86444	-30.61038	92.99428	-2.00897
readout_connector_pad_Q1	-0.22976	-37.29686	-19.21994	-2.00897	59.32747



Generate the hilberspace from the composite system, leveraging the scqubits package

```
hilbertspace = composite_sys.create_hilbertspace()
hilbertspace = composite_sys.add_interaction()
```

Print the results

```
hamiltonian_results = composite_sys.hamiltonian_results(hilbertspace, evals_count=30)
```

```
Finished eigensystem.

system frequencies in GHz:
--------------------------
{'transmon_alice': 6.053360688806868, 'transmon_bob': 4.7989883222888094, 'readout_alice': 8.0090548

Chi matrices in MHz
--------------------------
                transmon_alice  transmon_bob  readout_alice  readout_bob
transmon_alice     -353.239816     -0.542895      -4.132854    -0.003120
transmon_bob         -0.542895   -263.940098      -0.001154    -1.460416
readout_alice        -4.132854     -0.001154       4.283111    -0.000017
readout_bob          -0.003120     -1.460416      -0.000017     3.829744
```

From layout to time evolution simulation in one simple notebook

Time evolution simulation

```
system = lom_composite_sys_to_seq_sys(composite_sys, hilbertspace, levels=[3, 3, 10, 10])
alice = system.modes[1]
```

```
Finished eigensystem.
```

```
selective_sigma = 100 # ns

# tune selective qubit pulse using Rabi
with system.use_modes([alice]):
    with alice.temporarily_set(gaussian_pulse__sigma=selective_sigma):
        _, _, selective_qubit_amp = tune_rabi(
            system, system.fock(transmon_alice=0, transmon_bob=0, readout_alice=0, readout_bob
        )
```

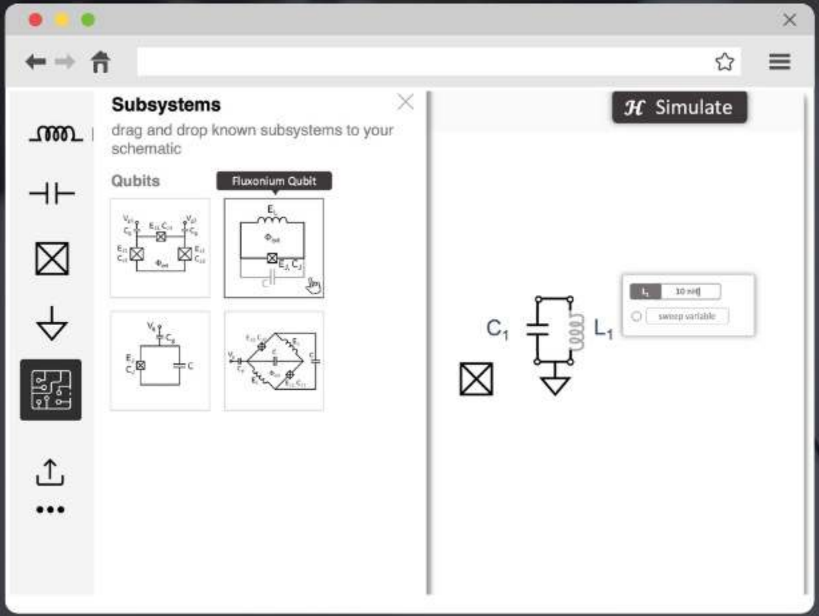
```
100%|
| 51/51 [00:01<00:00, 41.64it/s]
```

```
def selective_rotation(qubit, angle, phase=0, detune=0, sigma=selective_sigma):
    with qubit.gaussian_pulse.temporarily_set(sigma=sigma, amp=selective_qubit_amp):
        qubit.rotate(np.pi, phase, detune=detune)
```

Apply a selective pi pulse that is resonant with the qubit when the cavity is in |0>.

```
init_states = [
    (f'$|g{n}\\rangle$', system.fock(transmon_alice=0, readout_alice=n)) for n in range(4)
```

Qiskit | quantum device design



Near to medium-term action items

Expand adoption

Metal on the Cloud/Web app, GUI 2.0, Super-user adoptions

Enhance quantum analysis

LOM 2.0, time evolution simulation, impedance model

Boost core tech

Modularity of simulation and analysis, Keysight renderer