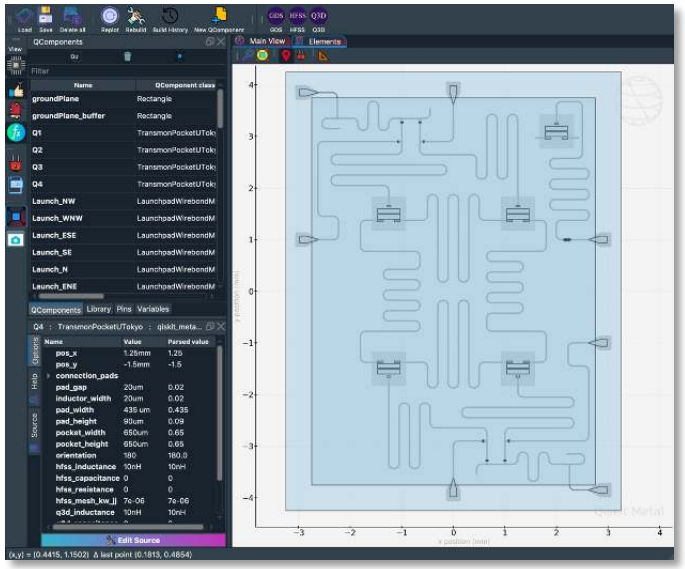
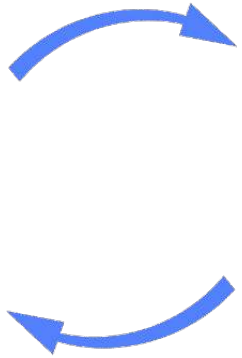
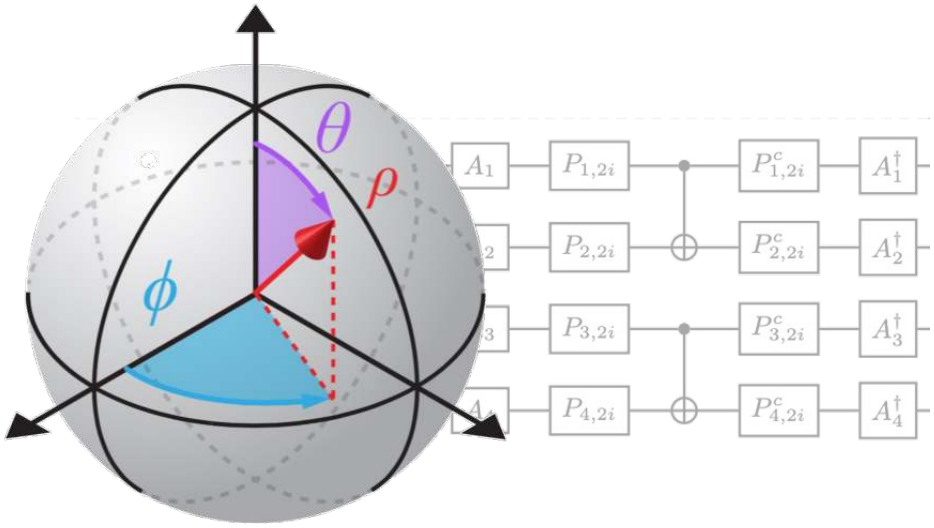
Overview of Quantum Hardware Design

Energy, Circuits, and Metal: Qiskit Metal





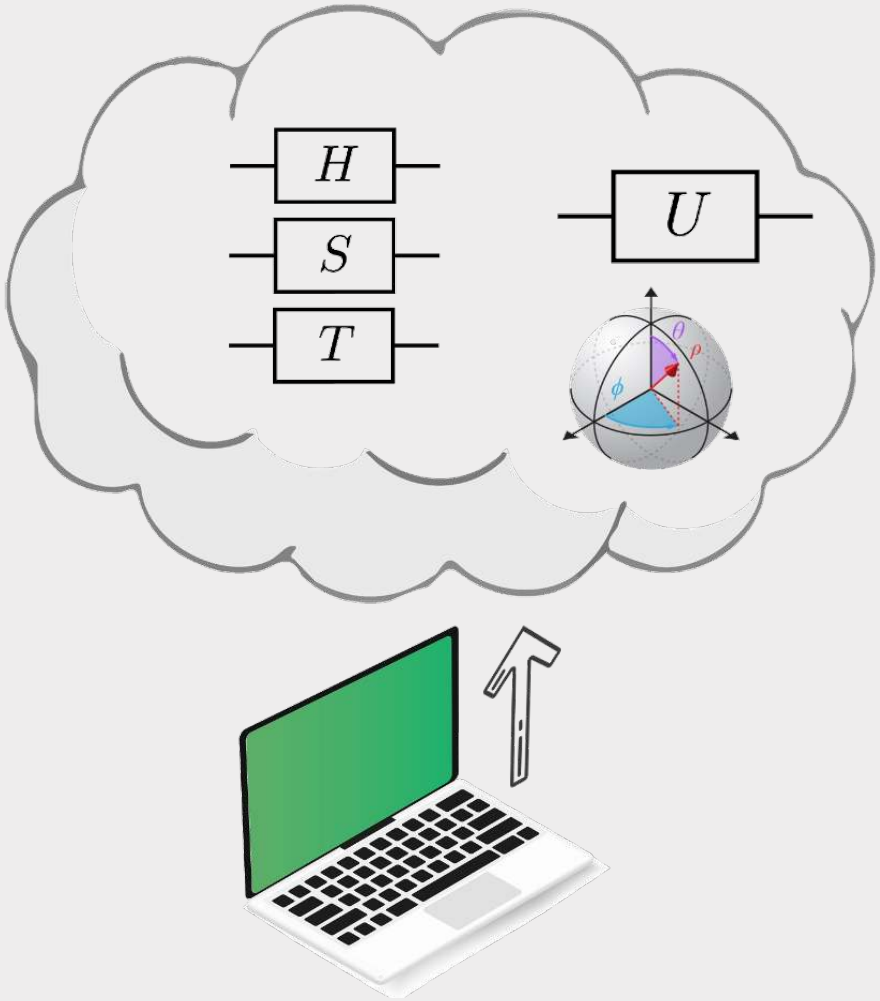


Zlatko K. Minev

IBM Quantum

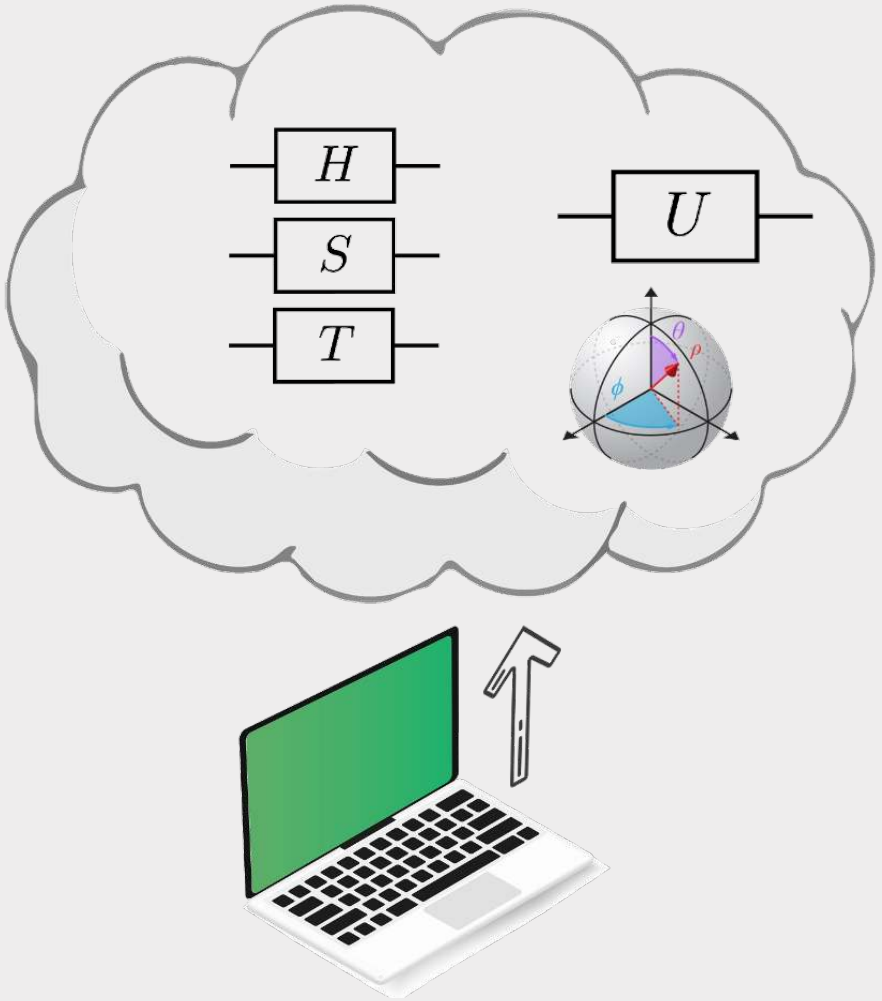
What is possible to demonstrate with quantum processors today?

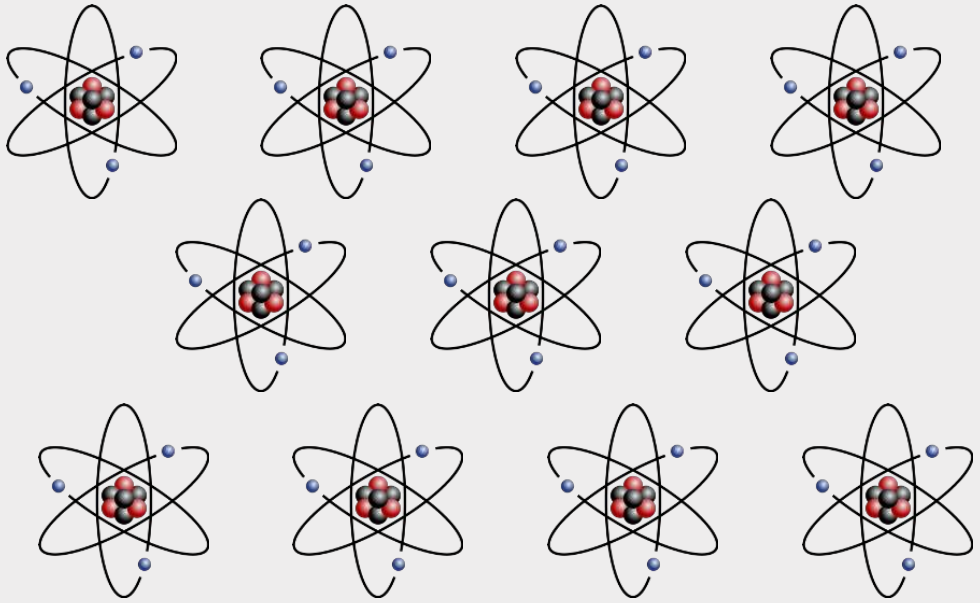
- *Qiskit Leap* research team -



Laptop image: rawpixel.com

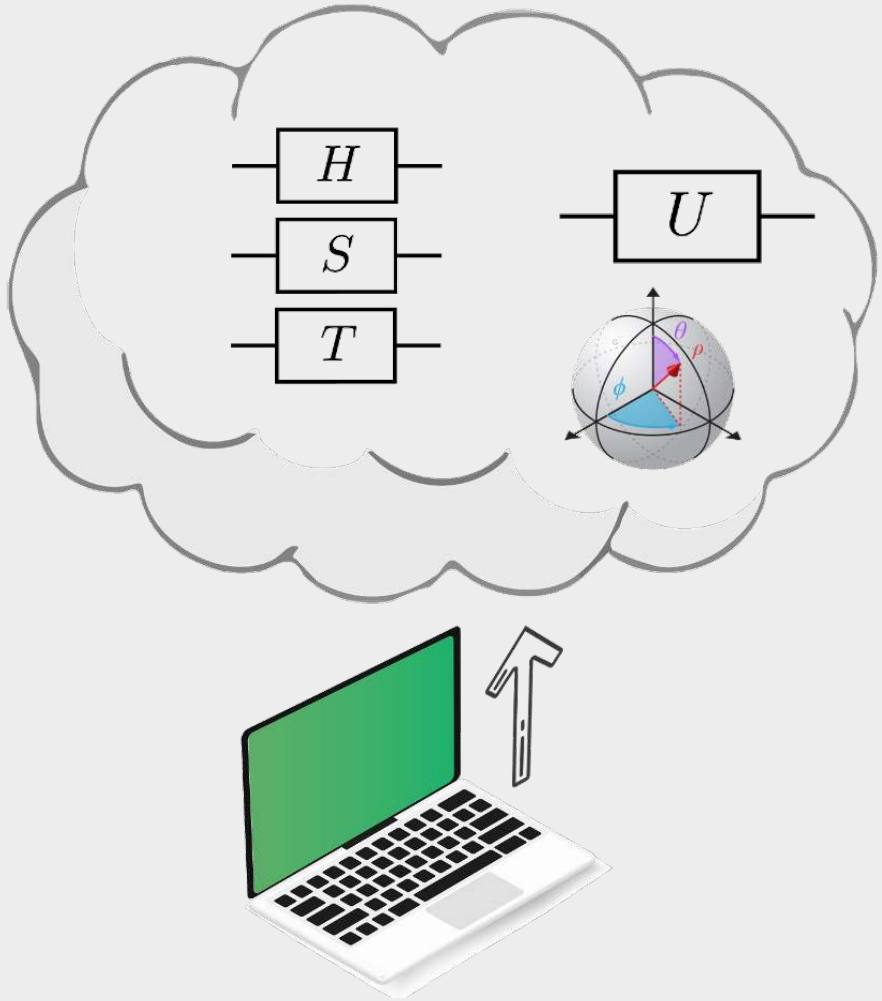
Zlatko Minev, IBM Quantum



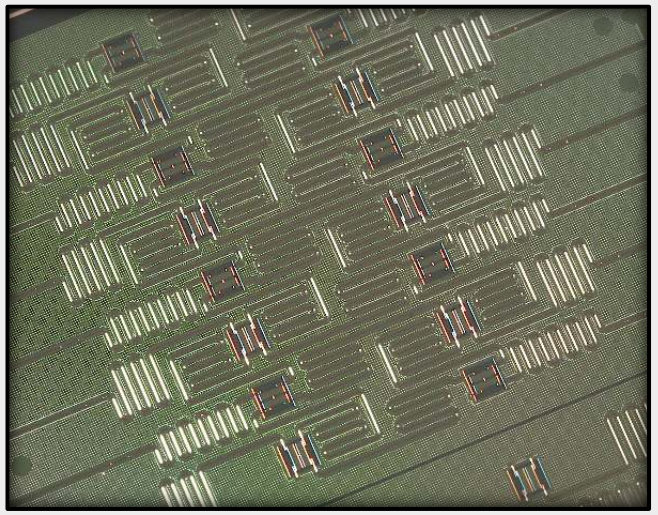


Laptop image: rawpixel.com

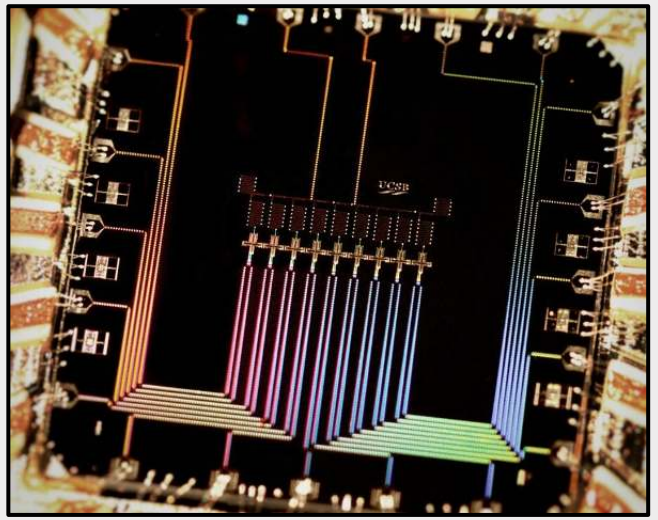
Zlatko Minev, IBM Quantum



How to design a quantum processor?



IBM 16-qubit processor (2018)



John Martinis group (2015)

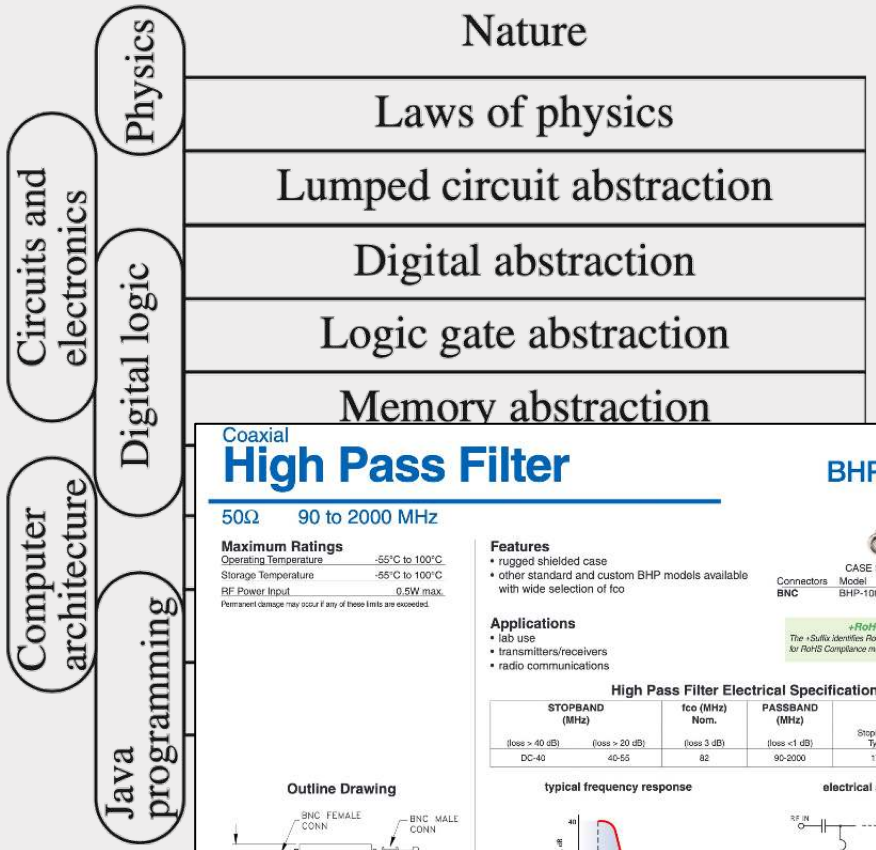
Laptop image: rawpixel.com

Zlatko Minev, IBM Quantum

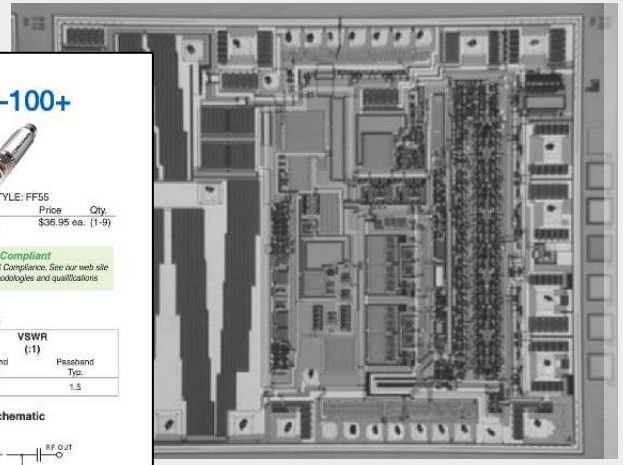
How to design a classical processor

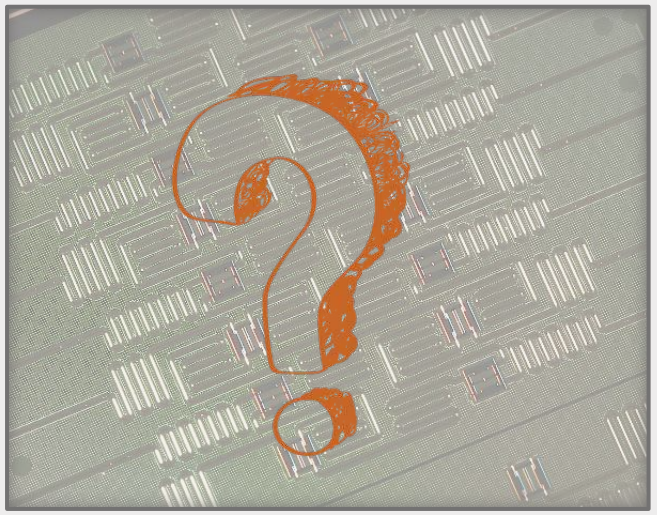
How to design a quantum processor?

Classical hierarchy of abstractions

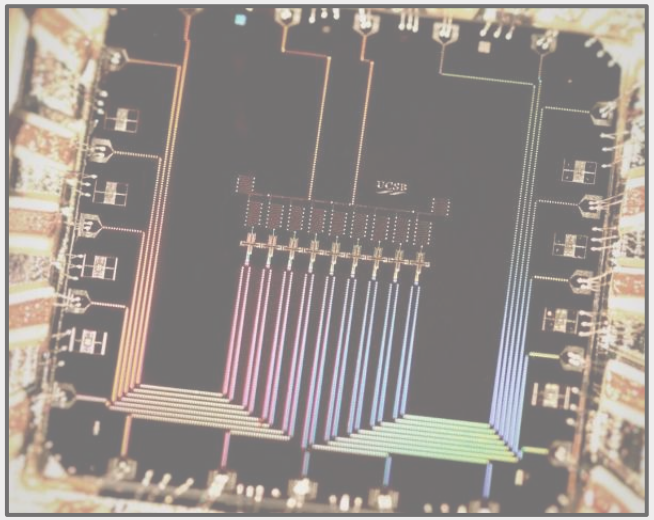


Abstraction layers
Effective models
Modularity
General simulators
…

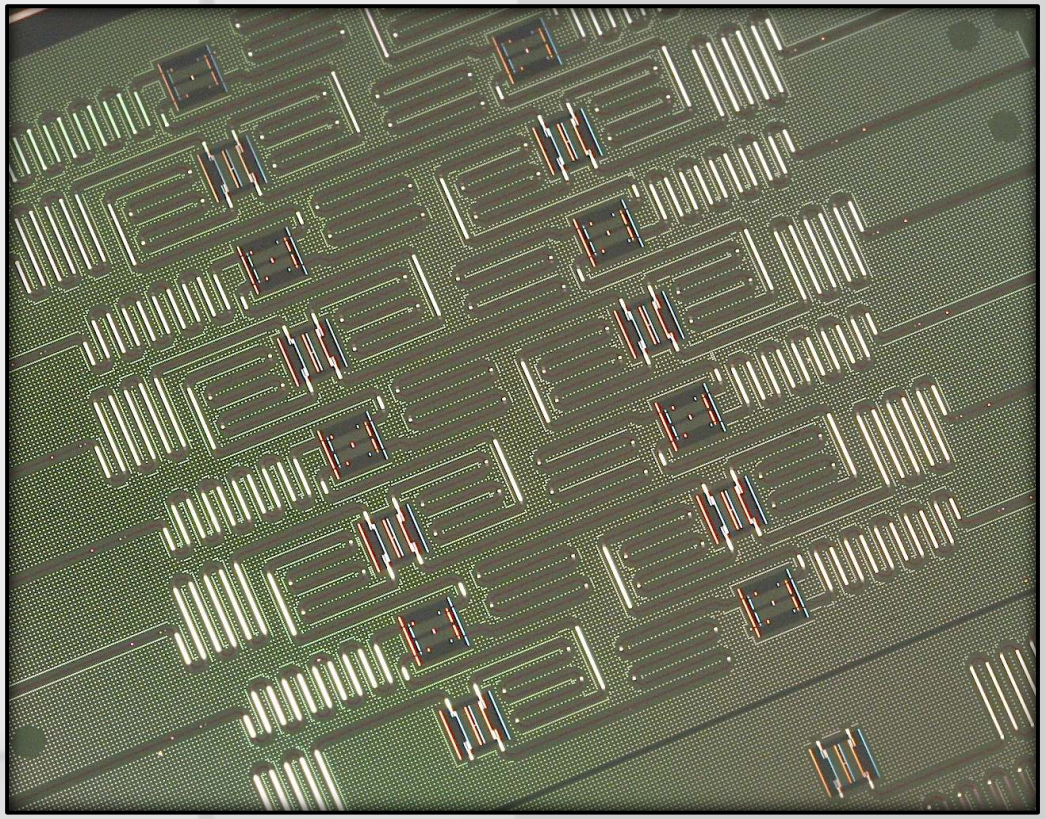


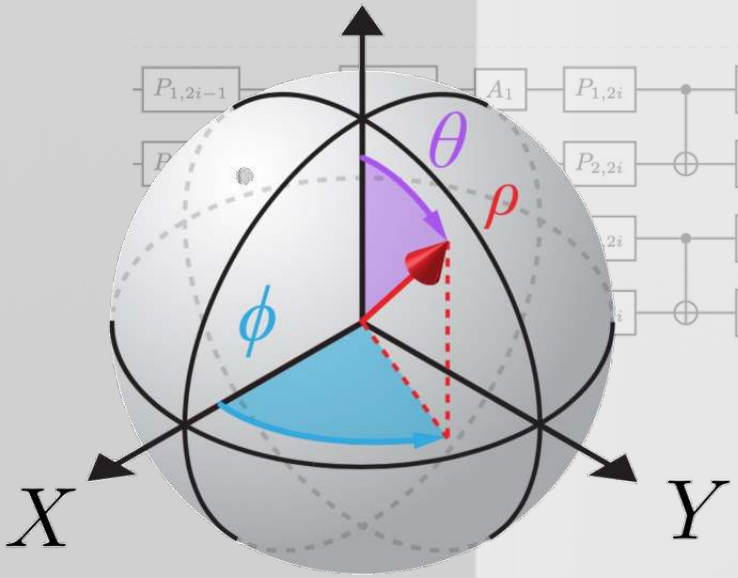


IBM 16-qubit processor (2018)



John Martinis group (2015)

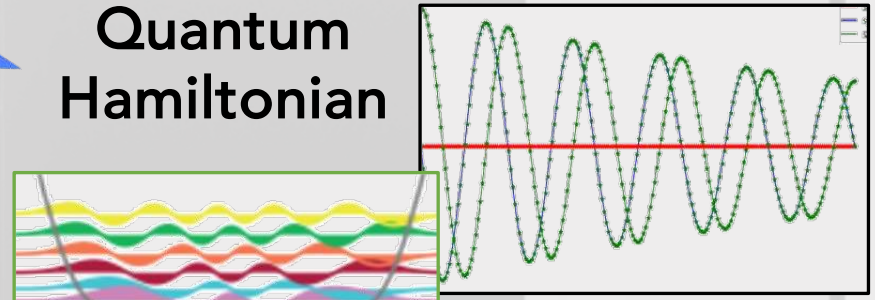



$P_{1,2i-1}$
A_1
$P_{1,2i}$
$P^c_{1,2i}$
$P_{2,2i}$
$P^c_{2,2i}$
$P^c_{3,2i}$
$P^c_{4,2i}$
θ
ρ
ϕ
X
Y

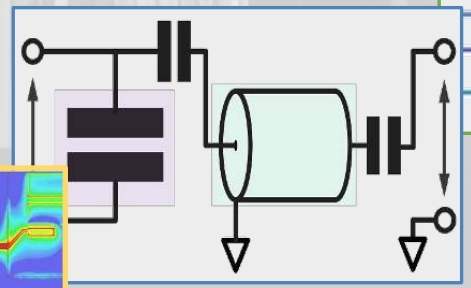
Quantum device design: layers & abstractions

Gates & time-evolution analysis

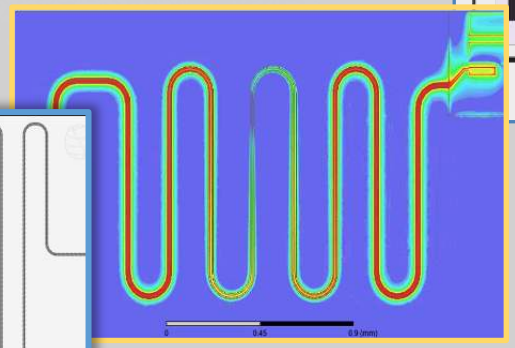
Quantum Hamiltonian



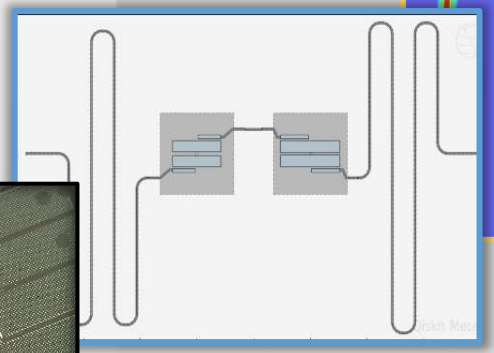
Quantization Methods



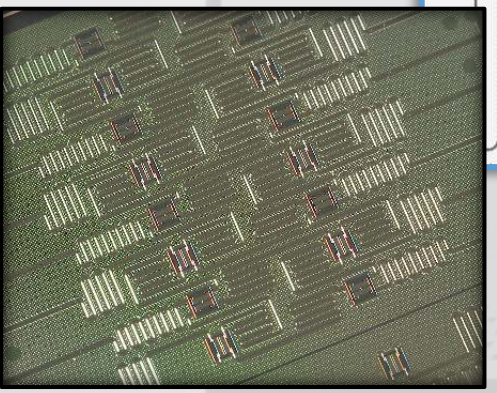
E&M Simulation



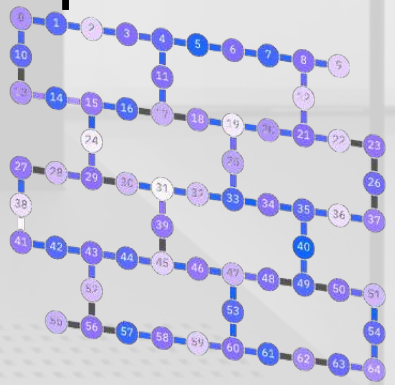
Layout



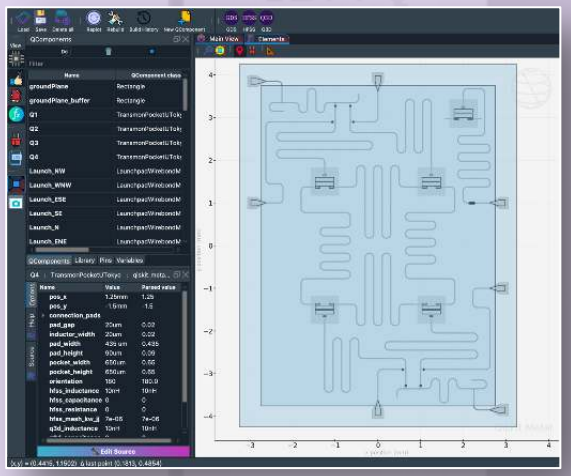
Fabrication



Quantum processor



Qiskit | quantum device design



\+ pyEP

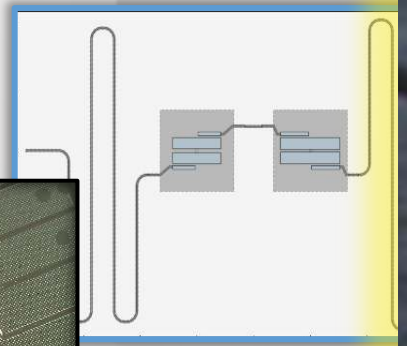
\+ scQ

\+ QuT

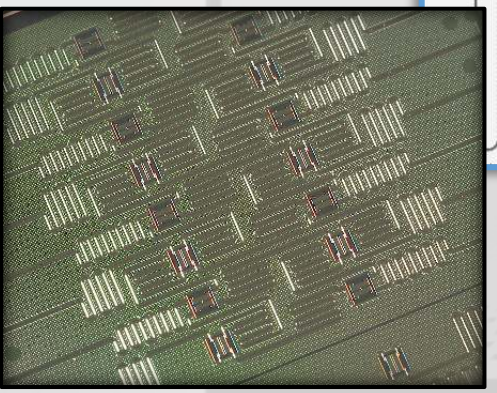
\+ ⋮

Open source

Layout

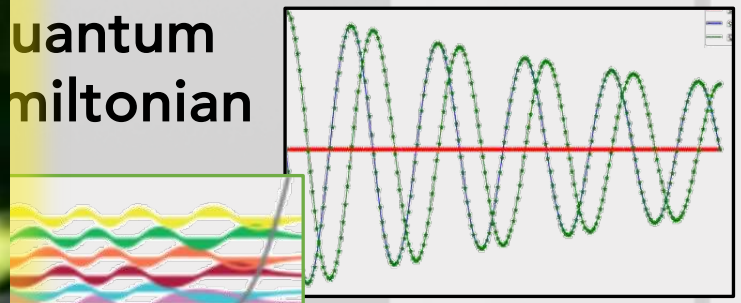


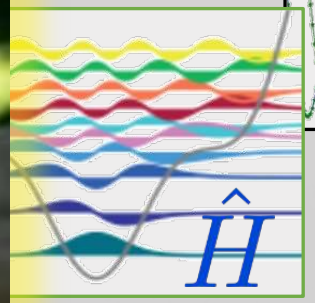
Fabrication



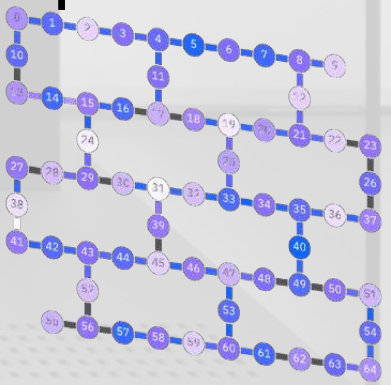
Gates & time-evolution analysis

uantum miltonian

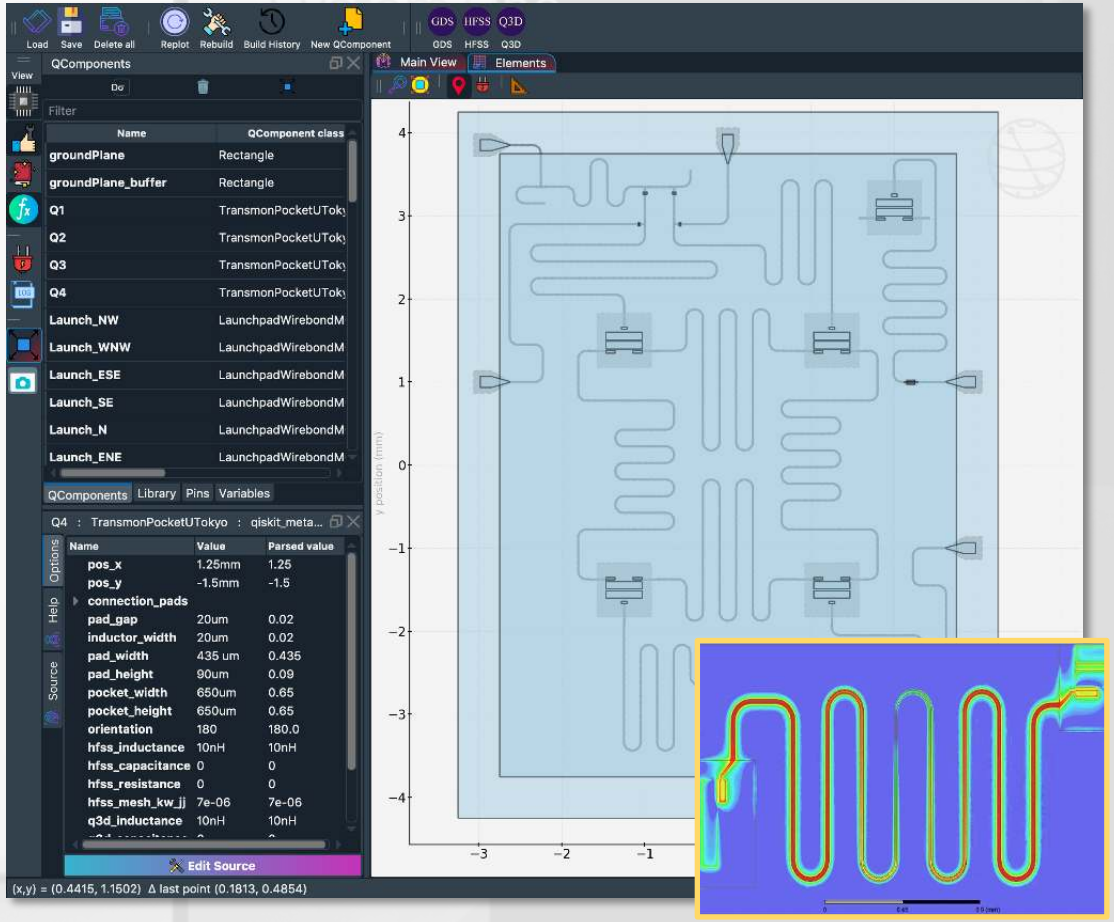


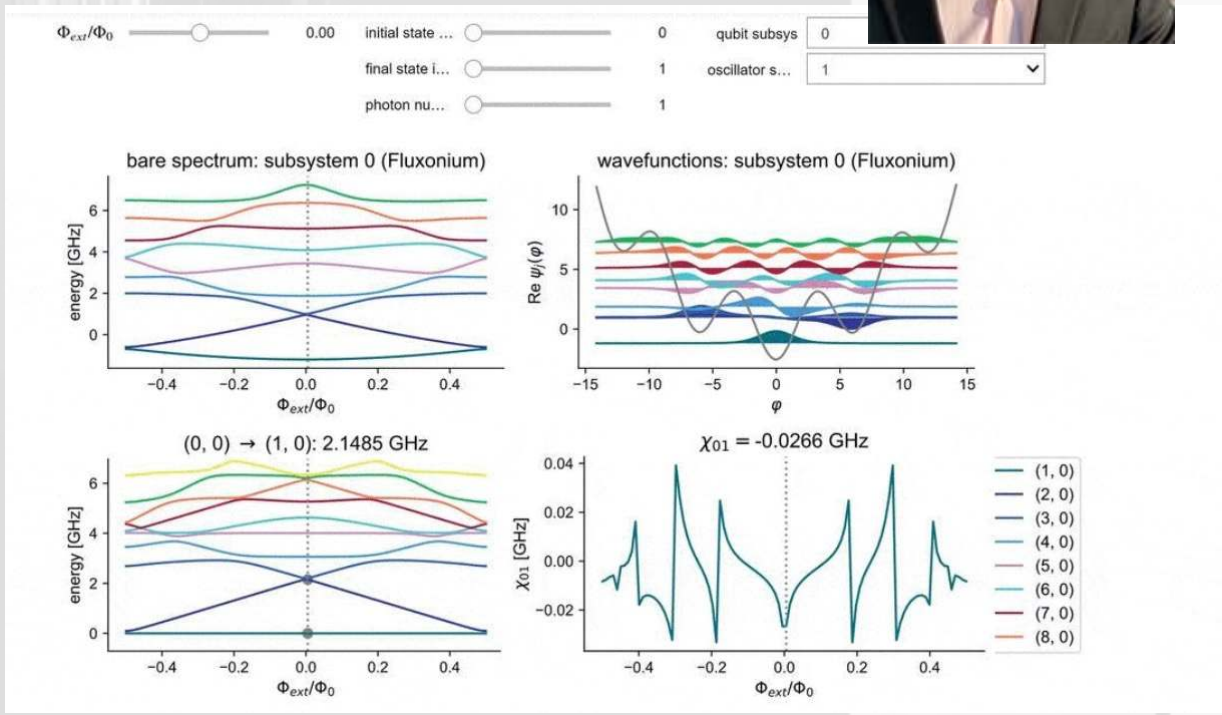


Quantum processor



Qiskit Metal ⟷ scQubits





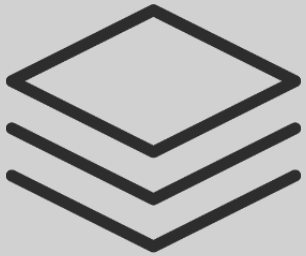
Open source

qiskit.org/metal

Open source

scqubits.readthedocs.io

A hardware designer’s wish list

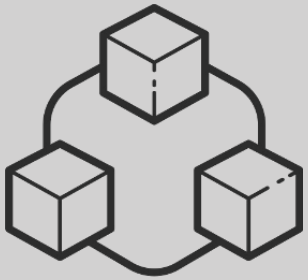


Full Stack

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

Speed

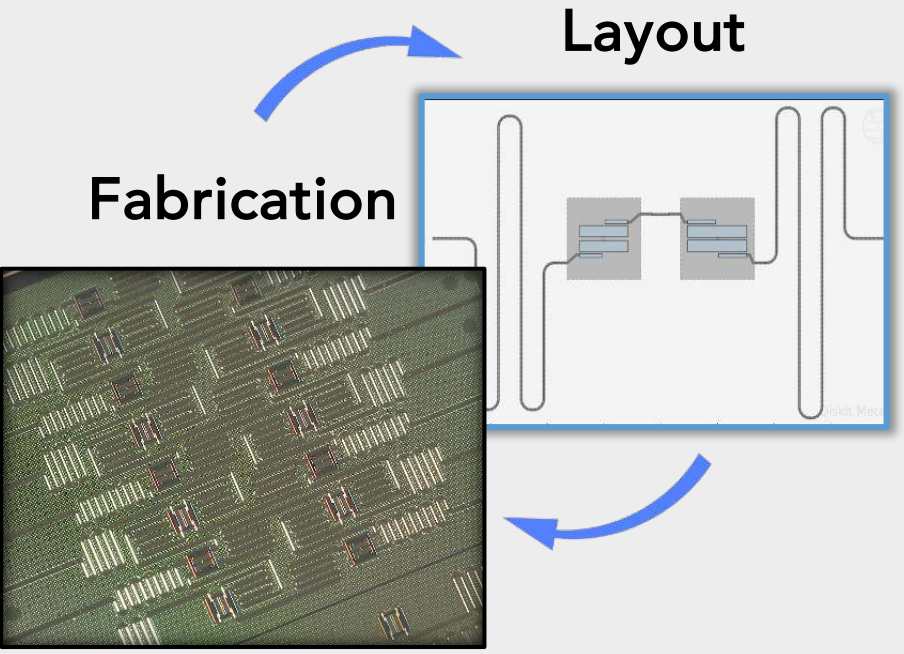
Good trade-off between speed and accuracy. Not every device requires hours of full wave simulations



Modular

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

Qiskit | quantum device design



Example with Qiskit Metal

"Classical"

icon: wanicon

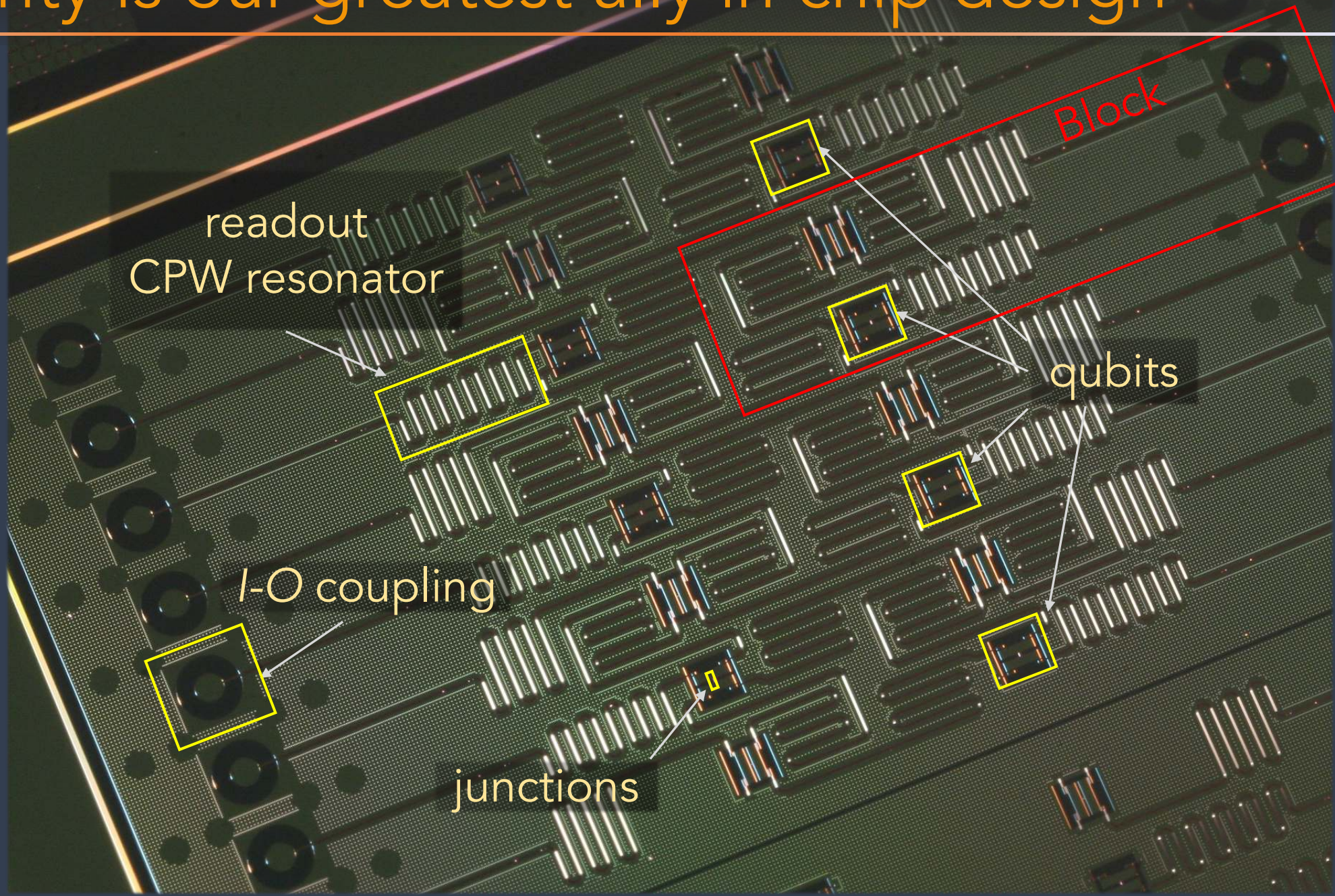
Regularity is our greatest ally in chip design

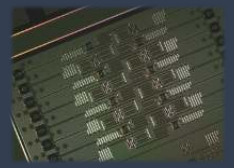
Reuse

Fine-tune

Automate

Extend

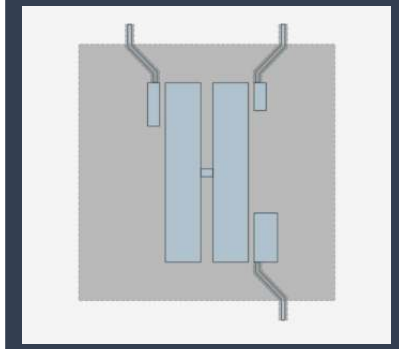




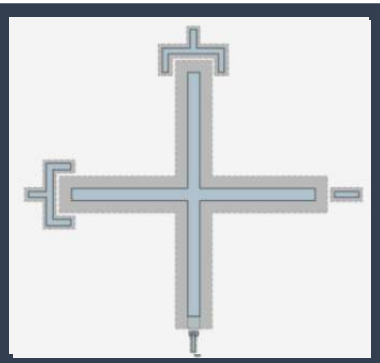
Device Library

Qiskit | quantum device design

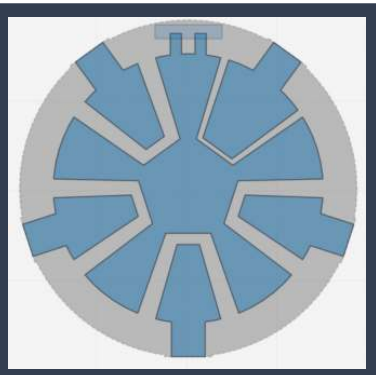
Qubits



Single transmon-floating

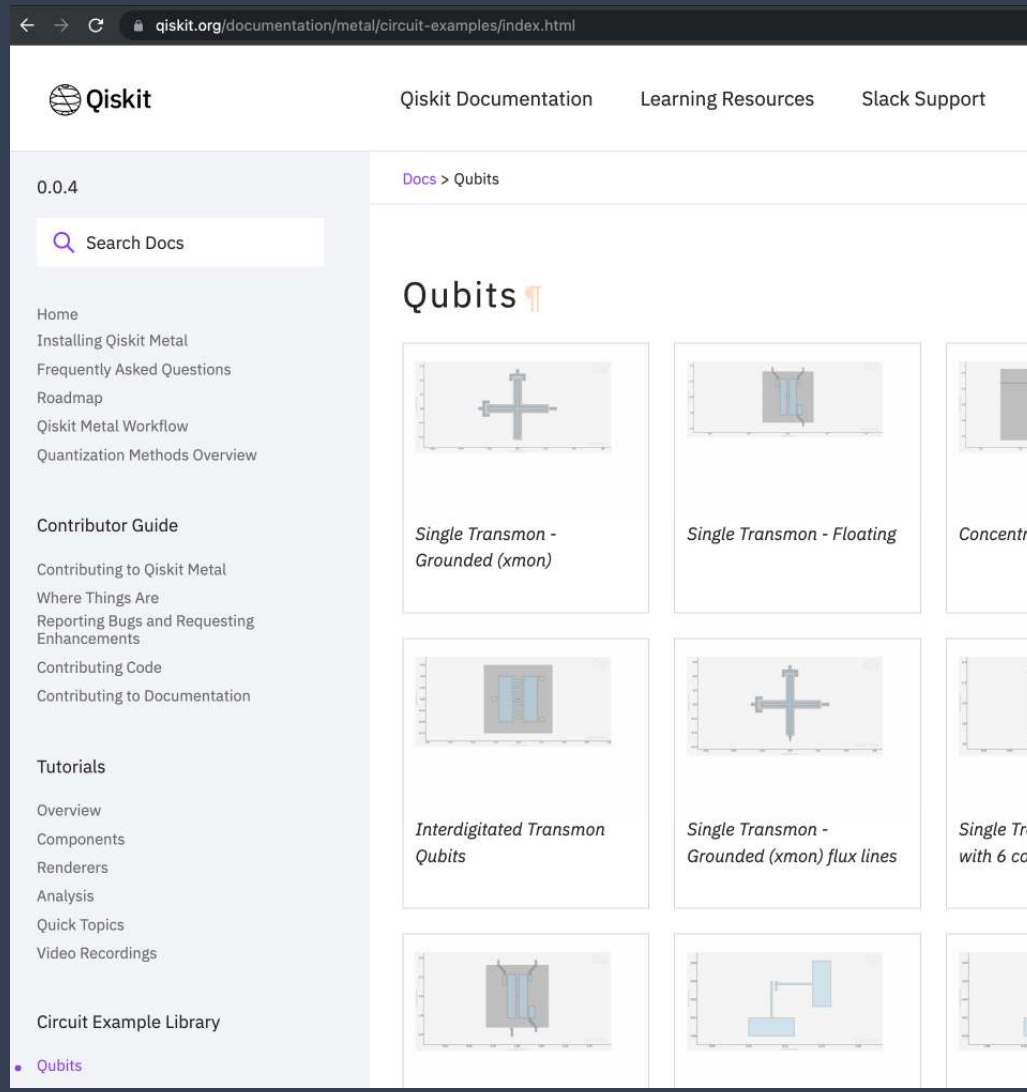


Single transmon-grounded flux lines



Star-shaped qubit

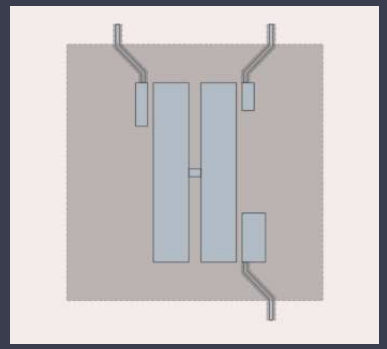
qiskit.org/documentation/metal



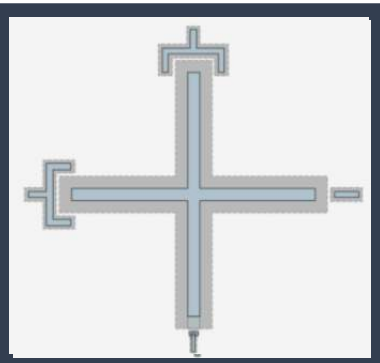
Device Library

Qiskit | quantum device design

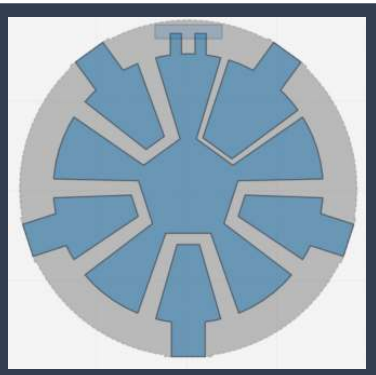
Qubits



Single transmon-floating

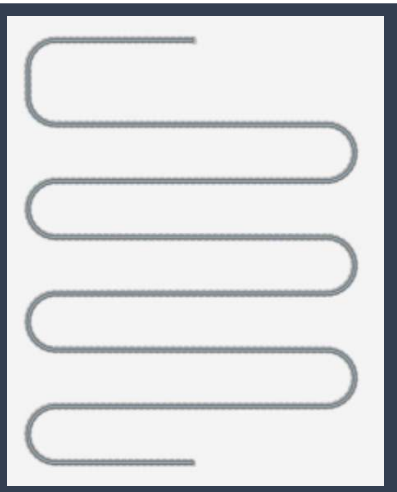


Single transmon-grounded flux lines



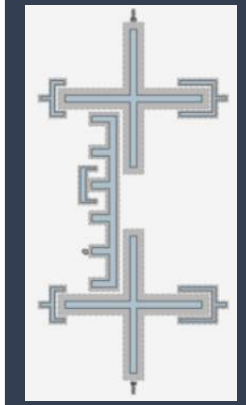
Star-shaped qubit

Resonators

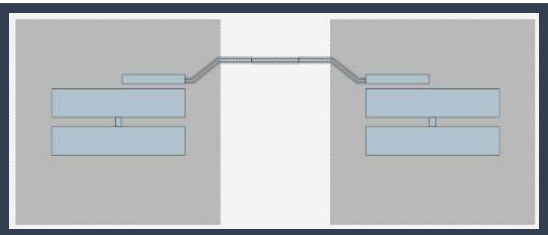


Simple coplanar waveguide (CPW) meander

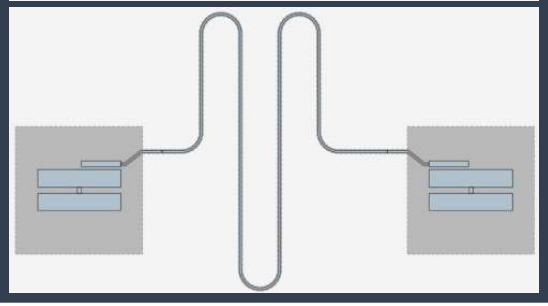
Qubit Couplers



Tunable coupler (MIT)

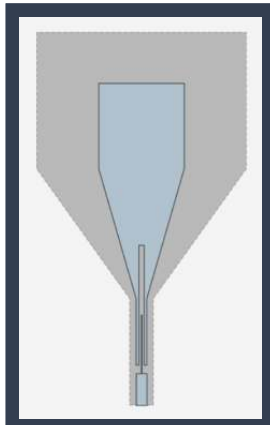


Direct coupler

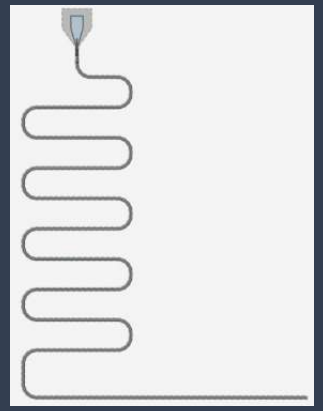


Bus resonator coupler

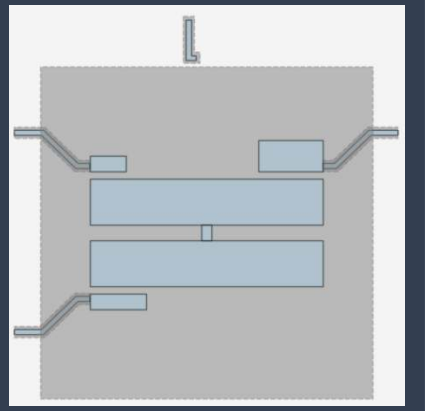
Input – Output Coupling



CPW launch pad



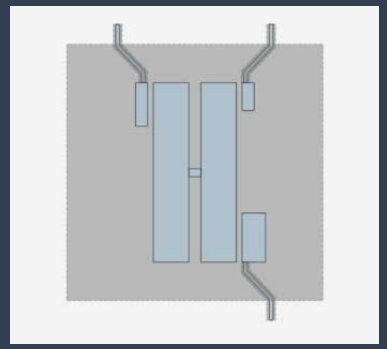
Readout line



Charge line

Device Library

Qiskit | quantum device design



Single transmon-floating

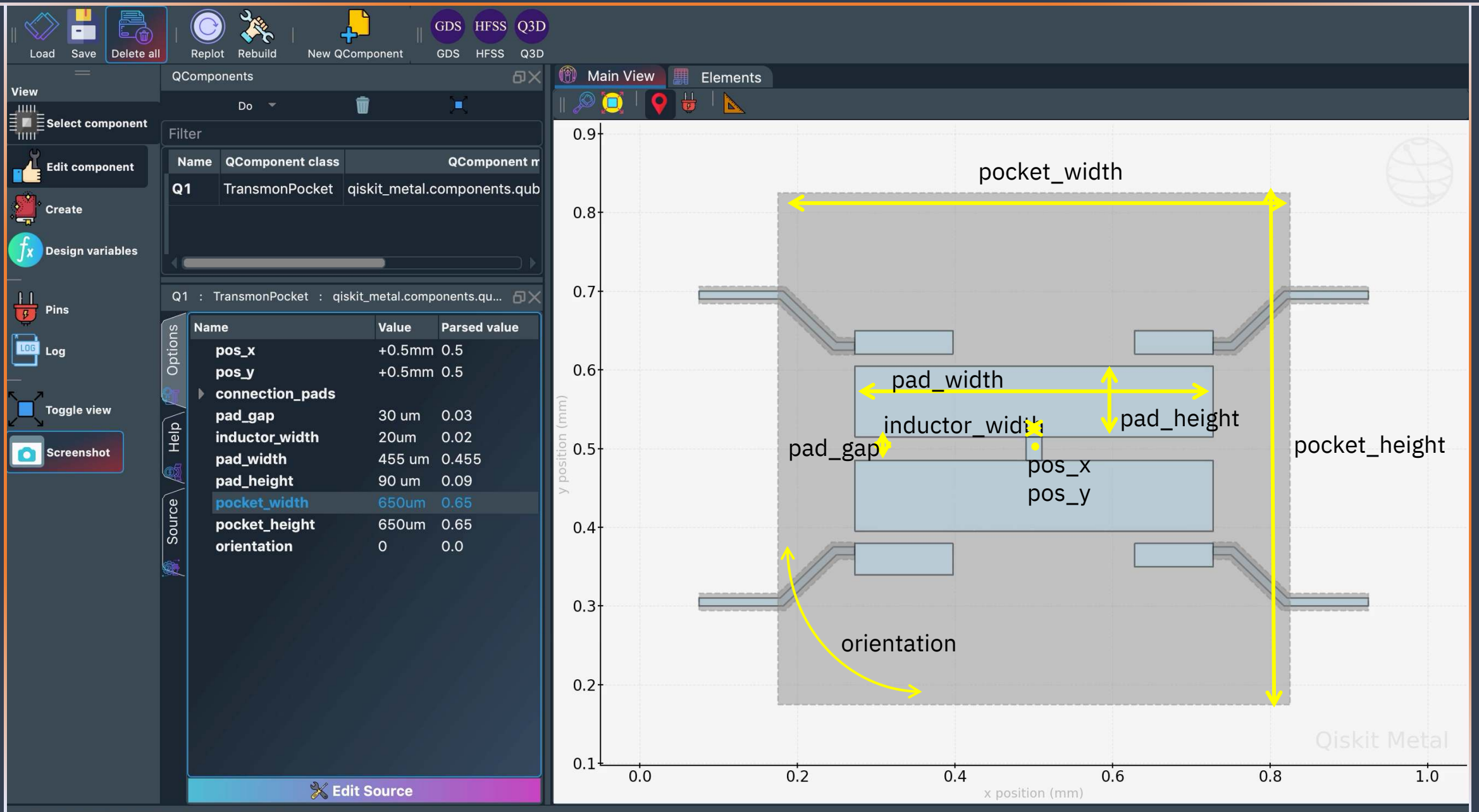
Code

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

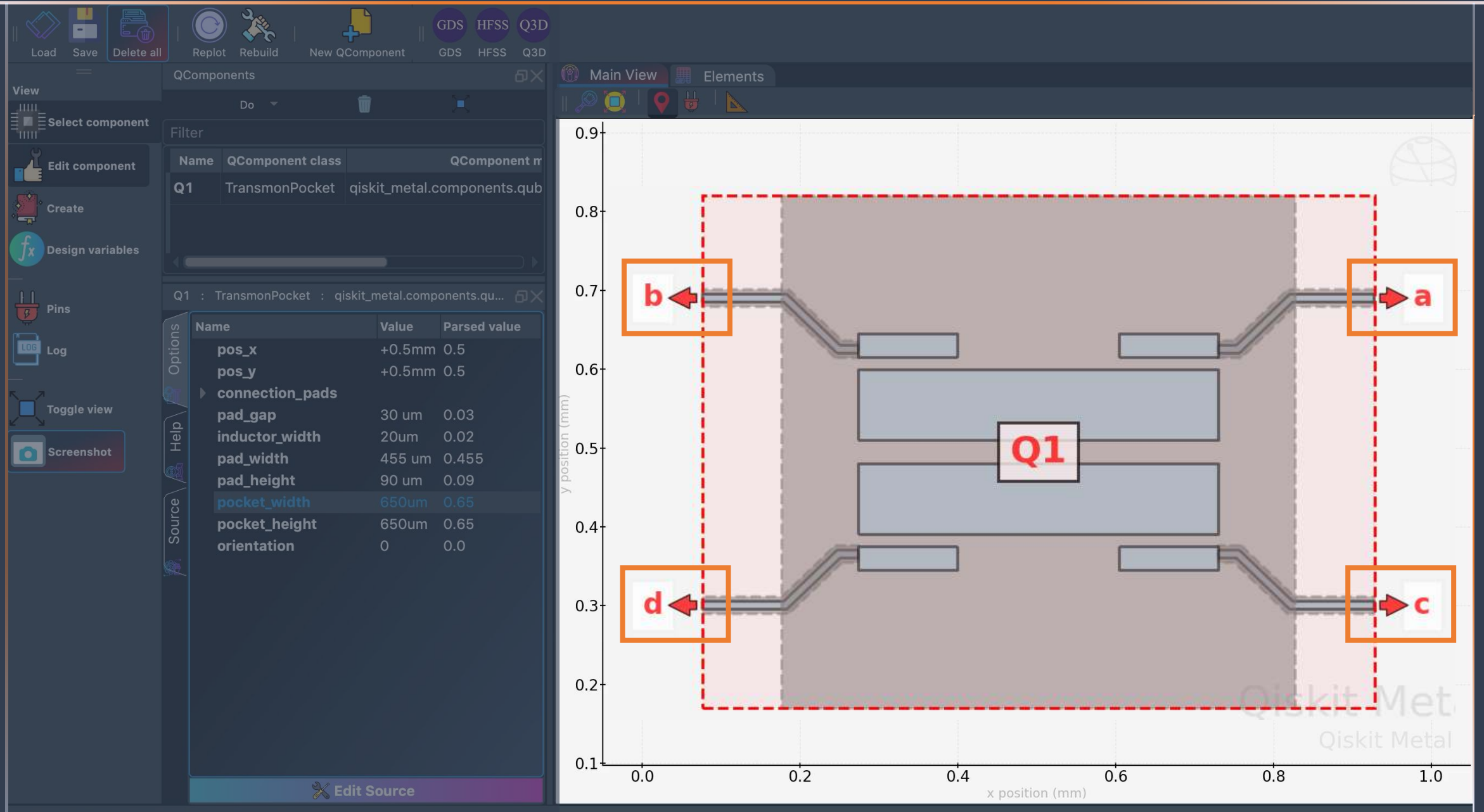
Synced GUI

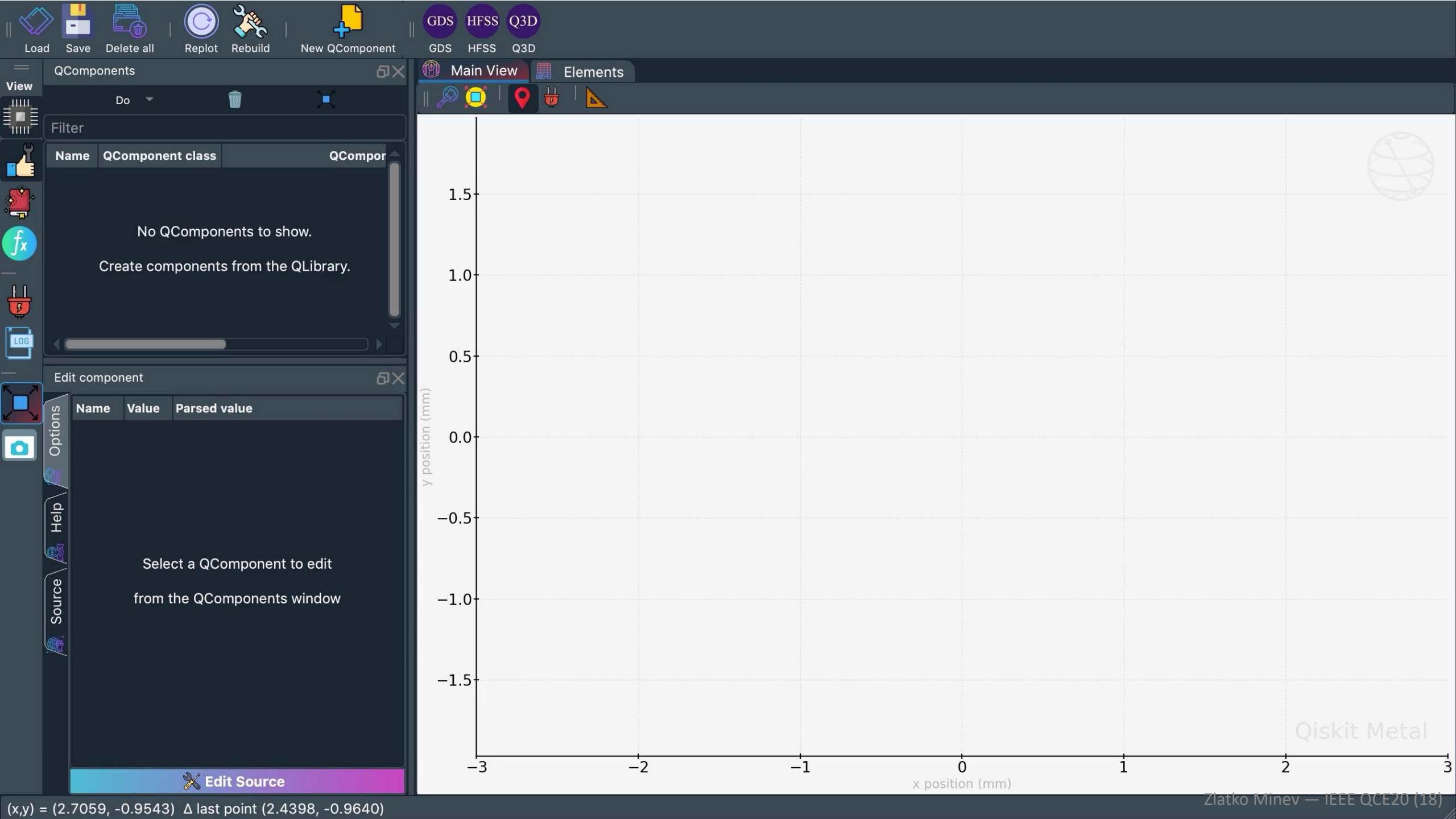
```
from qiskit_metal import MetalGUI
MetalGUI()
```

Fine-tune and automate parameters



Connecting quantum components dynamically




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)
Zlatko Minev — IEEE QCE20 (18)

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

Options Help Source

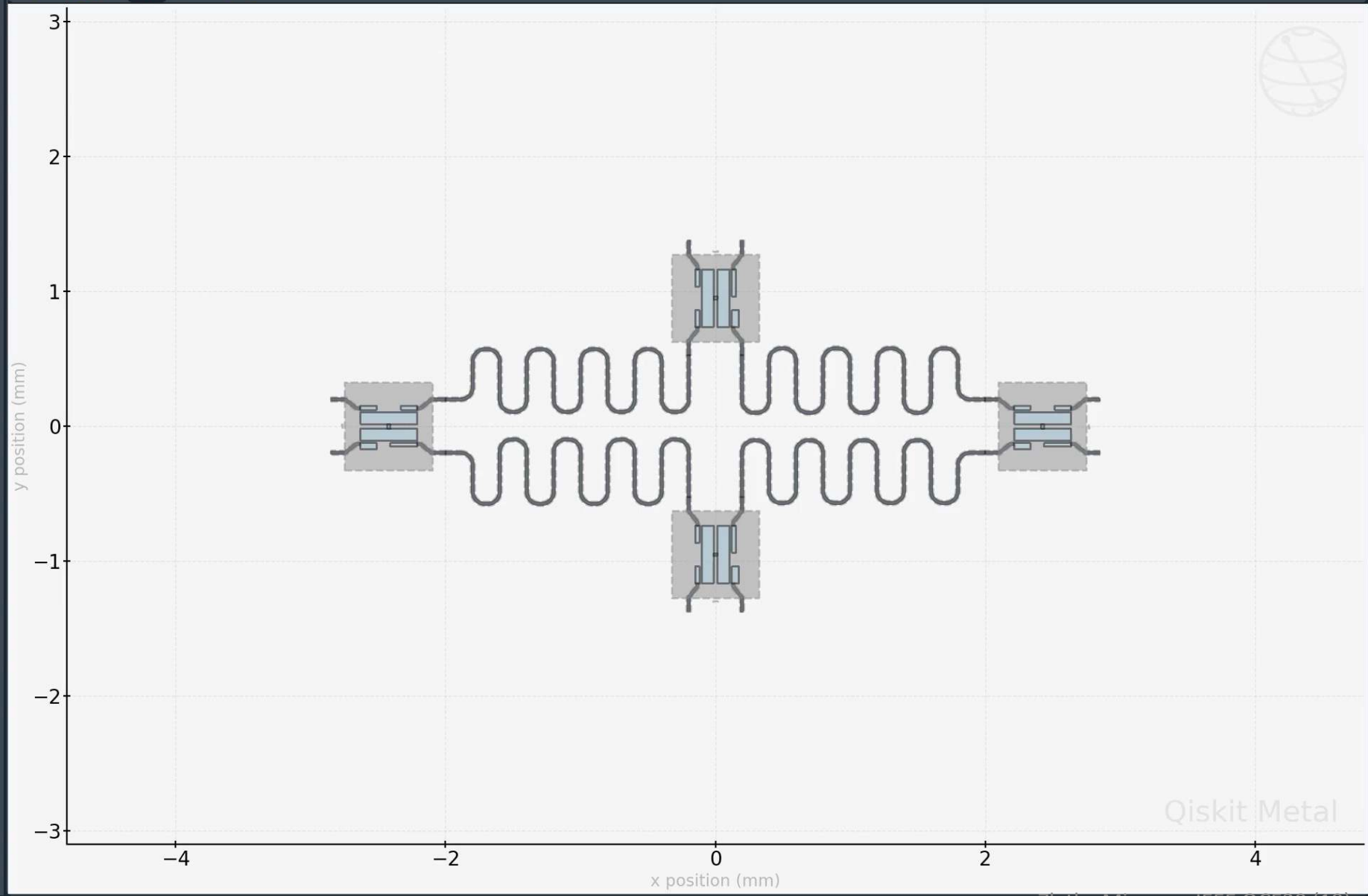
Name	Value	Parsed value

Select a QComponent to edit

from the QComponents window

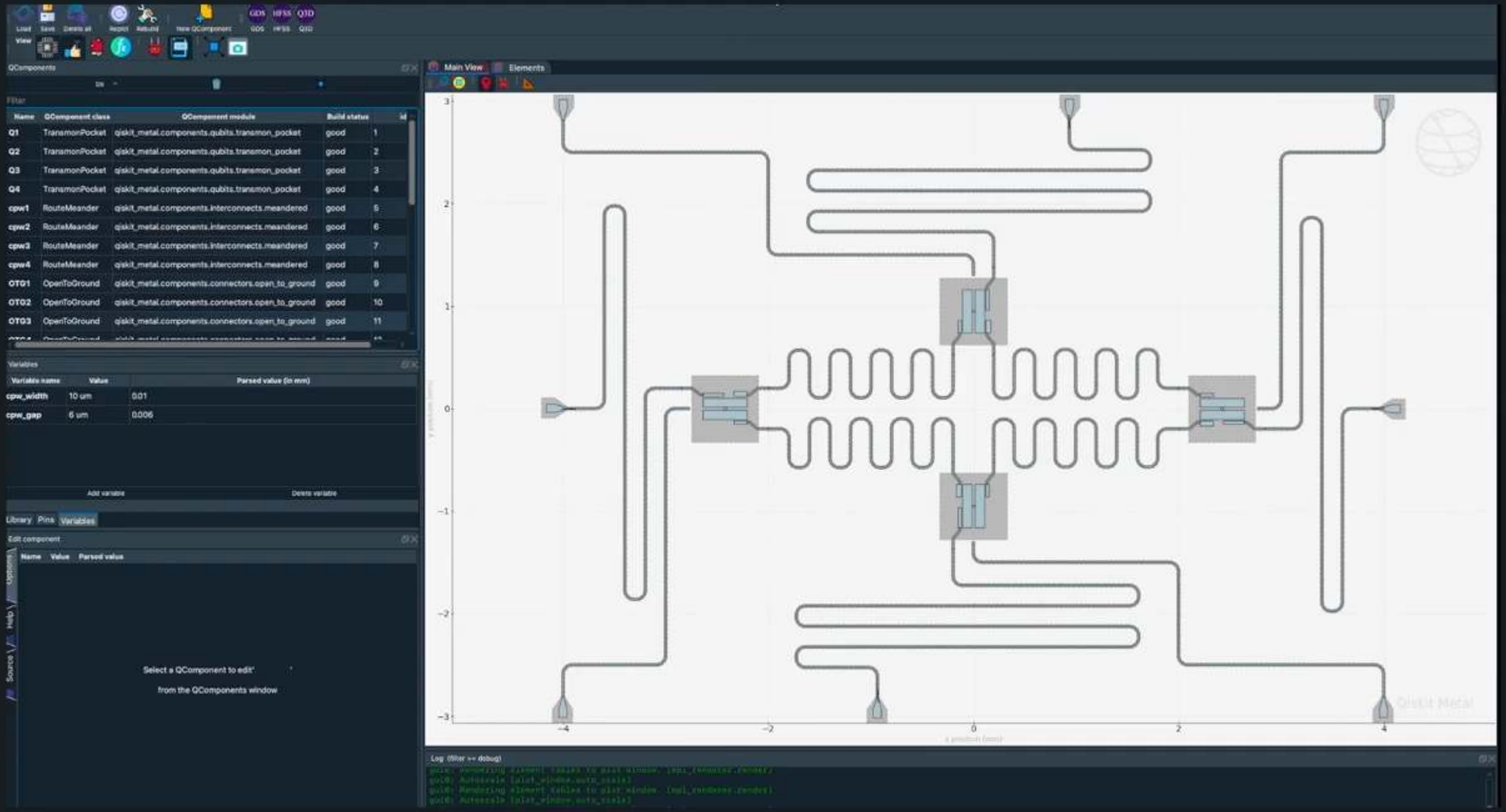
Edit Source

Main View Elements

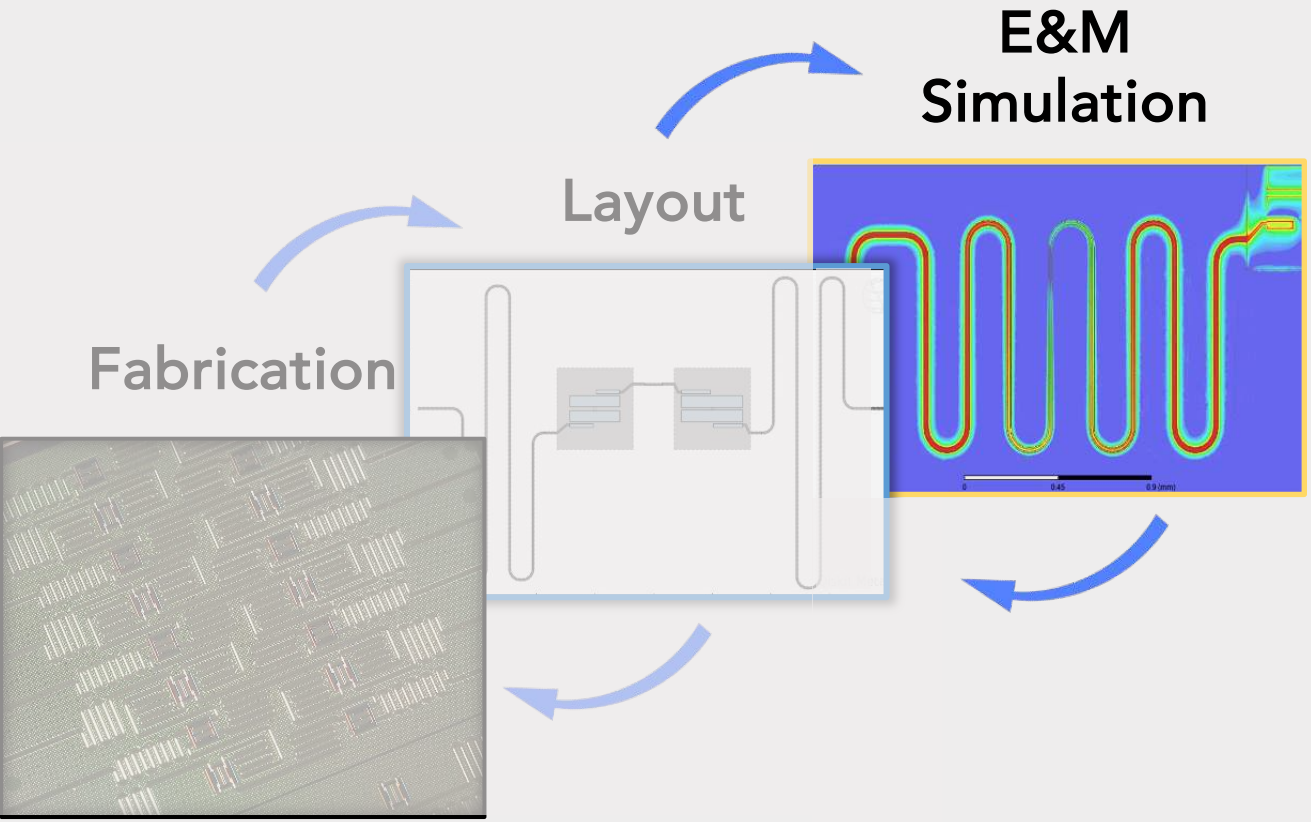


Qiskit Metal: Quantum Creator

Render your design as a GDS file


Load
Save
Delete all
Reset
Rebuild
New QComponent
GDS
HFSS
Q3D
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
OT01 OpenToGround qiskit_metal.components.connectors.open_to_ground good 9
OT02 OpenToGround qiskit_metal.components.connectors.open_to_ground good 10
OT03 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 | quantum device design

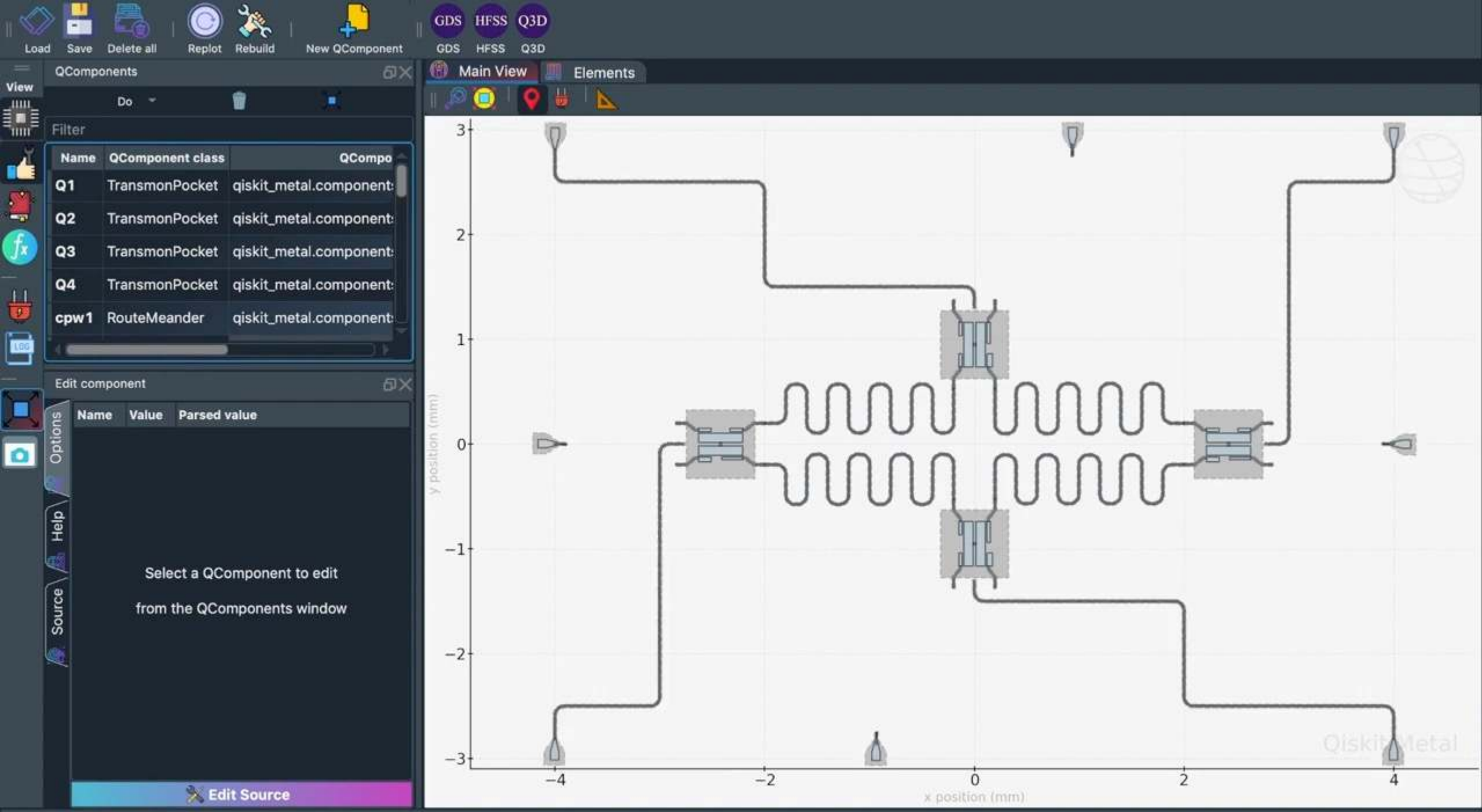


Render your design in an EM Solver

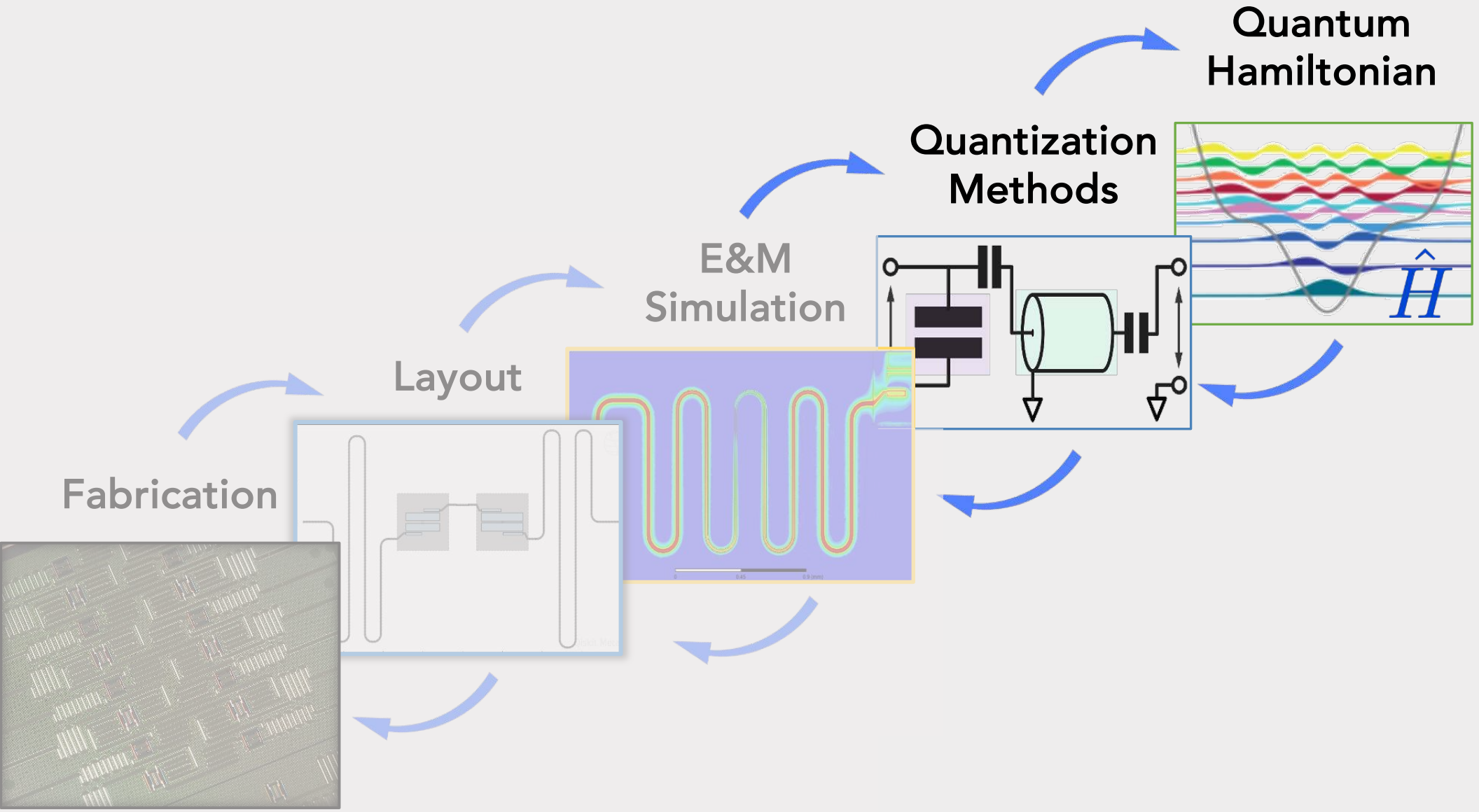
For example: Ansys HFSS

icon: wanicon

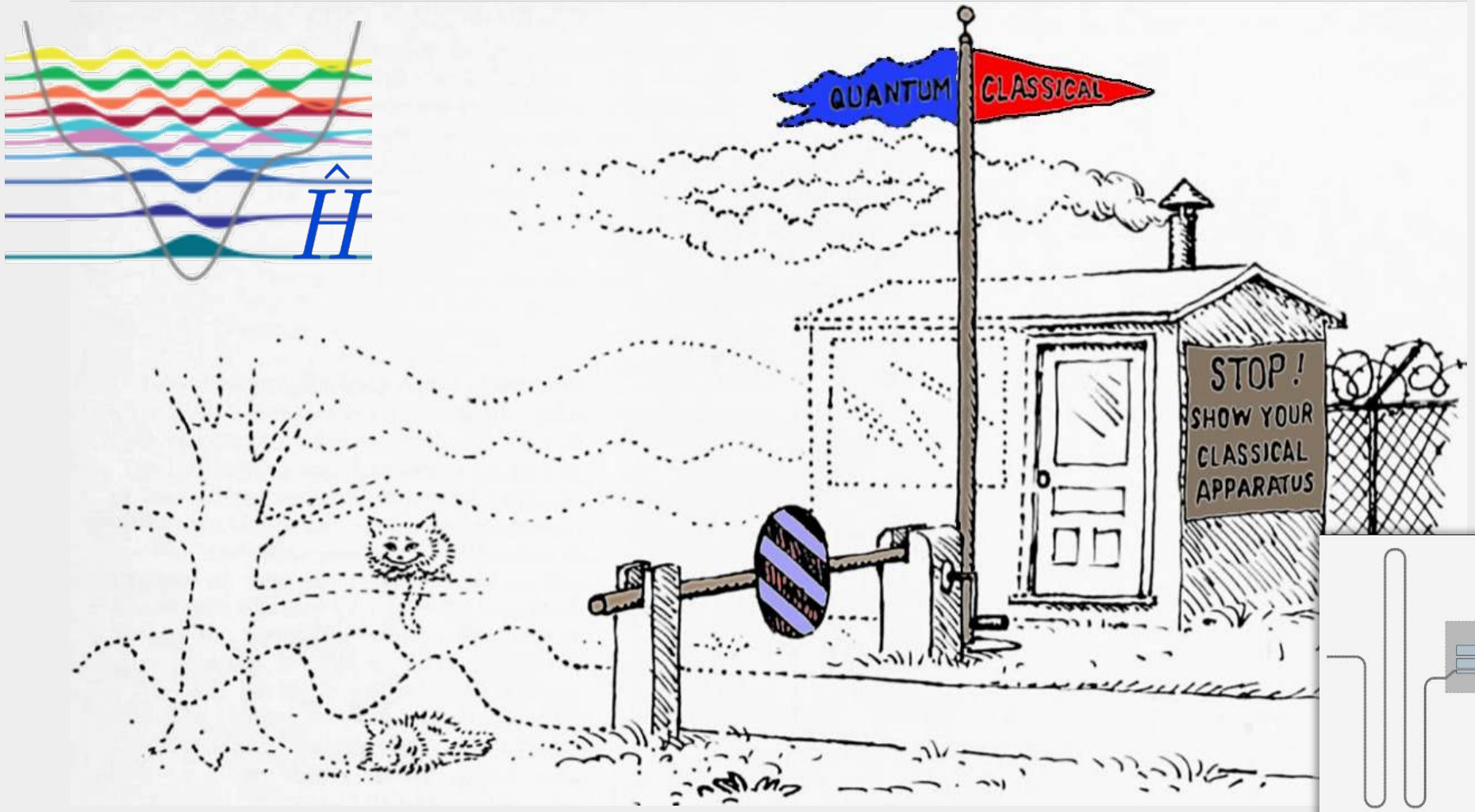
"Classical"


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
Options
Help
Source
Name
Value
Parsed value
Select a QComponent to edit
from the QComponents window
Edit Source
Main View
Elements
y position (mm)
x position (mm)
Design Dock

Qiskit | quantum device design

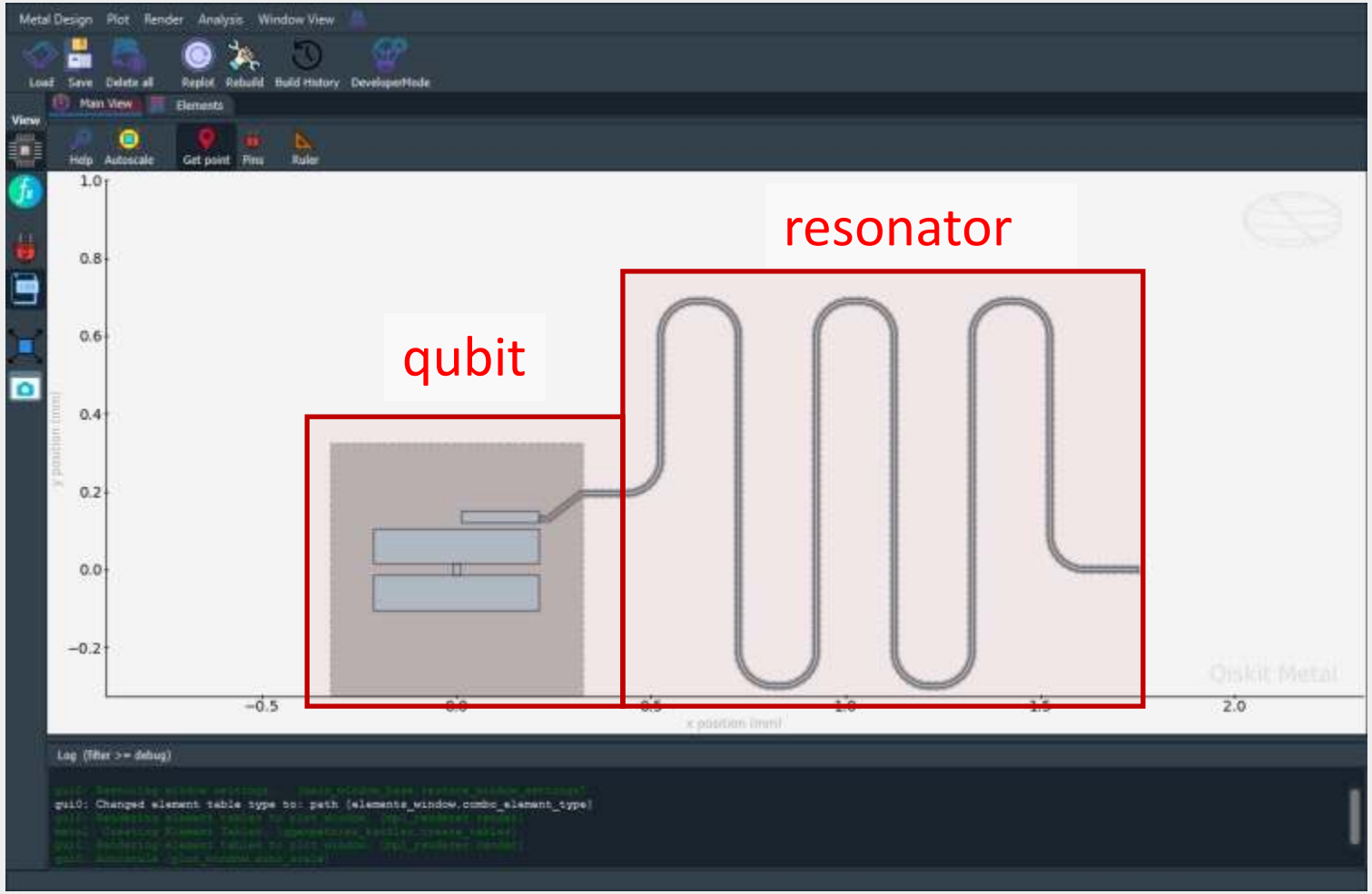


Quantum Analysis

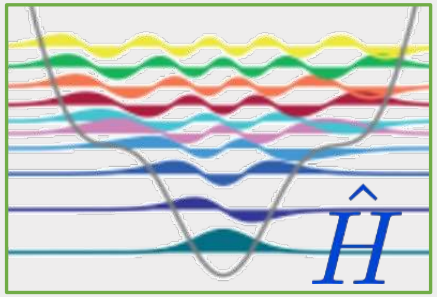


Drawing: Zurek, Physics Today (1991)

Simple example

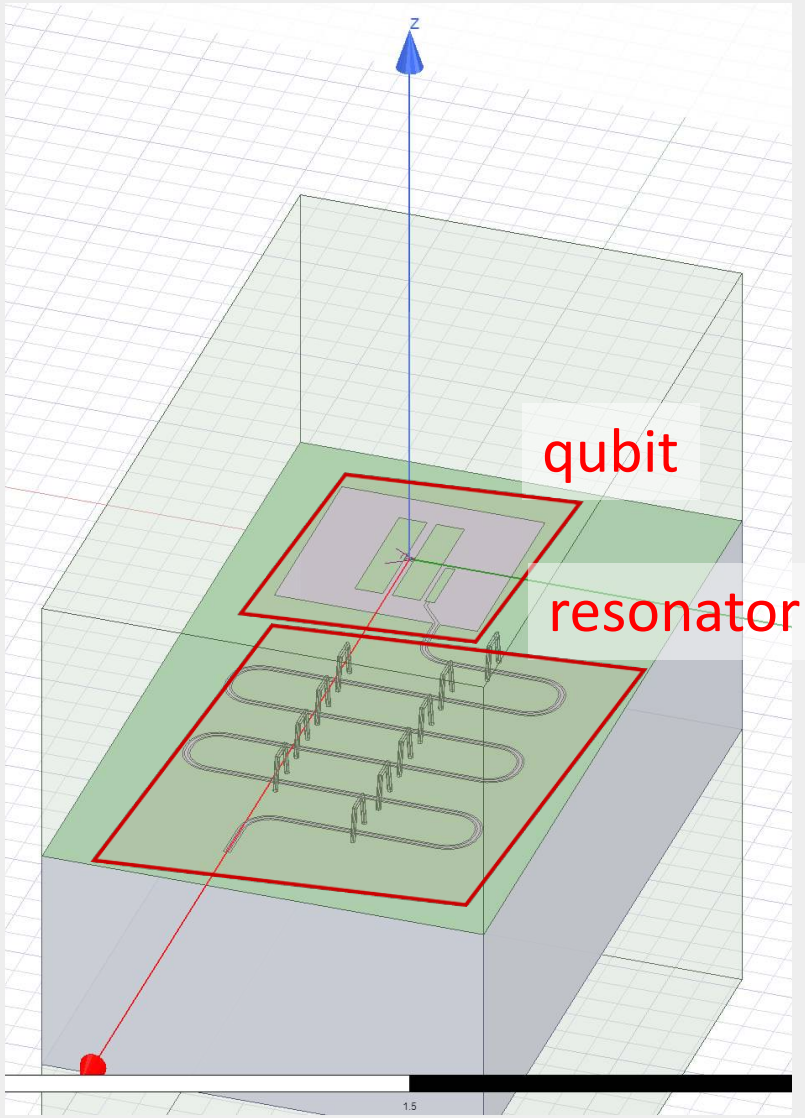


?

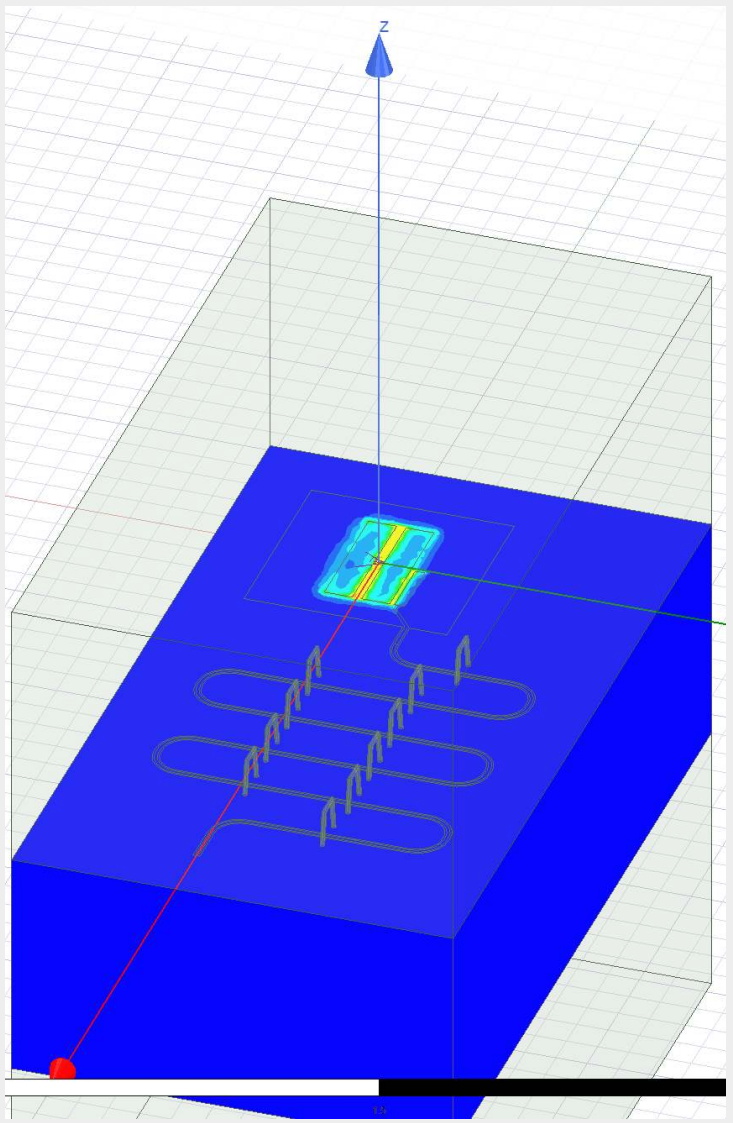


Modeling step 1: Classical E&M approximations

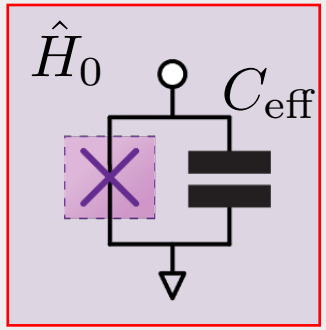
Physical (linear) model



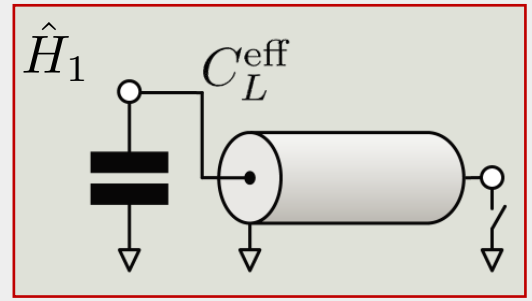
Eigenmodes



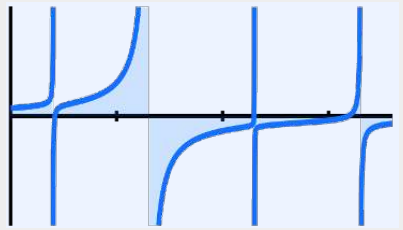
Quasi-lumped models



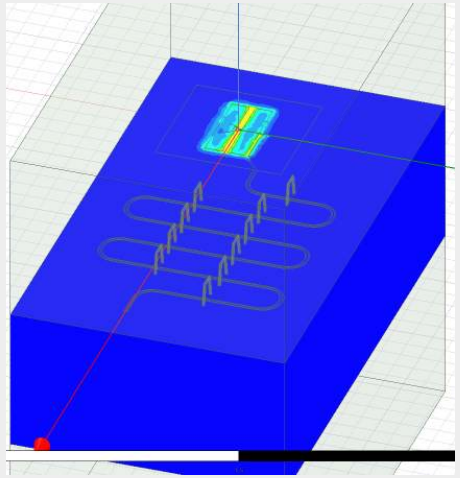
$$\begin{array}{c} \\ n_0 \\ n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \end{array} \begin{array}{c} \begin{array}{cccccc} n_0 & n_1 & n_2 & n_3 & n_4 & n_5 \end{array} \\ \begin{pmatrix} C_{0\Sigma} & -C_{01} & -C_{02} & -C_{03} & -C_{04} & -C_{05} \\ & C_{1\Sigma} & -C_{12} & -C_{13} & -C_{14} & -C_{15} \\ &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

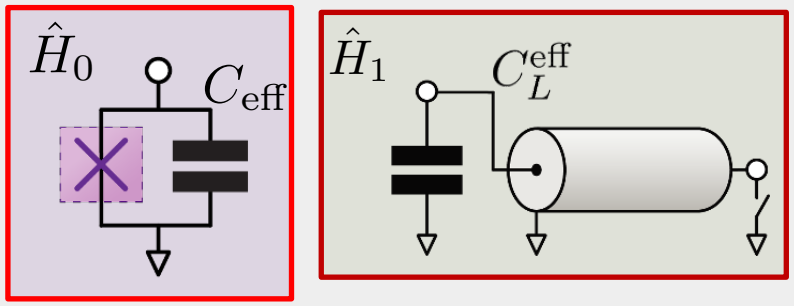


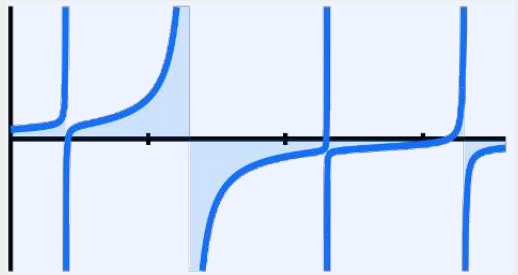
Impedance



Modeling step 2: Quantize







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

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

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

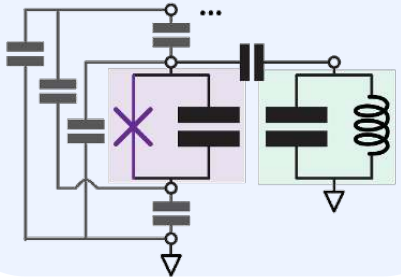
There's a description for every job!

Landscape of **quantization** methods

quasi-static

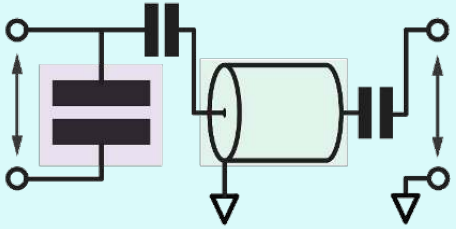
full-wave

lumped



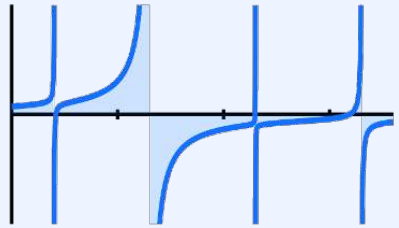
Yurke & Denker (1984), Devoret (1997), Burkard *et al.* (2004), Koch *et al.*…

quasi-lumped



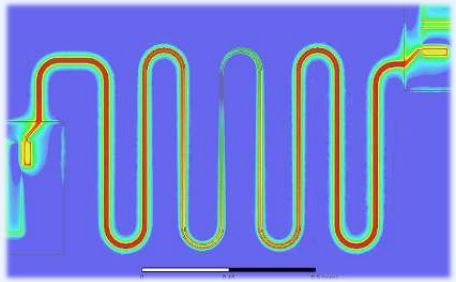
Malekakhlagh *et al.* (2017, 2019), Gely *et al.* (2019), Parra-Rodriguez *et al.* (2019), Minev *et al.* (2021), ...

impedance



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

energy



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

more speed

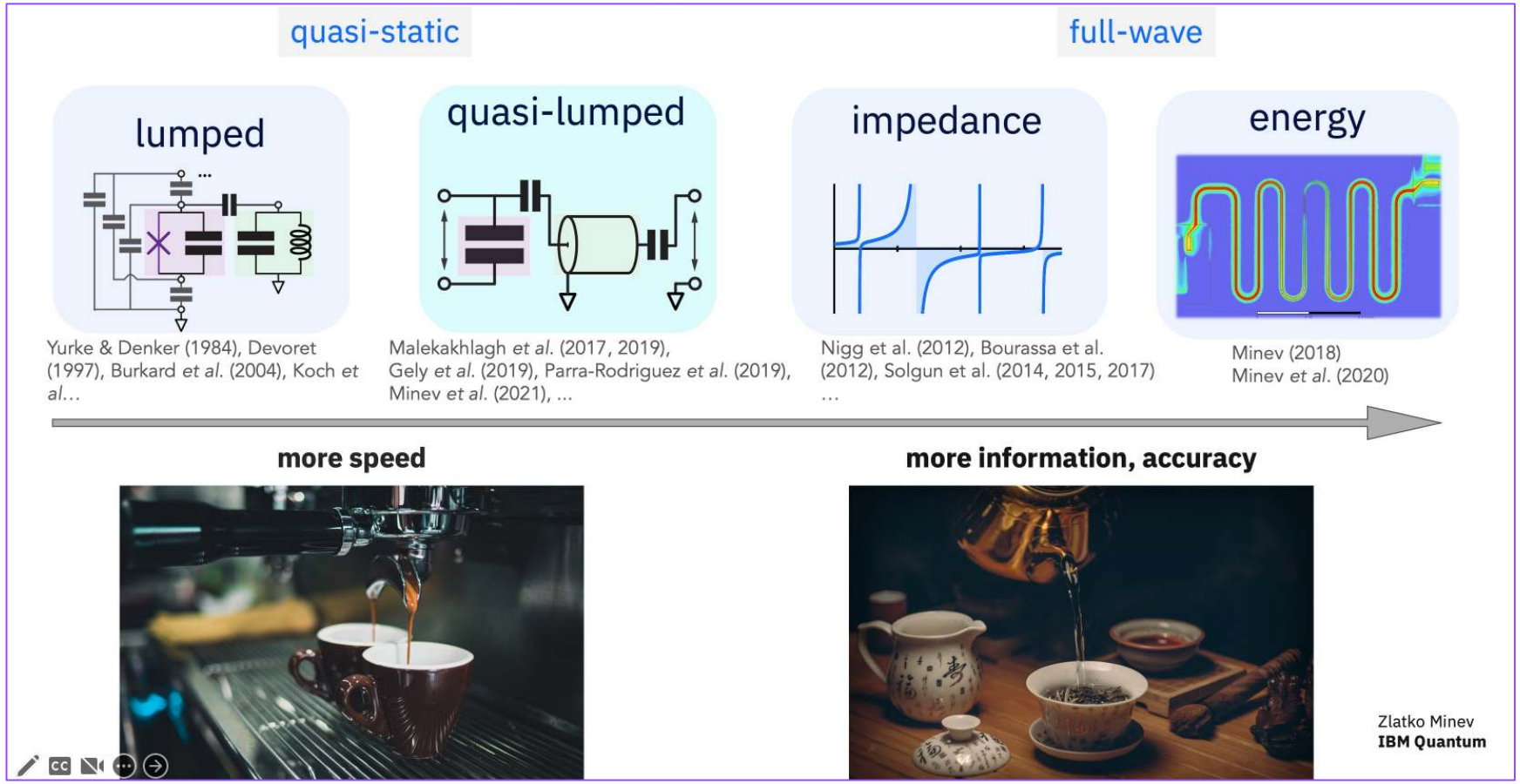
more information, accuracy

Zlatko Minev
IBM Quantum

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

Qiskit

Quantization methods library



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

Quantization methods library: Tutorials

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

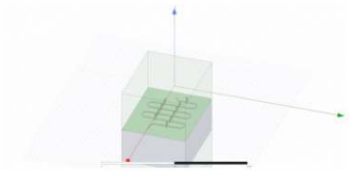
Code of Conduct

Qiskit

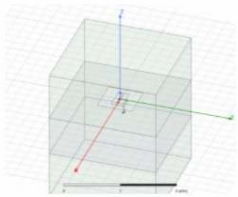
https://qiskit.org/documentation/metal/tut/index.html

Tutorials: Quantum analysis library

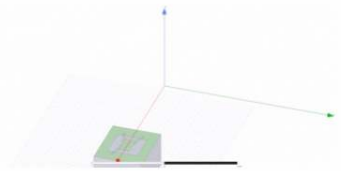
Qiskit



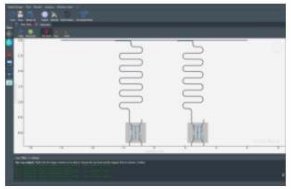
Eigenmode and EPR



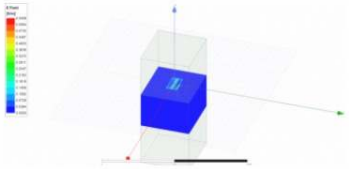
Impedance Analysis



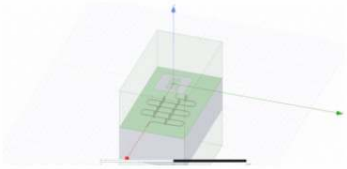
Capacitance Matrix and LOM Analysis



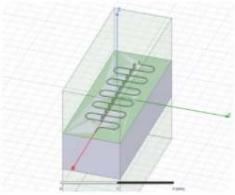
Sweep a option in QComponent, get Ansys HFSS driven-modal scatter matrix, impedance matrix, and admittance matrix.



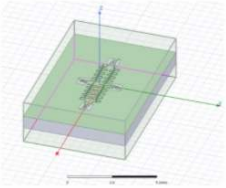
Analyzing and Tuning a Transmon Qubit



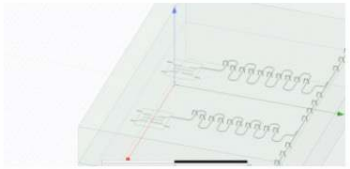
Analyzing a transmon and a resonator



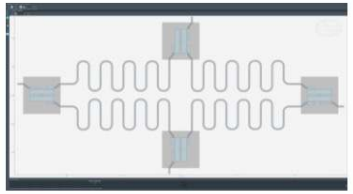
Analyzing a resonator



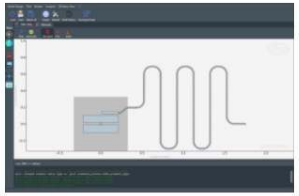
How to Render a Metal Design into Ansys



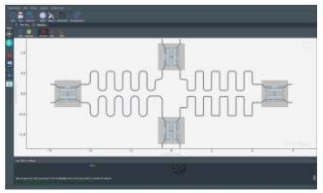
Driven Modal Simulation and S-Parameters



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

Tutorials: Quantum analysis library

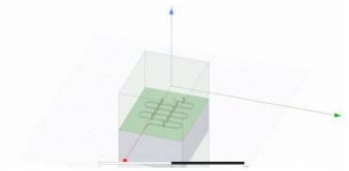
- Resonators
- Composite Bi-Partite Systems
- Qubit Couplers
- Input-Ouput Coupling
- Small Quantum Chips
- Design Flow

Libraries

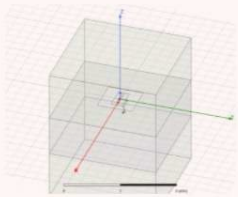
- All Quantum Devices

API References

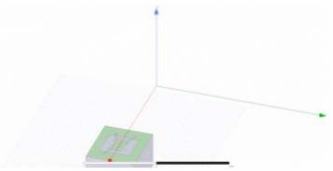
- Overview
- QDesigns
- QComponents



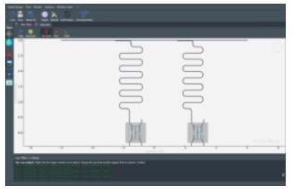
Eigenmode and EPR



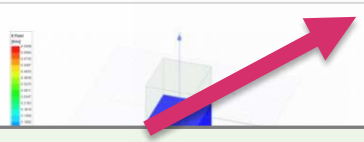
Impedance Analysis

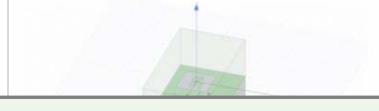


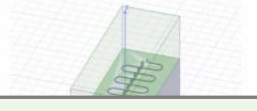
Capacitance Matrix and LOM Analysis

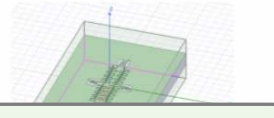


Sweep a option in QComponent, get Ansys HFSS driven-modal scatter matrix, impedance matrix, and admittance matrix.



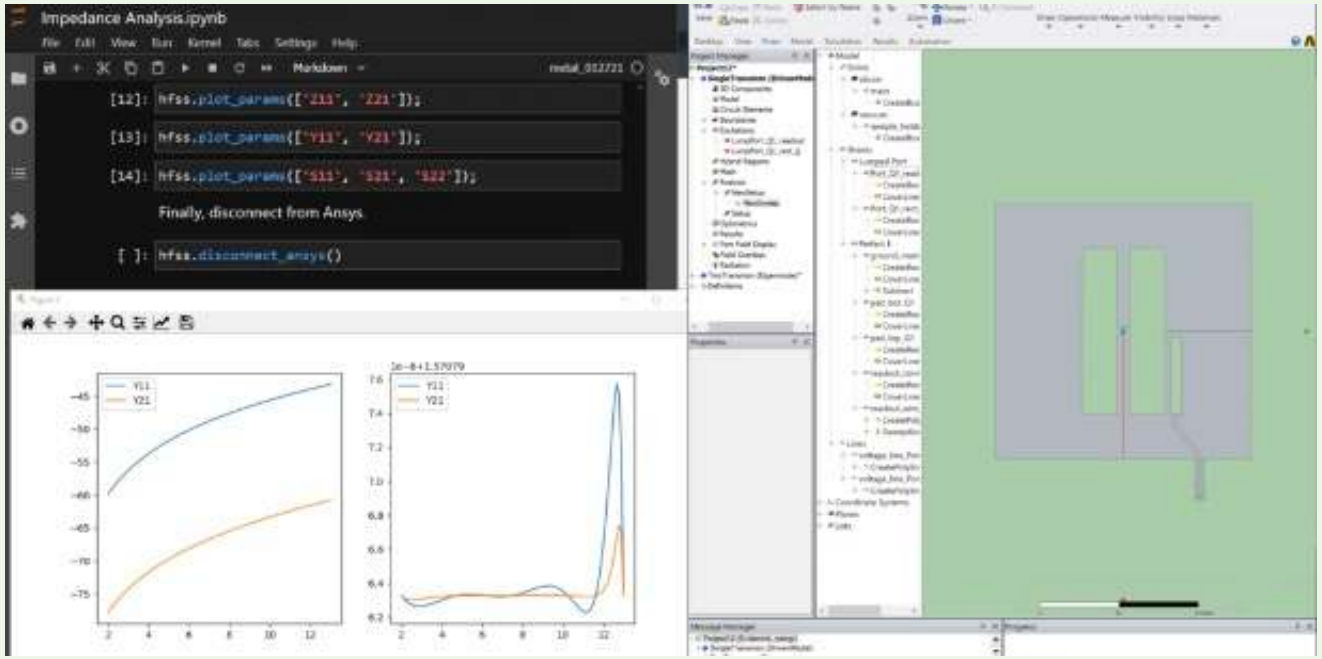






S, Z, Y Impedance Scattering

arXiv:1204.0587 ...



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

Tutorials: Quantum analysis library

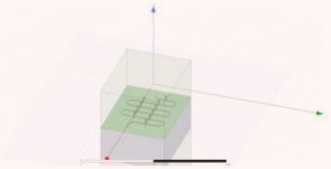
- Resonators
- Composite Bi-Partite Systems
- Qubit Couplers
- Input-Ouput Coupling
- Small Quantum Chips
- Design Flow

Libraries

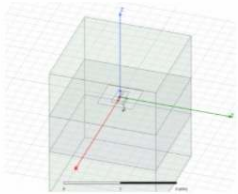
- All Quantum Devices

API References

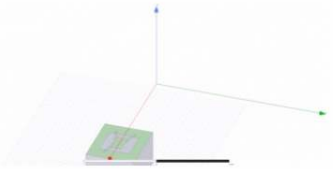
- Overview
- QDe



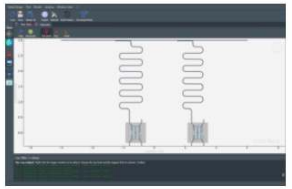
Eigenmode and EPR



Impedance Analysis



Capacitance Matrix and LOM Analysis



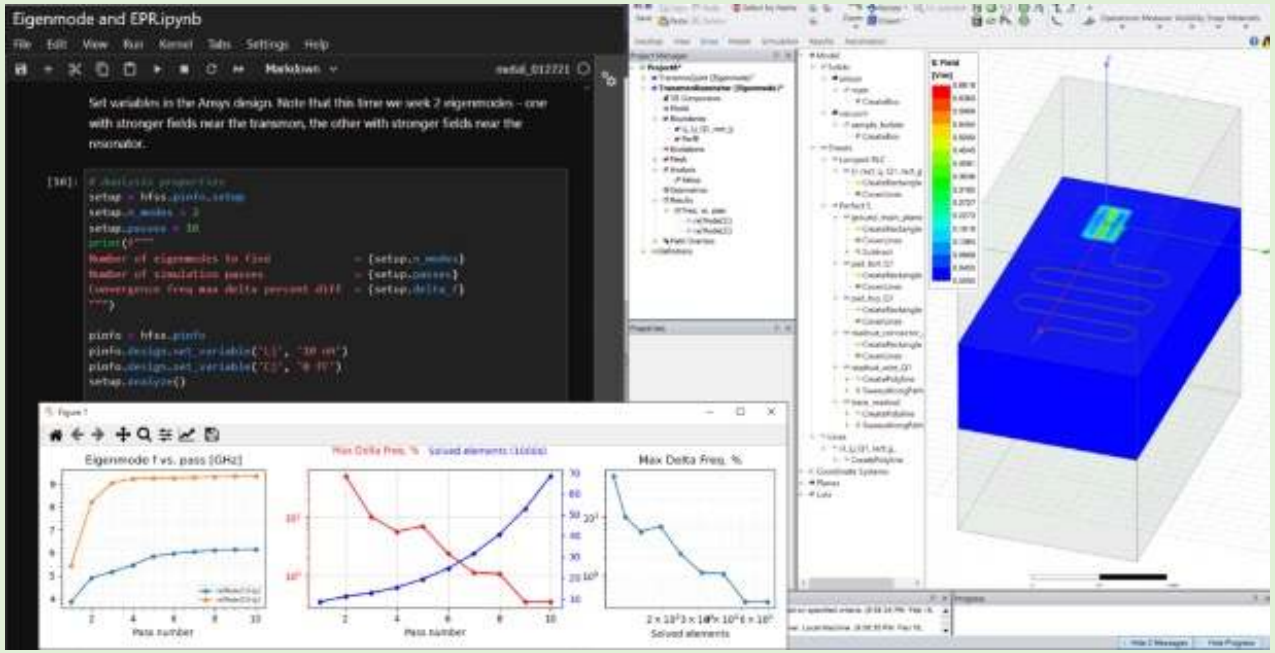
Sweep a option in QComponent, get Ansys HFSS driven-modal scatter matrix, impedance matrix, and admittance matrix.

Eigenmode

Energy-participation quantization of Josephson circuits

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

npj Quantum Information **7**, Article number: 131 (2021) | Cite this article



Minev arXiv: 1902.10355
Minev arXiv: 2010.00620

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

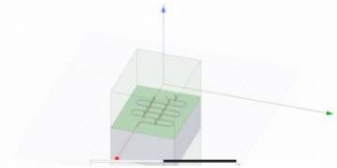
API References

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

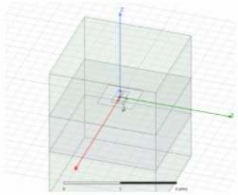
Code of Conduc

Qisk

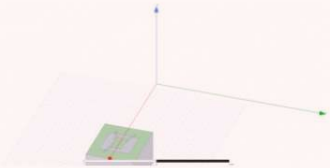
Tutorials: Quantum analysis library



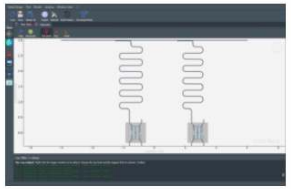
Eigenmode and EPR



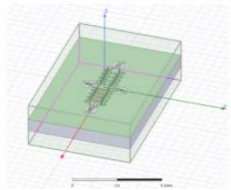
Impedance Analysis



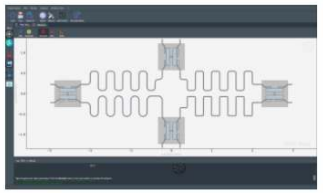
Capacitance Matrix and LOM Analysis



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



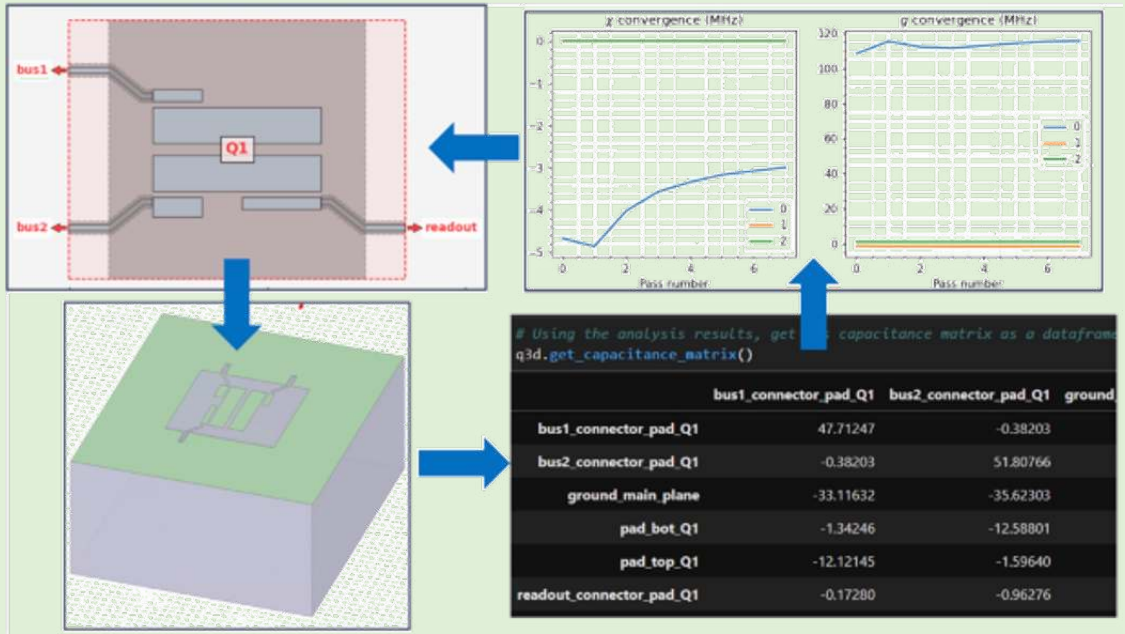
How to Render a Metal Design into Ansys



How to Render a Metal Design into GDS

Capacitive

arXiv:2103.10344 ...



	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

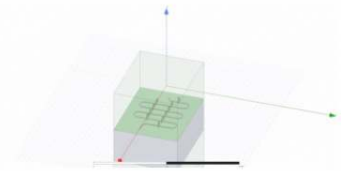
Libraries

- All Quantum Devices

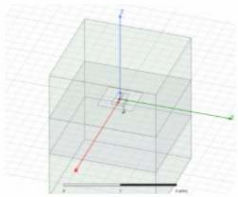
API References

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

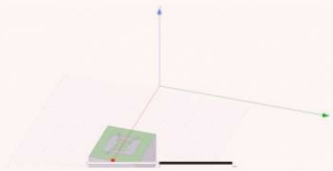
Code of Conduc



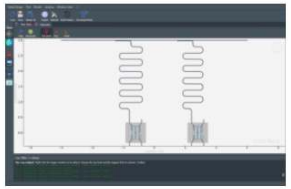
Eigenmode and EPR



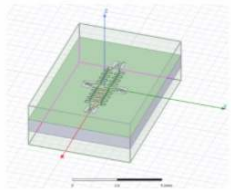
Impedance Analysis



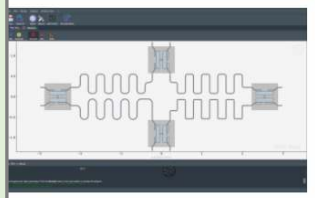
Capacitance Matrix and LOM Analysis



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

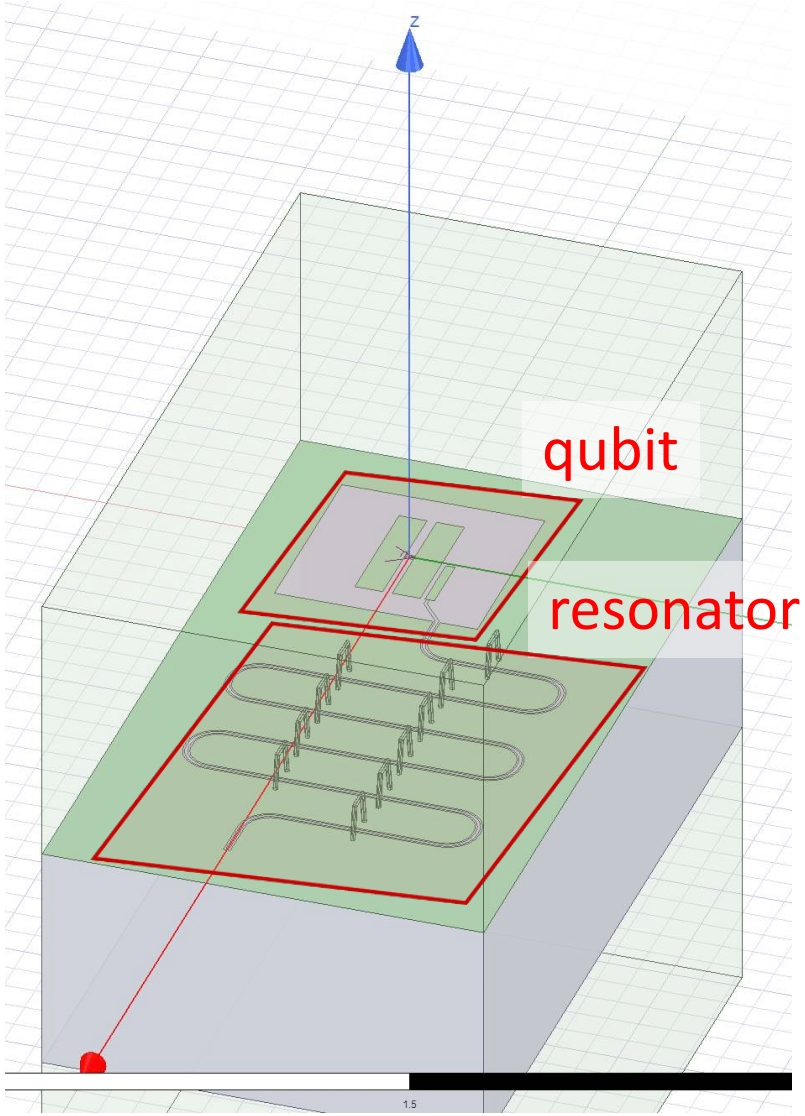
Circuit quantum electrodynamics (cQED) with modular quasi-lumped models

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

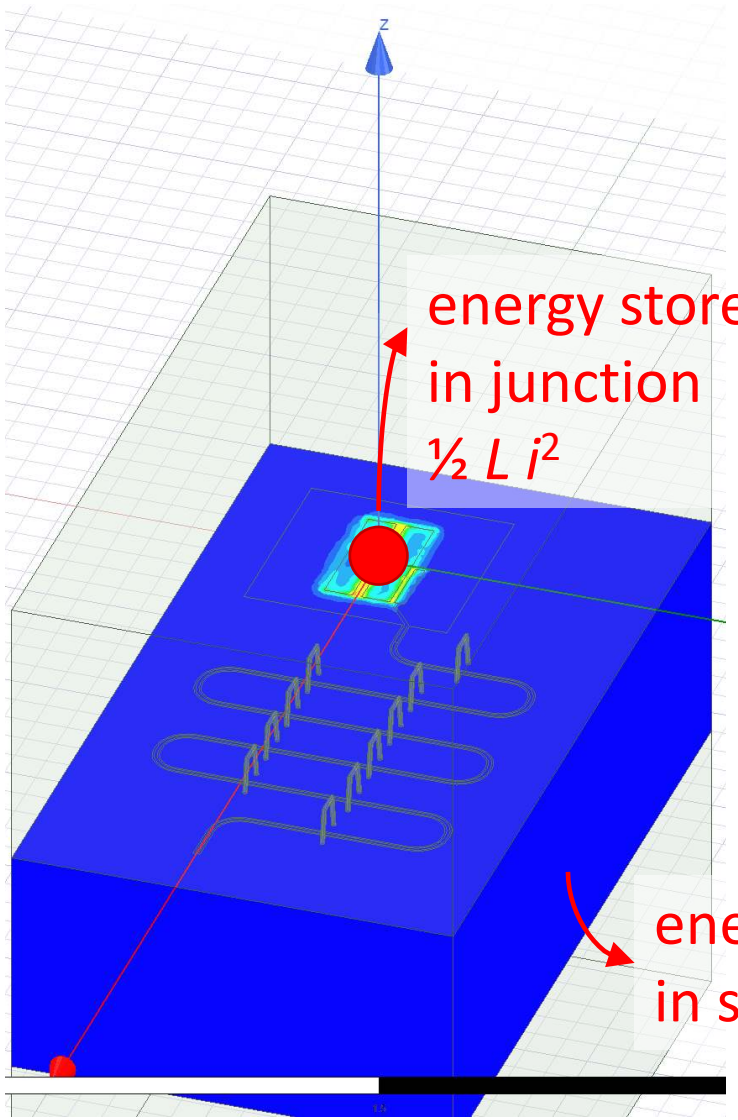
WIP: General capacitive analysis code
map to known building blocks: e.g.,
transmon, fluxonium, zero-pi, etc.

Example of step 2 with EPR

Physical (linear) model



Eigenmodes



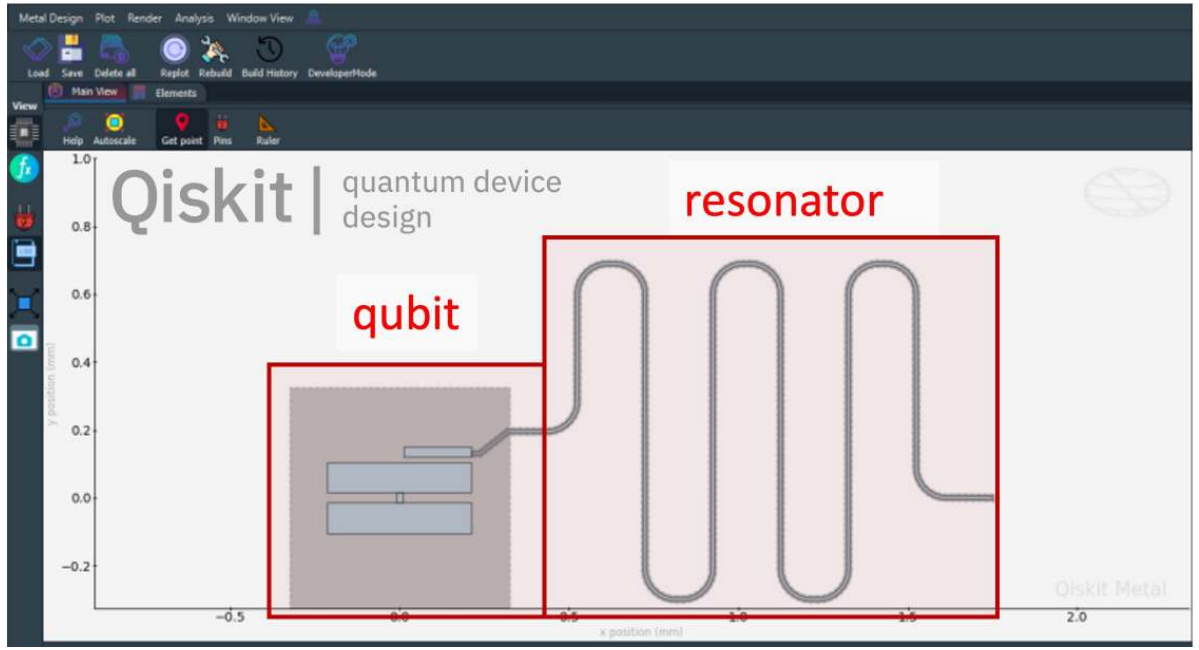
Non-linear device params

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

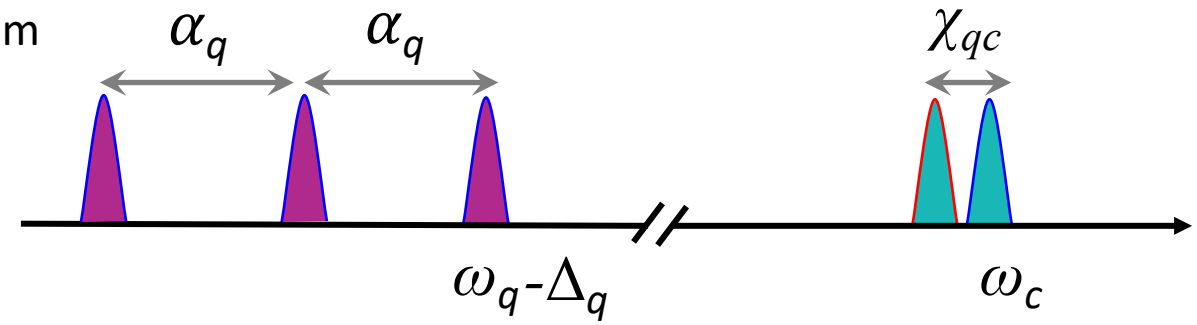
Dissipation budget

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

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

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

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

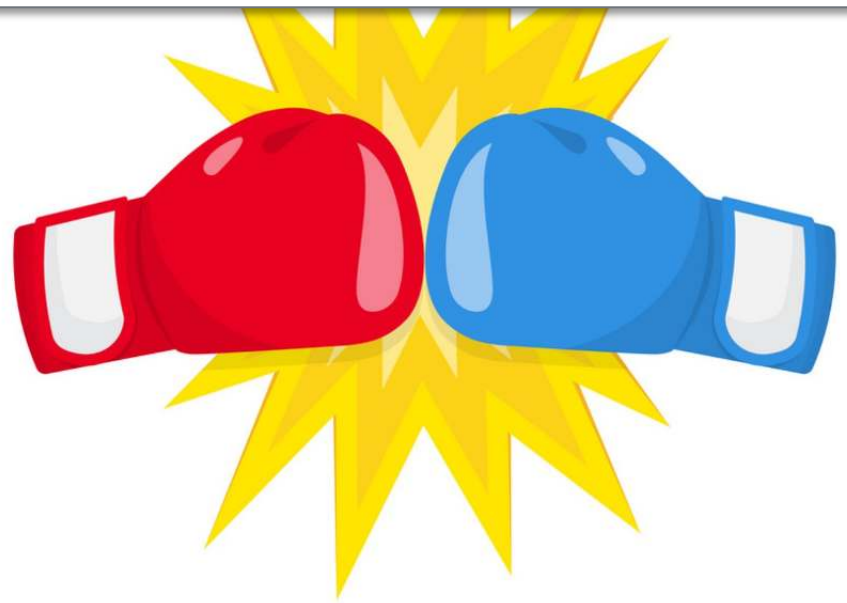
Planar devices & comparison to other methods

Quantum Physics

[Submitted on 2 Feb 2021]

Exploiting dynamic quantum circuits in a quantum algorithm with superconducting qubits

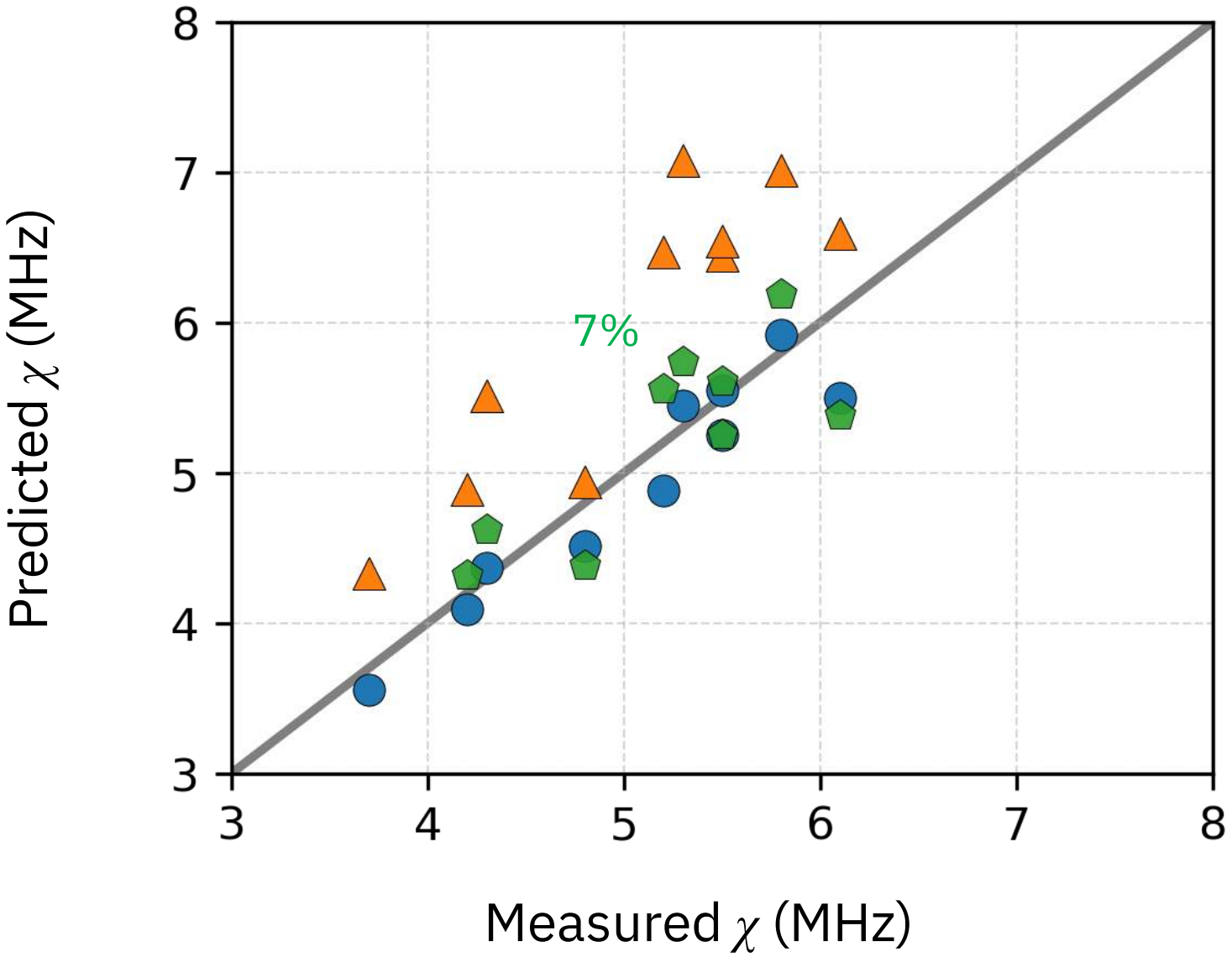
Antonio D. Corcoles, Maika Takita, Ken Inoue, Scott Lekuch, Zlatko K. Minev, Jerry M. Chow, Jay M. Gambetta

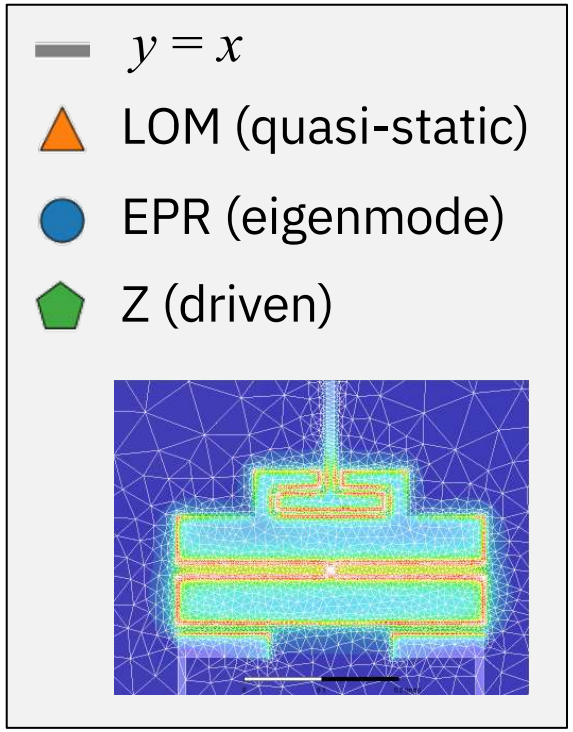


Automated with Qiskit | quantum device design

image: makc76

Measured vs. predicted: qubit-readout cross-Kerr





Marker size indicates error bars.

From layout to time evolution simulation in one simple notebook

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()
```

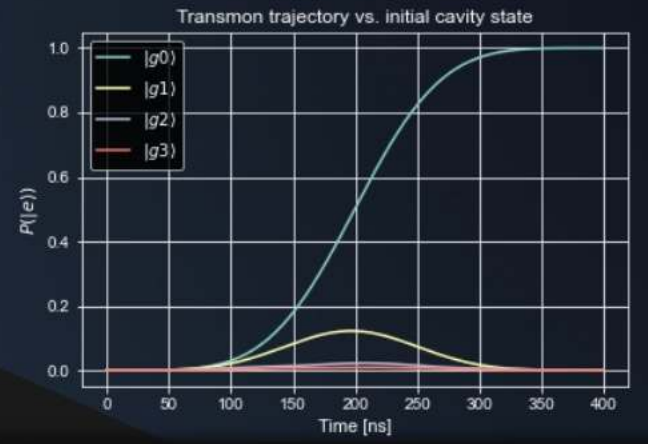
Print the results

```
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
```



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
        )
```

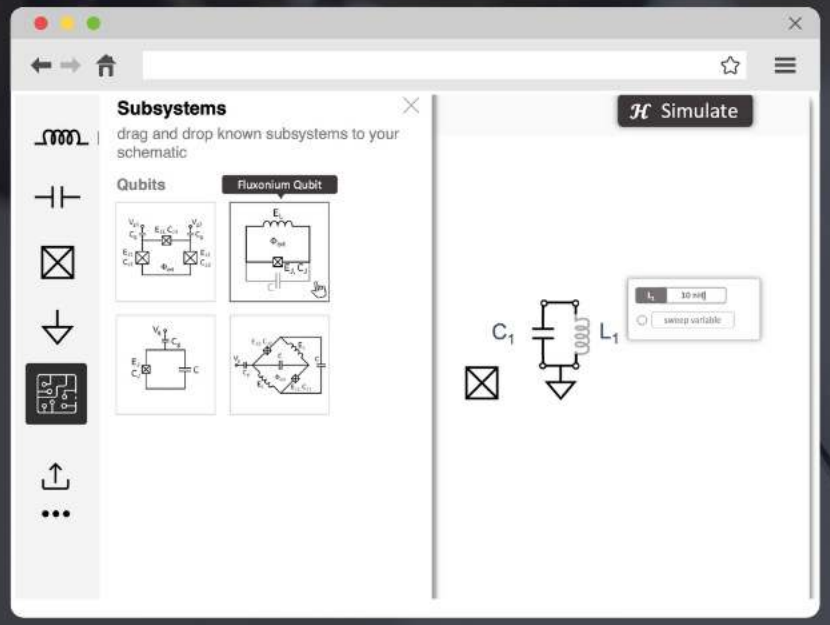
```
100%|
| 51/51 [00:01<00:00, 41.64it/s]
```

```
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



Cloud-based WebApp

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

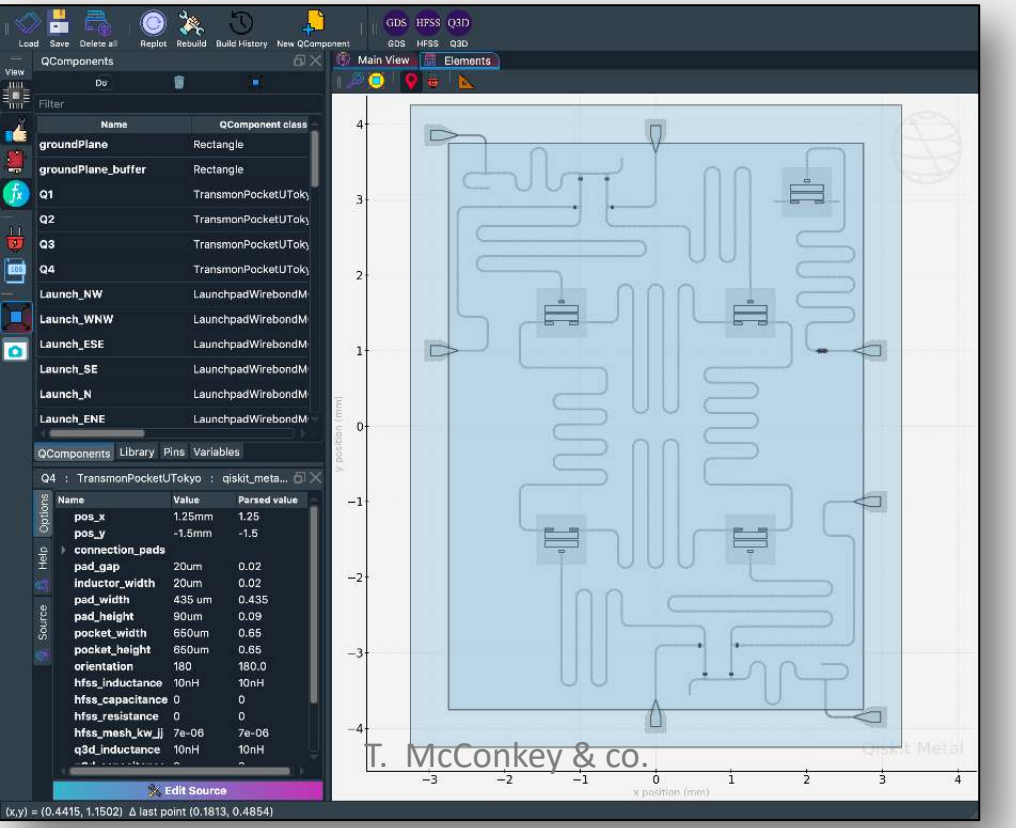
Community

Education

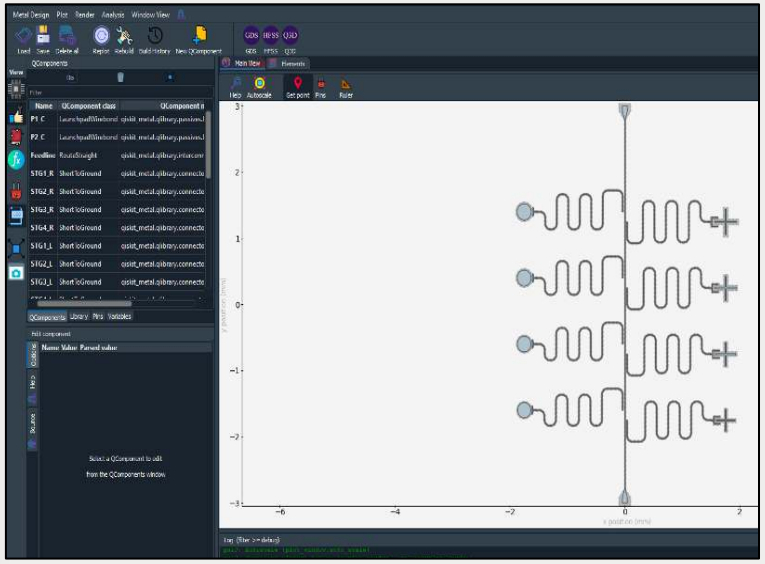
Workforce development

Qiskit | quantum device design

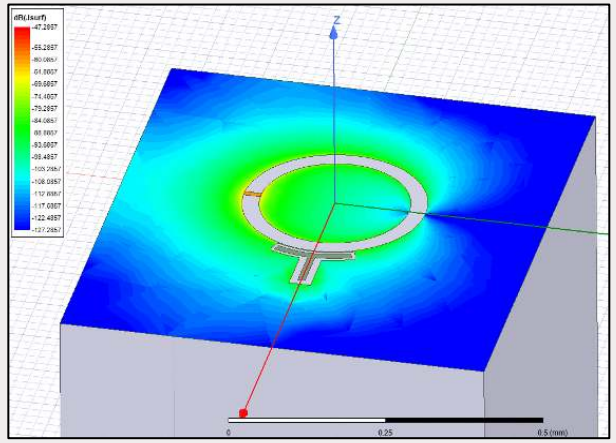
IBM 5Q Tsuru U Tokyo



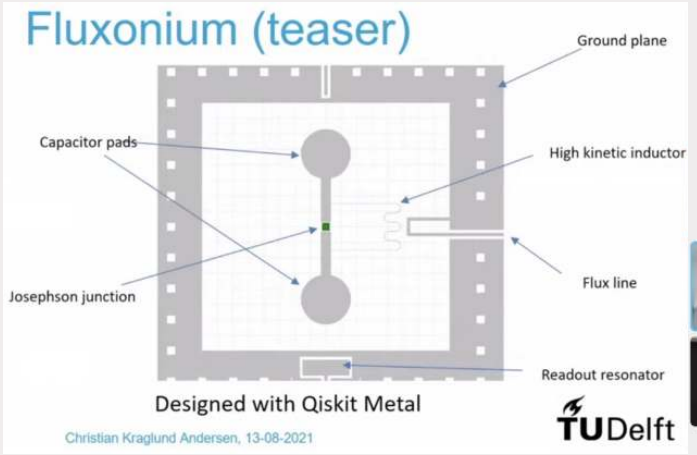
Chalmers 8Q

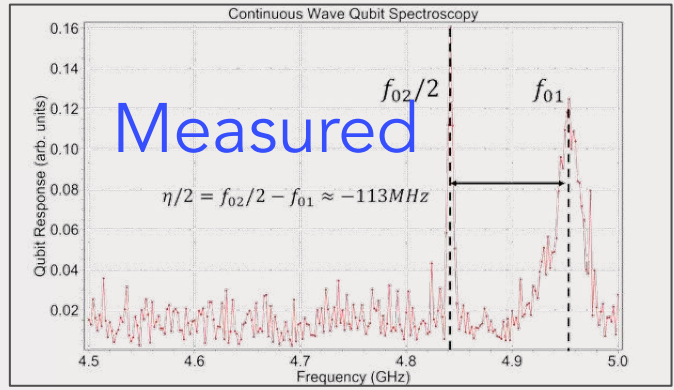


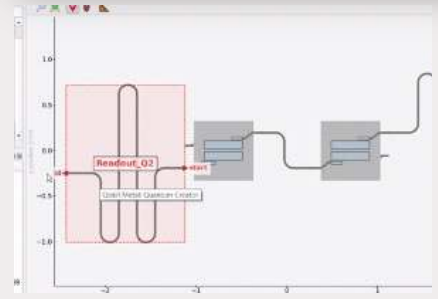
Simulated



TU Delft Fluxonium

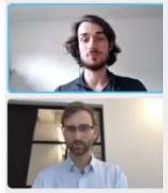






APS physics MARCH MEETING 2021
MARCH 15-19 ONLINE

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



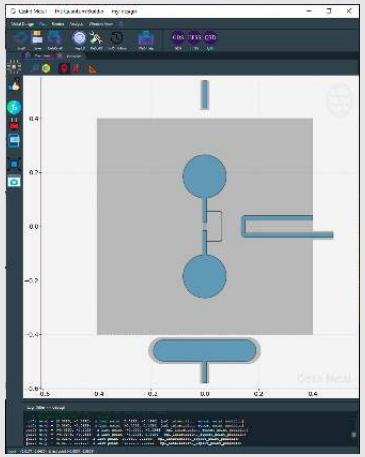
...

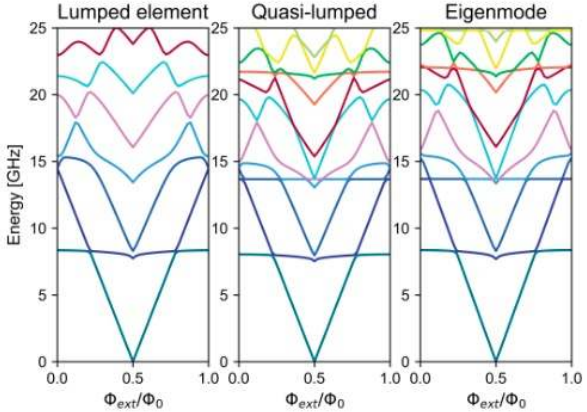
IEEE QUANTUM WEEK
IEEE International Conference on Quantum Computing and Engineering — QCE21

Use by researchers, continued

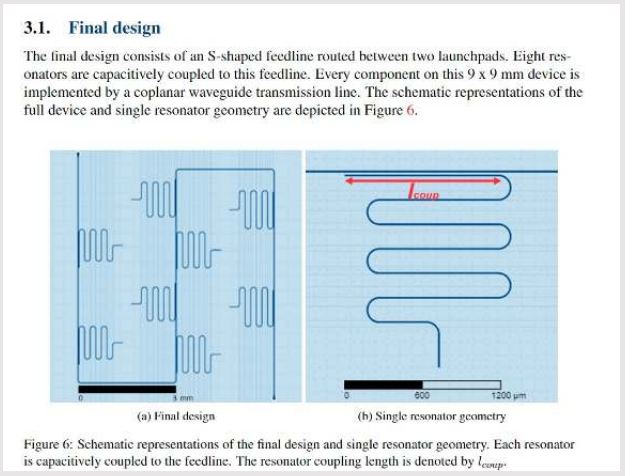
QuTech / **Technische Universiteit Delft**

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**

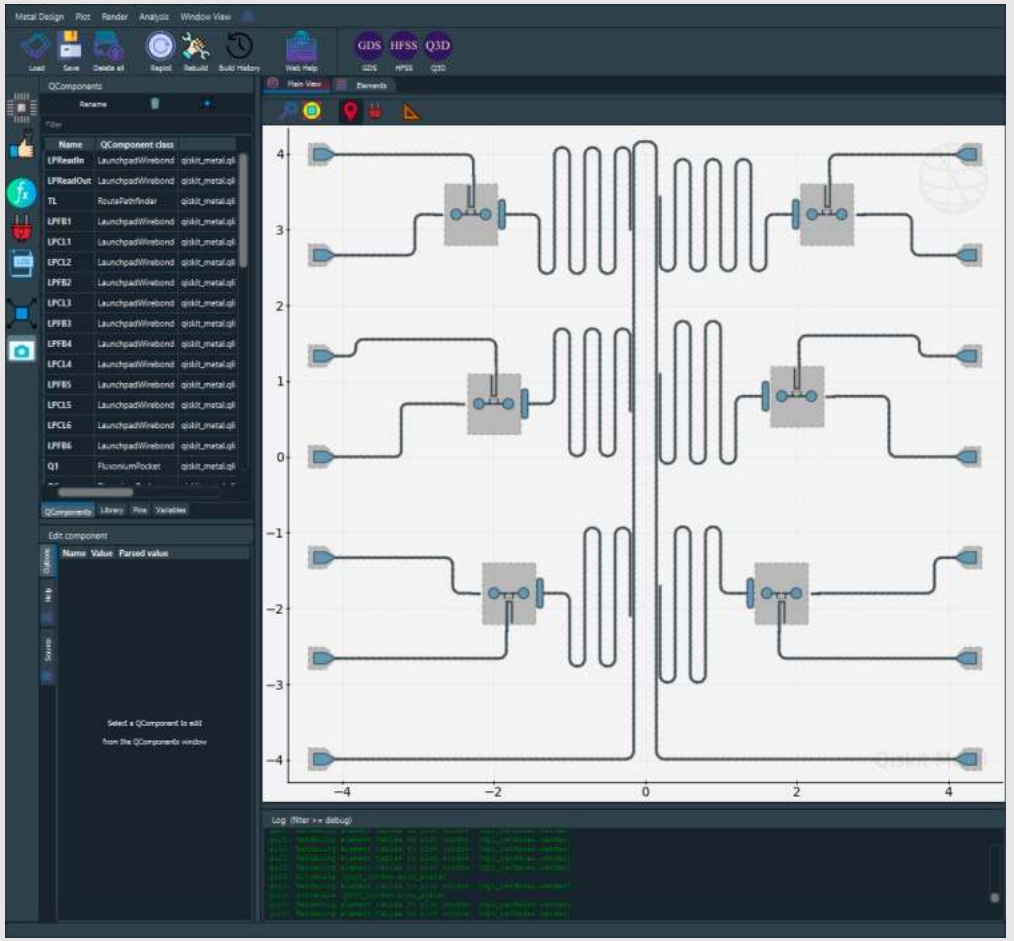
The final design consists of an S-shaped feedline routed between two launchpads. Eight resonators are capacitively coupled to this feedline. Every component on this 9 x 9 mm device is implemented by a coplanar waveguide transmission line. The schematic representations of the full device and single resonator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

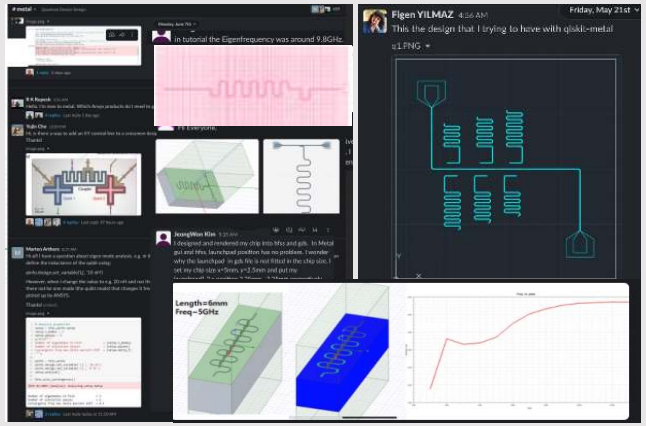


Qiskit | quantum device design

Active user base and content

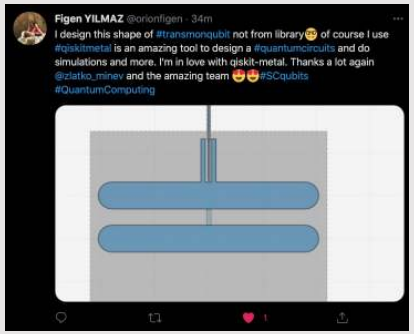
900+ users
4,500+ Q&A

Est. ½- 1/3 of quantum hardware people

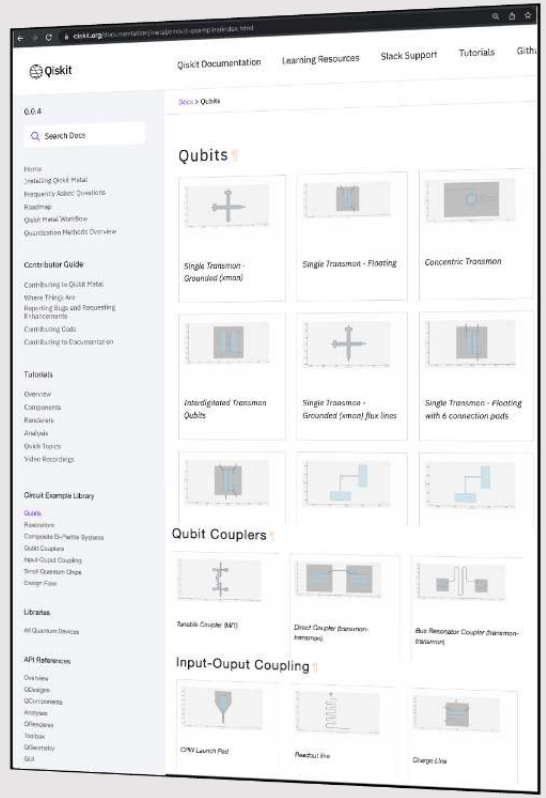


GitHub

	Star	Fork
/qiskit-metal	193	133
/qiskit-nature	155	139
/qiskit-optimization	122	82
/qiskit-finance	112	82



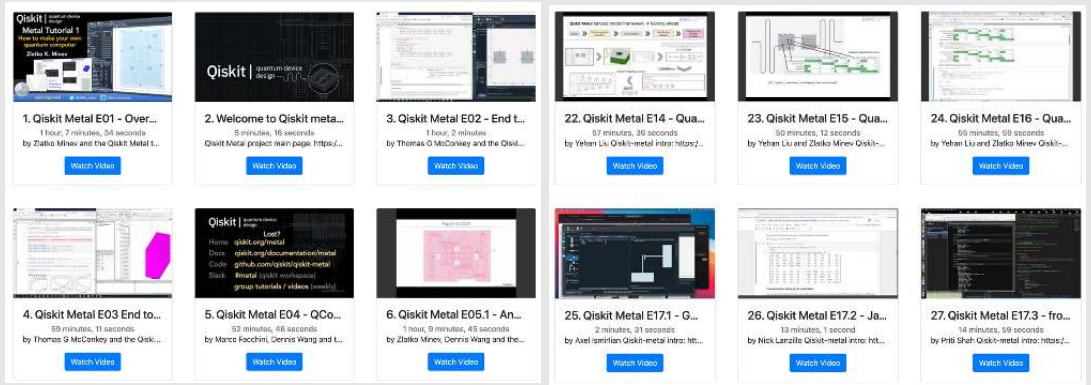
47 tutorials
+ 27 demos



...

23 hours video tutorials

29+ videos

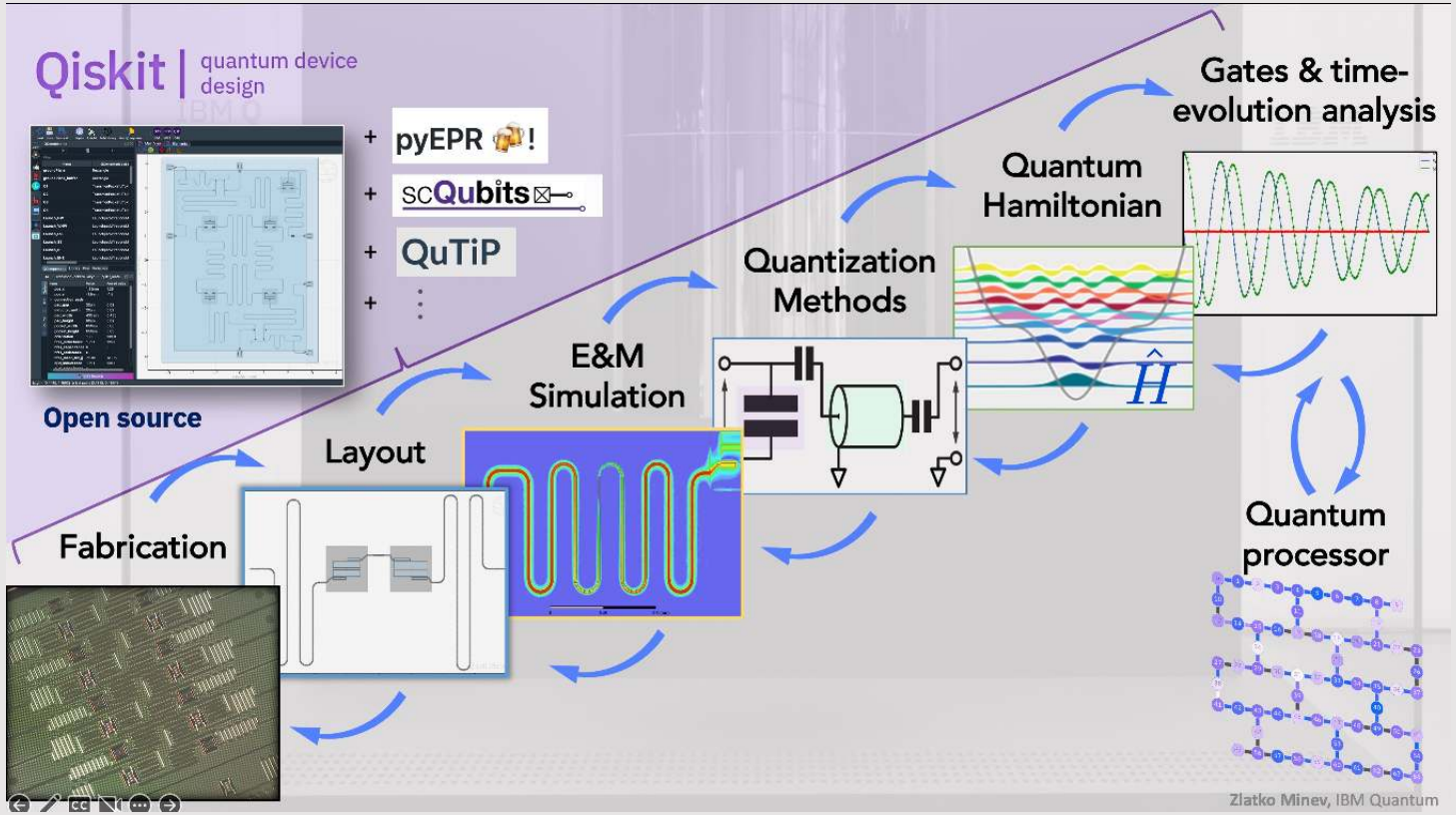


60+ user session

users + devs working groups

Zlatko Minev, IBM Quantum (48)

Building it together



Open source

Call for community participation

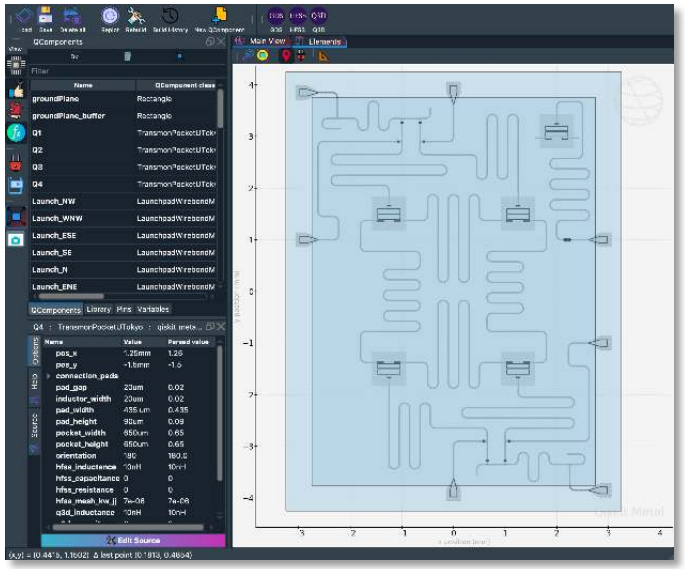
Education
- tutorials
- courses?
- summer schools with hands on (see QGSS 2020 Z. Minev from *Introduction to Quantum Computing and Quantum Hardware)*
- ...

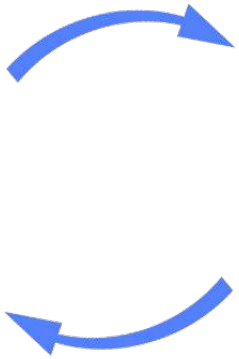
@zlatko_minev

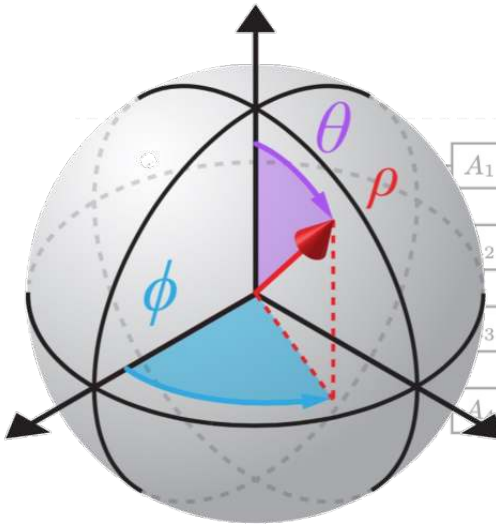
Overview of Quantum Hardware Design

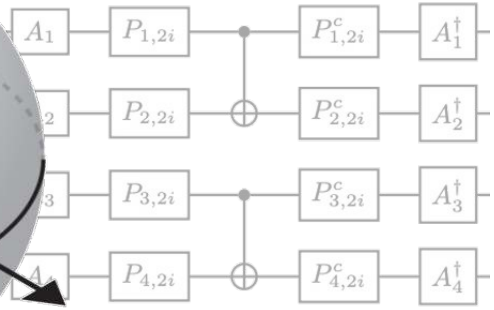
Energy, Circuits, and Metal: Qiskit Metal

Thank you!









Zlatko K. Minev

IBM Quantum

@zlatko_minev

zlatko-minev.com

qiskit.org/metal

EXTRA

Thank you!

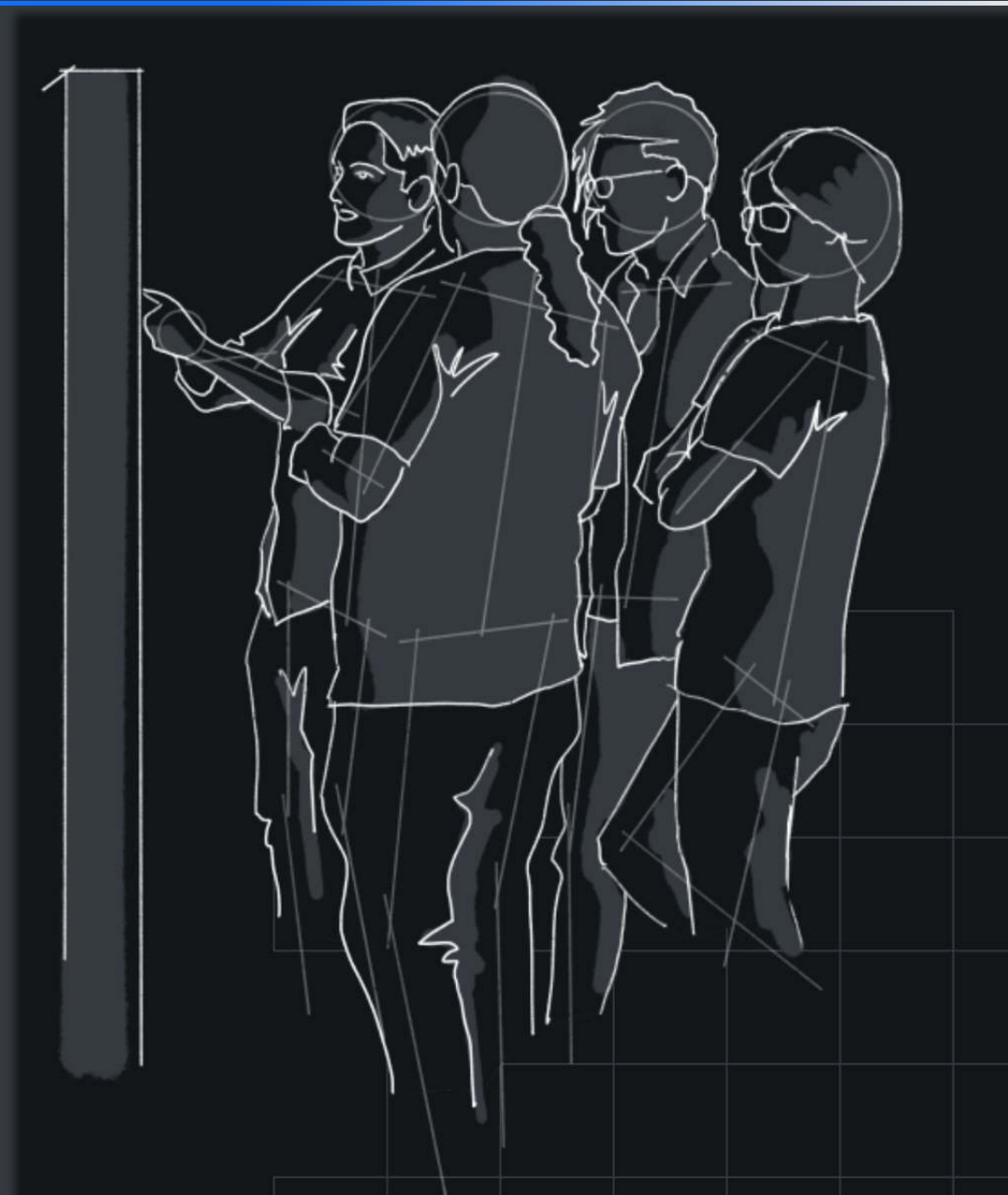
Building together

Call for community participation

Open source

Education
eg. Summer school lectures by Z. Minev from
Introduction to Quantum Computing and Quantum Hardware
...

qiskit.org/metal



@zlatko_minev

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

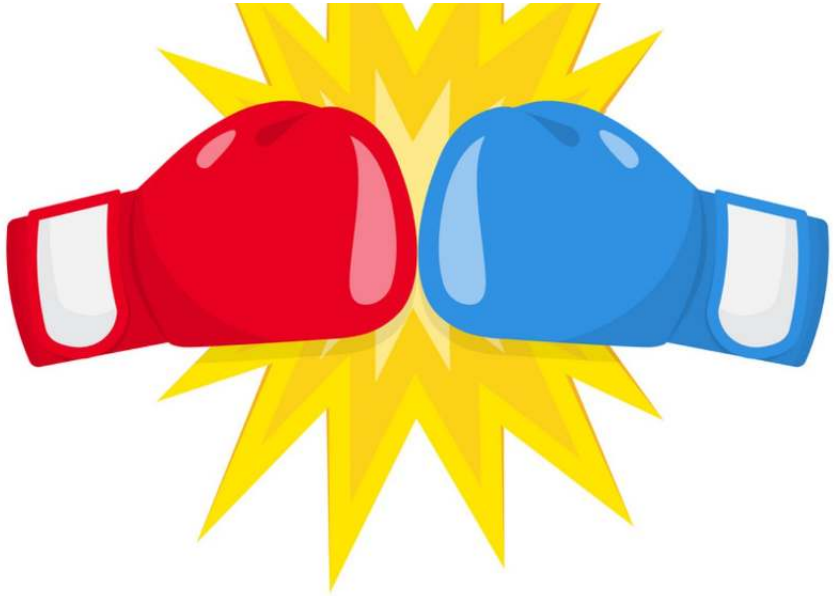
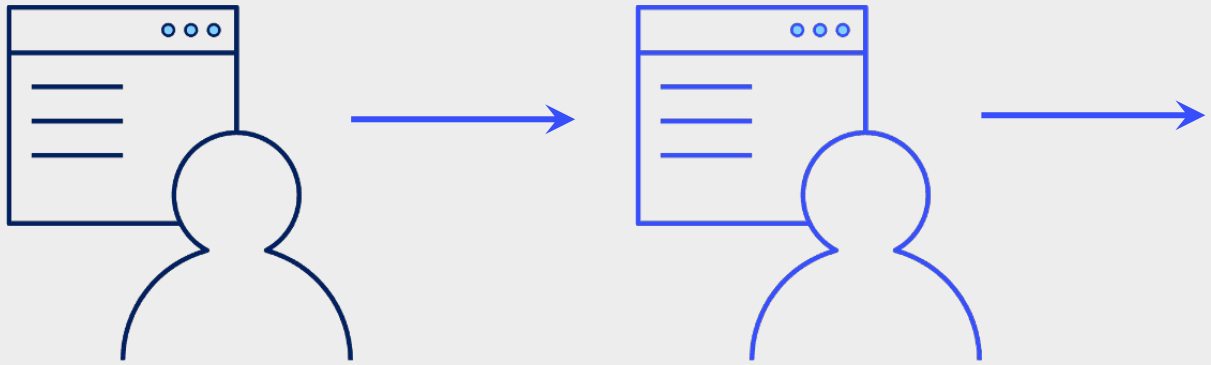


image: makc76

Quantum Device Design

Make it easy!



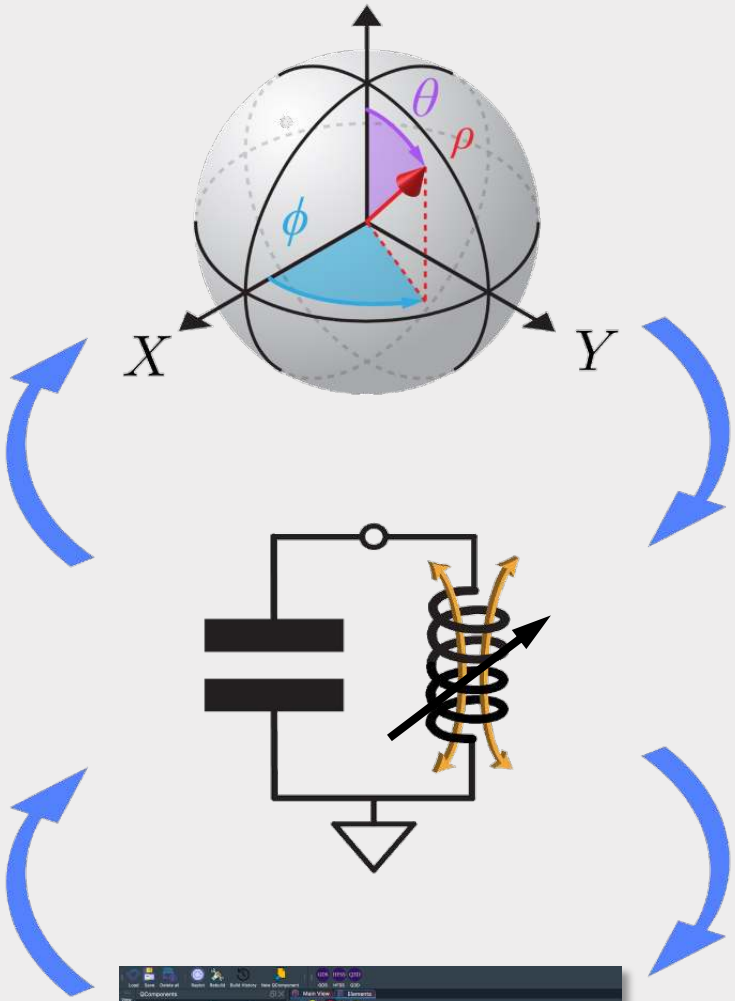
Classical developer

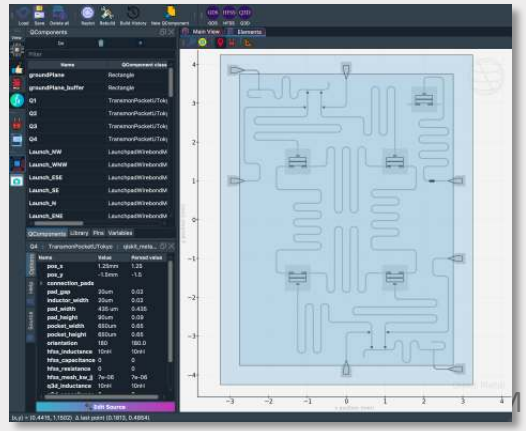
Enabling innovation of quantum computer hardware & analysis

Quantum

Engineering

API
UI
Layout



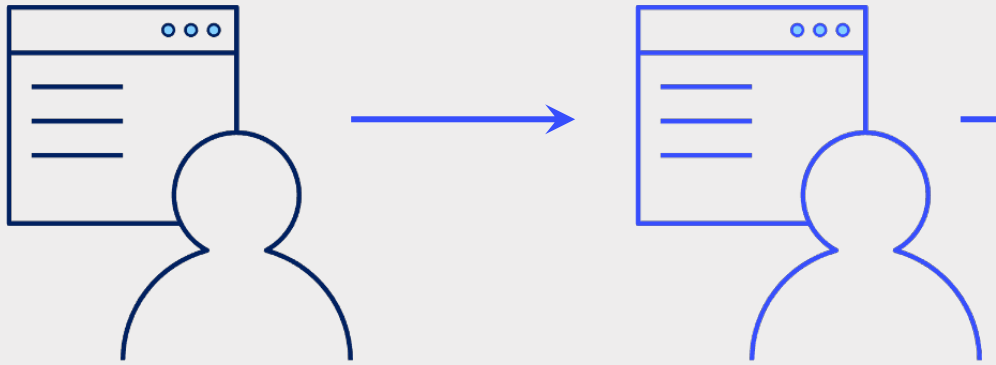


Open source

qiskit.org/metal

Fueling the fire of innovation

Make it easy!



Classical developer

Enabling innovation of quantum computer hardware & analysis

qiskit.org/metal

Forbes

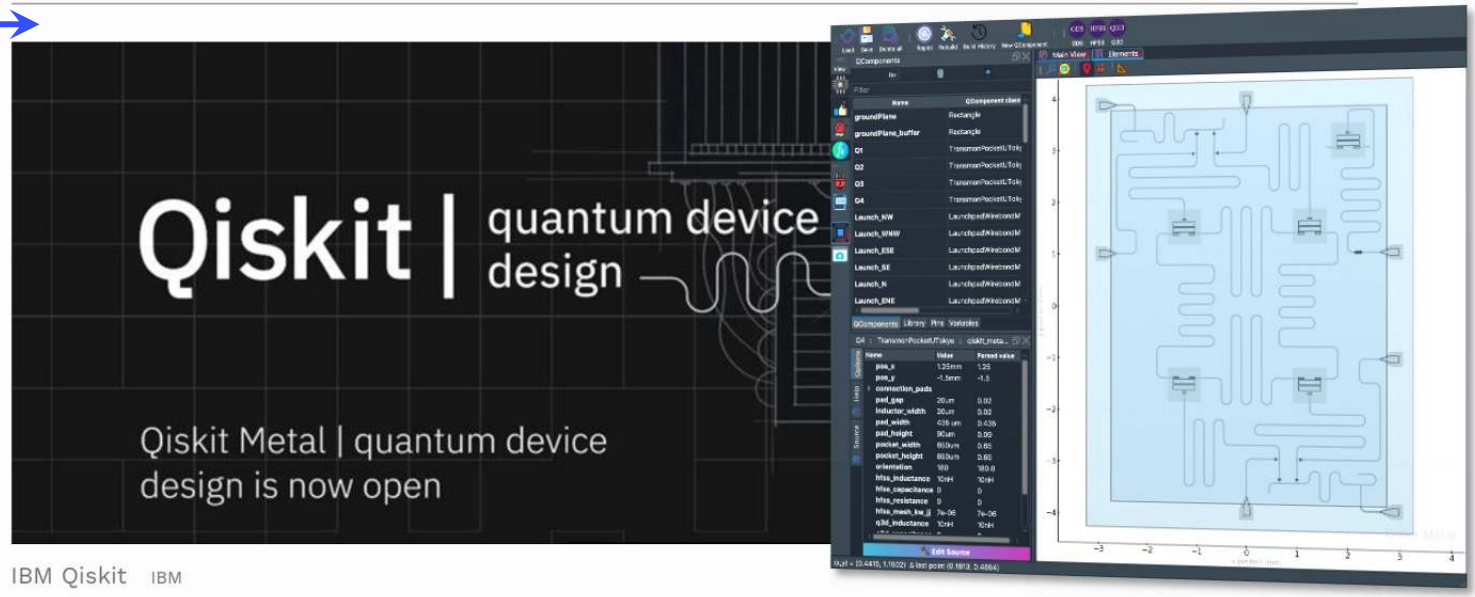
You Don't Have To Be A Rocket (Or Quantum) Scientist To Design A Quantum Computer Chip Using IBM's New Tool Called Qiskit Metal

Paul Smith-Goodson Contributor

Moor Insights and Strategy Contributor Group

Cloud

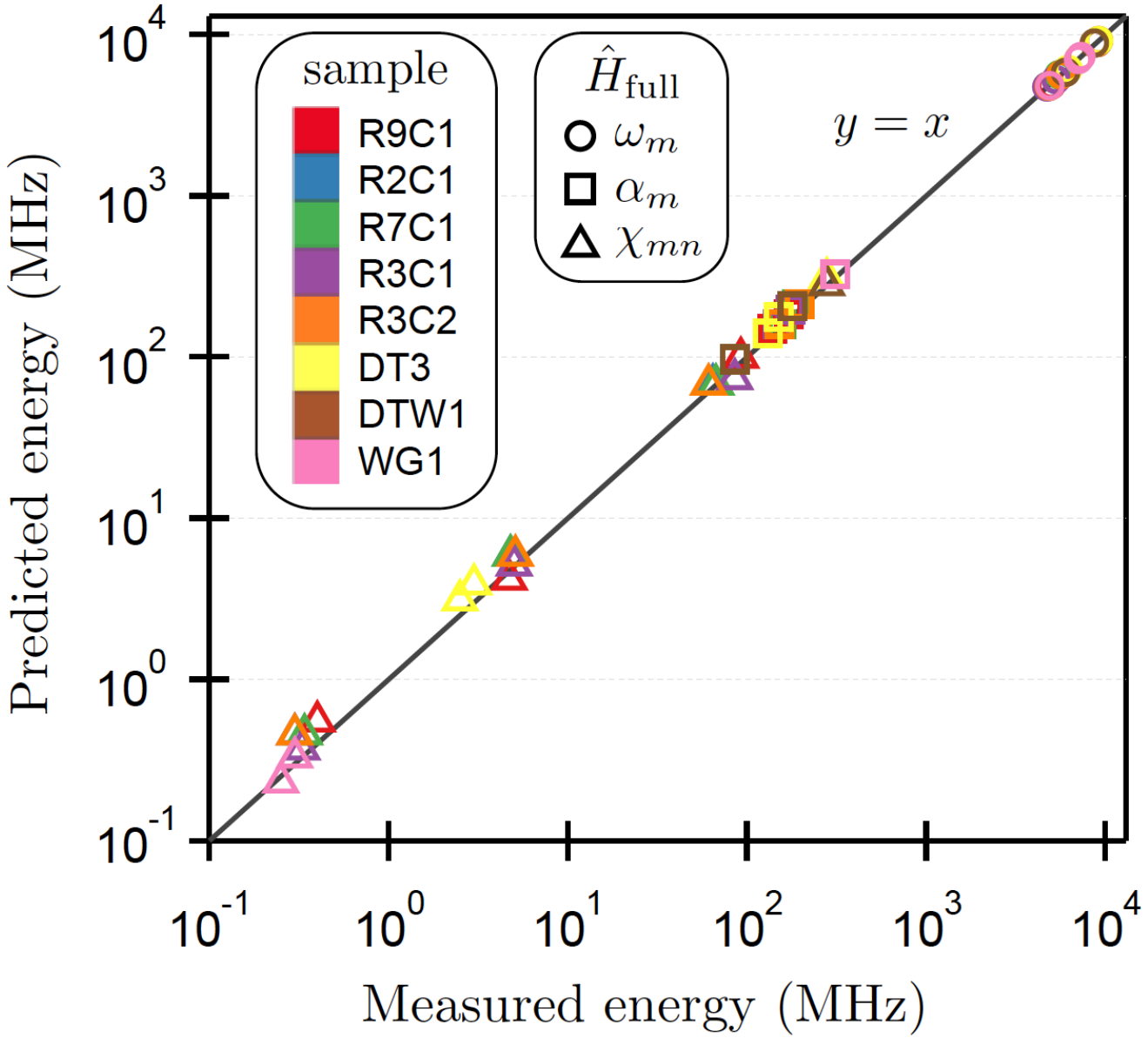
Analyst-in-residence, Quantum Computing

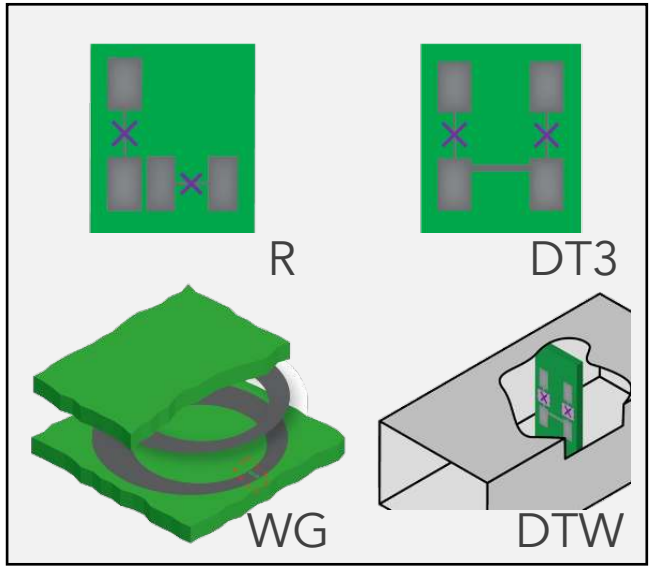


IBM Qiskit IBM

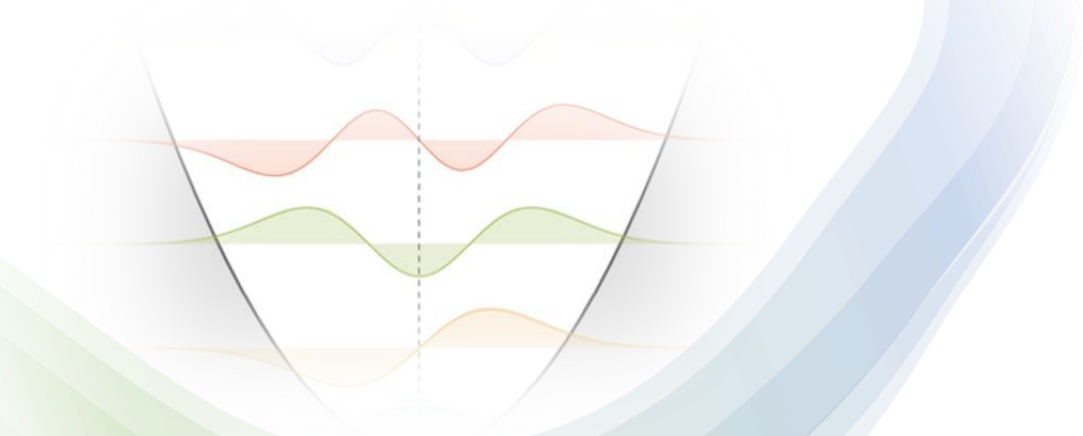
Intuitively, almost everyone can appreciate how difficult and knowledge-intensive it is to design, develop, analyze, and simulate a quantum computer chip. Without years

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)



"Quantum phenomena
do *not* occur in a Hilbert space,
they occur in a laboratory."

Asher Peres