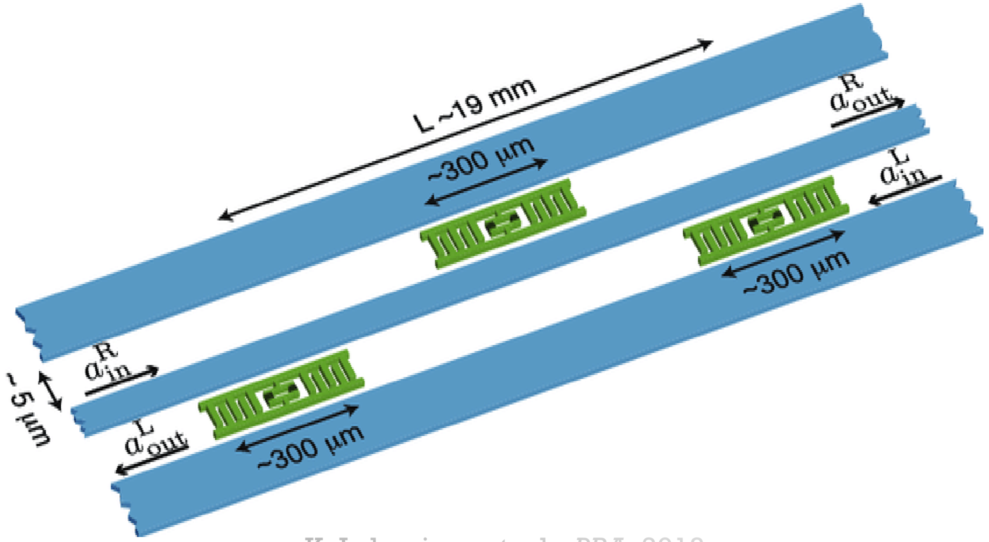
بسم الله الرحمن الرحیم



K. Lalumiere et. al., *PRA*, 2013

طراحی و شبیه سازی مدارهای رایانه های کوانتومی ابررسانا

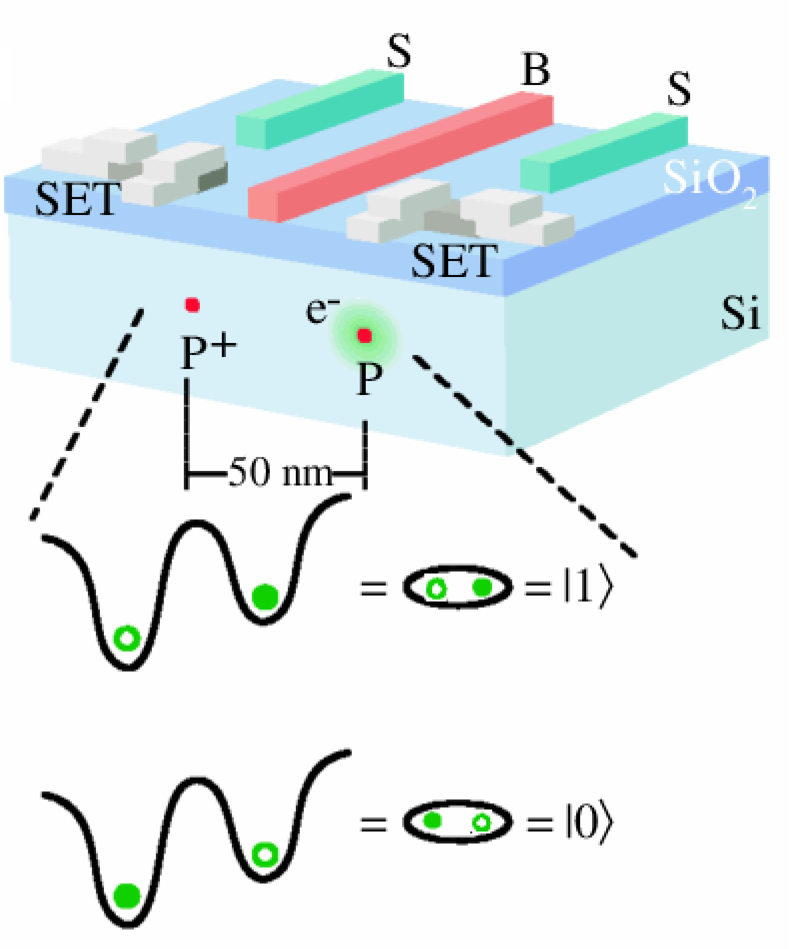
حسام زندی

۱۴۰۲/۱۱/۲۷

مرکز تحقیقات فناوری‌های کوانتومی ایران

Kharazmi University

دانشگاه صنعتی خواجه نصیرالدین طوسی

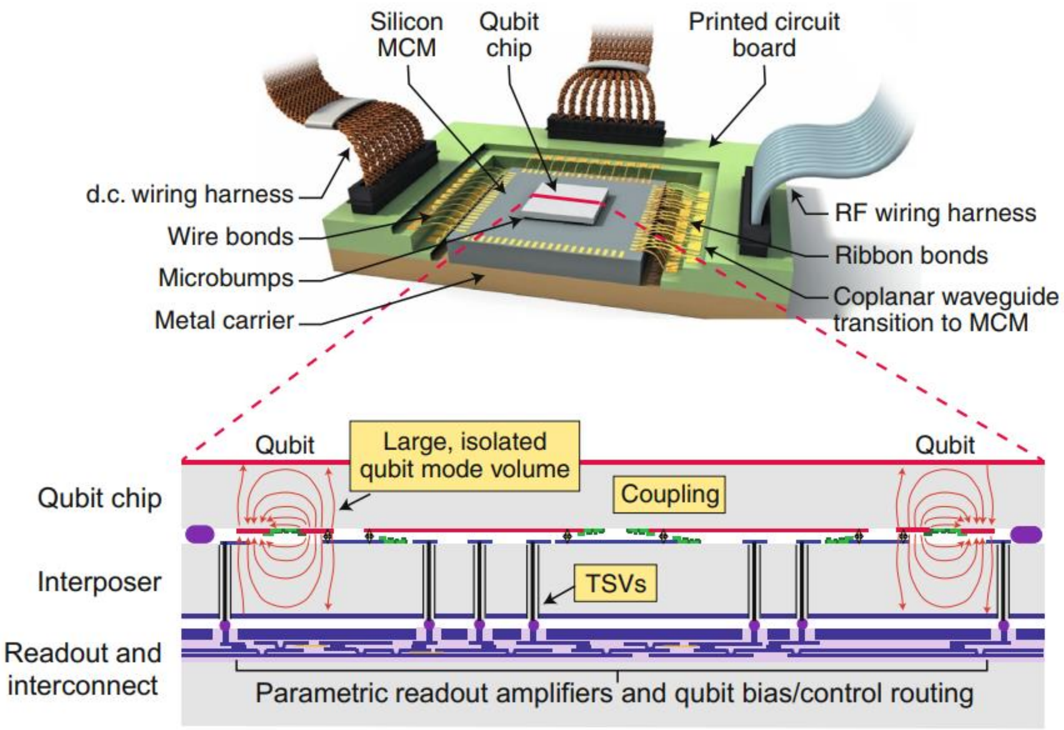


اهداف این ارایه

- آشنایی مختصر با بخش های مختلف یک تراشه محاسبات کوانتومی
- آشنایی با مبانی اولیه شبیه سازی در بخش های مختلف
- معرفی نمونه های طراحی و شبیه سازی در بخش های مختلف
- بررسی نحوه استفاده از برخی کتابخانه ها و نرم افزارها

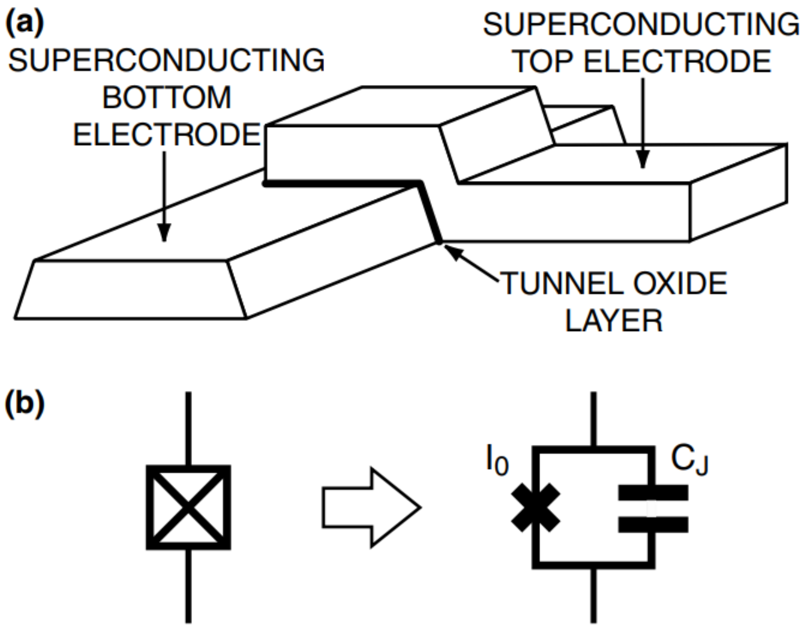
  در شبیه سازی مدارهای کوانتومی ابررسانا

فهرست مطالب



A. Blais et. al., *nature*, 2020

Specific Josephson inductance: $L_J = \frac{\Phi_0}{2\pi I_c}\frac{1}{\cos(\phi)}$



$E_J = \frac{\Phi_0 I_c}{2\pi}$: Josephson energy

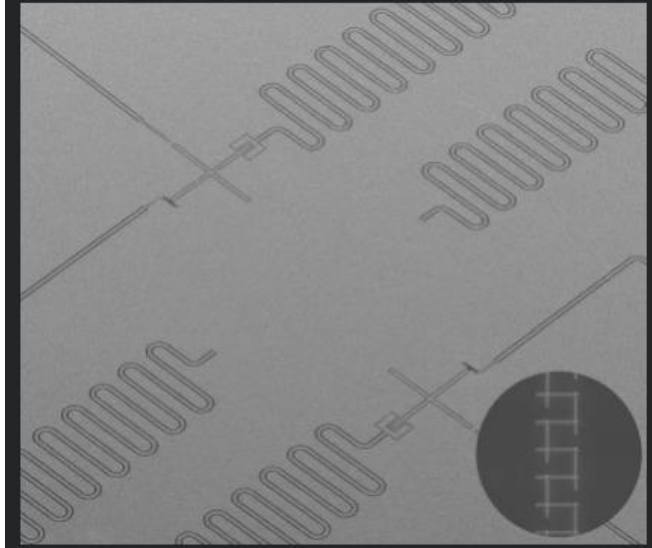
$E_c = \frac{e^2}{2C_j}$: charging energy

The Hamiltonian could be defined in terms of E_J and E_c:

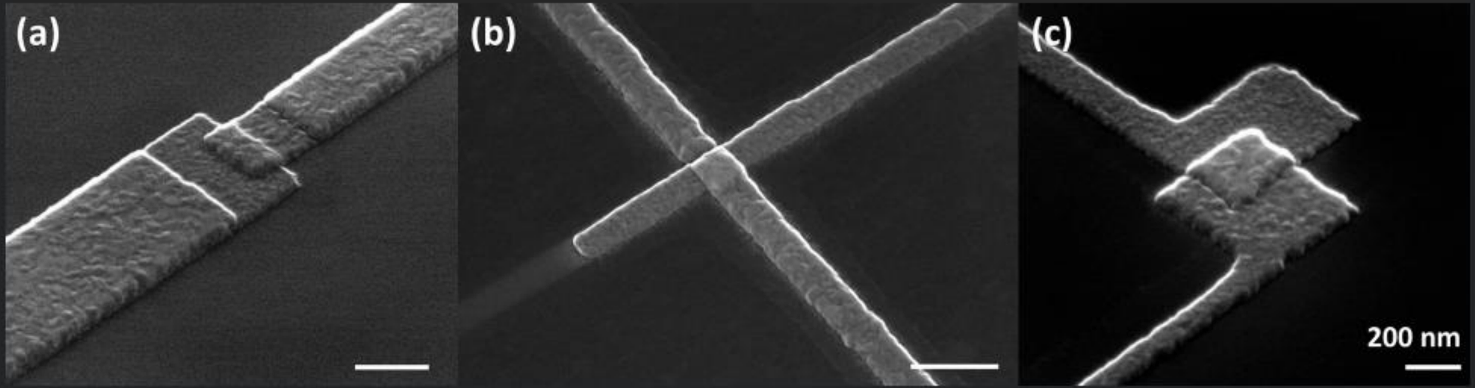
$$\mathcal{H} = 4E_c\hat{n}^2 + E_J\big(1 - \cos(\hat{\phi})\big)$$

$$\hat{Q} \leftrightarrow \hat{n} \quad , \quad \hat{\Phi} \leftrightarrow \hat{\phi}$$
$$C \leftrightarrow E_C \quad , \quad L \leftrightarrow E_J$$

پیوند جوزفسون



Xmon qubit with junction array.



Our capability of fabricating different types of junctions. (a) Junction with Dolan-bridge. (b) Cross junction (Manhattan style. (c) Large-area junction.

Institute of Physics · Academia Sinica · Taiwan

شبیه سازی پیوند جوزفسون

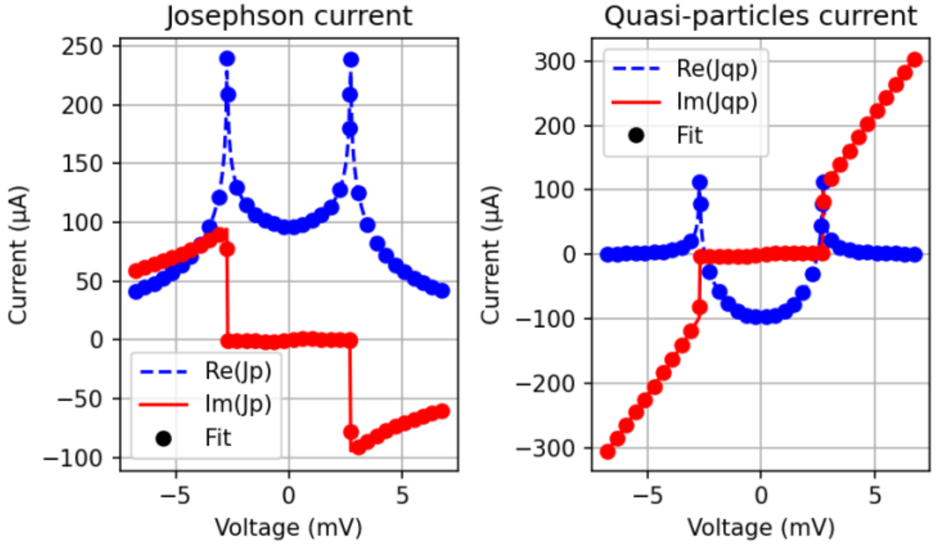
Jspice3 Circuit Simulator

The ***Jspice3*** built-in model library provides the following devices:

- Bipolar Transistor
- Capacitor
- Junction Diode
- Inductor and Coupled Inductors
- JFET
- **<u>Josephson Junction</u>**
- Lossy Transmission Line
- MOSFET
- CRYO-MOSFET
- MESFET
- Resistor
- Switch
- Lossless Transmission Line
- Lumped URC Line
- Dependent and Independent Current and Voltage Sources

External

Internal



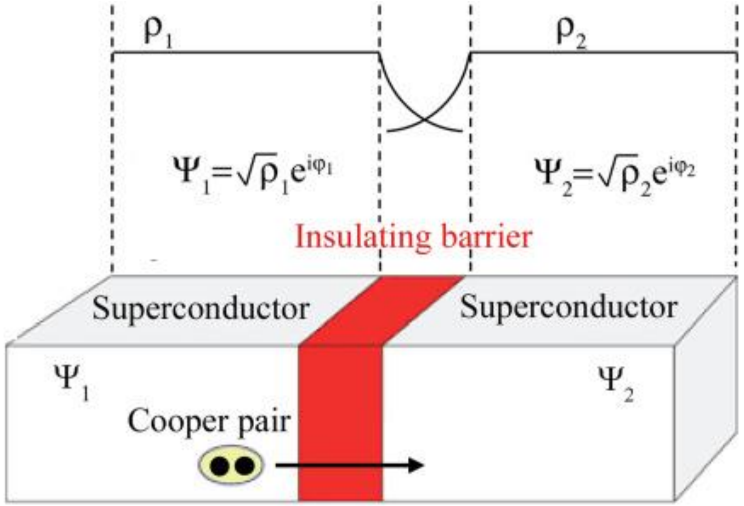
Time-domain simulator of Josephson junctions based on the BCS theory

L. Iwanikow, P. Febvre, Submitted to ArXiv – 17 January 2023

http://www.wrcad.com/jspice3.html

For Simulating a JJ in Comsol:

Setting up a mathematical model that describes the behavior of the junction and its interactions with the surrounding environment.

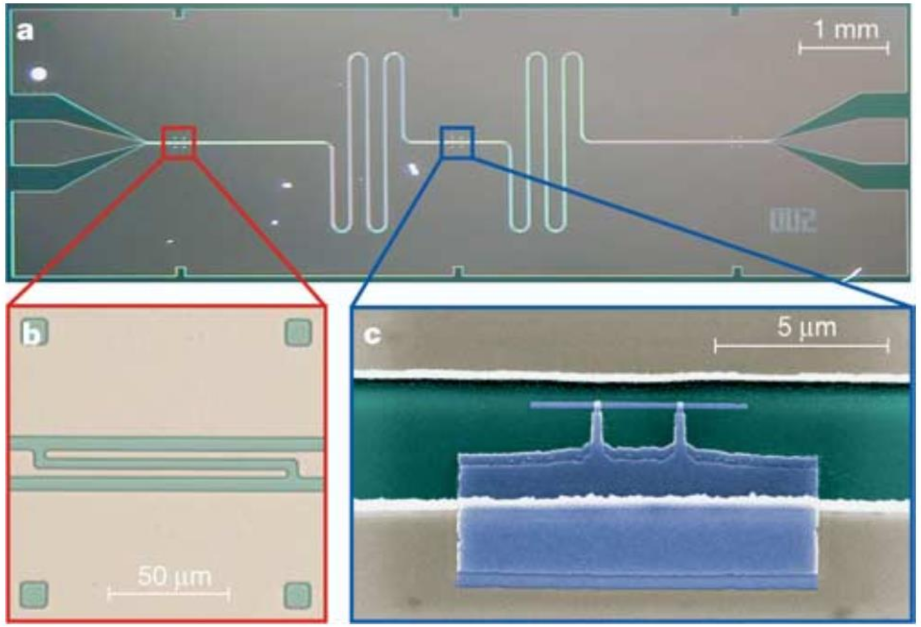


General steps:
1. Create a new model in Comsol.
2. Define the geometry of the junction.
3. Define the **material properties**.
4. Define the initial conditions and boundary conditions.

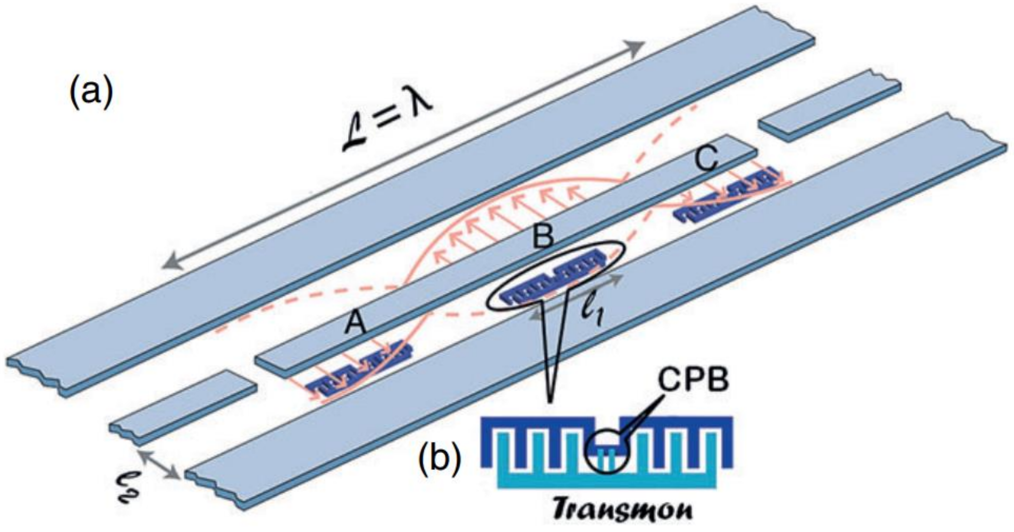
it may be necessary to include additional physics and coupled equations, such as heat transfer.

Related Book: Shortcut to superconductivity – superconducting electronics via comsol modeling - 1st ed Armen Gulian, 2020

بخش های مختلف مدار کوانتومی؛ موجبر و رزوناتور

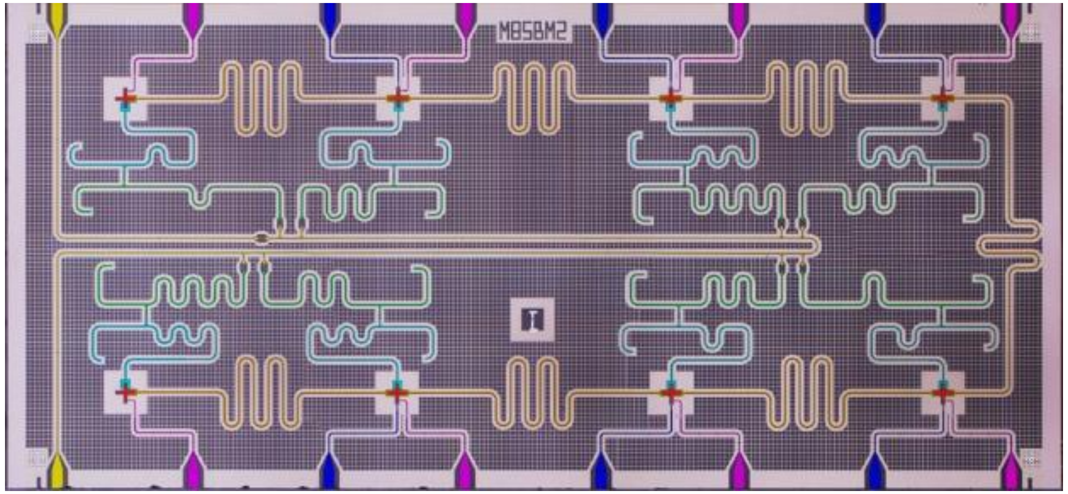


A. Wallraff et. al., *nature*, 2004

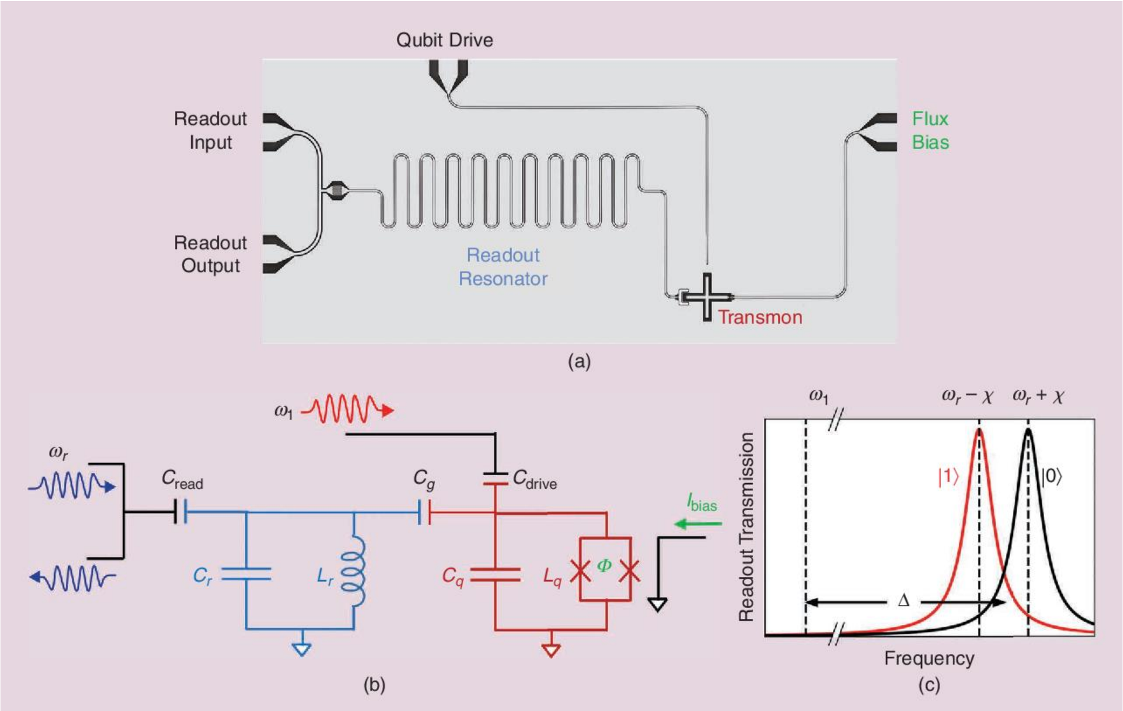


D. Chatterjee & A. Roy, *Prog. Theor. Exp. Phys.* , 2015

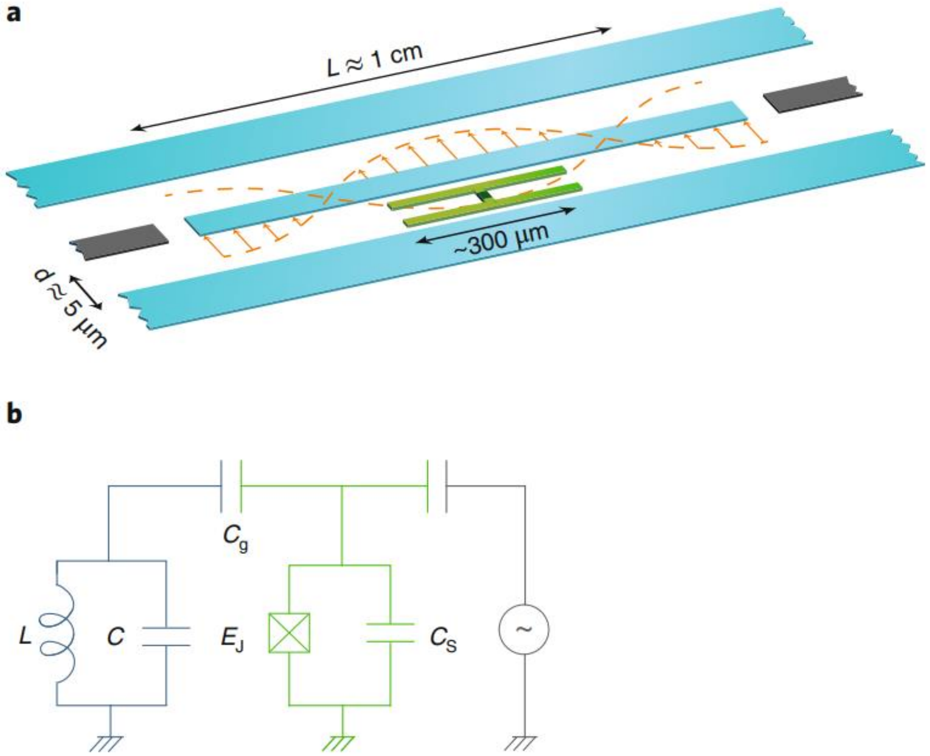
بخش های مختلف مدار کوانتومی؛ موجبر و رزوناتور



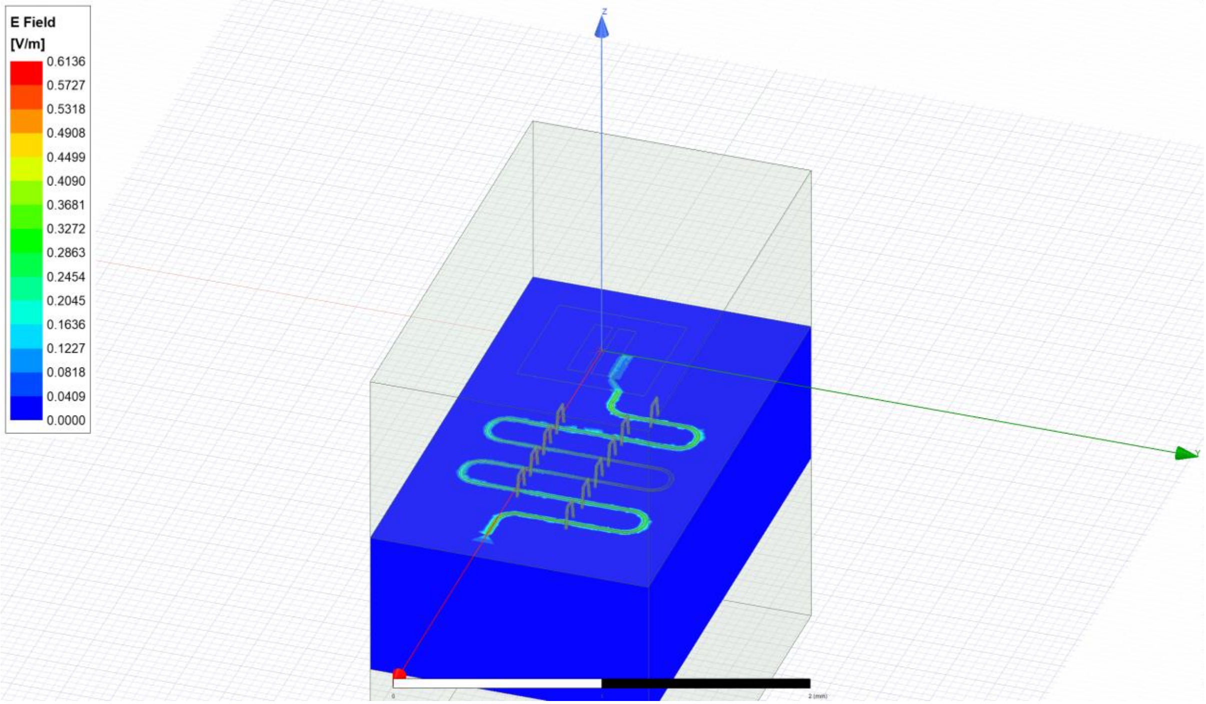
8-qubit superconducting quantum processor fabricated at ETH Zurich



T. E. Roth et. al., *IEEE Antennas and Propagation Magazine*, 2023



A. Blais et. al., *nature*, 2020



Analyzing and tuning a transmon qubit with a resonator

https://qiskit.org/ecosystem/metal/tut/4-Analysis/4.13-Analyze-transmon-and-resonator.html

رزوناتور سه بعدی

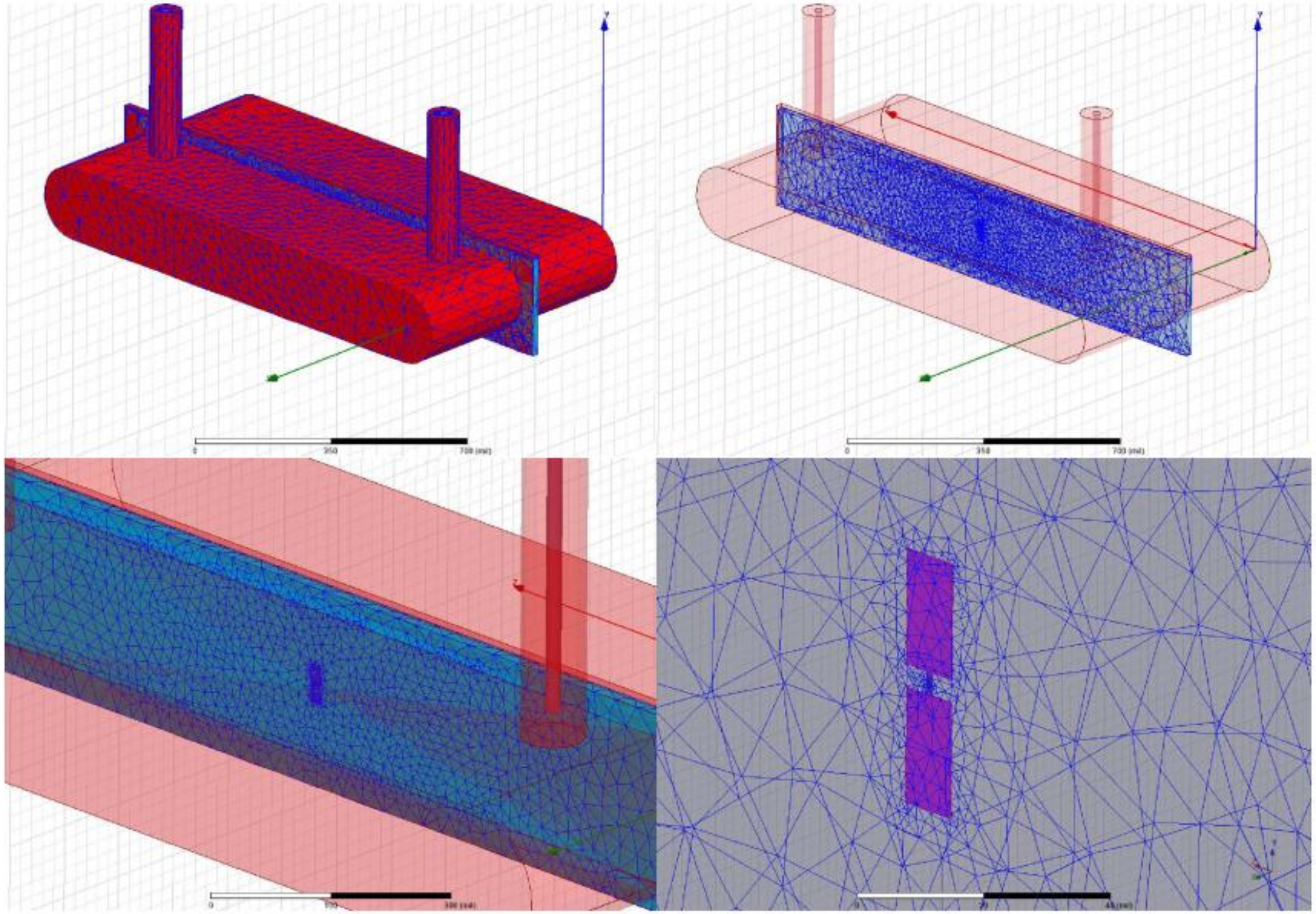
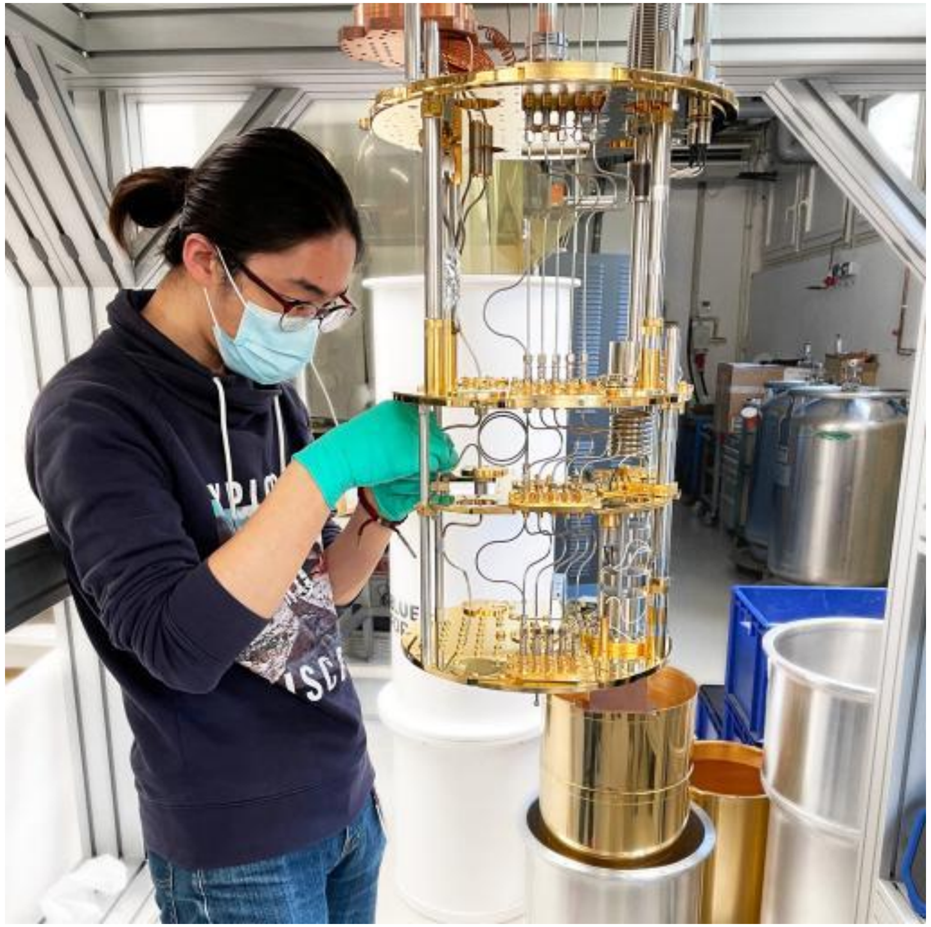


FIG. 3. (Color online) HFSS model of a 3D-transmon. (a) The 3D resonator with input and output ports. These are terminated by 50 Ohm ports. (b) Transparent view of the cavity showing the sapphire substrate. Because the electric field is concentrated in the dielectric, a finer mesh is used. (c) and (d) Zoom-ins on the antenna placed on top of the substrate. The mesh is finest around the antenna.

Simon E. Nigg, et. al., *PRL*, 2012.



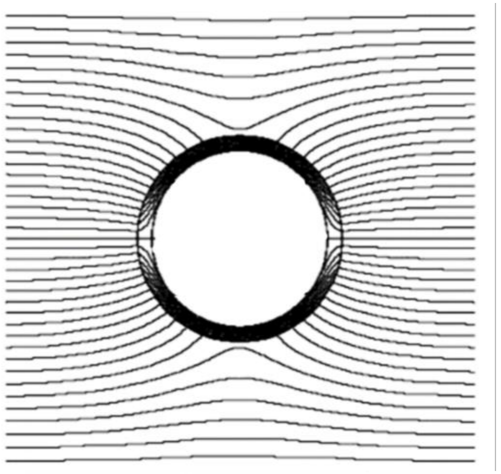
Superconducting Quantum Computing
in Jülich & Aachen

Oxford Becomes UK Partner in IBM's
Quantum Computing Network

AFRL/RITQ - Superconducting and Hybrid Quantum Systems

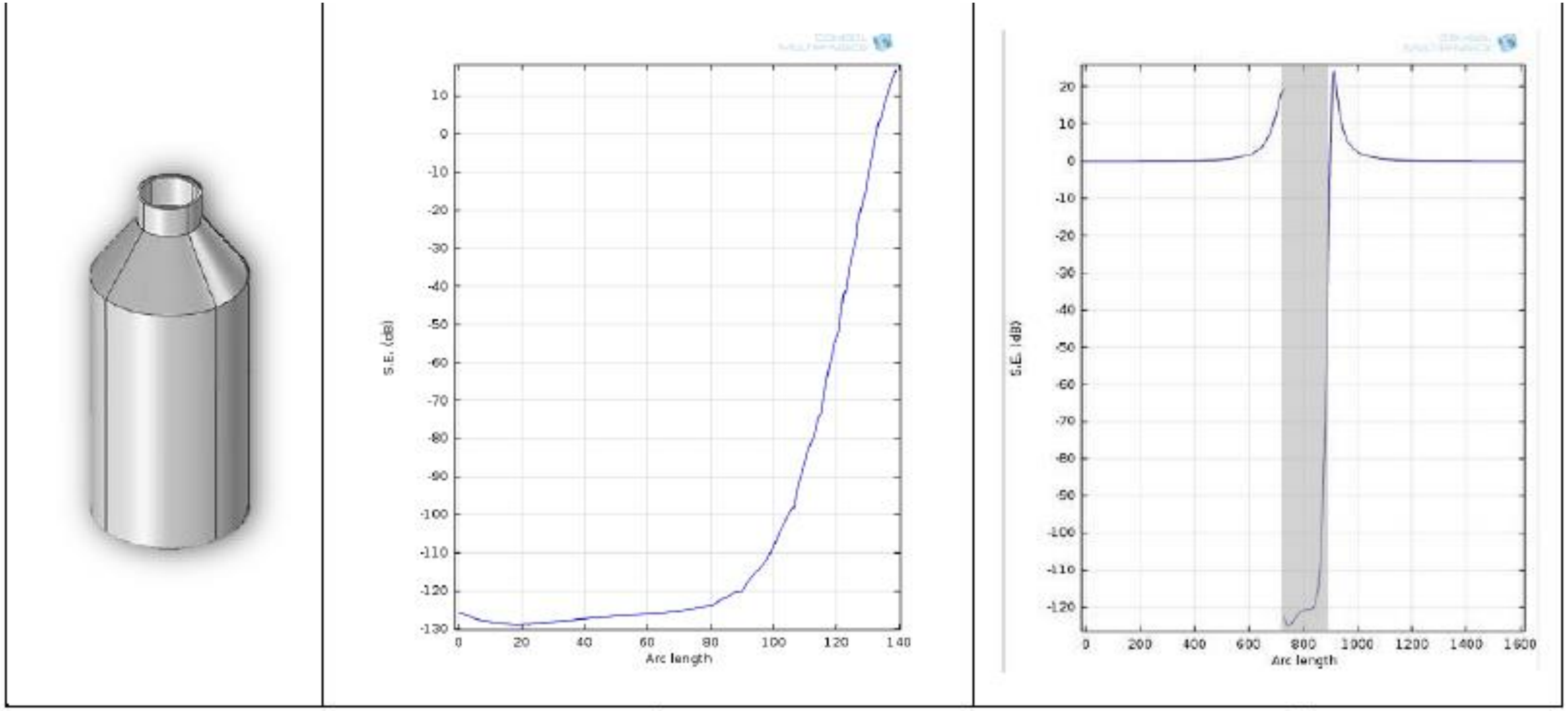
Dilution refrigerator set with an aluminum cavity in FermiLab

حفاظت الکترومغناطیسی با میومتال



$$\mathcal{R}_1 = \frac{L_1}{\mu A_1} = \frac{L_1}{\mu_r \mu_0 A_1}$$

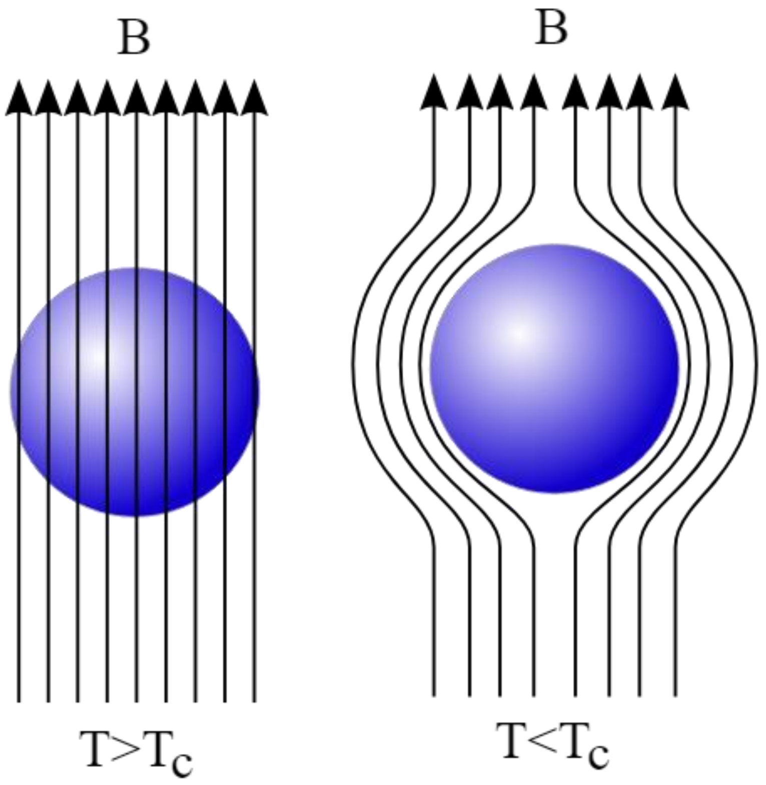
نمونه ای از محافظ cryoperm

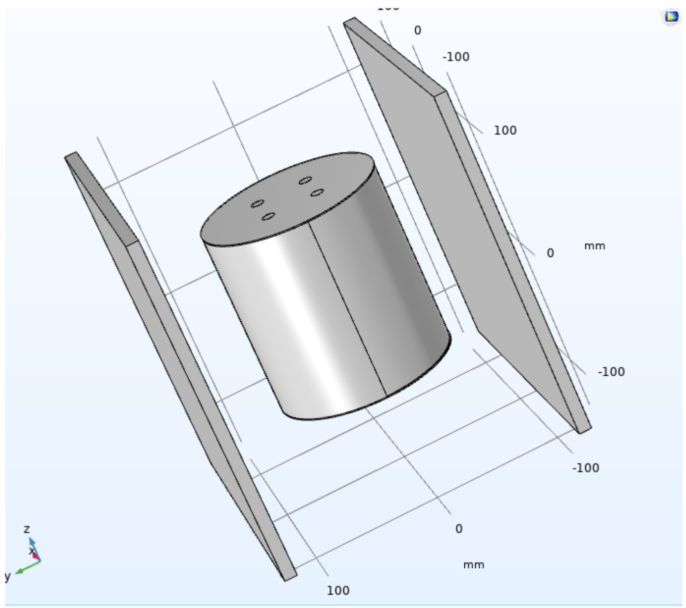


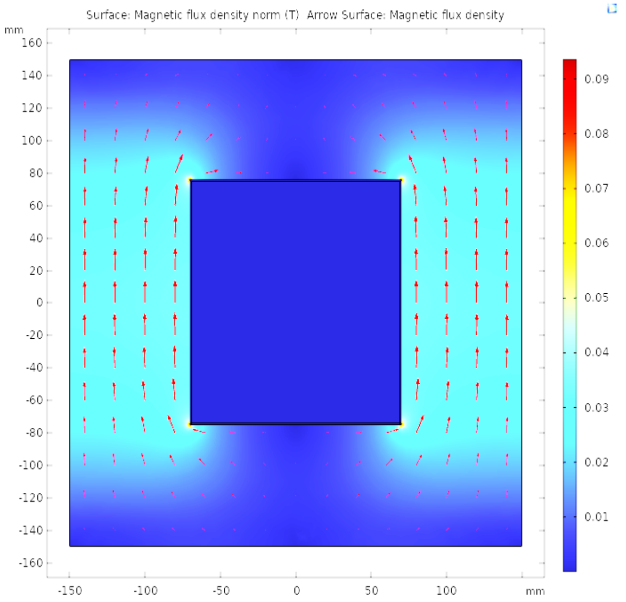
a) b)

مرکز تحقیقات فناوری‌های کوانتومی ایران

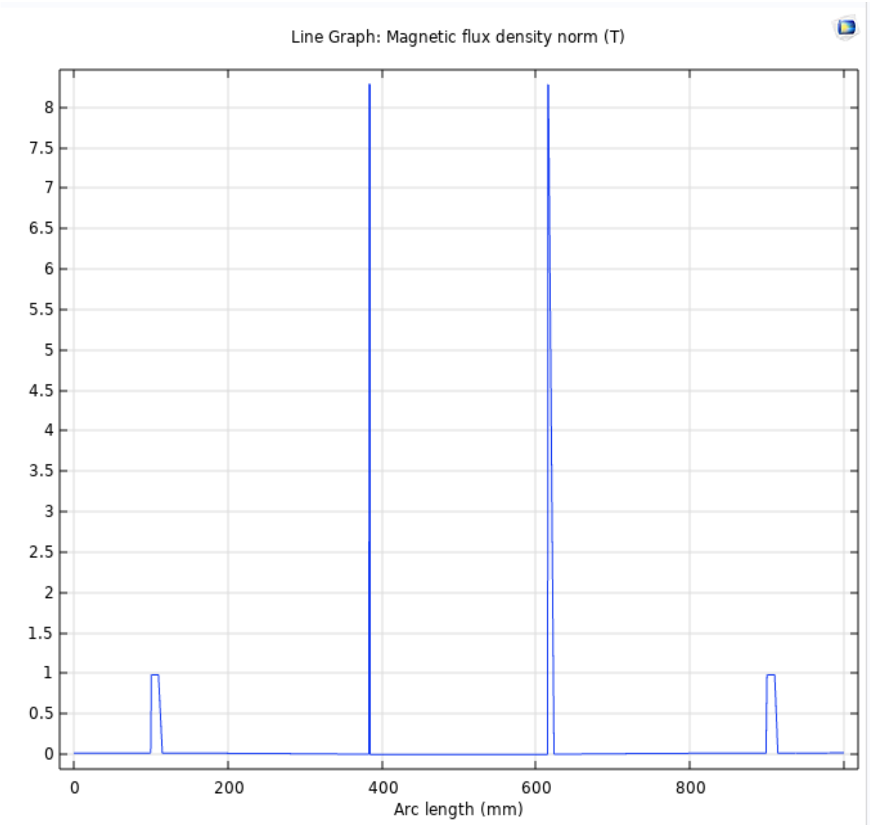
Kharazmi University

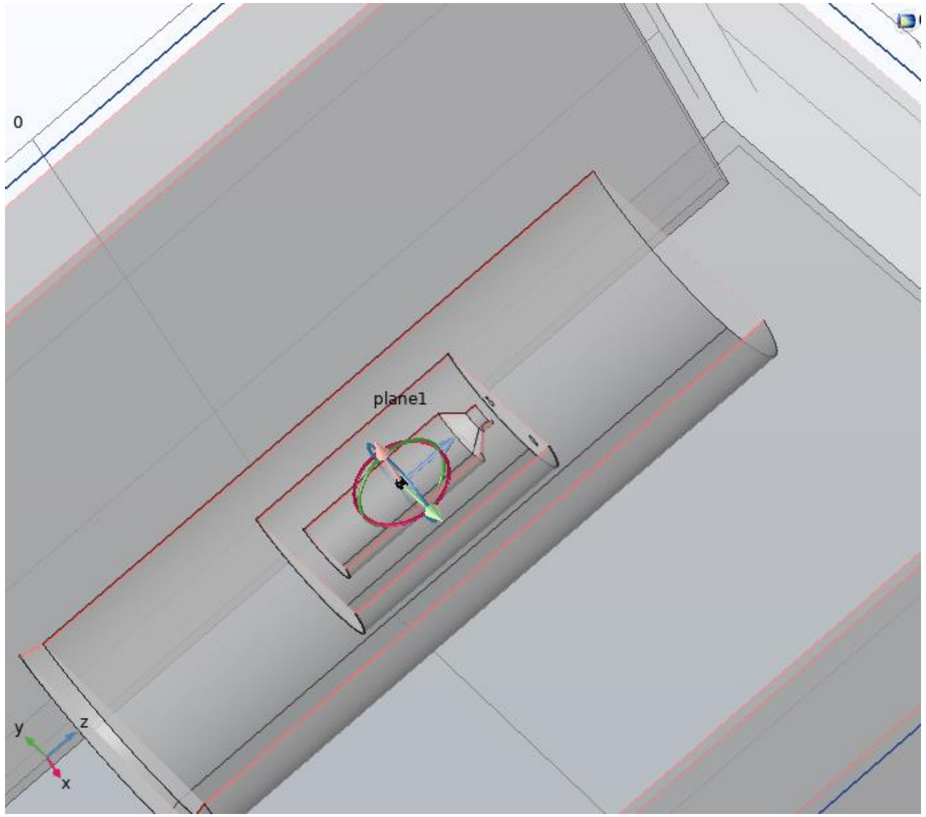






Aluminium shielding





Attenuation more than 10^{-7}

Qutip Library for Hamiltonian Analysis

Lecture-11-Charge-Qubits.ip ×

Markdown   Notebook   Python 3 (ipykernel)

QuTiP lecture: Superconducting Josephson charge qubits

J.R. Johansson (robert@riken.jp)

[1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
```

[2]:
```
from qutip import *
```

Introduction

The Hamiltonian for a Josephson charge qubit is

$$H = \sum_n 4E_C(n_g - n)^2 \left|n\right\rangle \left\langle n\right| - \frac{1}{2}E_J \sum_n (\left|n+1\right\rangle \left\langle n\right| + \left|n\right\rangle \left\langle n+1\right|)$$

where E_C is the charge energy, E_J is the Josephson energy, and $\left|n\right\rangle$ is the charge state with n Cooper-pairs on the island that makes up the charge qubit.

References

- J. Koch et al, Phys. Rec. A 76, 042319 (2007)
- Y.A. Pashkin et al, Quantum Inf Process 8, 55 (2009)

Helper functions

Below we will repeatedly need to obtain the charge qubit Hamiltonian for different parameters, and to plot the eigenenergies, so here we define two functions to do these tasks.

Lecture-11-Charge-Qubits.ip × +

Markdown   Notebook   Python 3 (ipykernel)

Helper functions

Below we will repeatedly need to obtain the charge qubit Hamiltonian for different parameters, and to plot the eigenenergies, so here we define two functions to do these tasks.

[3]:
```python
def hamiltonian(Ec, Ej, N, ng):
    """
    Return the charge qubit hamiltonian as a Qobj instance.
    """
    m = np.diag(4 * Ec * (np.arange(-N,N+1)-ng)**2) + 0.5 * Ej * (np.diag(-np.ones(2*N), 1) +
                                                              np.diag(-np.ones(2*N), -1))
    return Qobj(m)
```

[4]:
```python
def plot_energies(ng_vec, energies, ymax=(20, 3)):
    """
    Plot energy levels as a function of bias parameter ng_vec.
    """
    fig, axes = plt.subplots(1,2, figsize=(16,6))

    for n in range(len(energies[0,:])):
        axes[0].plot(ng_vec, energies[:,n])
    axes[0].set_ylim(-2, ymax[0])
    axes[0].set_xlabel(r'$n_g$', fontsize=18)
    axes[0].set_ylabel(r'$E_n$', fontsize=18)

    for n in range(len(energies[0,:])):
        axes[1].plot(ng_vec, (energies[:,n]-energies[:,0])/(energies[:,1]-energies[:,0]))
    axes[1].set_ylim(-0.1, ymax[1])
    axes[1].set_xlabel(r'$n_g$', fontsize=18)
    axes[1].set_ylabel(r'$(E_n-E_0)/(E_1-E_0)$', fontsize=18)
    return fig, axes
```

کتابخانه Qutip

Lecture-11-Charge-Qubits.ip

Markdown | Notebook | Python 3 (ipykernel

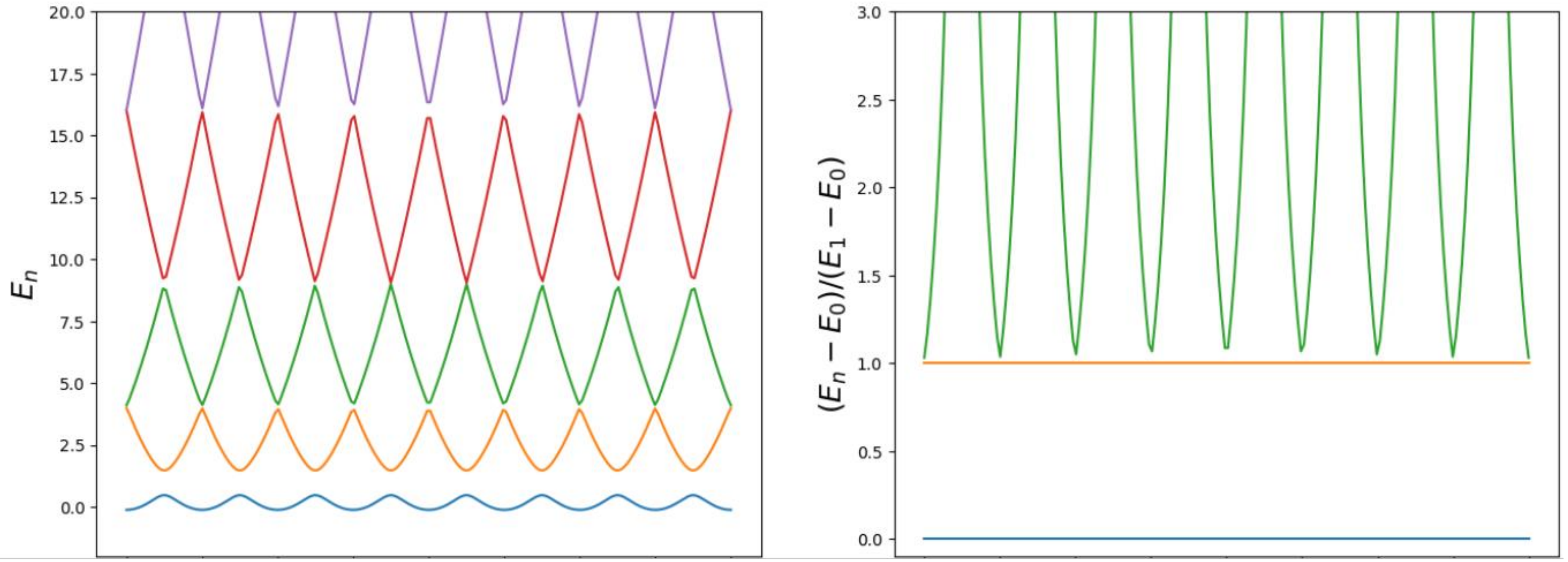
Charge qubit regime

[6]:
```
N = 10
Ec = 1.0
Ej = 1.0
```

[7]:
```
ng_vec = np.linspace(-4, 4, 200)

energies = np.array([hamiltonian(Ec, Ej, N, ng).eigenenergies() for ng in ng_vec])
```

[8]:
```
plot_energies(ng_vec, energies);
```



Lecture-11-Charge-Qubits.ip

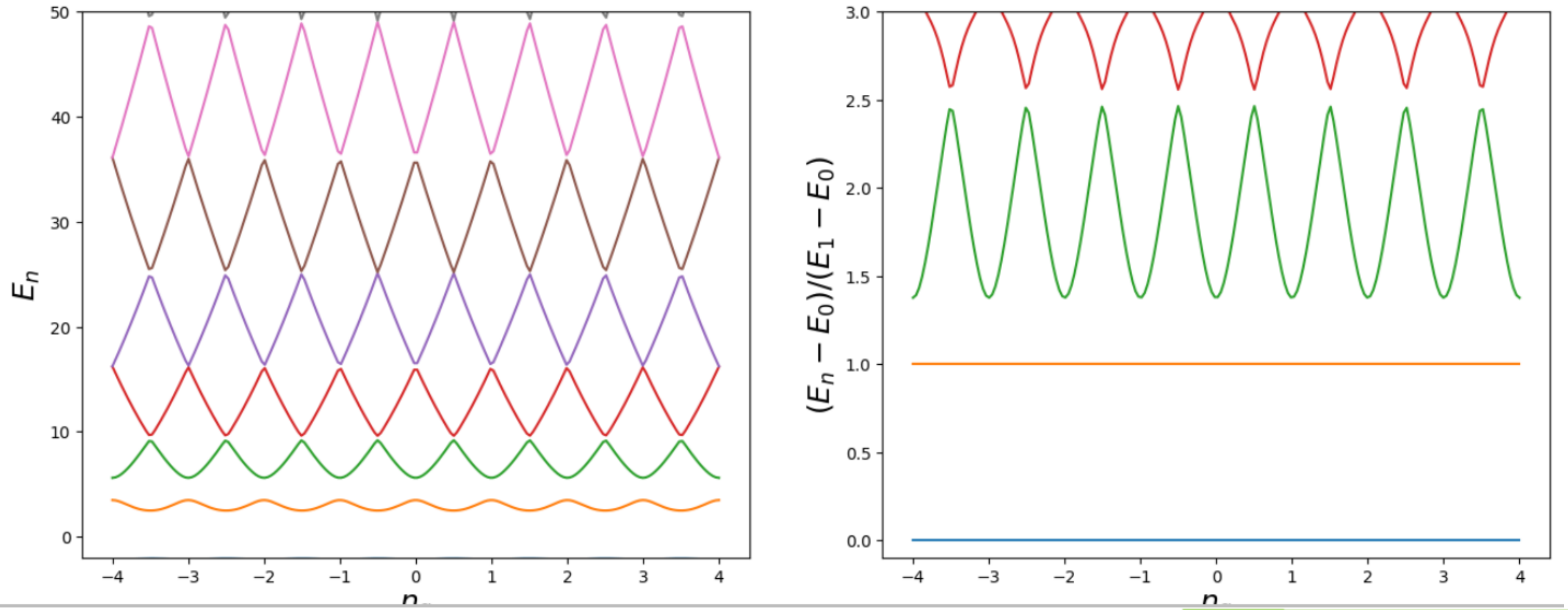
Intermediate regime

```
[12]: ng_vec = np.linspace(-4, 4, 200)
```

```
[13]: Ec = 1.0
      Ej = 5.0
```

```
[15]: energies = np.array([hamiltonian(Ec, Ej, N, ng).eigenenergies() for ng in ng_vec])
```

```
[16]: plot_energies(ng_vec, energies, ymax=(50, 3));
```



کتابخانه Qutip

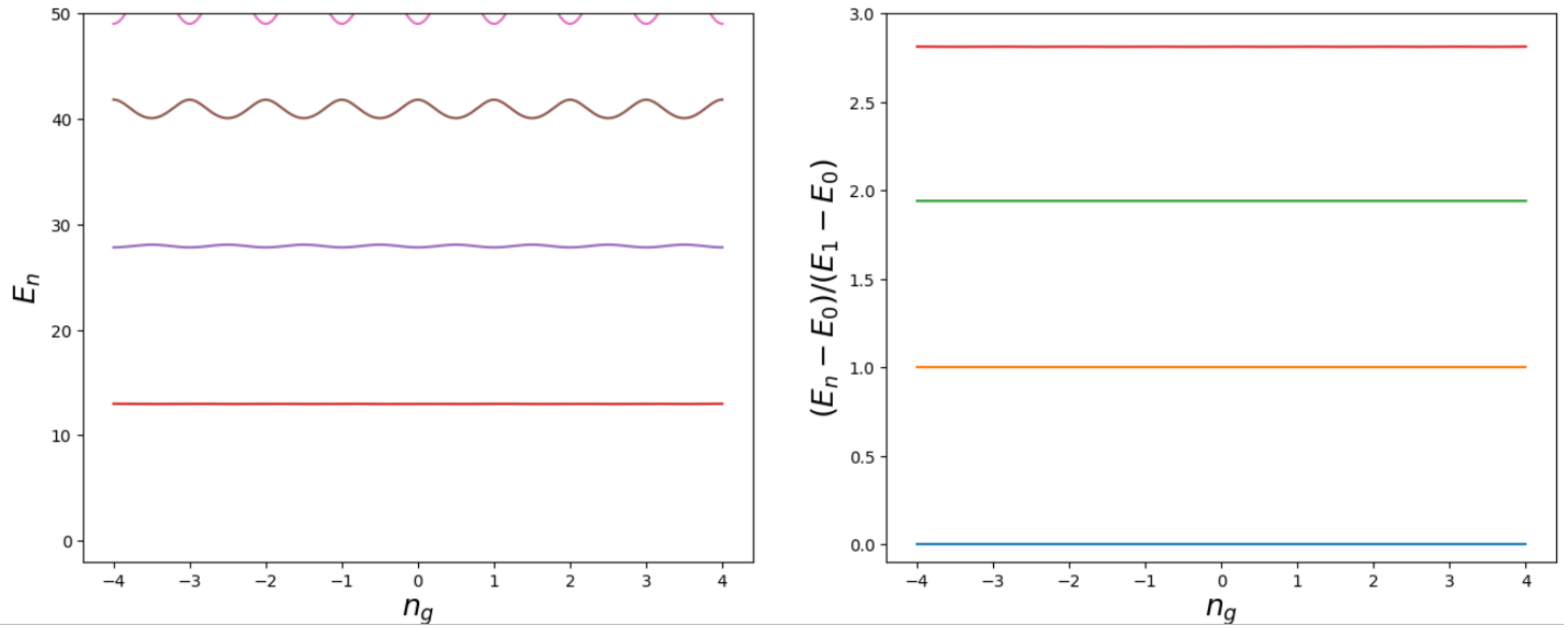
Lecture-11-Charge-Qubits.ip

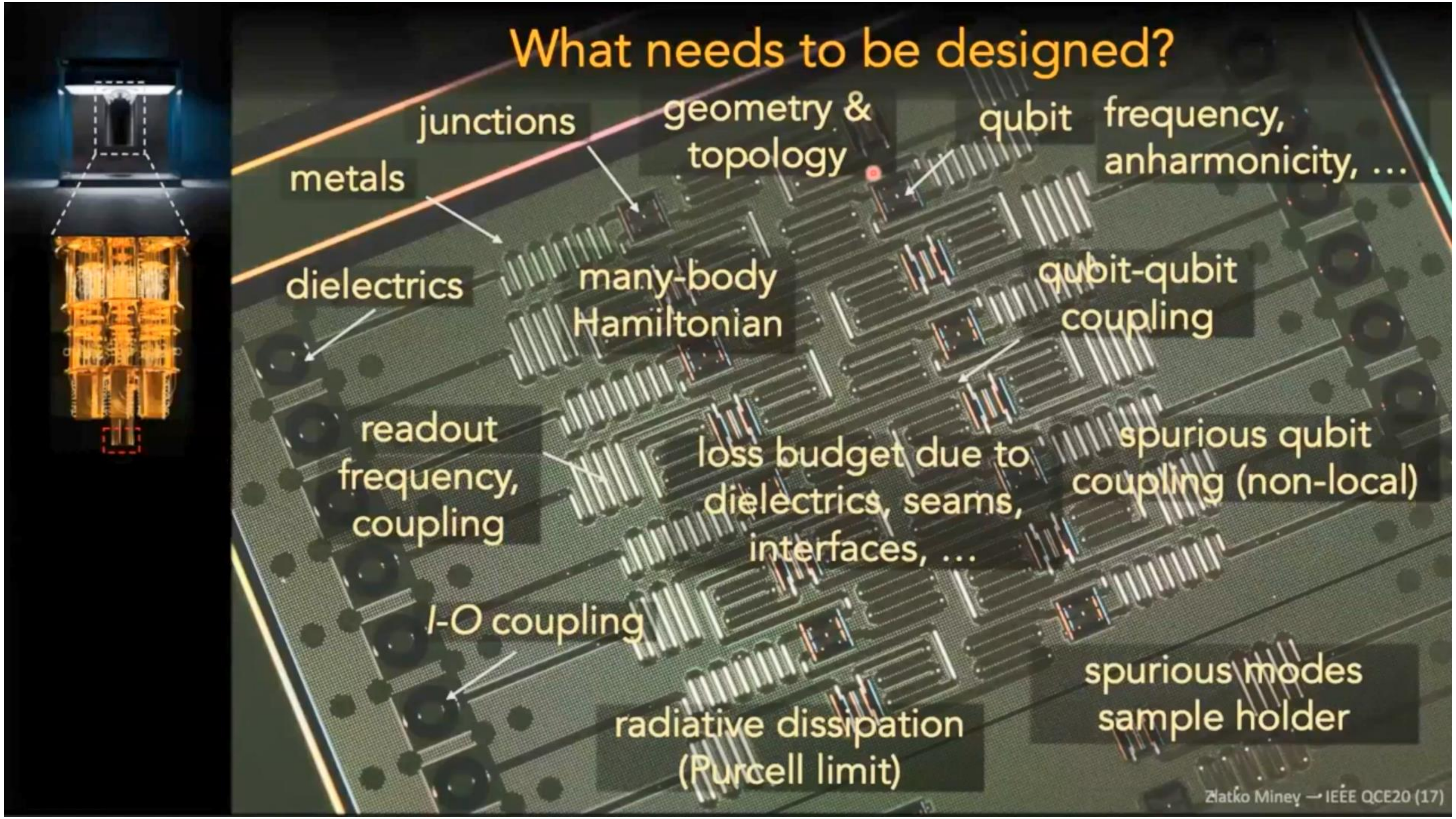
Transmon regime

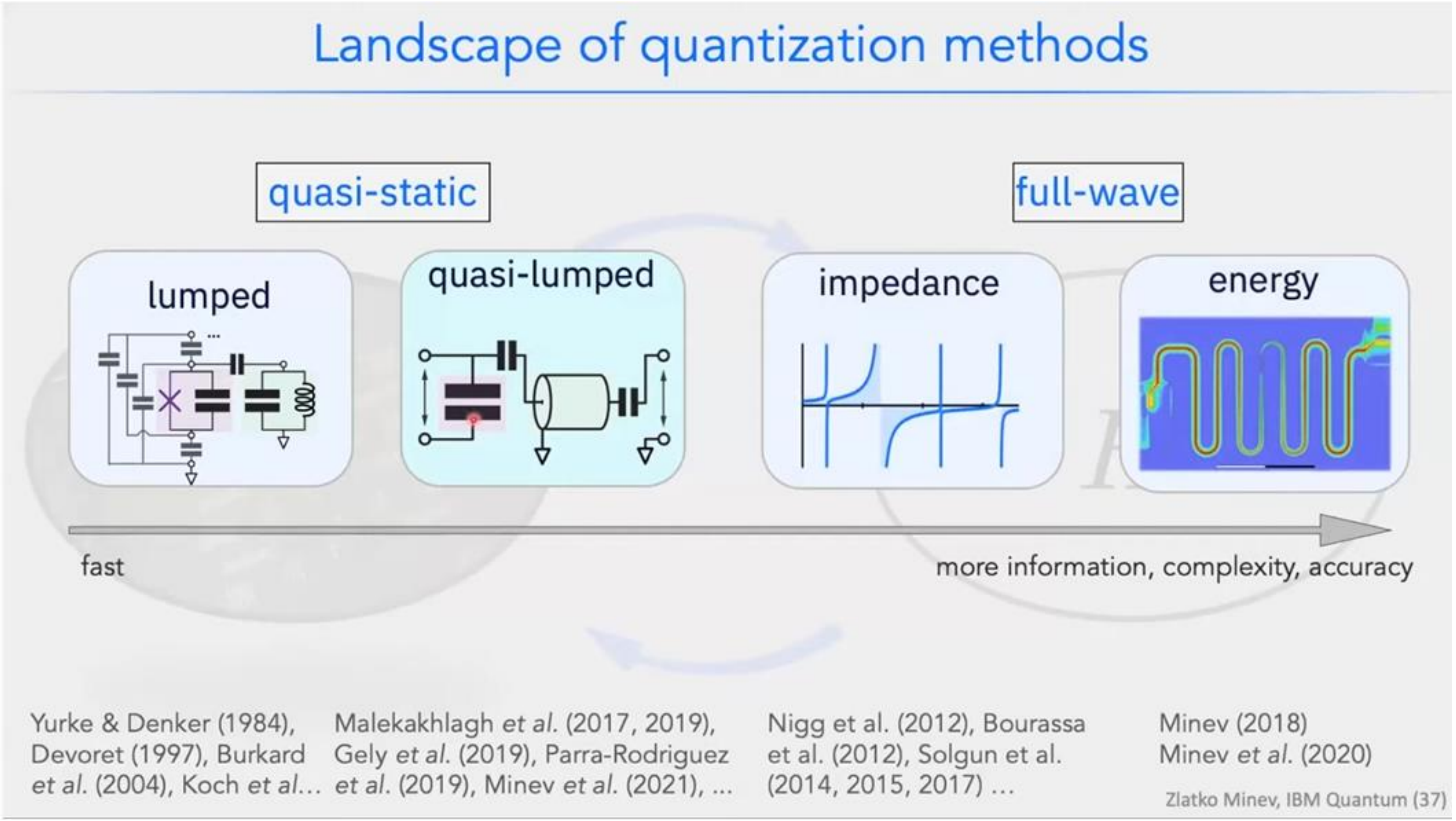
```
[21]: Ec = 1.0
      Ej = 50.0
```

```
[23]: energies = np.array([hamiltonian(Ec, Ej, N, ng).eigenenergies() for ng in ng_vec])
```

```
[24]: plot_energies(ng_vec, energies, ymax=(50, 3));
```




What needs to be designed?
junctions
geometry &
topology
qubit
frequency,
anharmonicity, …
metals
dielectrics
many-body
Hamiltonian
qubit-qubit
coupling
readout
frequency,
coupling
loss budget due to
dielectrics, seams,
interfaces, …
spurious qubit
coupling (non-local)
I-O coupling
spurious modes
sample holder
radiative dissipation
(Purcell limit)
Zlatko Minev — IEEE QCE20 (17)



مرکز تحقیقات فناوری‌های کوانتومی ایران

Kharazmi University

Lumped element model

Hamiltonian $\mathcal{H} = \frac{\Phi^2}{2L} + \frac{Q^2}{2C} = \hbar\omega_r\left(a^\dagger a + 1\right)$

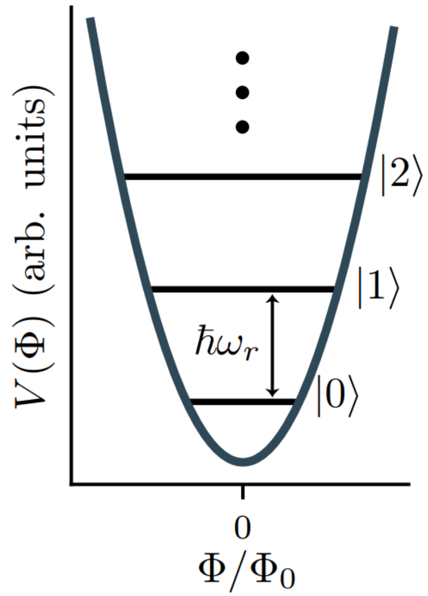
$$\omega_r = \frac{1}{\sqrt{LC}}$$

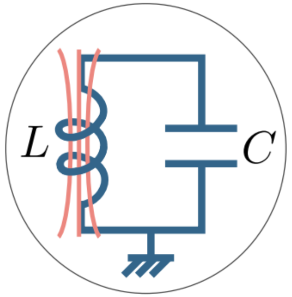
$$\hat{\phi} = \left(\frac{2E_c}{E_J}\right)^{\frac{1}{4}}\left(\hat{a} + \hat{a}^\dagger\right)$$

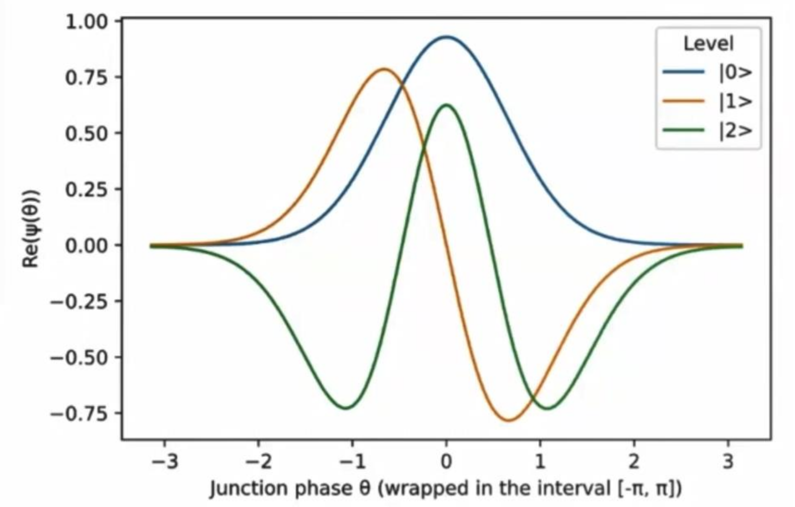
$$\hat{n} = \left(\frac{E_J}{32E_c}\right)^{\frac{1}{4}}\left(\hat{a}^\dagger - \hat{a}\right)$$

Zero point fluctuation of phase ϕ_{zpf}

Zero point fluctuation of charge number: n_{zpf}

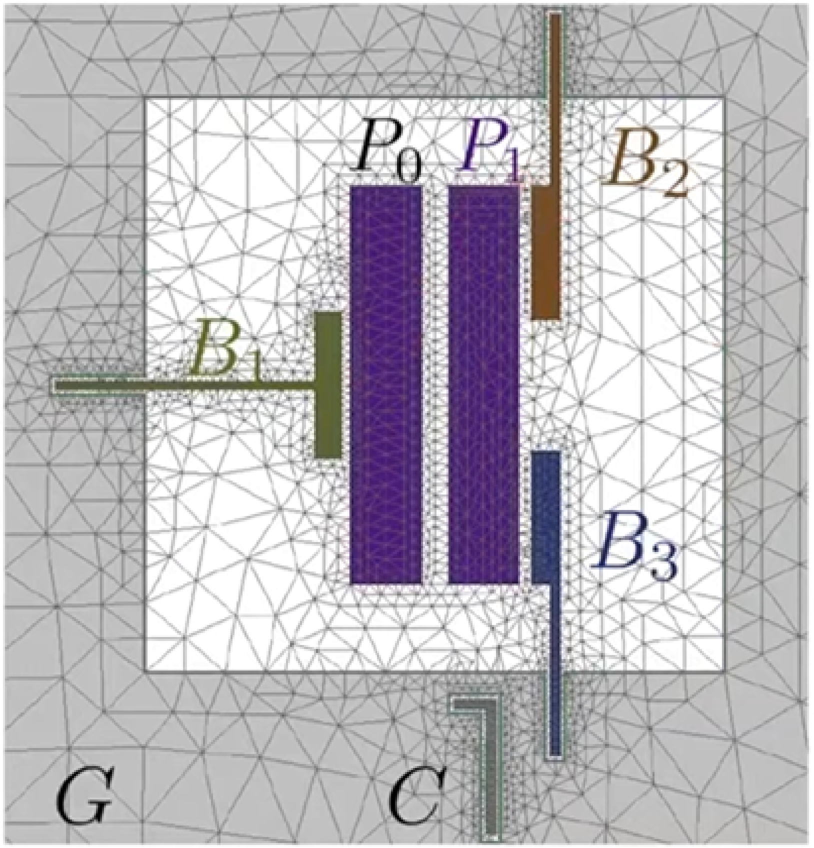






Extracting cell parameters

netlist & simulation



Maxwell capacitance matrix

$$
\begin{array}{c} \\ n_0 \\ n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \end{array}
\begin{array}{cccccc}
n_0 & n_1 & n_2 & n_3 & n_4 & n_5 \\
\end{array}
$$

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

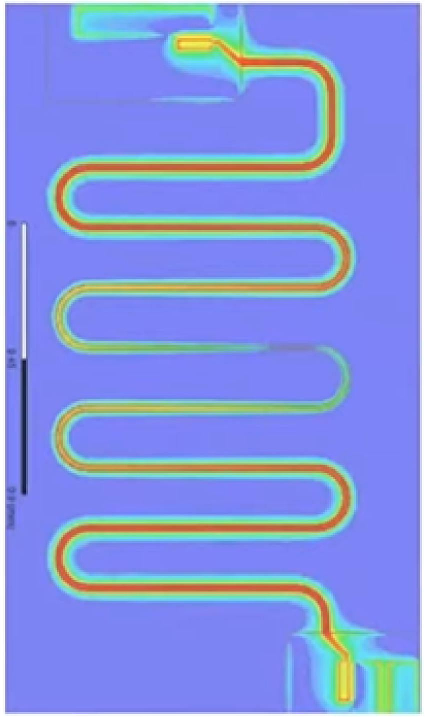
Automated with
Qiskit | quantum device design

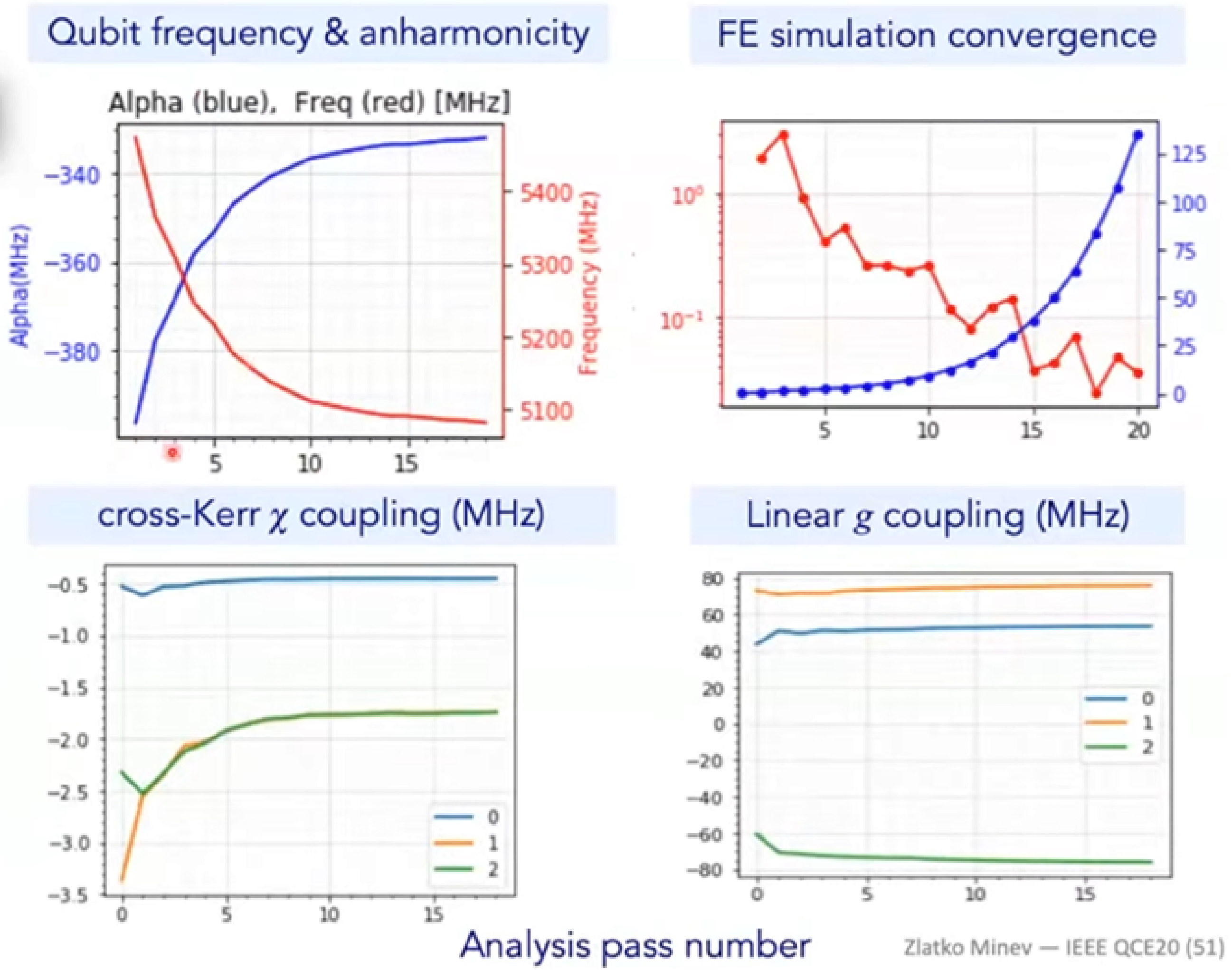
arXiv:2103.10344

Zlatko Minev, IBM Quantum (42)

Automated analysis and reports

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





Energy participation ratio analysis (EPR)

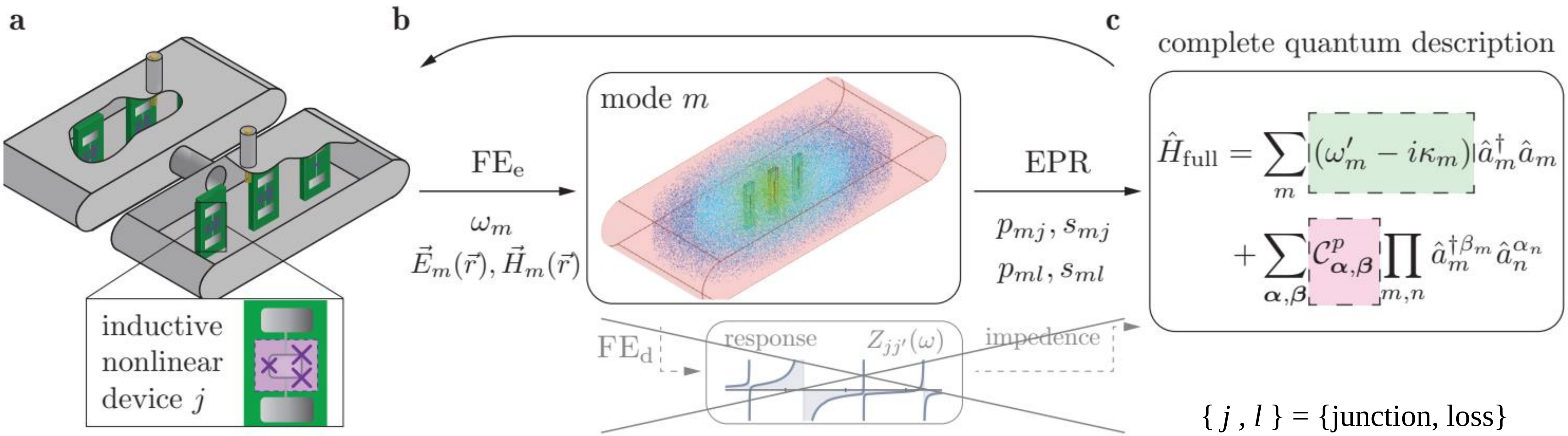
$$\hat{H}_{\text{tot}}$$

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

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

- $H_{\text{lin}} = \hbar\omega_c a^\dagger a + \hbar\omega_q a^\dagger a$
- $H_{nl} = -E_J\left(\cos(\Phi_J) + \frac{\Phi_J^2}{2}\right)$
- $\Phi_J = \phi_q(a_q + a_q^\dagger) + \phi_c(a_c + a_c^\dagger)$

Energy participation ratio analysis (EPR)



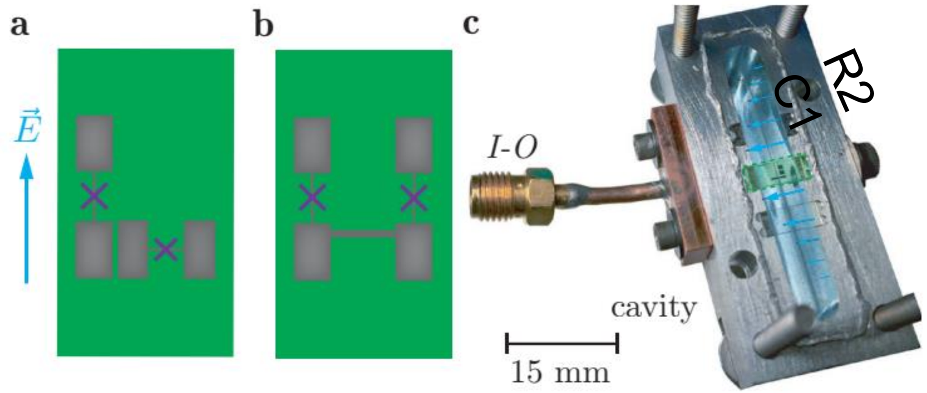
$$\hat{H}_{\mathrm{full}} = \sum_m (\omega'_m - i\kappa_m)\hat{a}_m^\dagger \hat{a}_m + \sum_{\alpha,\beta} \mathcal{C}^p_{\alpha,\beta} \prod_{m,n} \hat{a}_m^{\dagger\beta_m} \hat{a}_n^{\alpha_n}$$

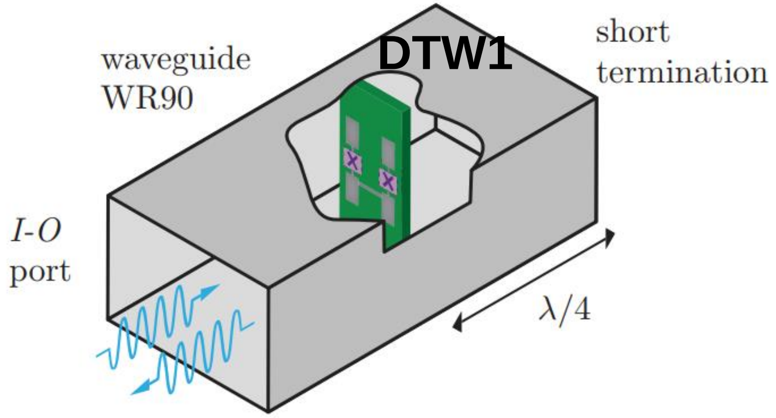
{ j , l } = {junction, loss}

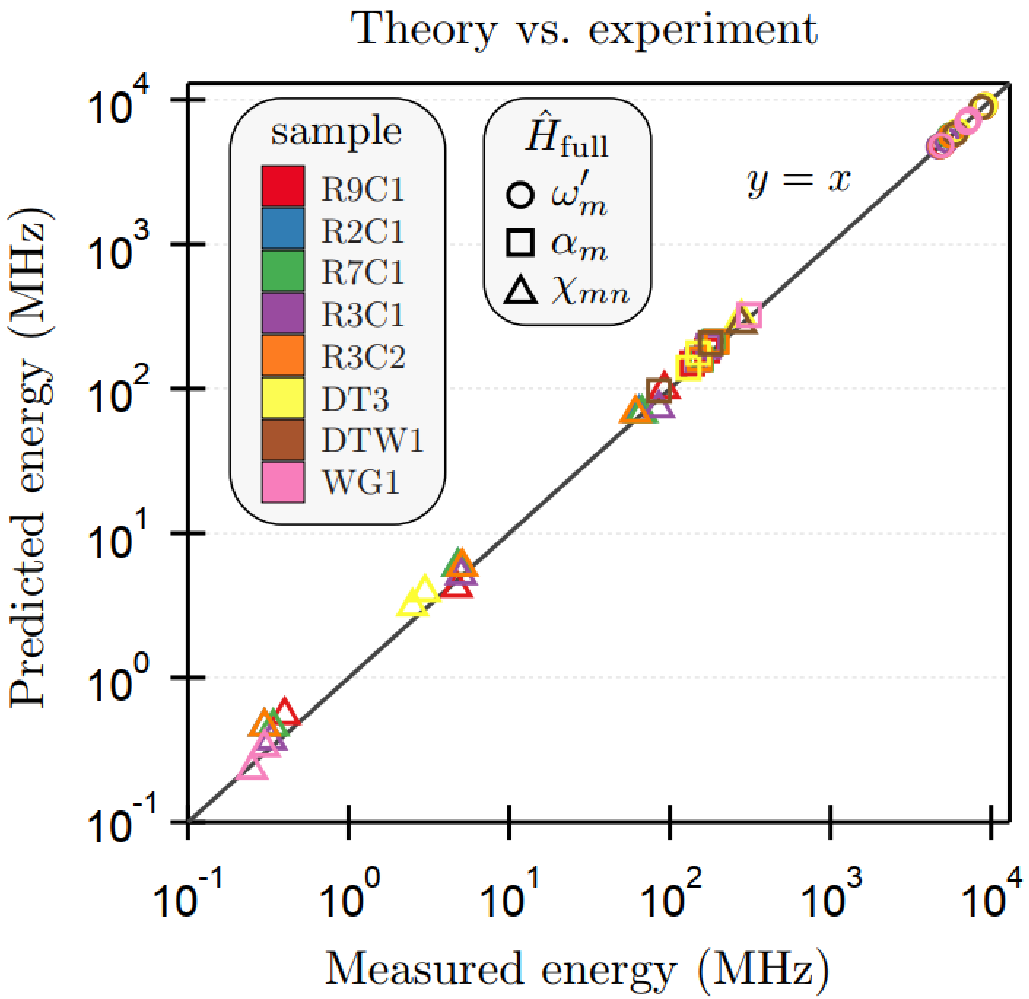
$$p_m := \frac{\text{Inductive energy stored in the junction}}{\text{Total inductive energy stored in mode } m}$$

$$p_m = \frac{\langle \psi_m | \frac{1}{2} E_J \hat{\varphi}_J^2 | \psi_m \rangle}{\langle \psi_m | \frac{1}{2} \hat{H}_{\mathrm{lin}} | \psi_m \rangle}$$

Measured vs. predicted







Bird's eye view of Qiskit Metal

Using Qiskit Metal in 4 stages

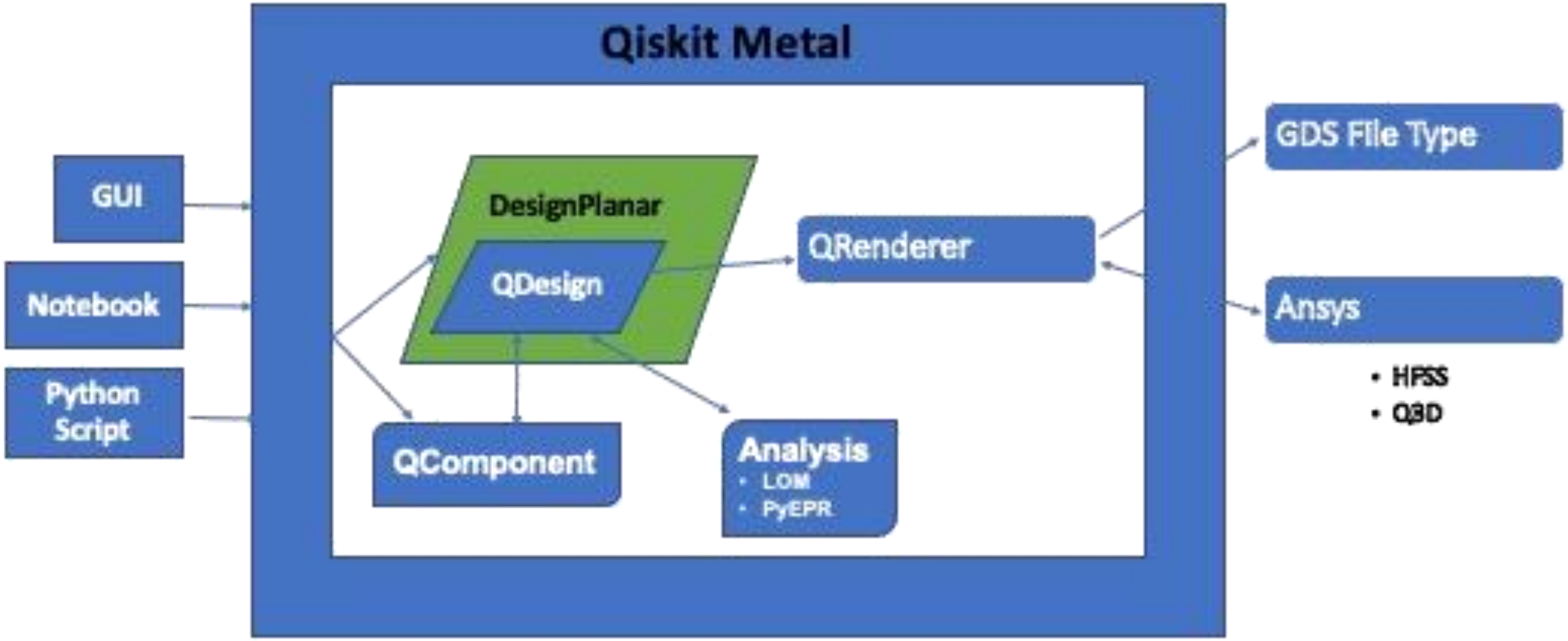
1. Choose a design class to instantiate.

2. Add and modify pre-built components (qubits, coplanar wave guides, etc.) from the QComponent library to your design. (Or, create your own components)

3. Render to Simulate & Analyze
   - Rendering Options:
     - Ansys
       - HFSS Renderer - for high frequency simulations (eigenmode, modal, terminal)
         - EPR Analysis - Uses eigenmode simulation to perform energy participation ratio analysis
       - Q3D Renderer - for extracting equivalent circuit values of a layout, such as capacitance
         - LOM Analysis - Uses the capacitance matrix from Q3D to determine the parameters of a transmon qubit

4. Render for Fabrication
   - Current Rendering Options:
     - GDS

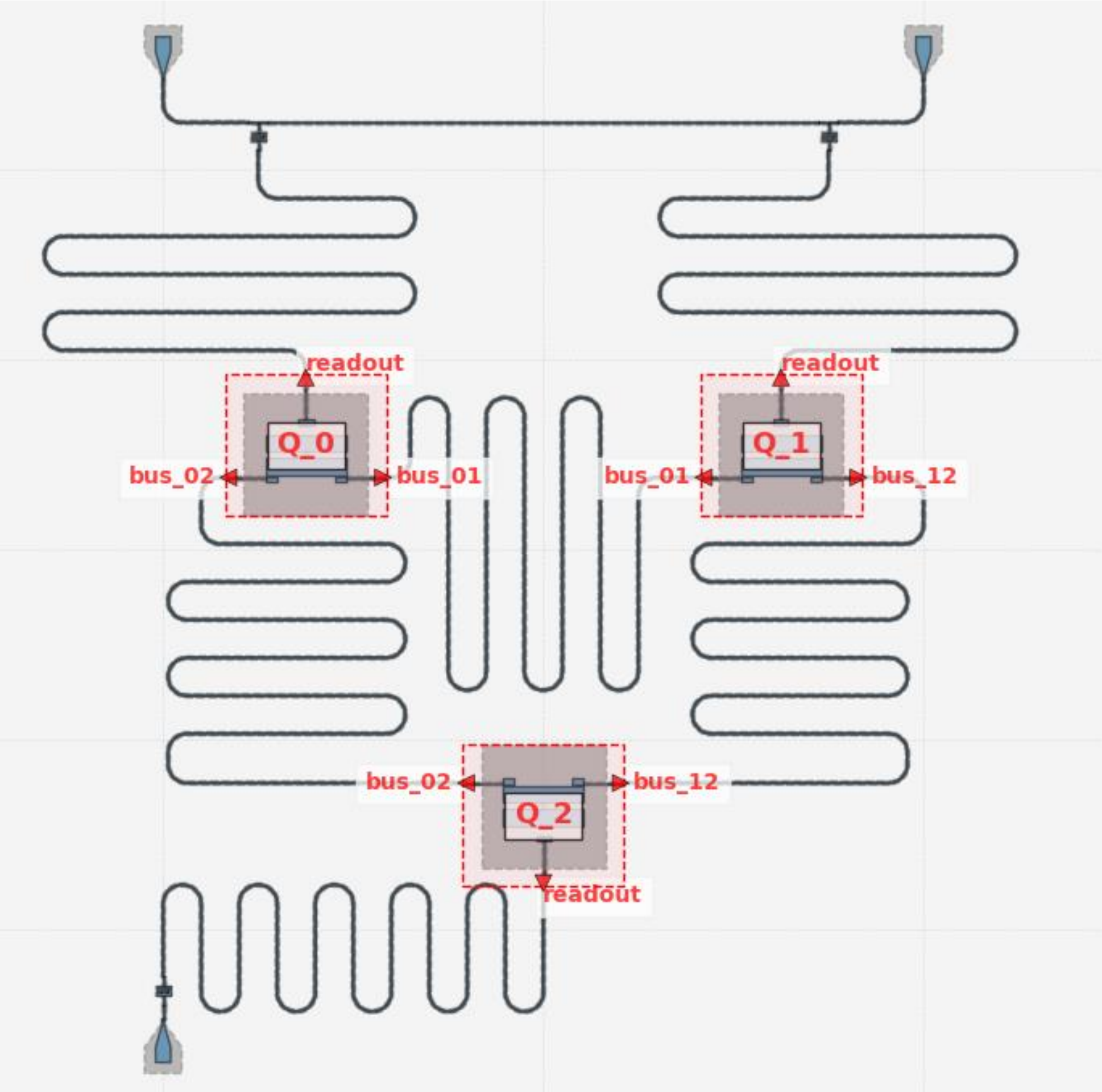
Example: QDesign Connections



مثال کتابخانه Qiskit Metal

- Goal:

- qubit frequency => 4.8, 5, 5.2 GHz
- qubit anharmonicity => 300 MHz
- qubit-bus (g) => 80 MHz
- qubit-readout (chi) => 1 MHz
- bus frequency (bus_01, bus_02, bus_12) => 5.8, 6.0, 6.2 GHz
- readout frequencies => 6.8, 7, 7.2 GHz
- readout Q_external => 2000



• Goal:

- qubit frequency => 4.8, 5, 5.2 GHz
- qubit anharmonicity => 300 MHz
- qubit-bus (g) => 80 MHz
- qubit-readout (chi) => 1 MHz
- bus frequency (bus_01, bus_02, bus_12) => 5.8, 6.0, 6.2 GHz
- readout frequencies => 6.8, 7, 7.2 GHz
- readout Q_external => 2000

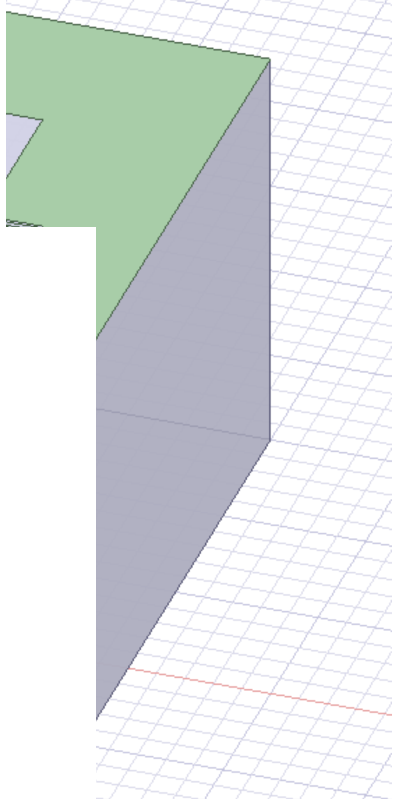
- First Shot for Qubit_0:
- LOM simulation results

```
{'fQ': 5.51159667165667,
 'EC': 313.4922216648471,
 'EJ': 13.616300010297985,
 'alpha': -364.8076551869923,
 'dispersion': 39.7165771074295,
 'gbus': array([-52.94824454,  49.75418056,  51.4568782 ]),
 'chi_in_MHz': array([-0.9742291 , -9.60195311, -4.65108037])}
```

$$\omega_q = \sqrt{8E_jEc} - \mathrm{E_c}$$

$$\alpha = E_c$$

$$\chi = \frac{g^2}{\Delta}$$



- **Quality factor:**

$$Q = \frac{\omega_q}{\delta\omega_q} \approx 10^4$$

- **Photon decay rate:**

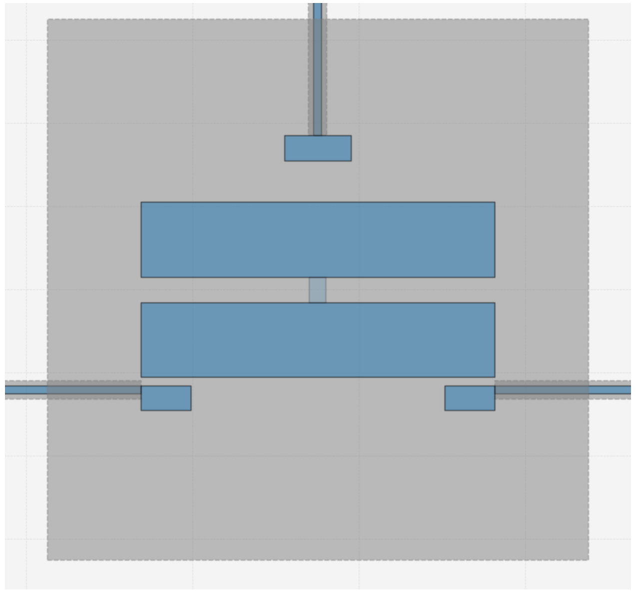
$$\frac{\kappa}{2\pi} = \frac{\omega_q}{Q} = 0.8\ \mathrm{MHz}$$

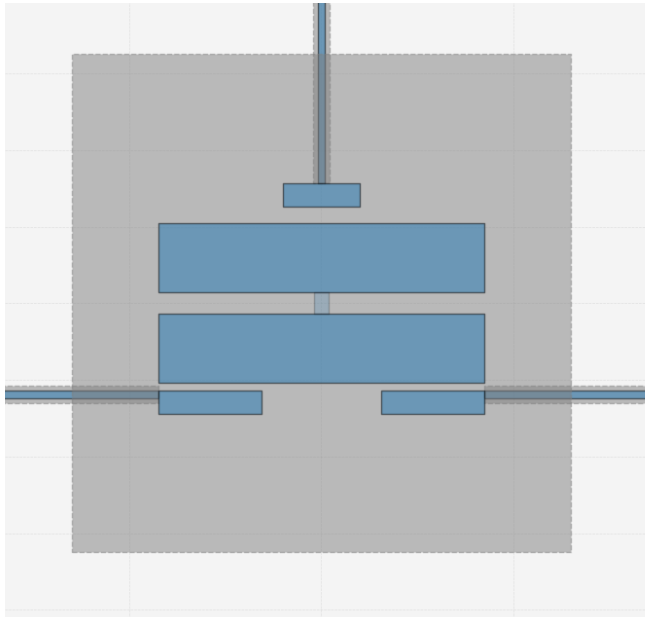
- **Photon lifetime:**

$$T_1^{\kappa} = \frac{1}{\kappa} \approx 200\ \mathrm{ns}$$

$$g_{ij} = \frac{E_{c_{ij}}}{2}\sqrt{\left(\frac{(\omega_i+\chi_{ii})(\omega_j+\chi_{jj})}{\chi_{ii}\chi_{jj}}\right)}$$

Modifying Qubit parameter:





- q_0.options.pad_gap = ‘30um’ to'28um’
- q_0.options.connection_pads.readout.pad_gap = ‘50um’ to '22um’
- q_0.options.connection_pads.readout.pad_width = ‘80um’ to '100um’
- q_0.options.connection_pads.bus_01.pad_width = ‘60um’ to '135um’
- q_0.options.connection_pads.bus_02.pad_width = ‘60um’ to '135um’
- q_0_LOM.setup.junctions.Lj = 12 to 14.9

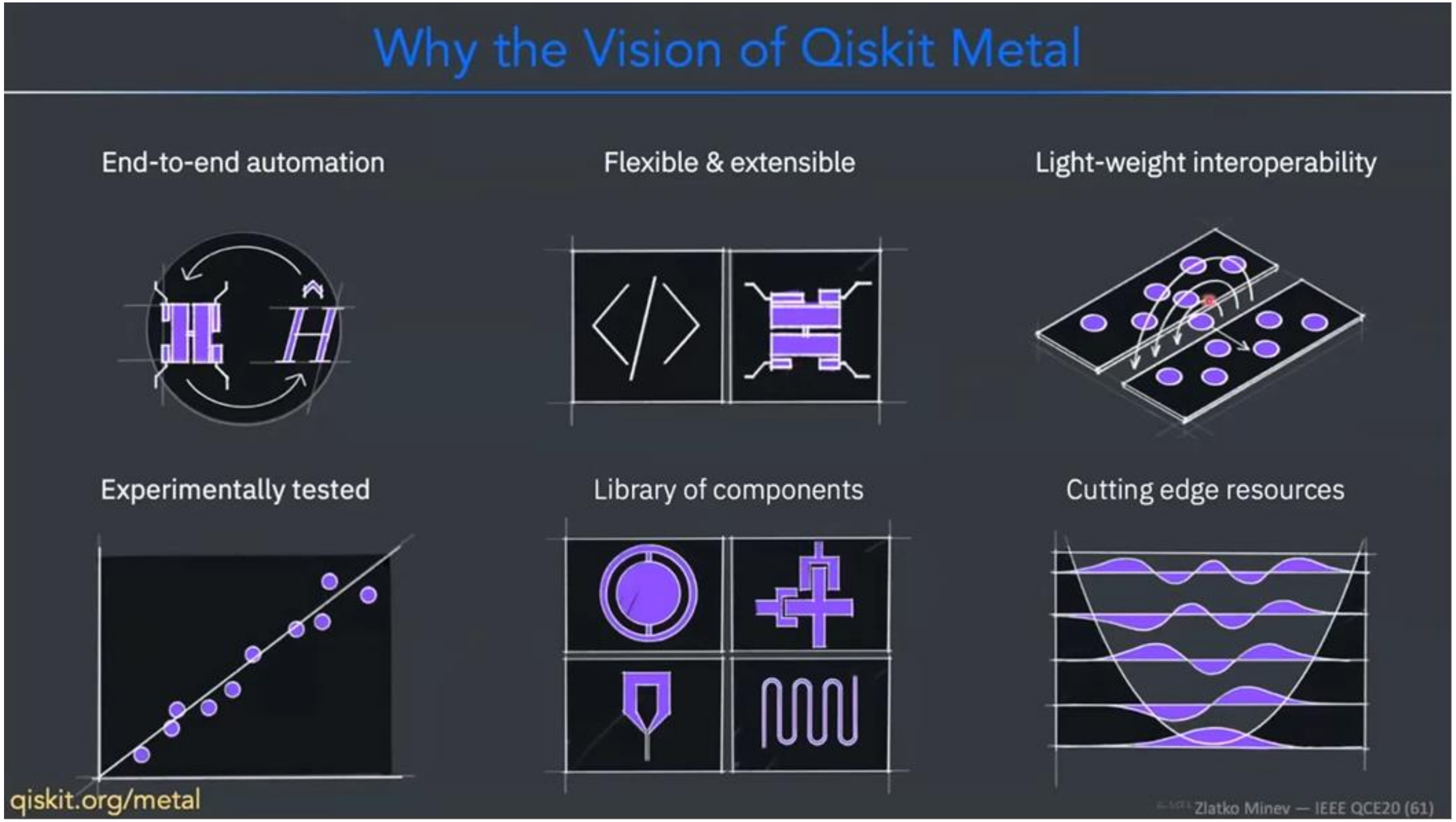
• Goal:

- qubit frequency => 4.8, 5, 5.2 GHz
- qubit anharmonicity => 300 MHz
- qubit-bus (g) => 80 MHz
- qubit-readout (chi) => 1 MHz
- bus frequency (bus_01, bus_02, bus_12) => 5.8, 6.0, 6.2 GHz
- readout frequencies => 6.8, 7, 7.2 GHz
- readout Q_external => 2000

- **Final Results for Qubit_0:**

```
{'fQ': 4.787226695506869,
 'EC': 296.92044408559445,
 'EJ': 10.966147659300391,
 'alpha': -351.528599604318,
 'dispersion': 128.87533861351014,
 'gbus': array([-83.55898953,  76.48193454,  79.11029219]),
 'chi_in_MHz': array([-1.0692286 , -3.01431109, -2.35838373])}
```

مثال کتابخانه Qiskit Metal



کتابخانه های متعددی برای تحلیل مدارهای کوانتومی ابررسانا معرفی شده اند، مانند:

SCQubits

QuCAT

SuperQUANT

QuEST

BQSKit

...

پایان

سپاس از توجه شما

مرکز تحقیقات
فناوری‌های
کوانتومی ایران

Kharazmi University