Introduction to Quantum Computing
Day 4 – QC with Neutral Atoms and NISQ devices

Mengoni Riccardo, PhD

24 June 2021

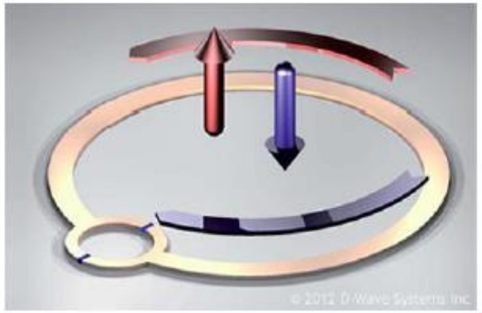
CINECA QUANTUM COMPUTING LAB

Content

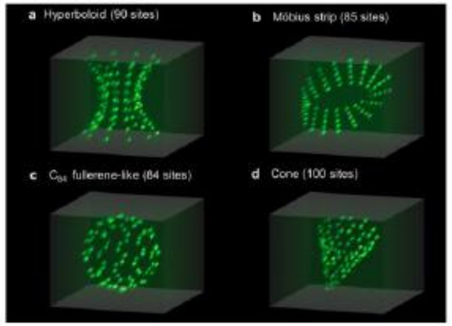
- Intro and Recap
- Pasqal Quantum Hardware: QC with Neutral Atoms
- Pulser: Control Software for Pasqal QC
- Quantum algorithms for NISQ Devices
- Application: QAOA & MIS problem

Intro and Recap

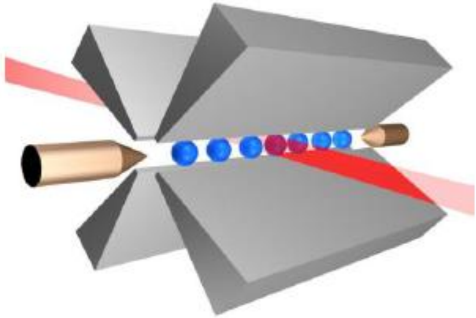
Hardware state of the art – qubit physical realization



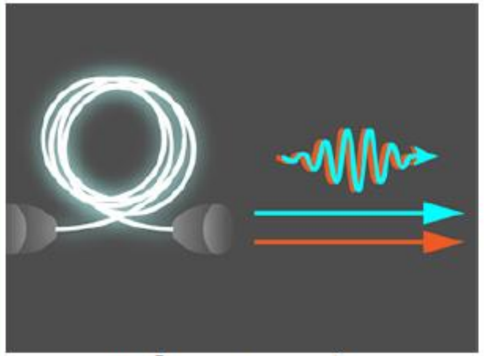
Superconducting



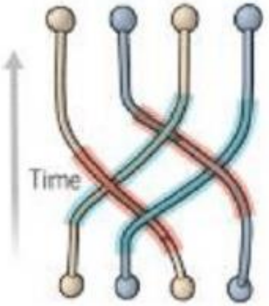
Neutral Atoms



Trapped-Ions



Photonic



Topological

Diamond Center

Hardware state of the art – qubit physical realization

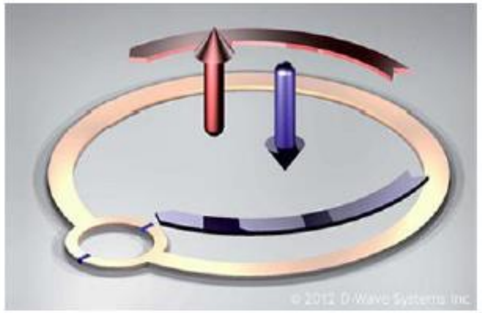
PASQAL

IBM Q

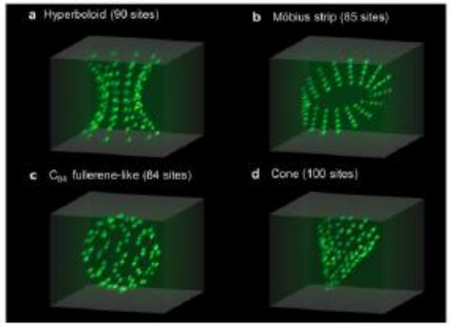
rigetti

D:WAVE
The Quantum Computing Company™

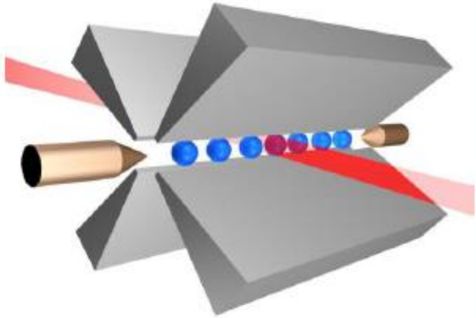
Google AI Quantum



Superconducting



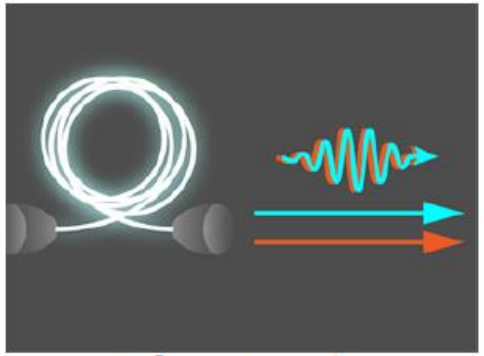
Neutral Atoms



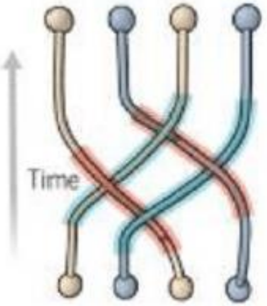
Trapped-Ions

IONQ

XANADU



Photonic



Topological

Diamond Center

QUANTUM BRILLIANCE

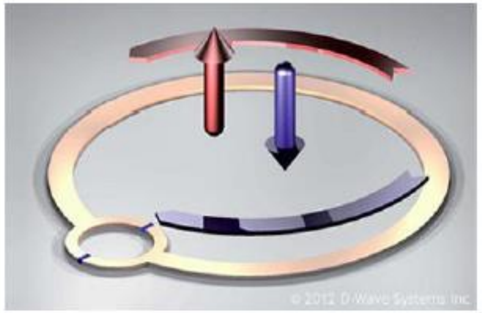
Microsoft

CINECA QUANTUM COMPUTING LAB

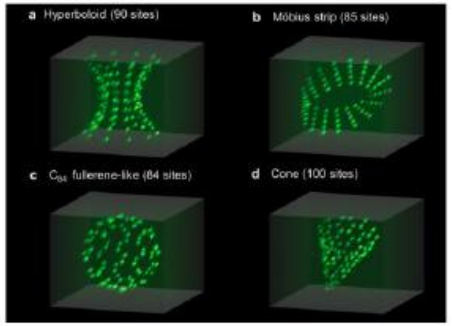
Hardware state of the art – qubit physical realization

PASQAL

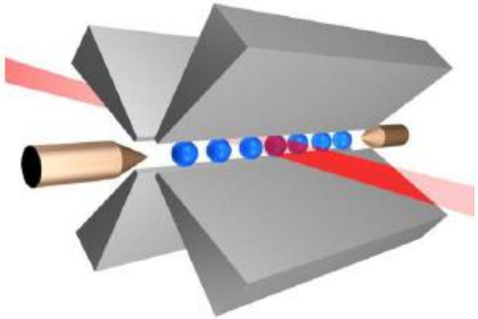
IBM Q
rigetti
D:WAVE
The Quantum Computing Company™
Google AI Quantum



Superconducting



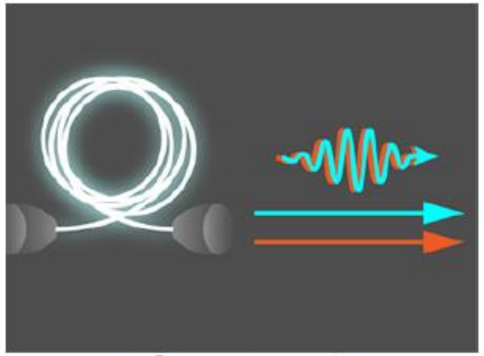
Neutral Atoms



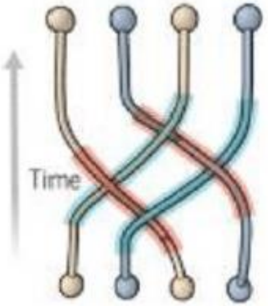
Trapped-Ions

IONQ

XANADU



Photonic



Topological

Microsoft

Diamond Center

QUANTUM BRILLIANCE

EuroHPC call 2020: Pilot on Quantum Simulator

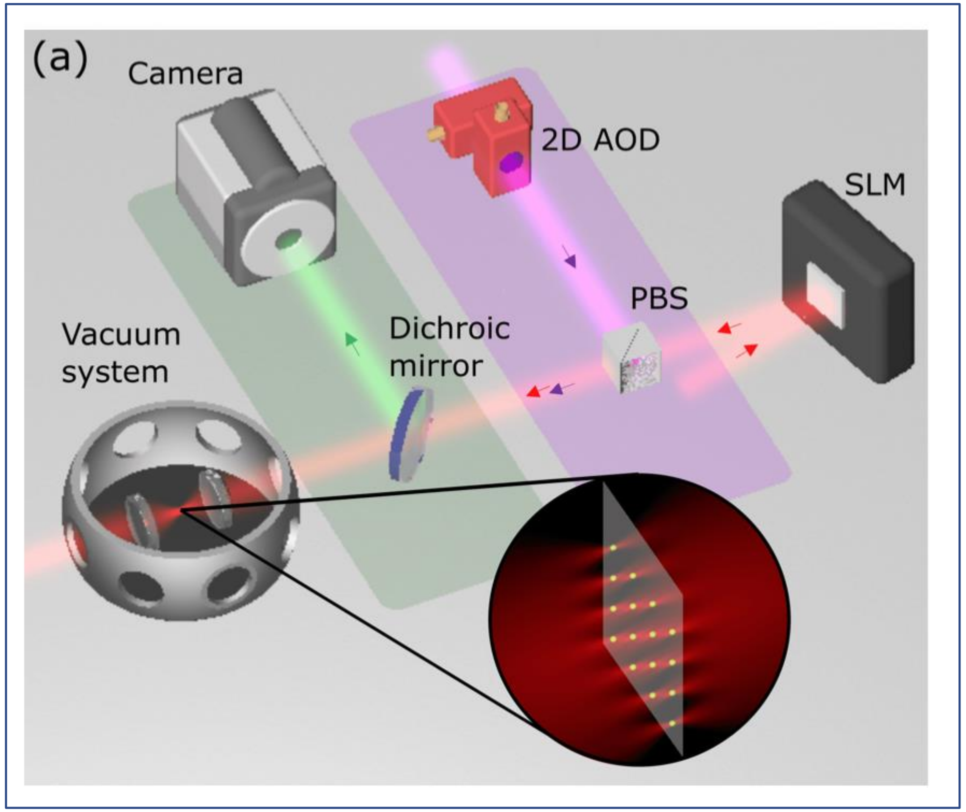
- The **project** will last 4 years, during which it will be created the **conditions** to **integrate quantum simulators with the European HPC network**.

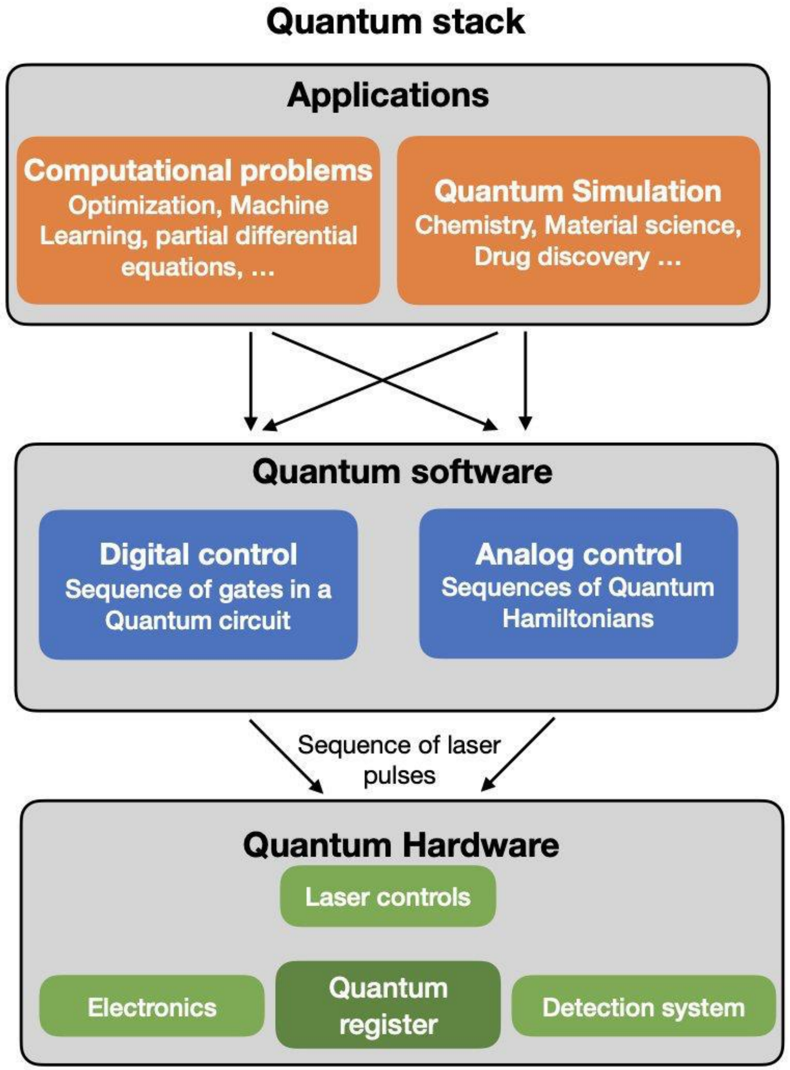
- The **aim** is to create an **integrated ecosystem**.

- **PASQAL** announced that it already has a **quantum simulator prototype** with **100 qubits** (scalable up to 1000).

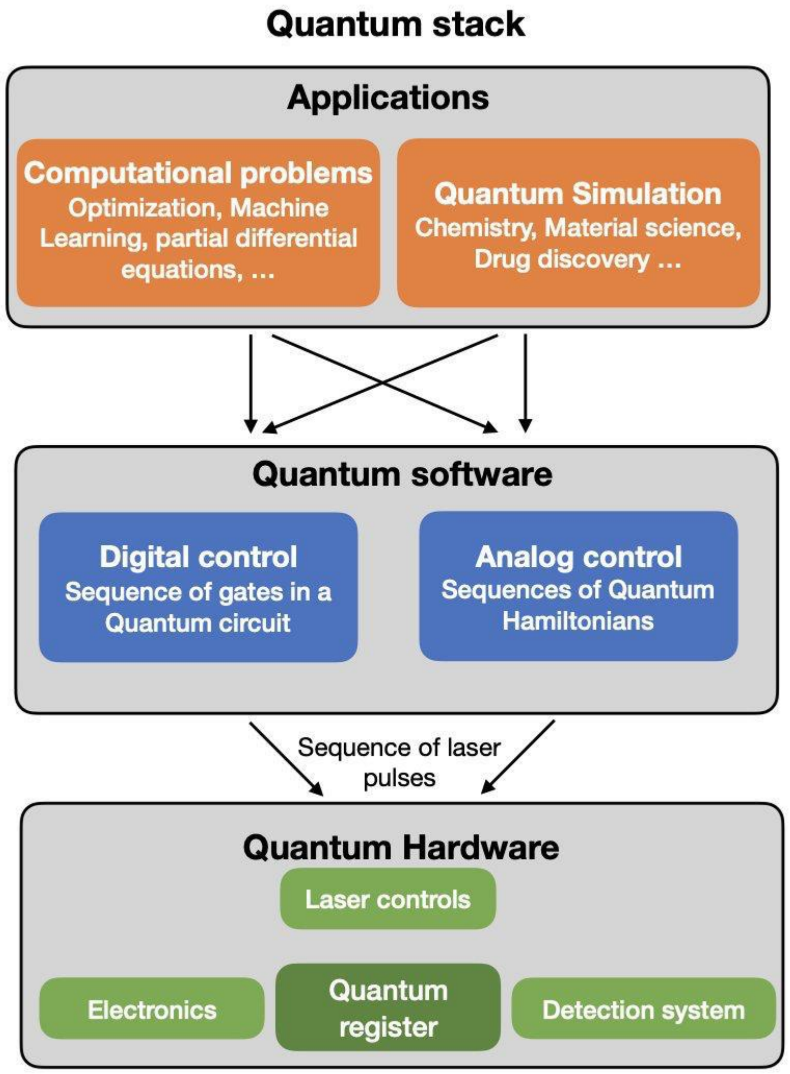
Hybrid Ecosystem

EuroHPC call 2020: Pilot on Quantum Simulator





EuroHPC call 2020: Pilot on Quantum Simulator

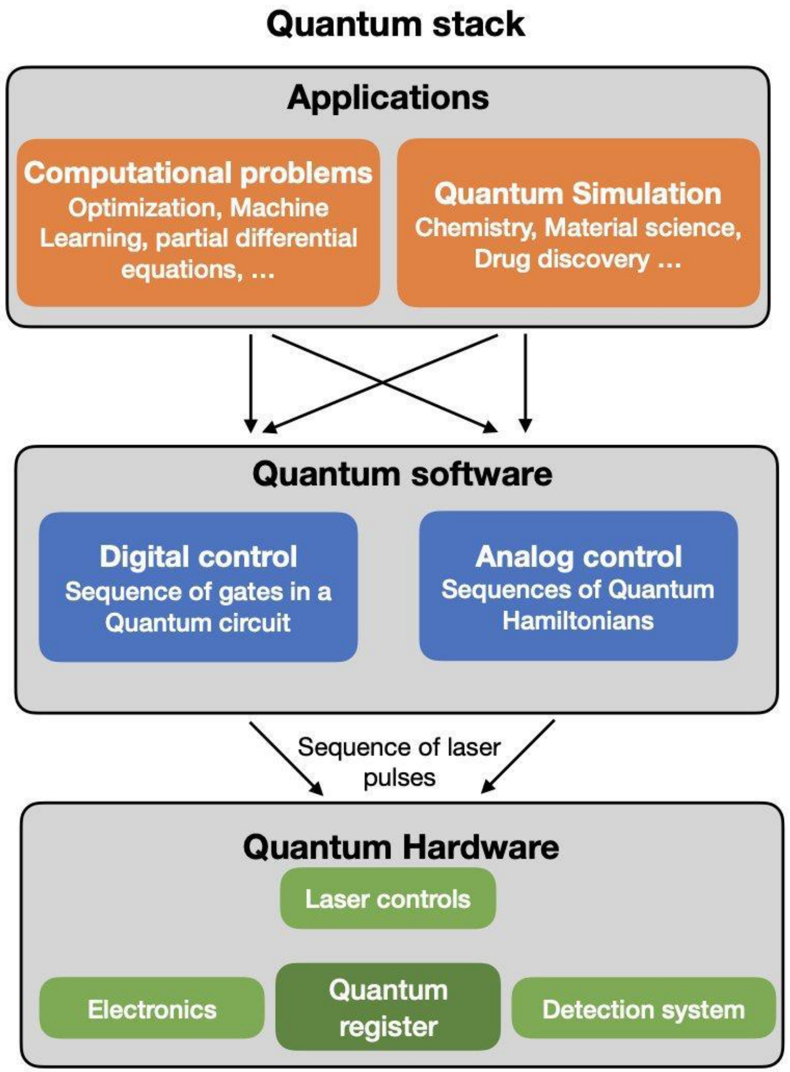


Pulser

Cirq

EuroHPC call 2020: Pilot on Quantum Simulator

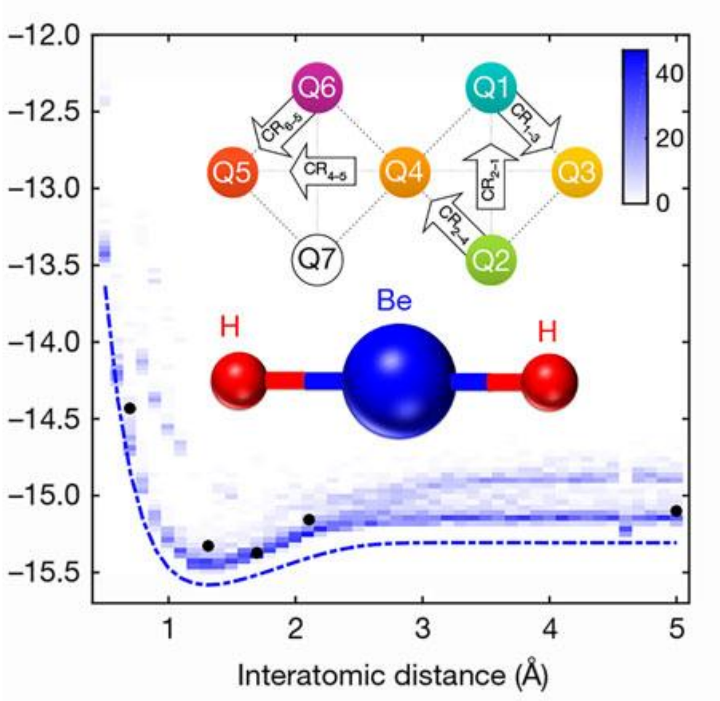
NISQ Algorithms
(Noisy Intermidiate Scale Quantum)



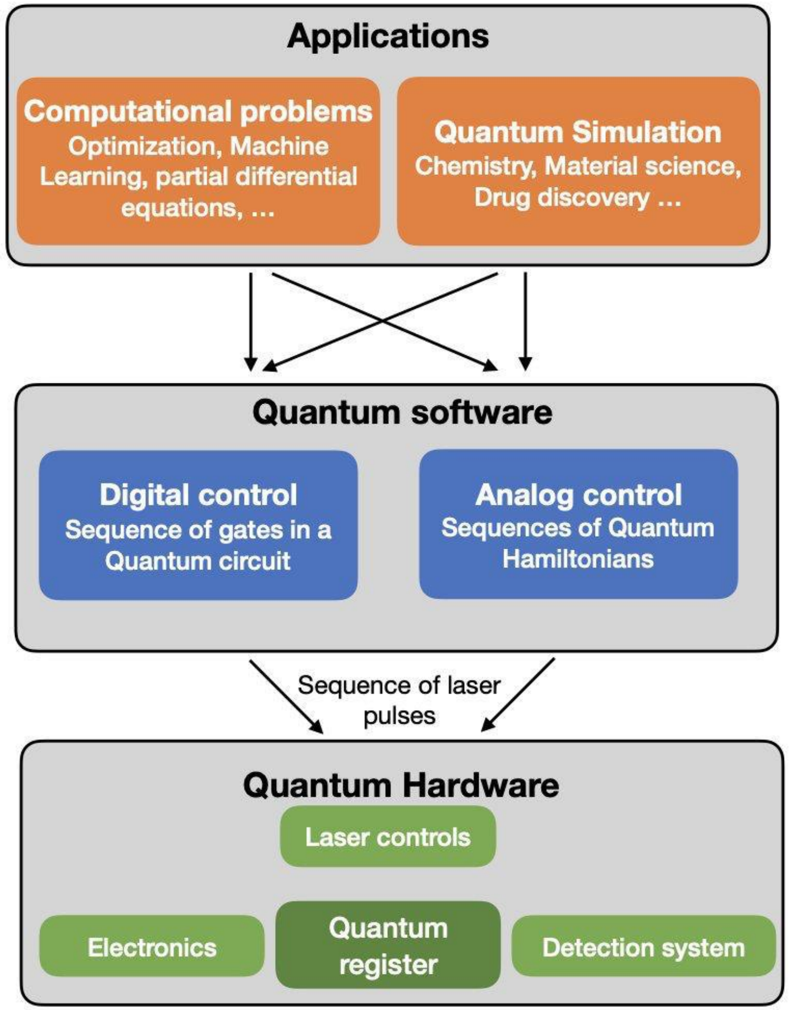
EuroHPC call 2020: Pilot on Quantum Simulator

NISQ Algorithms
(Noisy Intermidiate Scale Quantum)

VQE



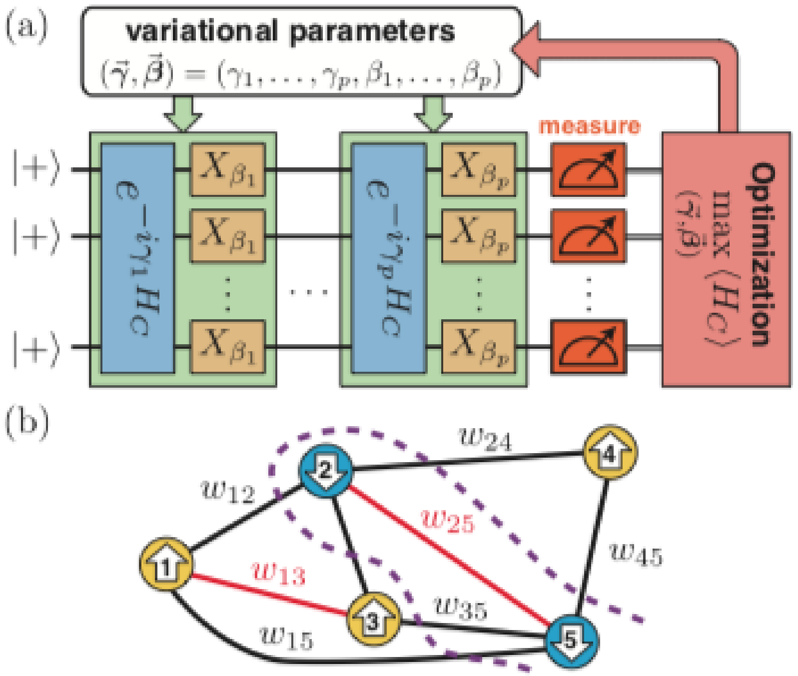
Quantum stack

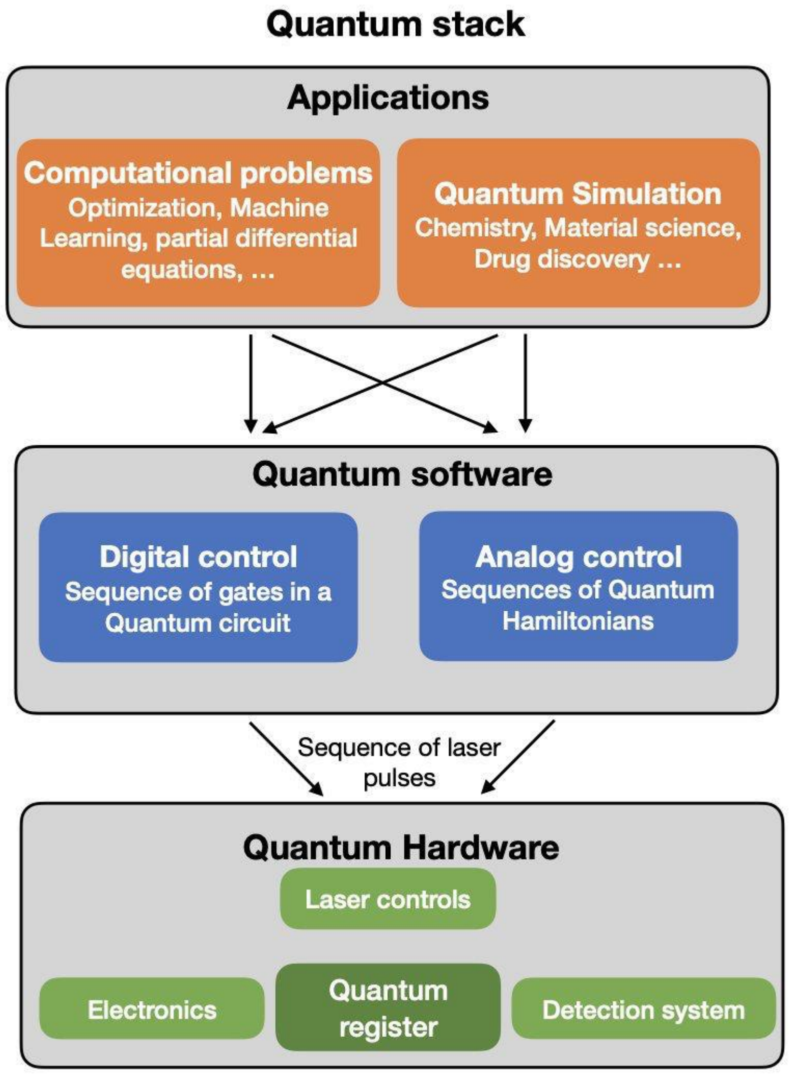


EuroHPC call 2020: Pilot on Quantum Simulator

NISQ Algorithms
(Noisy Intermidiate Scale Quantum)

QAOA

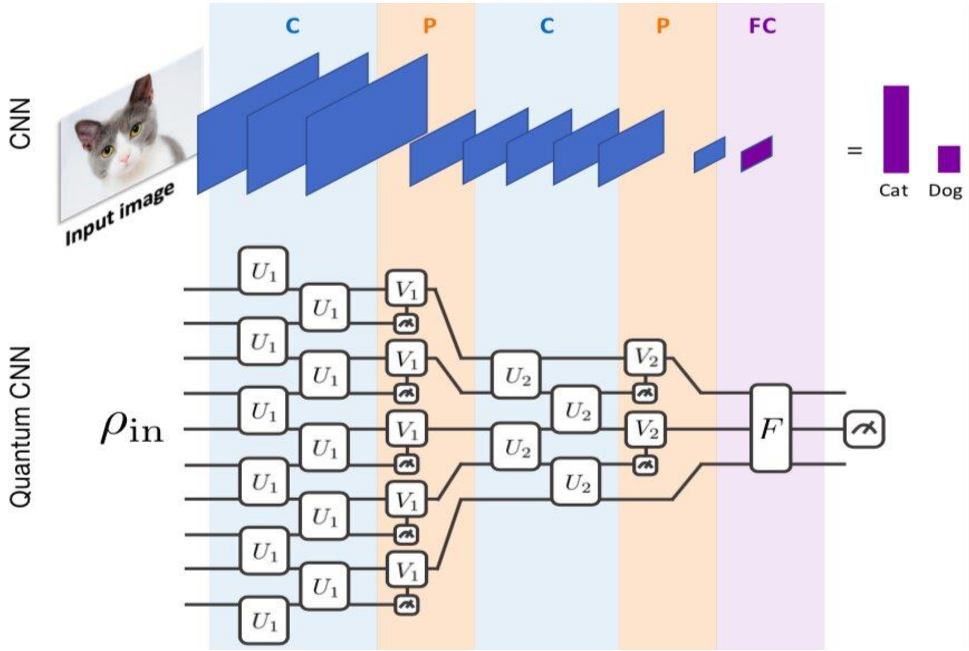


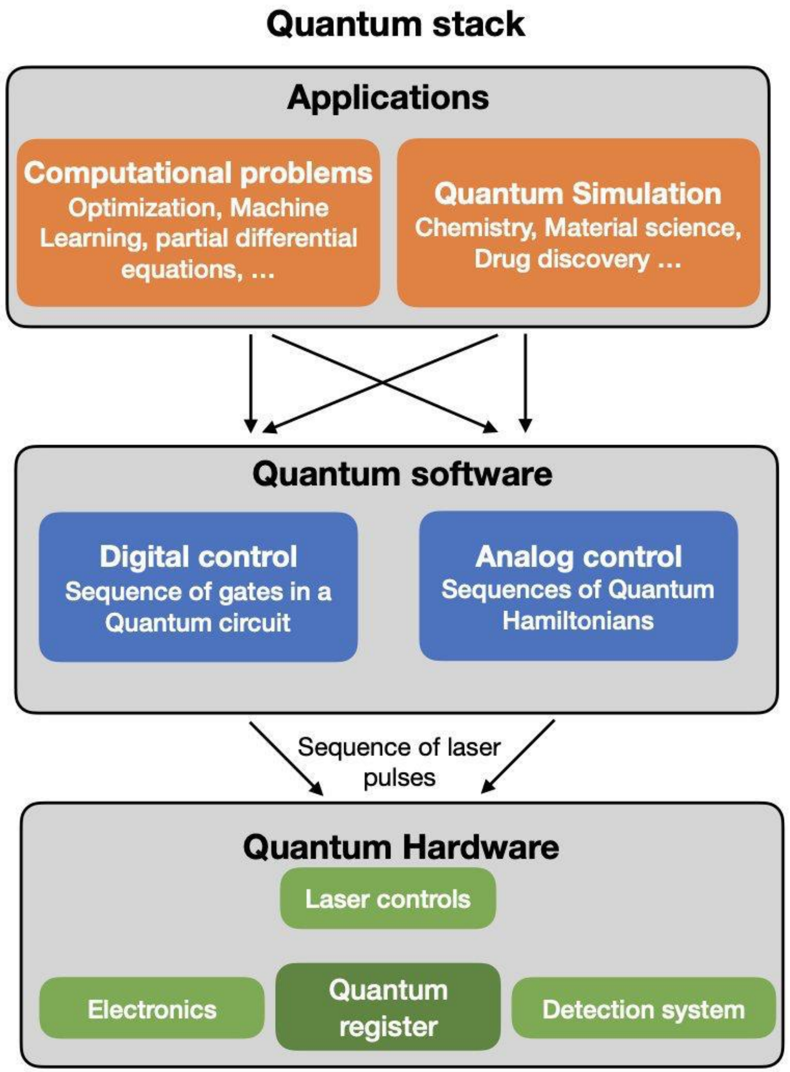


EuroHPC call 2020: Pilot on Quantum Simulator

NISQ Algorithms
(Noisy Intermidiate Scale Quantum)

QNN



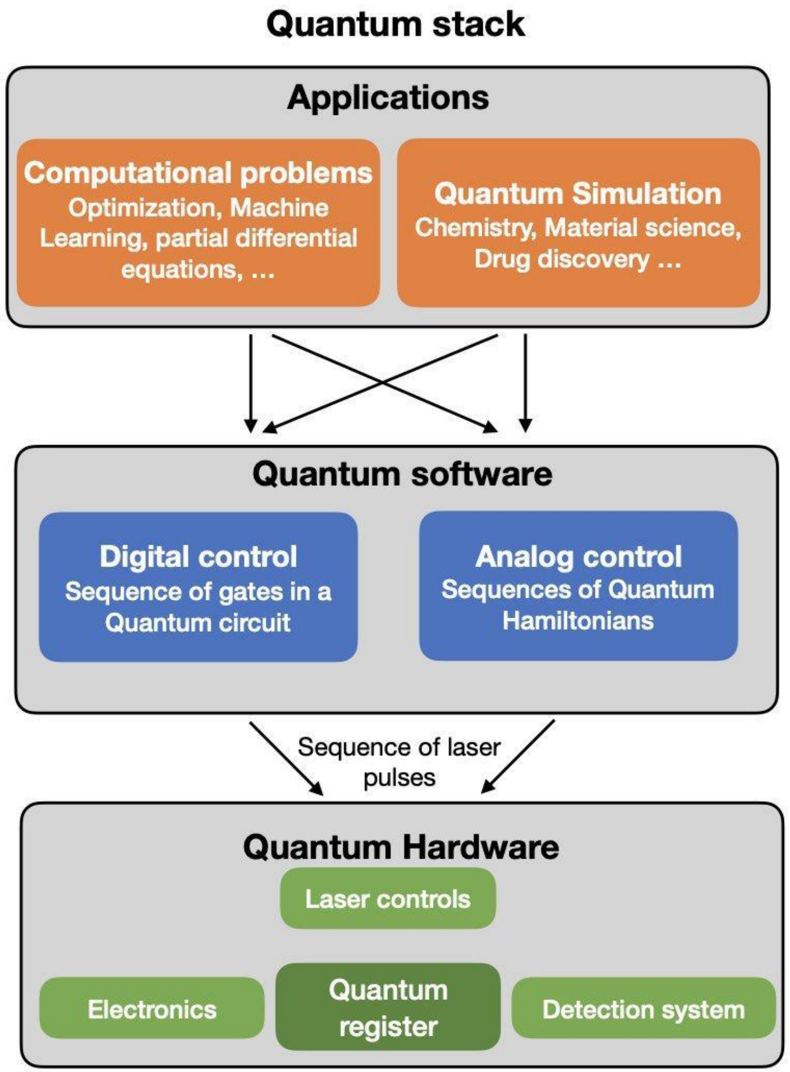


CINECA QUANTUM COMPUTING LAB

EuroHPC call 2020: Pilot on Quantum Simulator

NISQ Algorithms
(Noisy Intermidiate Scale Quantum)

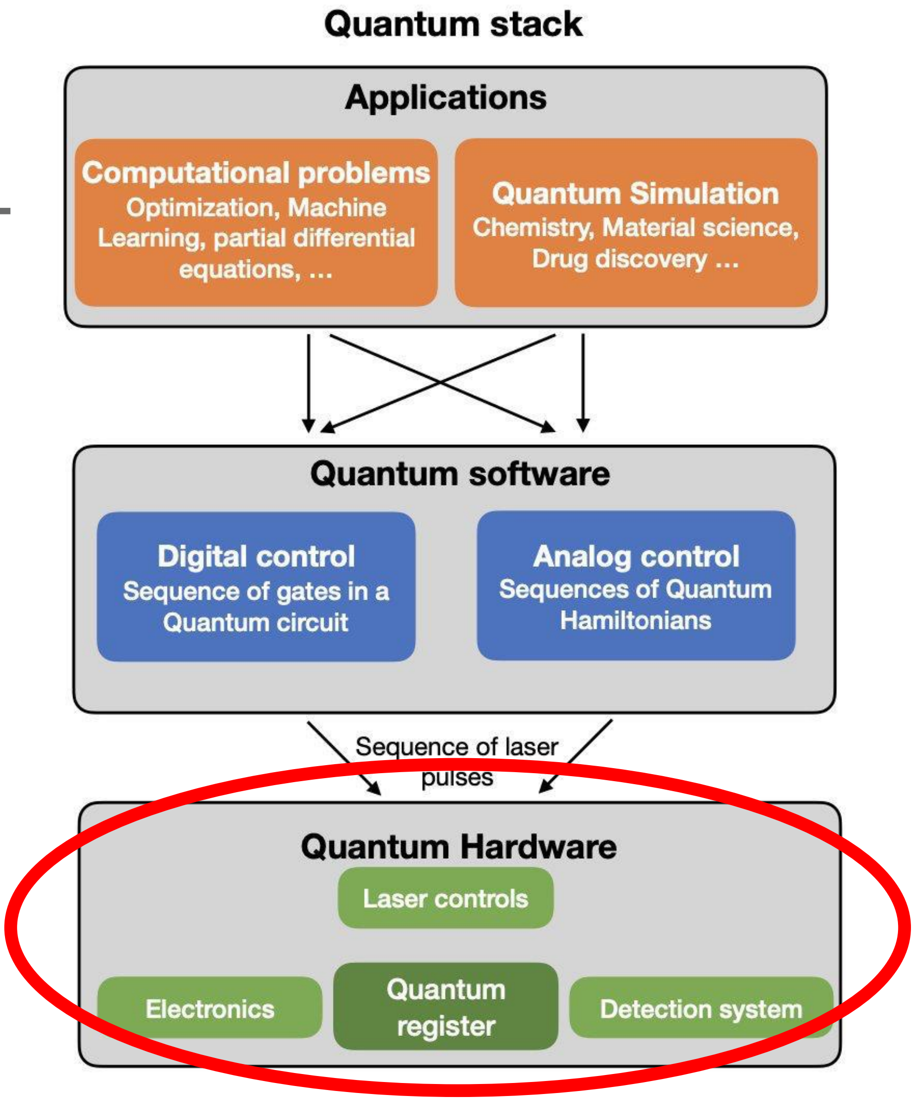
HPC



EuroHPC call 2020: Pilot on Quantum Simulator

NISQ Algorithms
(Noisy Intermidiate Scale Quantum)

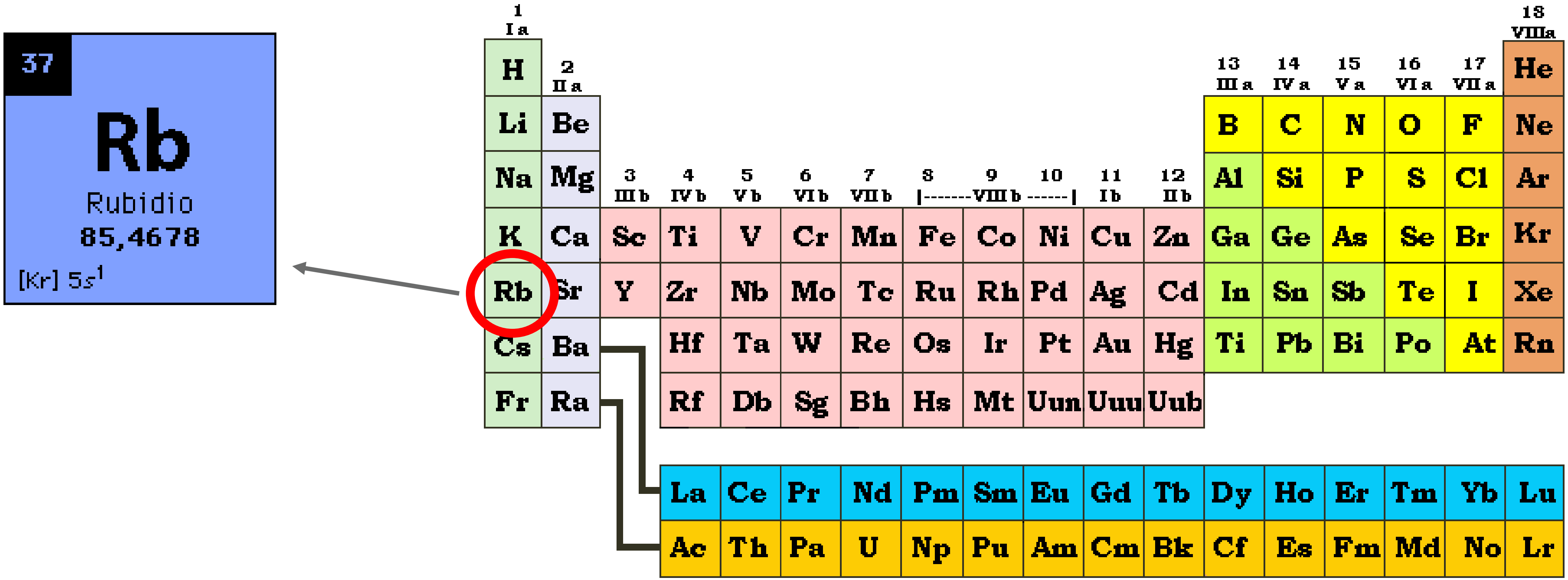
HPC



Pasqal Quantum Hardware: QC with Neutral Atoms

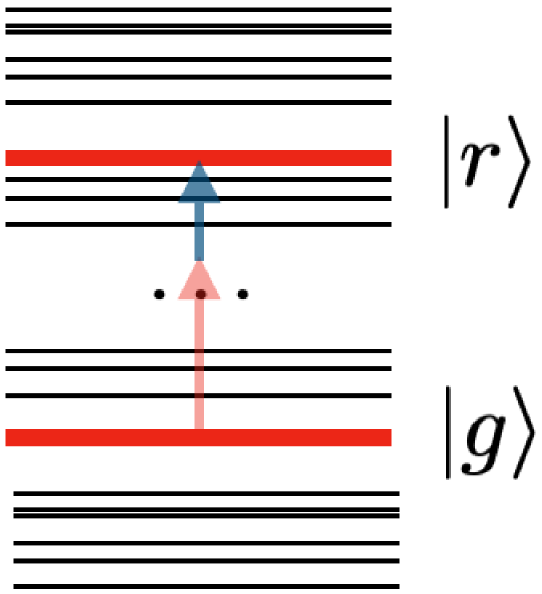
Pasqal Quantum Hardware: QC with Neutral Atoms

Pasqal employs **Rubidium Atoms** for its Neutral Atoms Quantum Computer



Pasqal Quantum Hardware: QC with Neutral Atoms

Pasqal employs **Rubidium Atoms** in the construction of the **QPU**



State of a Qubit

$$\{|\mathbf{0}\rangle, |\mathbf{1}\rangle\}$$

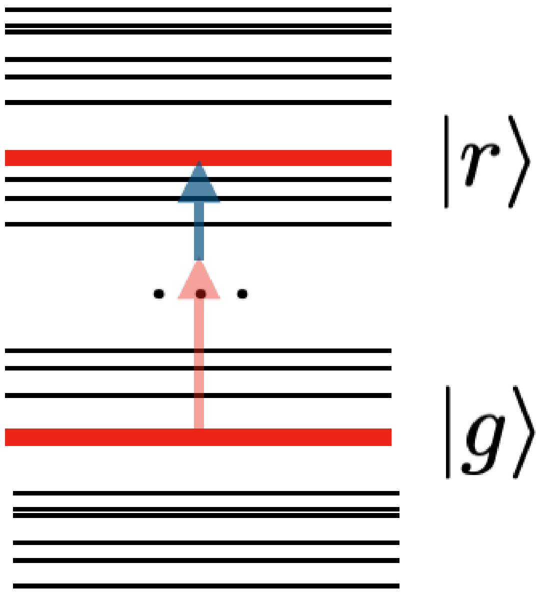
encoded in two electronic levels of the Rubidium Atom

$$\{|\boldsymbol{g}\rangle, |\boldsymbol{r}\rangle\}$$

Since the **atoms** are **indistinguishable**, even the **qubits are strictly identical**. This is a **great advantage** for obtaining **low error levels** when calculating.

Pasqal Quantum Hardware: QC with Neutral Atoms

Pasqal employs **Rubidium Atoms** in the construction of the **QPU**

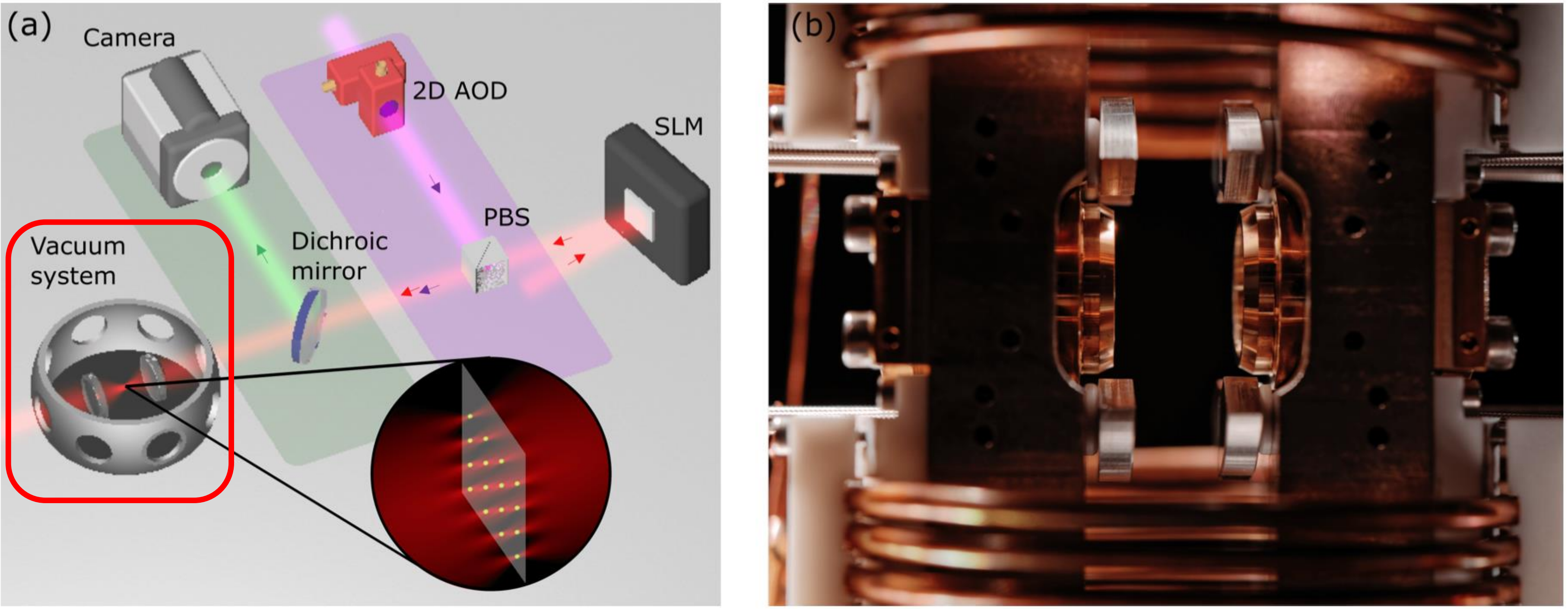


$\{|g\rangle, |r\rangle\}$ are ground and «Rydberg» states characterized by:

- **Long decay time:** if excited to the state $|r\rangle$, the atom tends to stay in that state and does not decays immediately in ground state $|g\rangle$
- **Strong interaction between atoms**

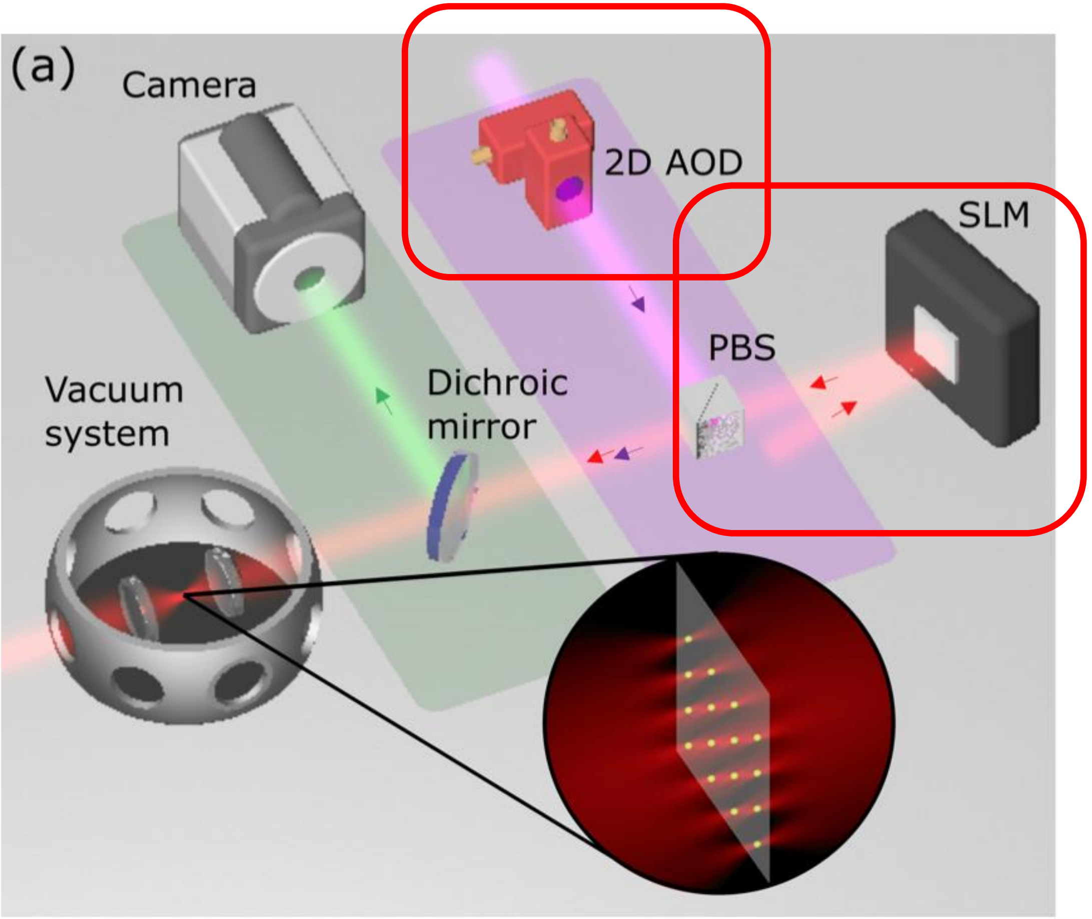
Pasqal Quantum Hardware: QC with Neutral Atoms

Pasqal employs **Rubidium Atoms** in the construction of the **QPU**



The atomic vapor is introduced into an ultra-high vacuum system operating at room temperature

Pasqal Quantum Hardware: QC with Neutral Atoms

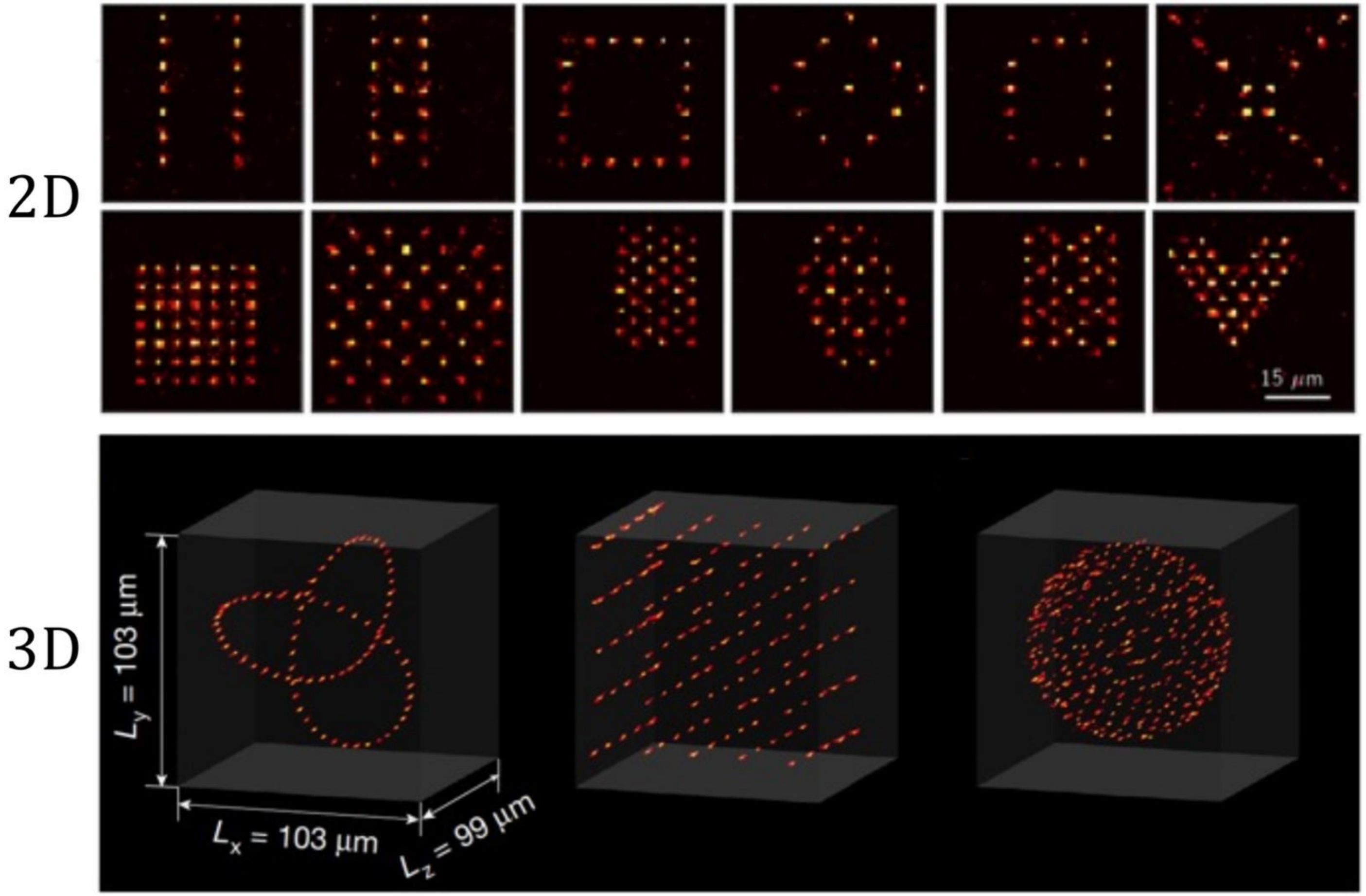


Rubidium atoms are trapped and held by laser beams, in particular:

- **Optical Tweezers (purple beam)** controlled by 2D acousto-optic laser deflector (AOD)
- **Laser (red beam)** reflected by spatial light modulator (SLM) which gives the correct phase

<u>Every Tweezers traps a single atom</u>

Pasqal Quantum Hardware: QC with Neutral Atoms

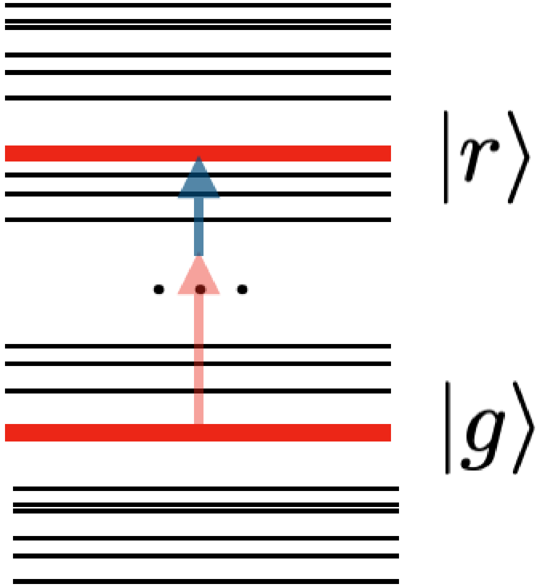


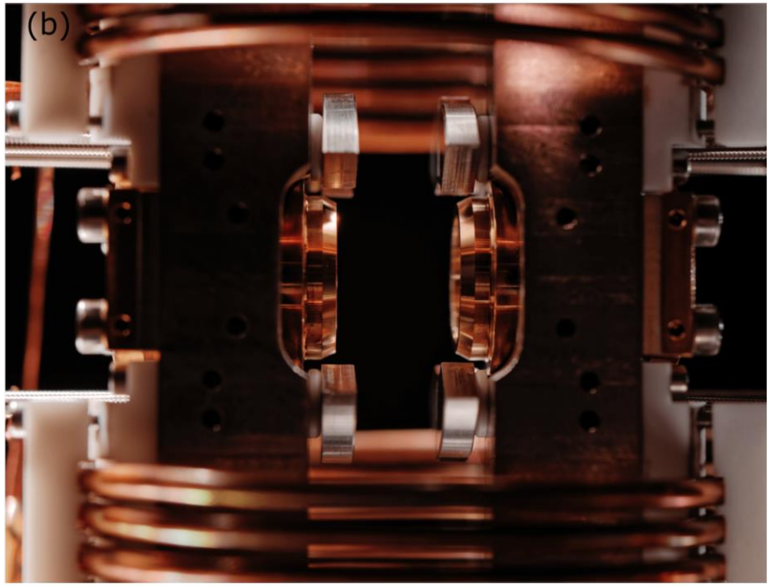
By **moving** the **optical tweezers** it is possible to arrange the **topology** of the Rubidium atoms and therefore of the qubits

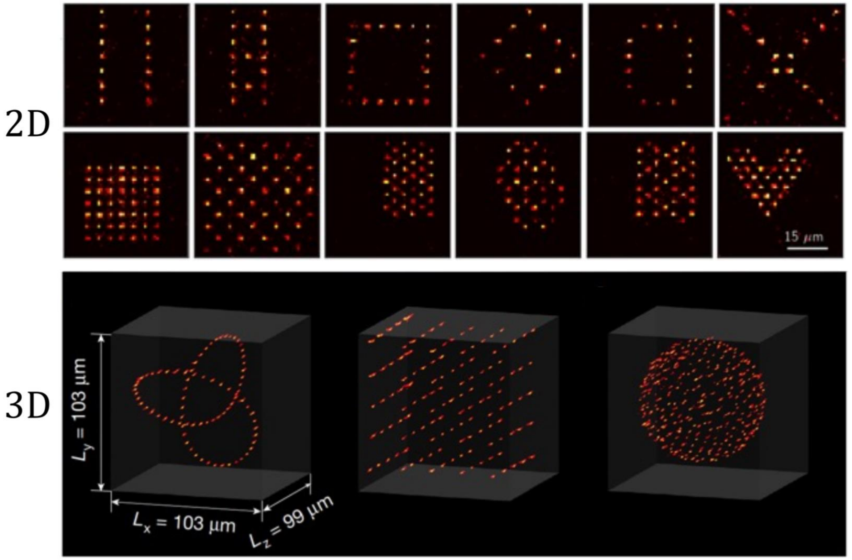
Depending on the application, it is useful to vary the Topology which can be 1D, 2D or even 3D

Pasqal Quantum Hardware: QC with Neutral Atoms

Pasqal employs **Rubidium Atoms** in the construction of the **QPU**



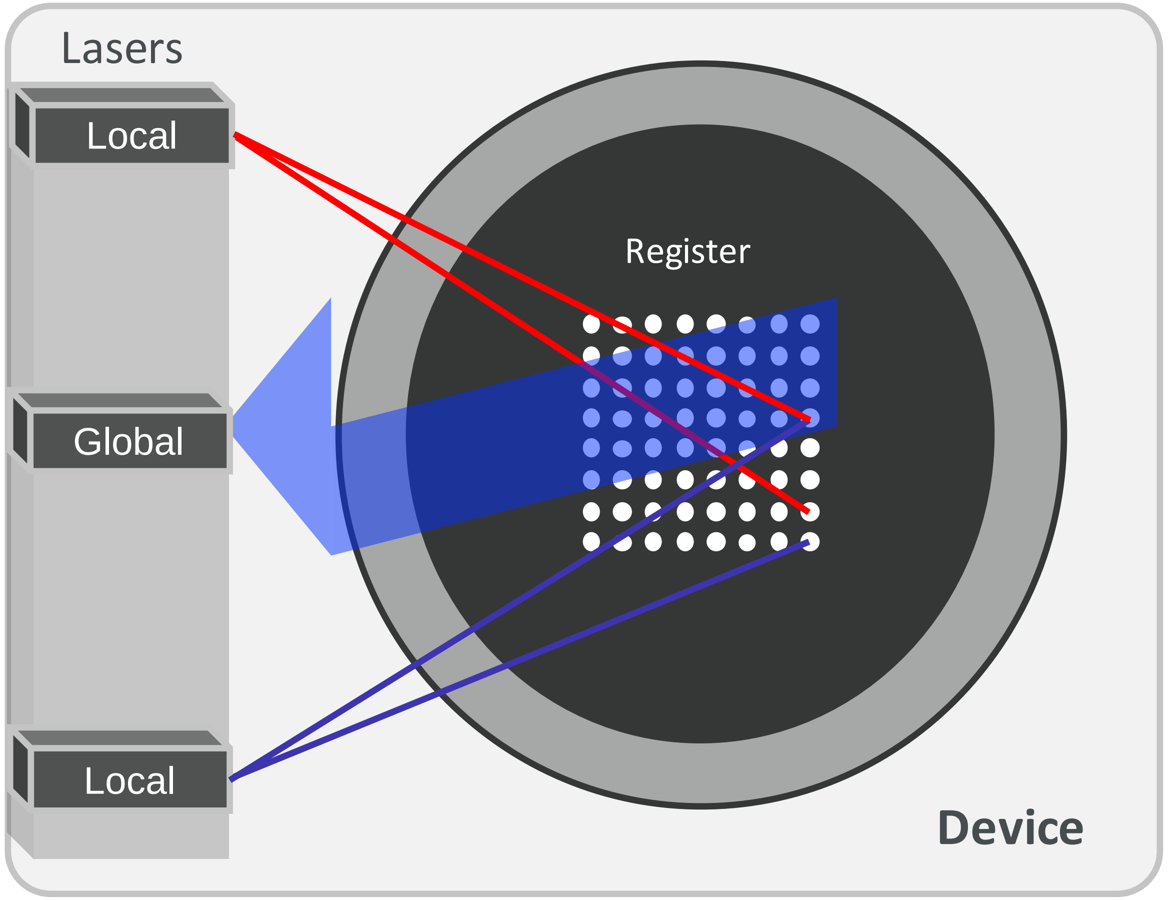




How does quantum computation happen?

CINECA QUANTUM COMPUTING LAB

Pasqal Quantum Hardware: QC with Neutral Atoms



Local and global laser beams control the state of qubit registers and allow to:

- **Act on single qubit**

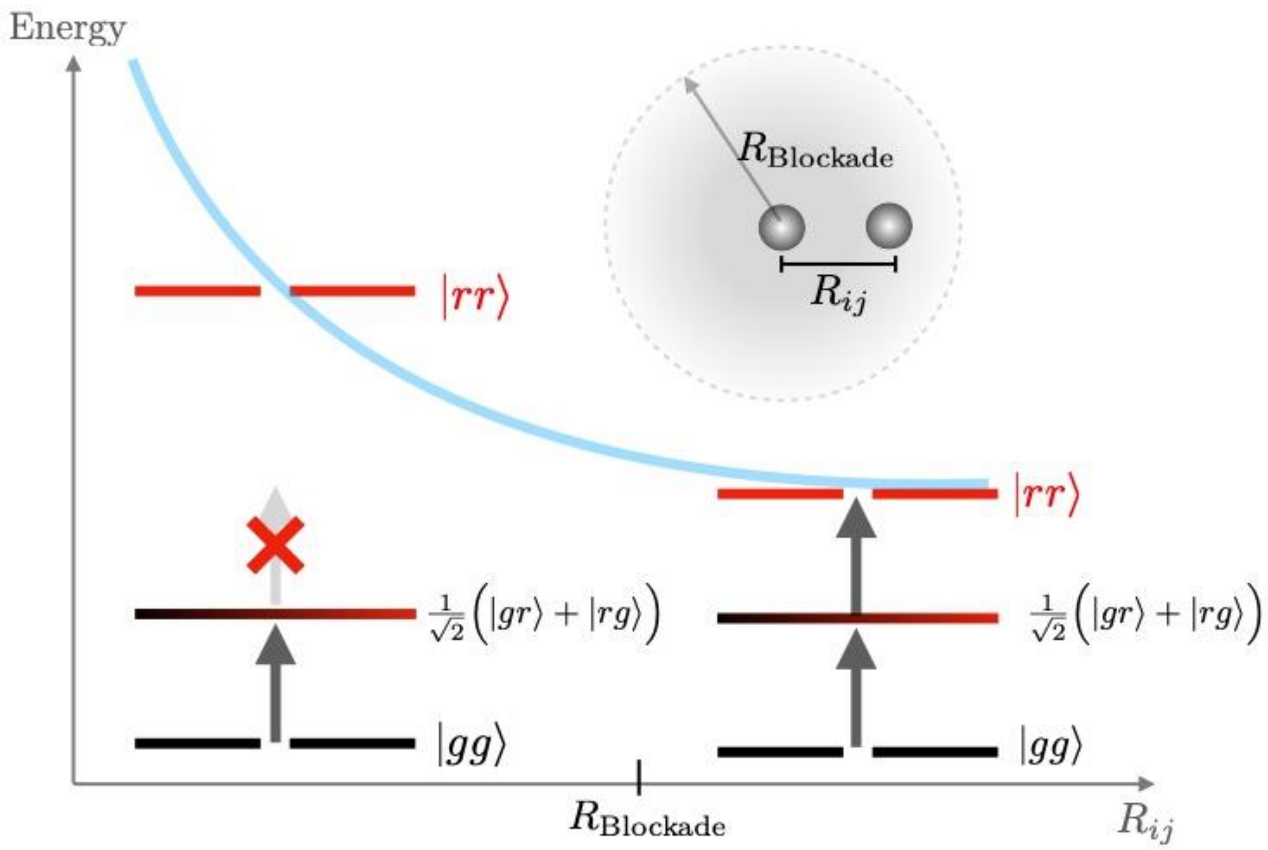
  e.g. $|g\rangle \rightarrow |r\rangle$

- **Make qubit interact**

  e.g. $|gg\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|gr\rangle + |rg\rangle\right)$

Pasqal Quantum Hardware: QC with Neutral Atoms

Rydberg Blockade: principle used to create entangled states



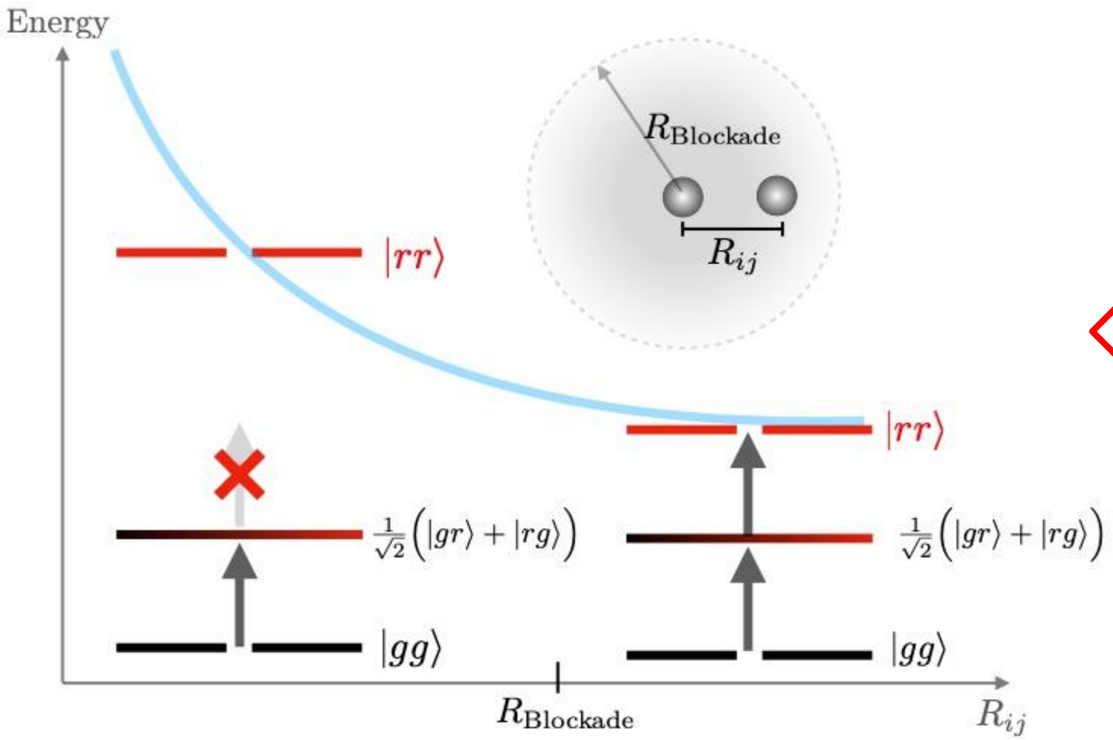
Qubits that are **within range** of the Rydberg blockade **interact** with each other.

The interaction within this radius is strong enough to make the **state** $|\mathbf{rr}\rangle$ **inaccessible**

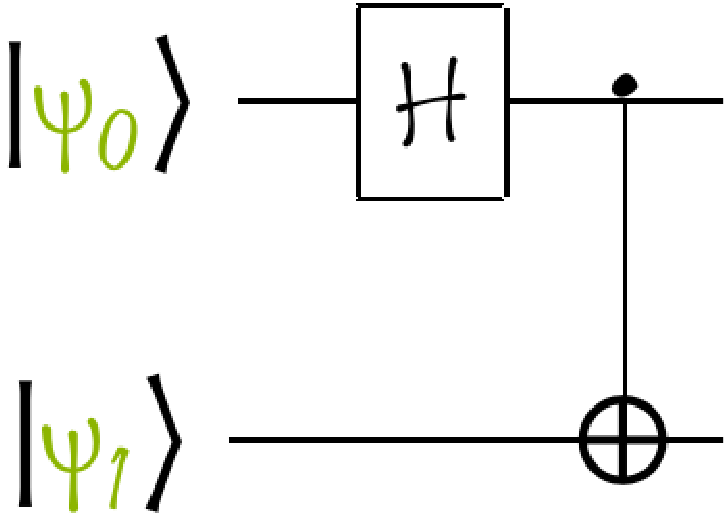
The resulting state is an **entangled state**, the same as obtained after a Hadamard gate followed by a CNOT

Pasqal Quantum Hardware: QC with Neutral Atoms

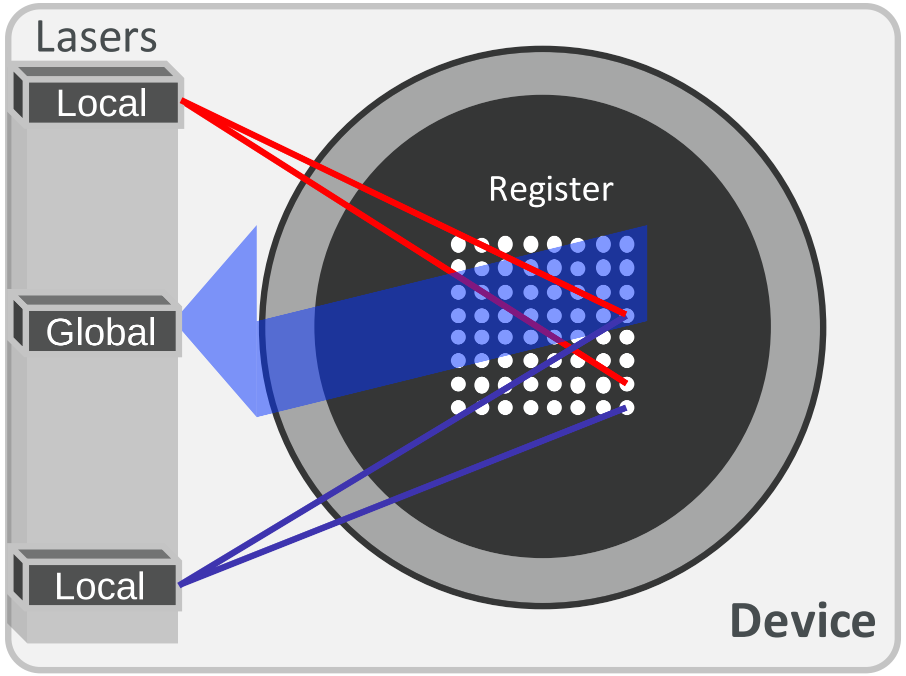
Rydberg Blockade: principle used to create entangled states



⇔



Pasqal Quantum Hardware: QC with Neutral Atoms



Mathematically, **lasers** interact with qubits, **modifying** the **Hamiltonian**, which is a function that describes the **energy** of the entire **qubit system**

$$H = \sum_i \frac{\hbar}{2}\left(\Omega(t)\sigma_i^x - \delta(t)\sigma_i^z\right) + \sum_{i<j} U_{ij}\hat{n}_i\hat{n}_j$$

CINECA QUANTUM COMPUTING LAB

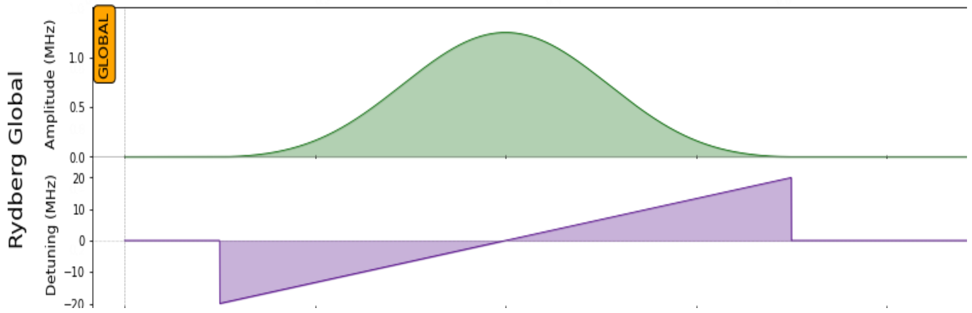
Pasqal Quantum Hardware: QC with Neutral Atoms

$$H = \sum_i \frac{\hbar}{2}\left(\Omega(t)\sigma_i^x - \delta(t)\sigma_i^z\right) + \sum_{i<j} U_{ij}\hat{n}_i\hat{n}_j$$

$$\hat{n}_j = (\mathbb{I} + \sigma_j^z)/2$$

Rabi Frequency **Detuning**

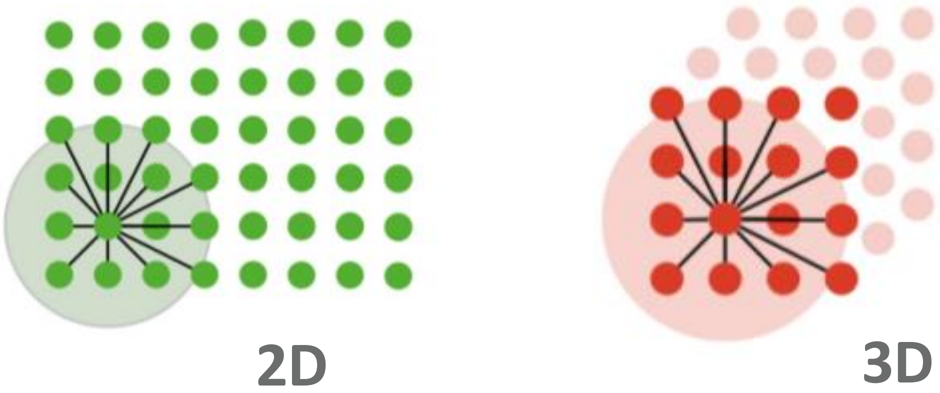
They vary by changing the **intensity** and **frequency** of the laser



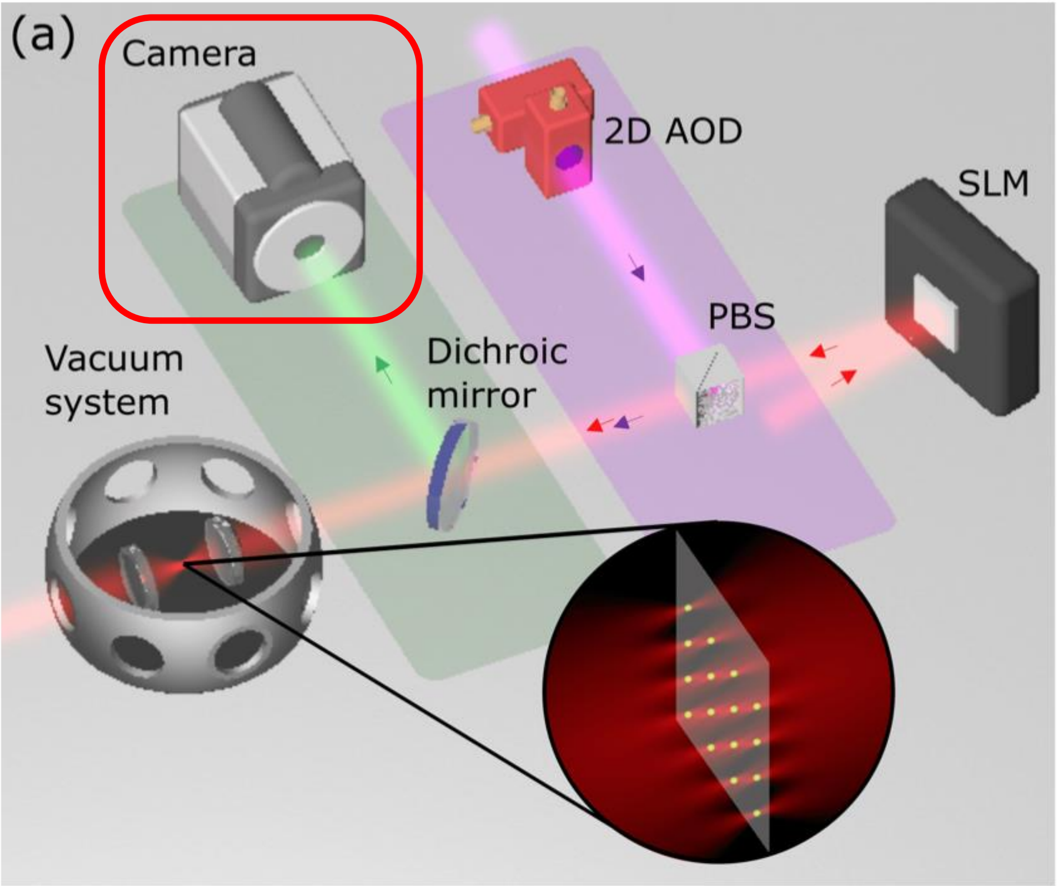
$$U_{ij} = \frac{C_6}{r_{ij}^6}$$

Modulates interaction between qubits

Vary with **Topology**



Pasqal Quantum Hardware: QC with Neutral Atoms



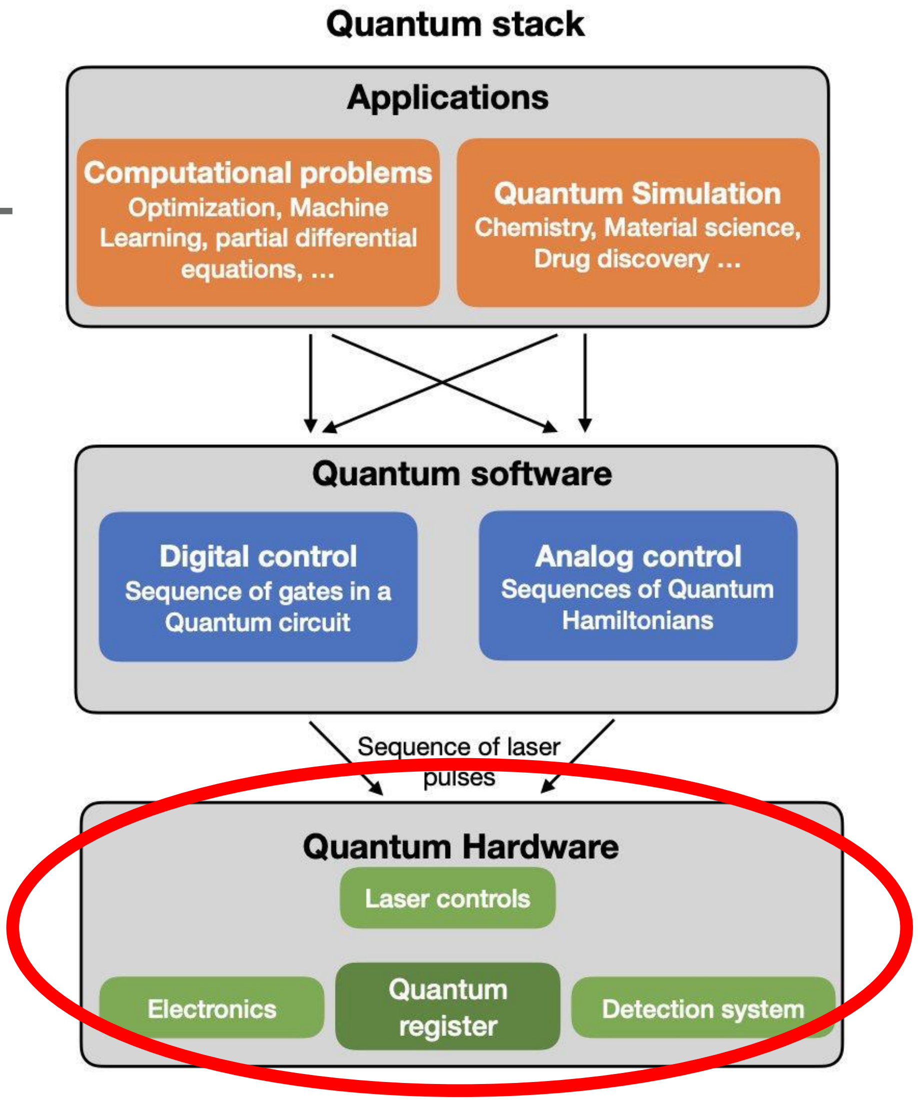
At the **end of the computation**, the qubit register is **measured** by **observing the final fluorescence image** (green beam).

The measurement process is performed in such a way that **each atom in the qubit state | 0⟩ appears bright,** while **the atoms in the qubit state | 1⟩ remain dark**.

Pasqal Quantum Hardware: QC with Neutral Atoms

NISQ Algorithms
(Noisy Intermidiate Scale Quantum)

HPC

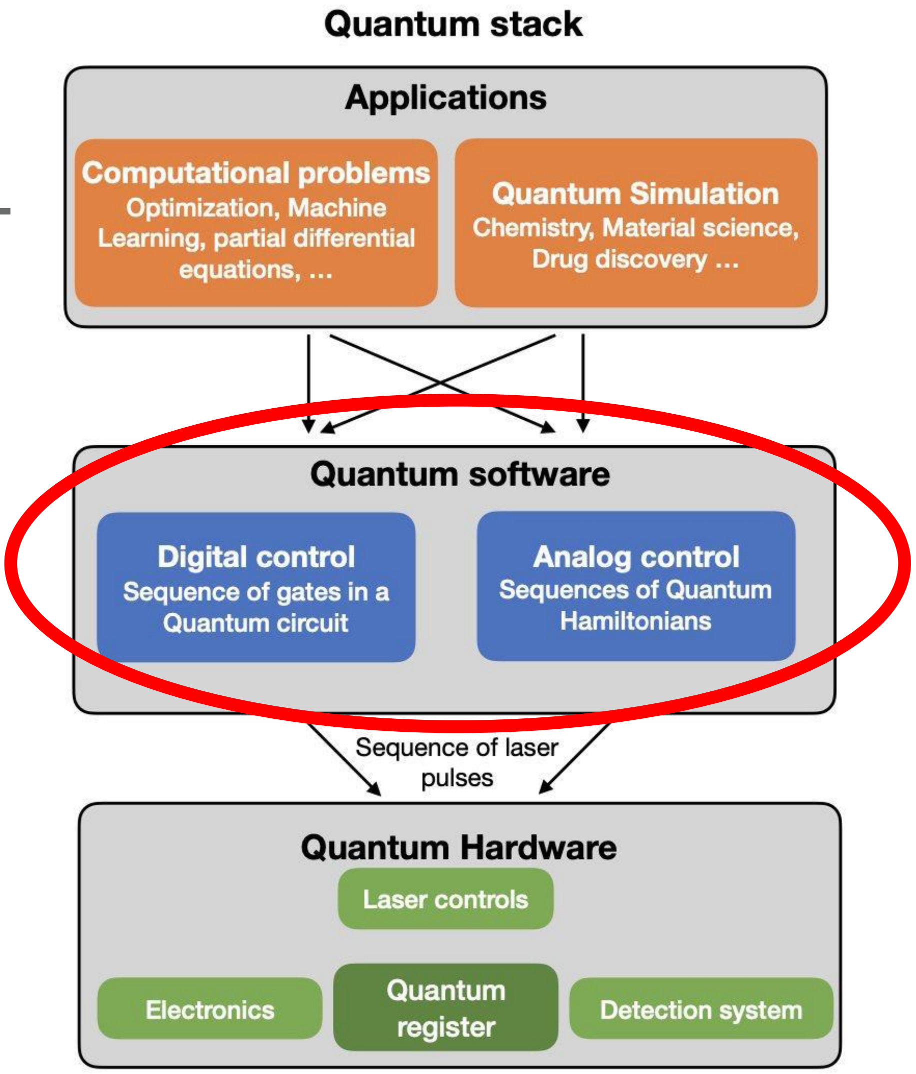


CINECA QUANTUM COMPUTING LAB

Pasqal Quantum Hardware: QC with Neutral Atoms

NISQ Algorithms
(Noisy Intermidiate Scale Quantum)

HPC

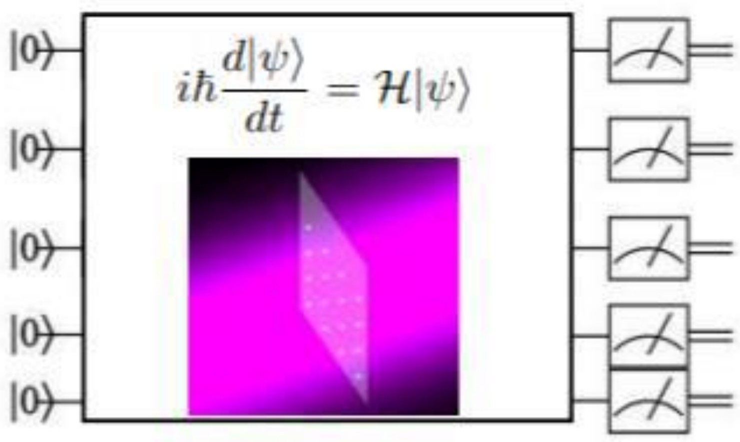


Pulser: Control Software for Pasqal QC

Pulser: Control Software for Pasqal QC

Lower level programming

(b) Analog processing



Pulser

Quantum computing is carried out by **directly manipulating** the mathematical operator (**Hamiltonian**) that **describes the evolution** of the **quantum system**

$$H = \sum_i \frac{\hbar}{2}\left(\Omega(t)\sigma_i^x - \delta(t)\sigma_i^z\right) + \sum_{i<j} U_{ij}\hat{n}_i\hat{n}_j$$

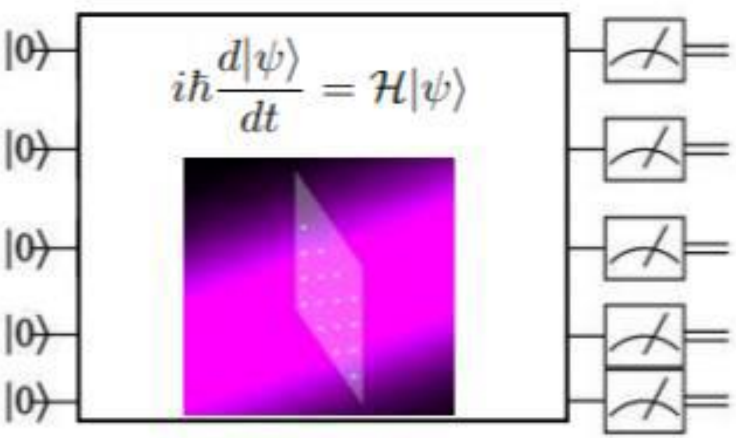
Possible by **varying**:

- **Intensity** and **frequency** of lasers
- Qubit register **topology**

Pulser: Control Software for Pasqal QC

Lower level programming

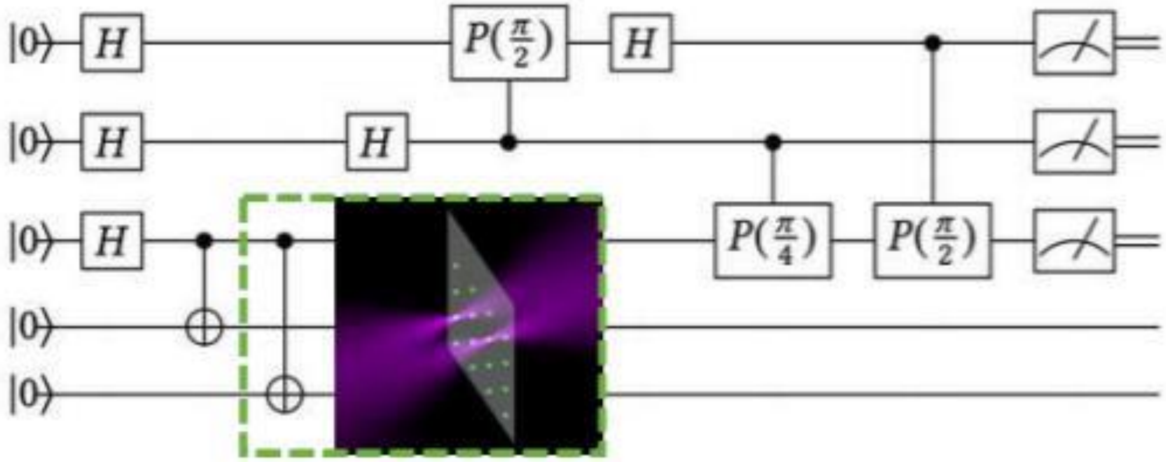
(b) Analog processing



Pulser

Higher level programming

(a) Digital processing



Cirq

Pulser: Control Software for Pasqal QC

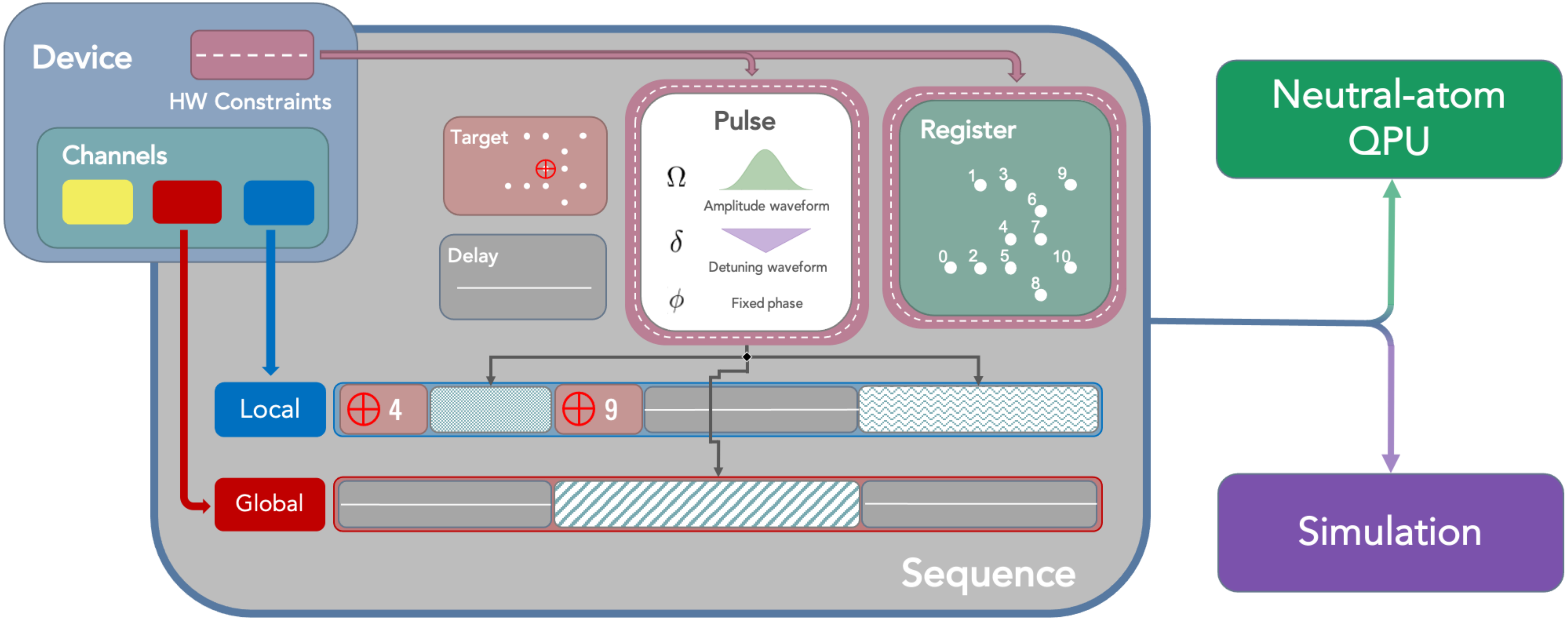
Python software library for **programming Pasqal devices** at the **laser pulse level**.

It allows to **design pulse sequences** that represent the physical parameters relevant to the computation.

The **sequences** can be **read** and **executed** by the **QPU** or by an **emulator**

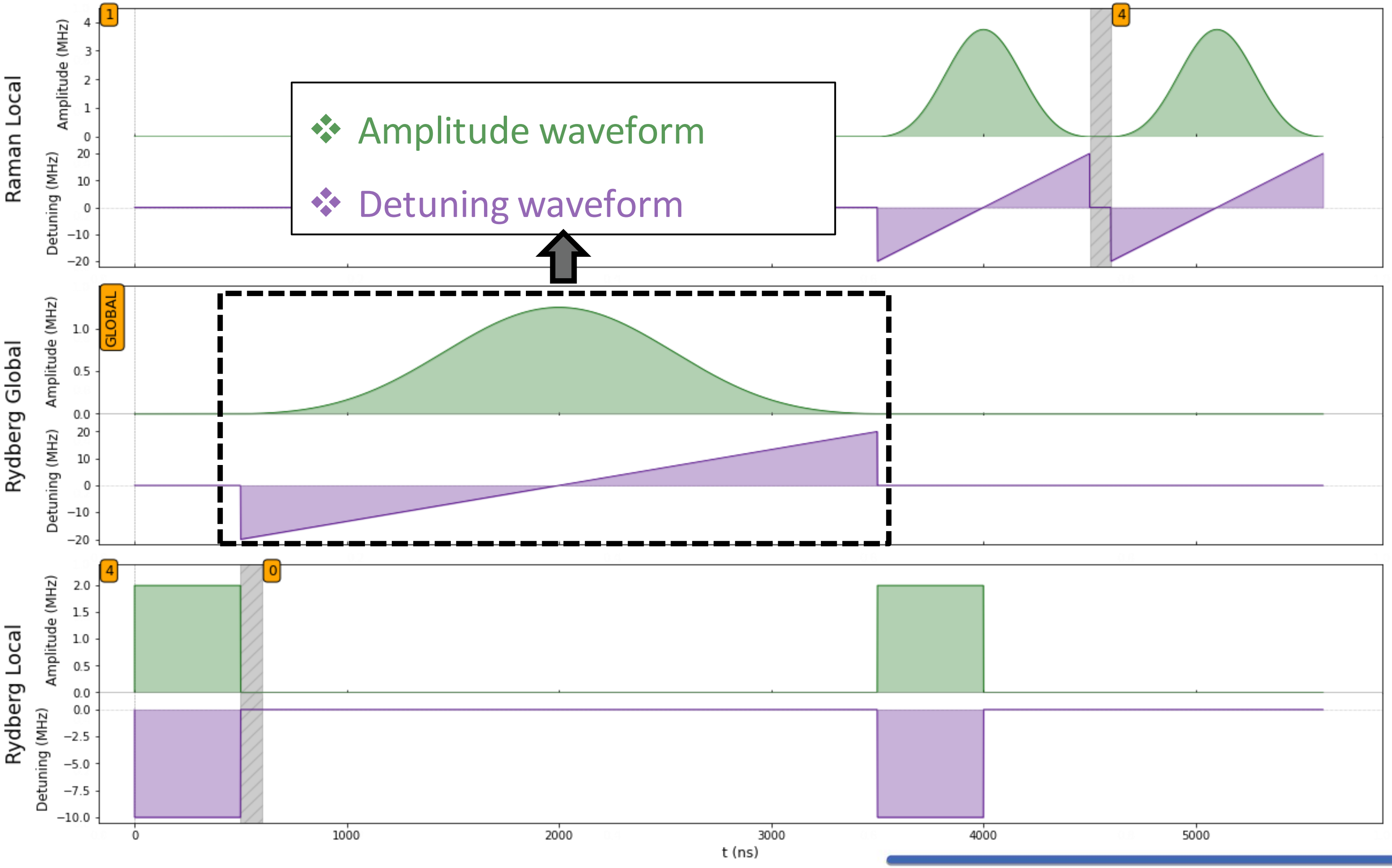
Pulser: Control Software for Pasqal QC

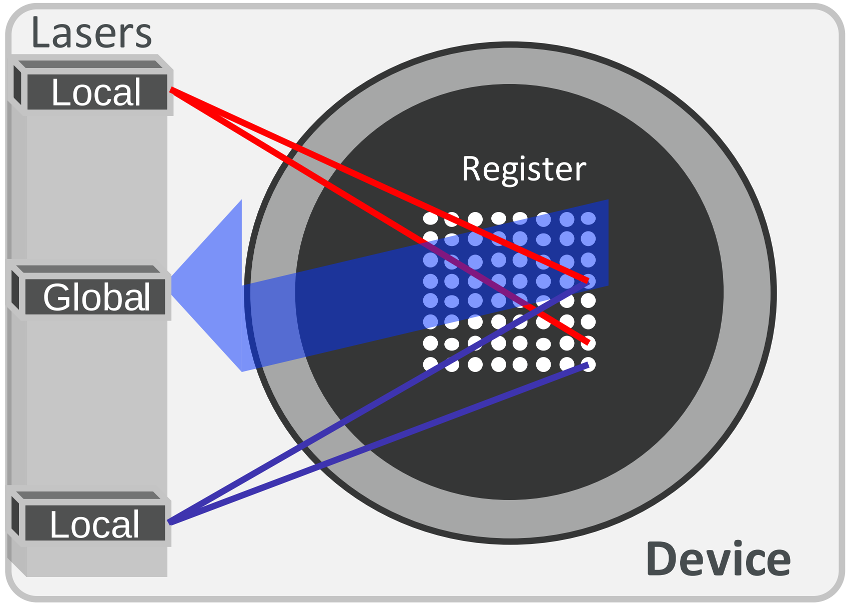
In Pulser, local and global pulse sequences can be defined



Pulser: Control Software for Pasqal QC

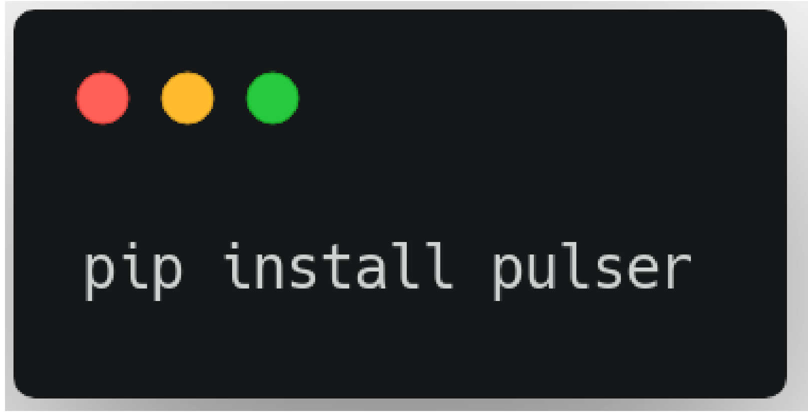
Practice Session

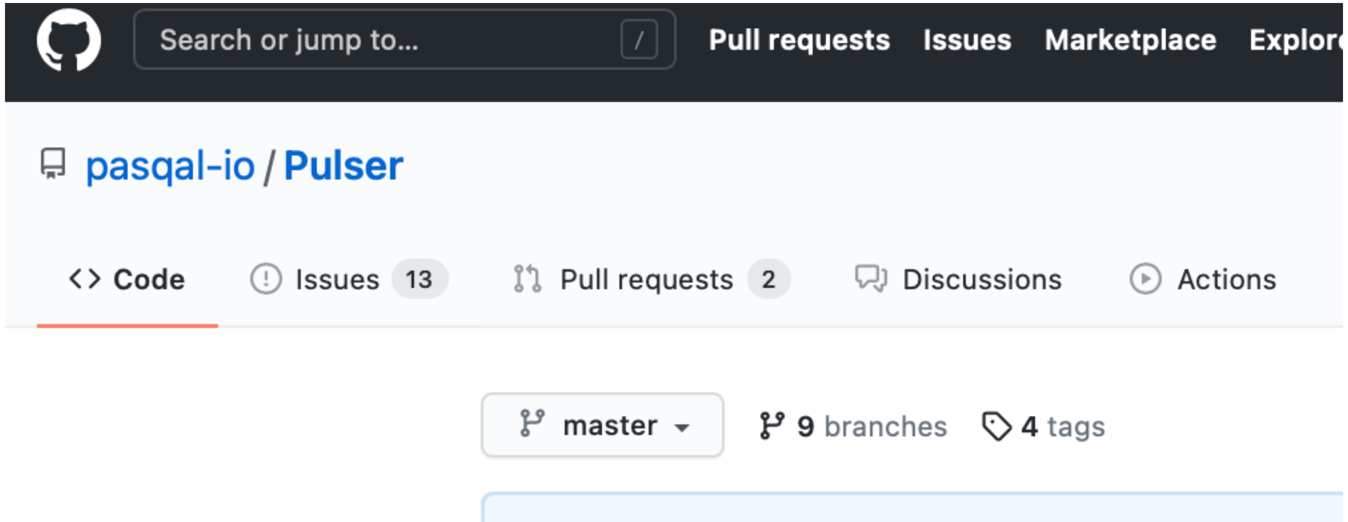




Pulser: Control Software for Pasqal QC

Practice Session



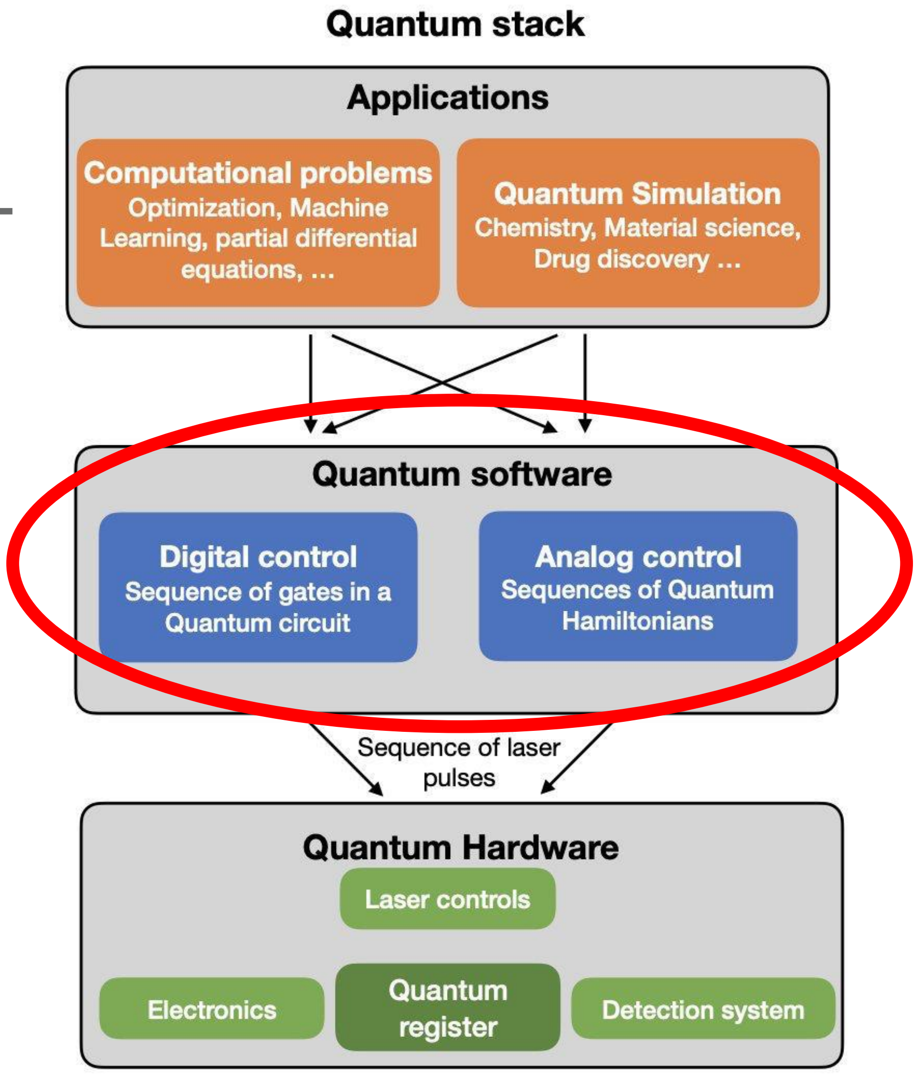


https://github.com/pasqal-io/Pulser

Pulser: Control Software for Pasqal QC

NISQ Algorithms
(Noisy Intermidiate Scale Quantum)

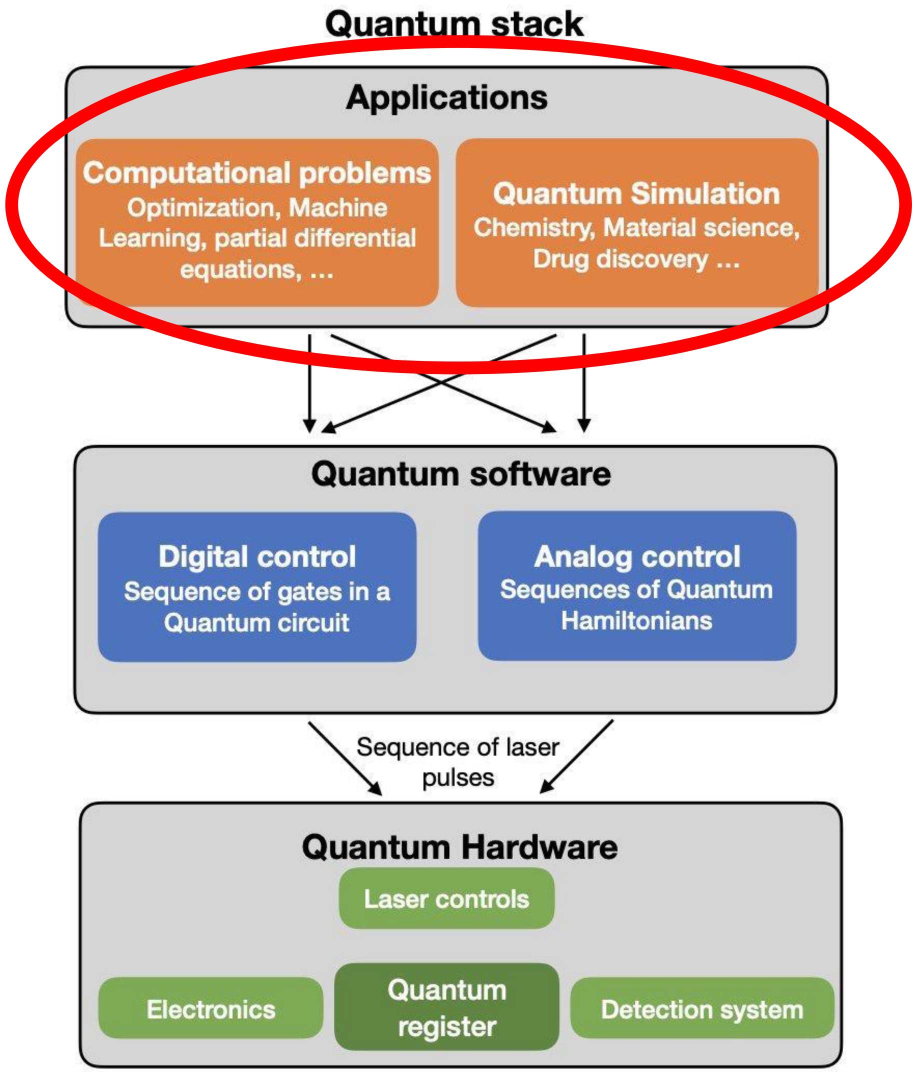
HPC



Pulser: Control Software for Pasqal QC

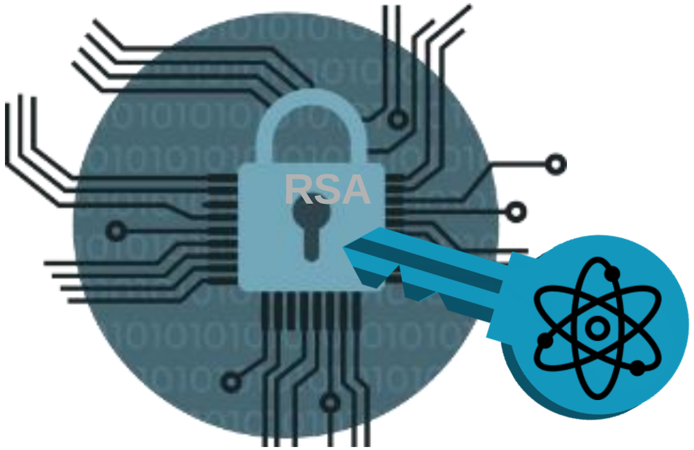
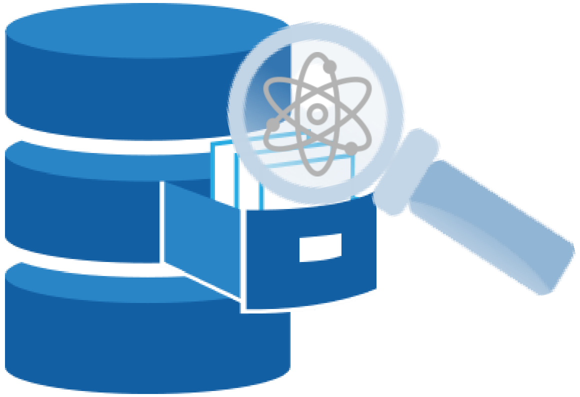
NISQ Algorithms
(Noisy Intermidiate Scale Quantum)

HPC



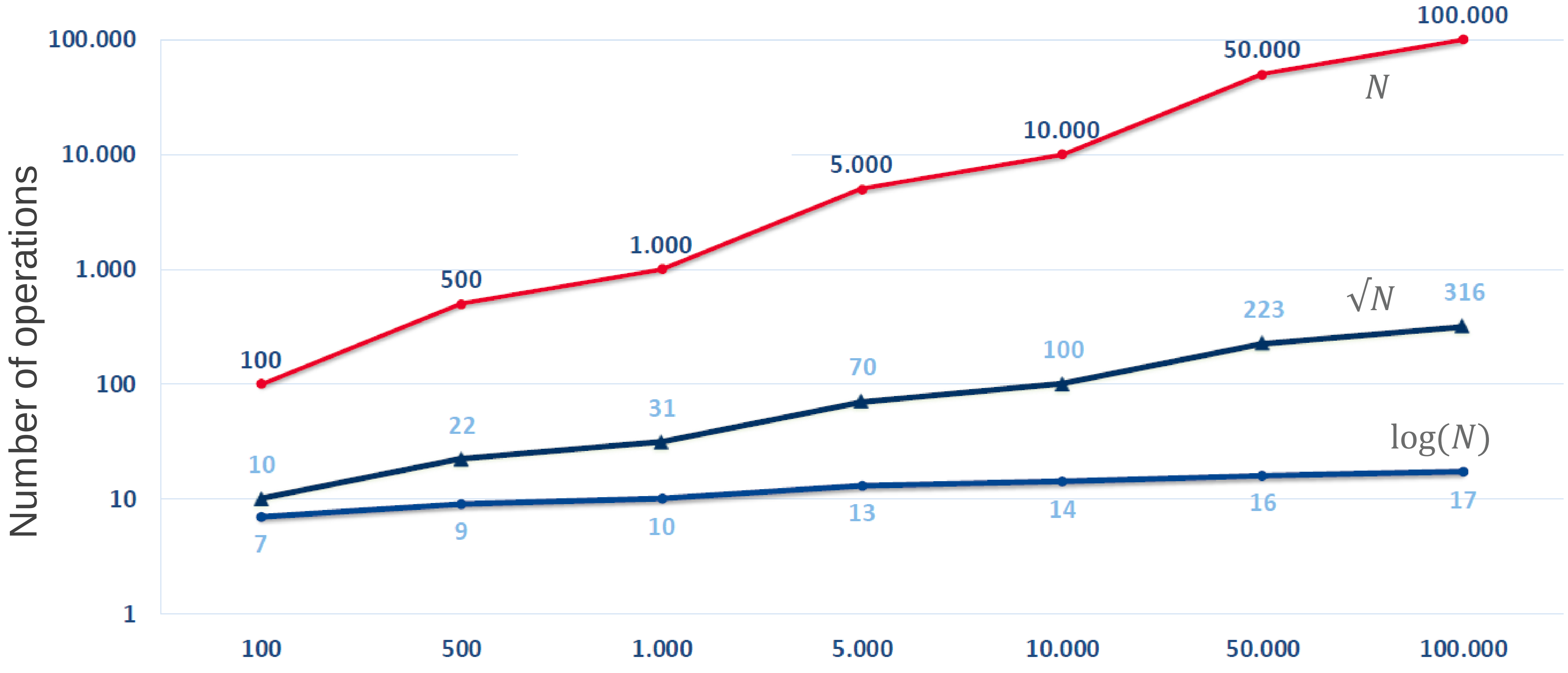
Quantum algorithms for NISQ Devices

Before NISQ – Old School Quantum Algorithms

Shor Algorithm Exponential Speedup	Grover Search Quadratic Speedup
	
Factorization	Optimization

Quantum algorithms for NISQ Devices

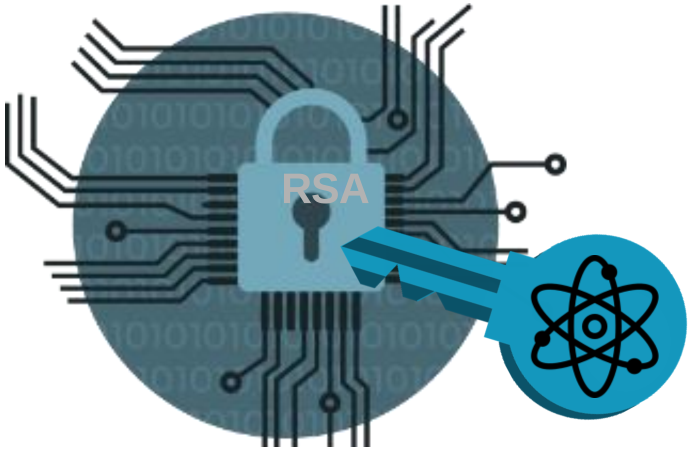
Before NISQ – Old School Quantum Algorithms



Quantum algorithms for NISQ Devices

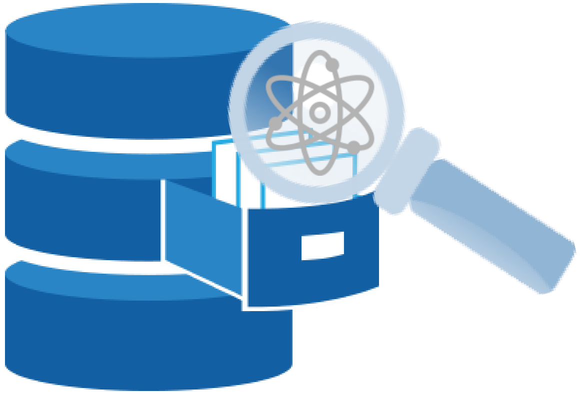
Before NISQ – Old School Quantum Algorithms

Shor Algorithm
Exponential Speedup



Factorization

Grover Search
Quadratic Speedup



Optimization

- Assume qubits are ideal and not subjected to noise

↓

- Require **error corrected quantum computers**

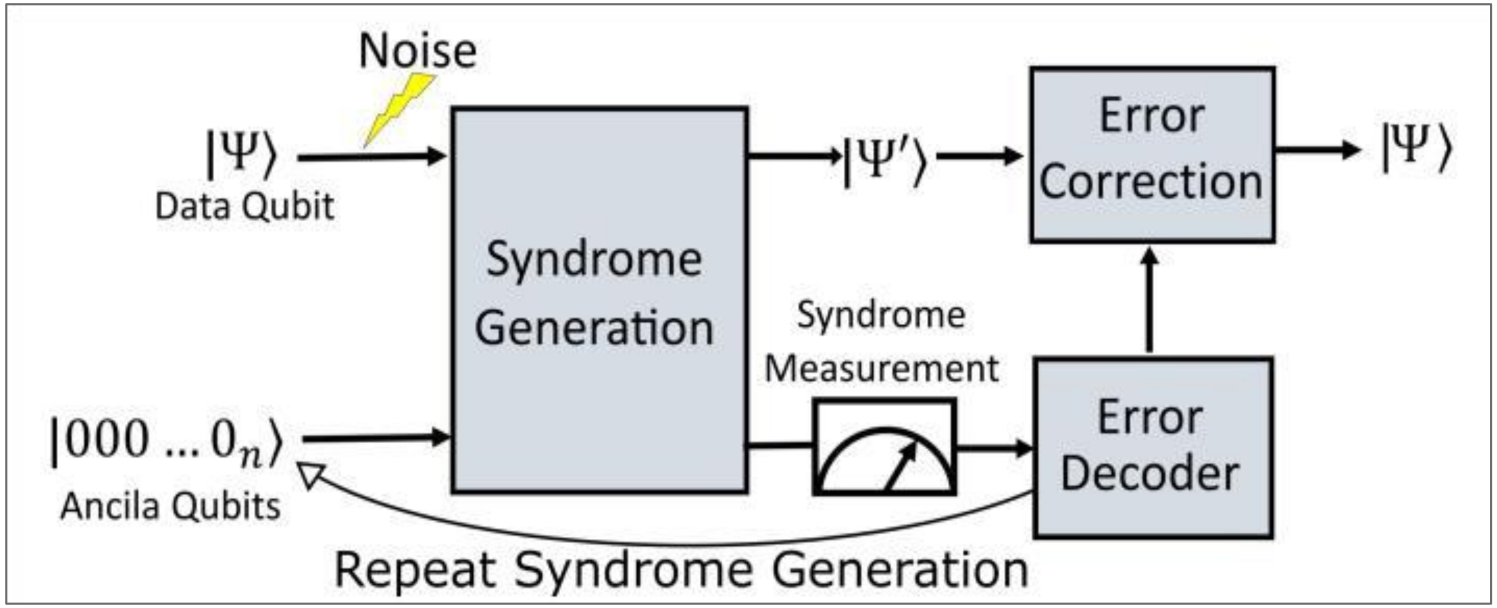
↓

- Require around **1 milion or 100 thousands qubits**

↓

- Available in **10 - 20 years**

Before NISQ – Old School Quantum Algorithms



- Assume qubits are ideal and not subjected to noise
- Require **error corrected quantum computers**
- Require around **1 milion or 100 thousands qubits**
- Available in **10 - 20 years**

We entered the NISQ era

NISQ = Noisy Intermediate-Scale Quantum

Intermediate-Scale

General Purpose QC

50 up to hundreds of qubits

IBM: 65 Qubits
Google: 53 Qubits

Quantum Annealers

Thousands of qubits

D-Wave Advantage: 5000Q

We entered the NISQ era

NISQ = Noisy Intermediate-Scale Quantum

Noisy - noise due to interaction with environment

General Purpose QC

- **No Quantum Error Correction**: overhead in number of qubit
- **Error rate per single gate affects the depth of the circuit**: error rate of 0.1% means that we can run circuits with at most 1000 elementary gates **(shallow circuits)**

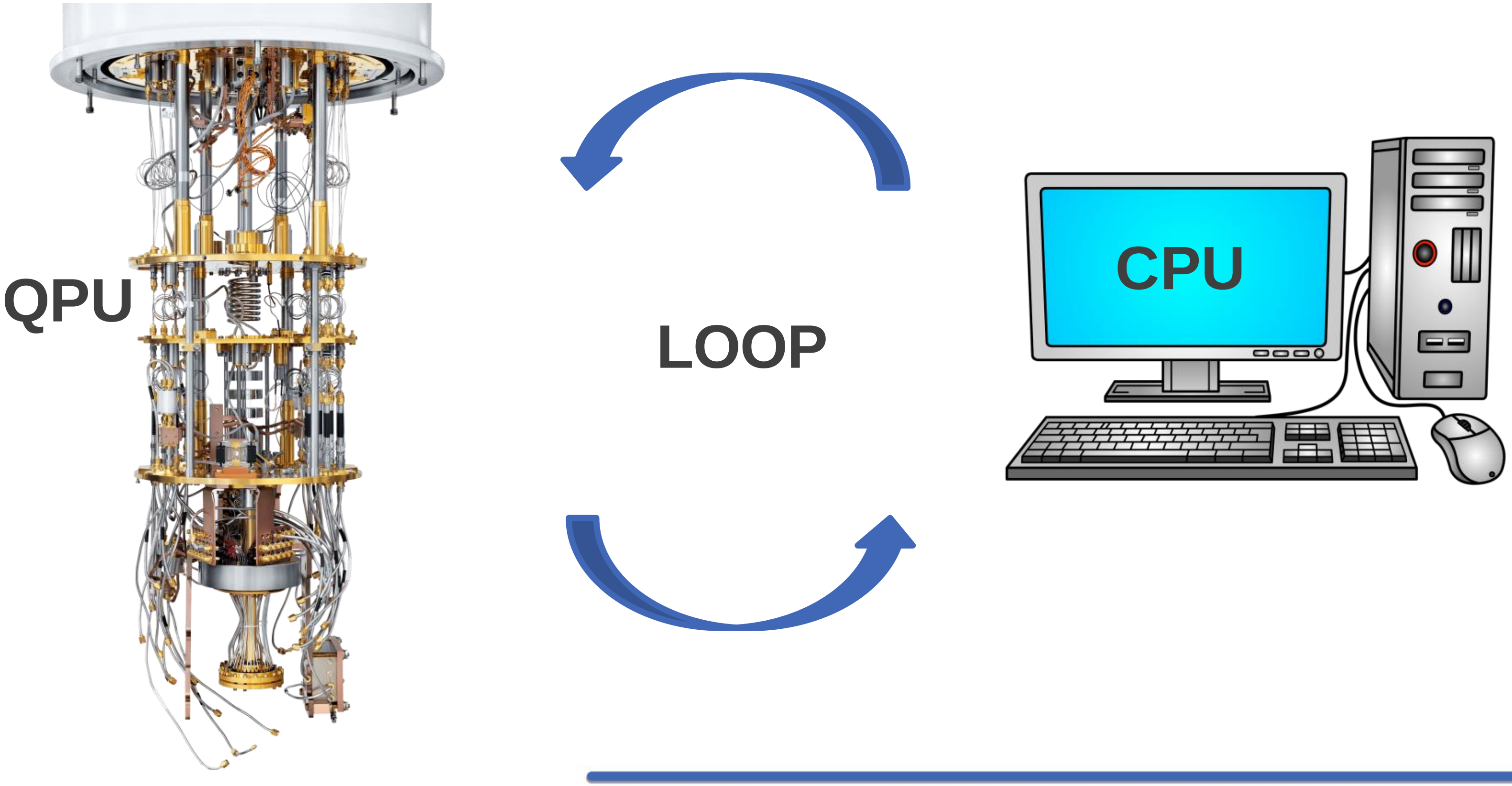
Quantum Annealers

- **No need for Quantum Error Correction**
- **Still unclear:** noise due to qubit quality could affect scalability (i.e. performance related to large problems)

Quantum algorithms for NISQ Devices

NISQ-ready algorithms for general purpose QPU

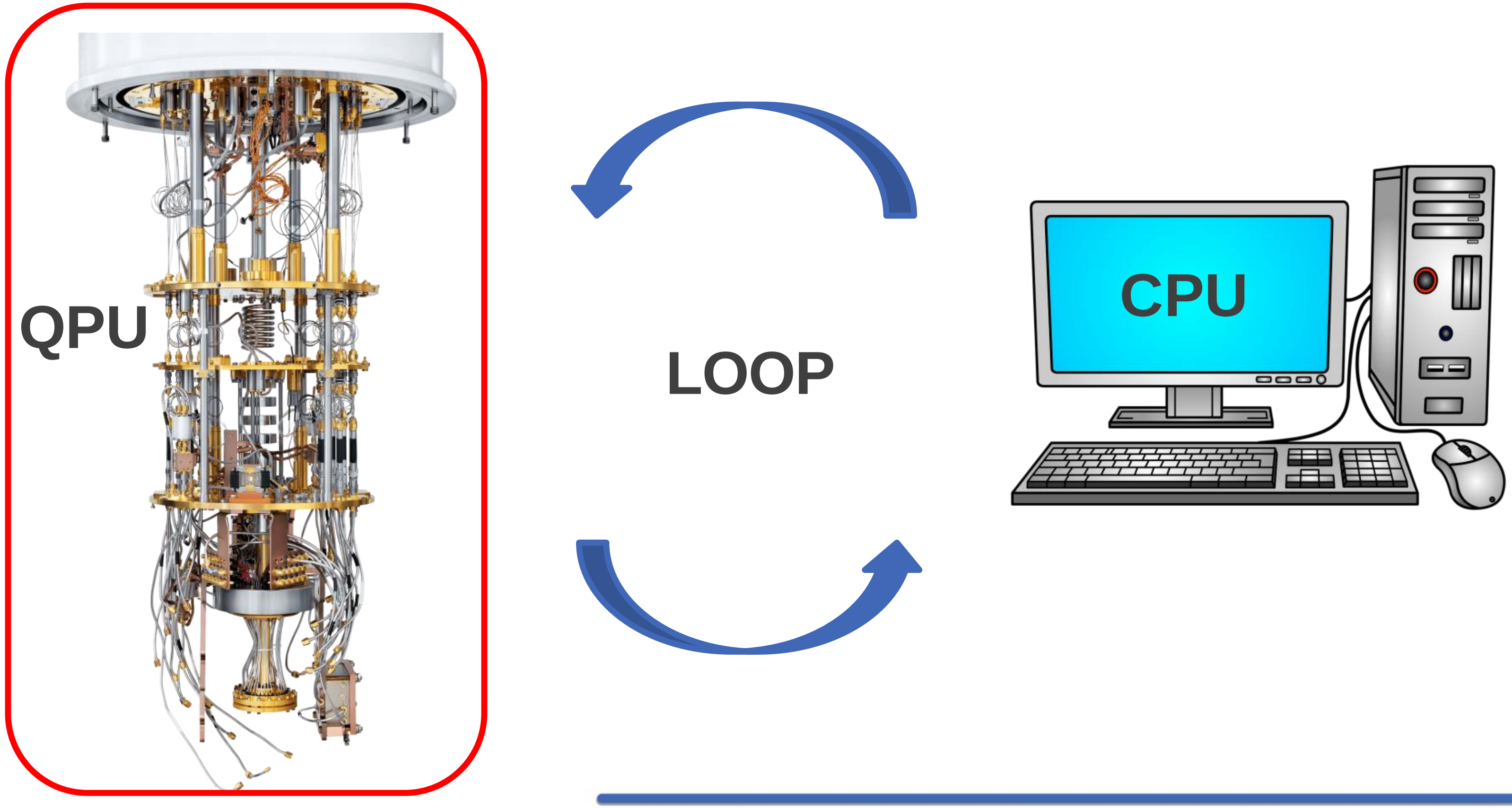
Hybrid Quantum-Classical algorithms



Quantum algorithms for NISQ Devices

NISQ-ready algorithms for general purpose QPU

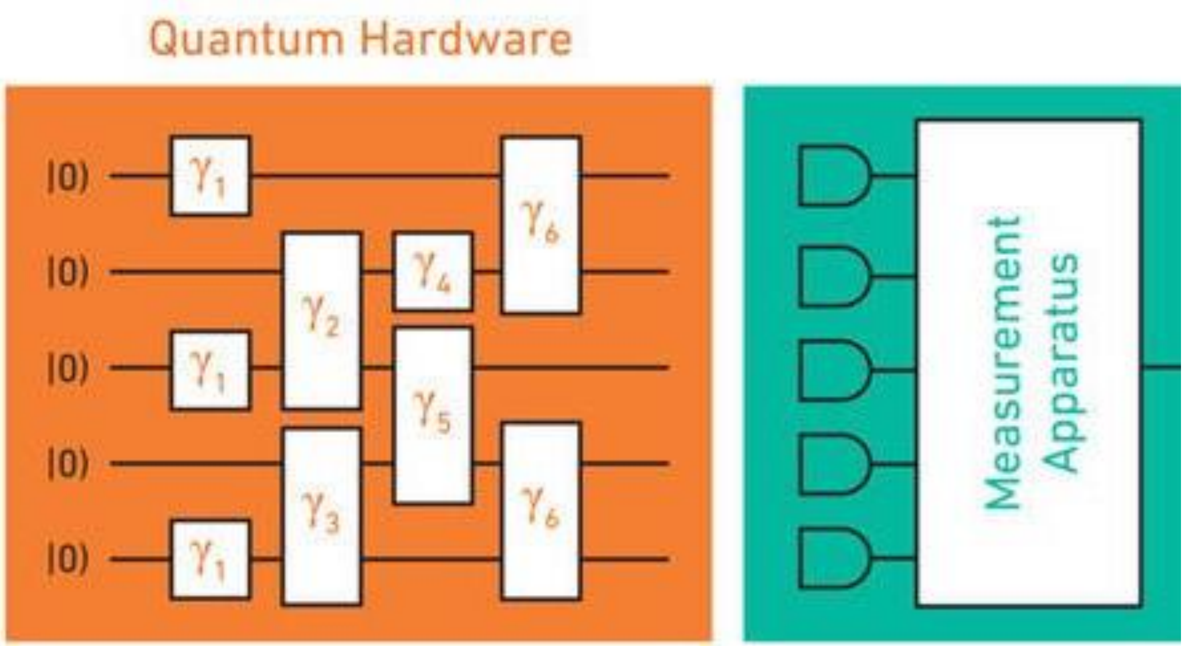
Hybrid Quantum-Classical algorithms



Quantum algorithms for NISQ Devices

NISQ-ready algorithms for general purpose QPU

Parametric Quantum Circuits



- Circuits that **use gates**, or in general, that apply **parameter-dependent operations** to qubits (e.g. Arbitrary rotations of angle γ)
- Circuits in which the **error is not corrected**
- **Shallow circuits**, i.e. of **limited depth** (1000 gates maximum, due to limited coherence times)

NISQ-ready algorithms for general purpose QPU

Working principle

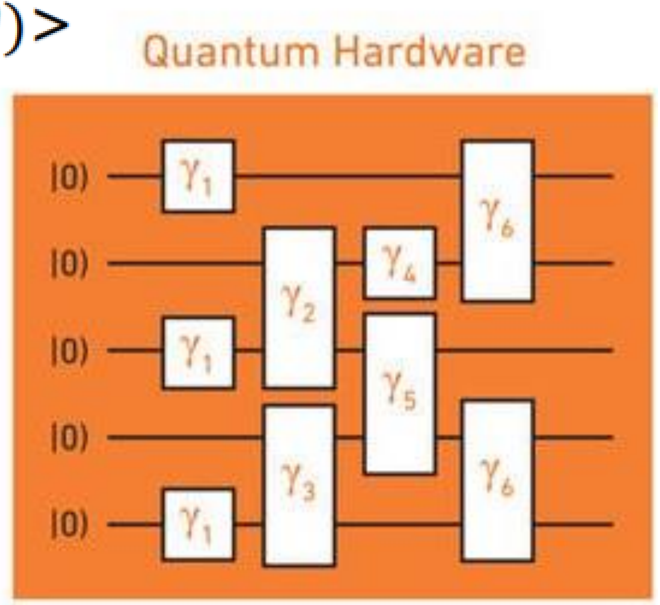
NISQ-ready algorithms for general purpose QPU

Working principle

1. Choose the parametric circuit you want to use (Variational Ansatz)

2. Implement Variational Ansatz on the QPU

$|\Psi(\vec{\theta})>$



Quantum algorithms for NISQ Devices

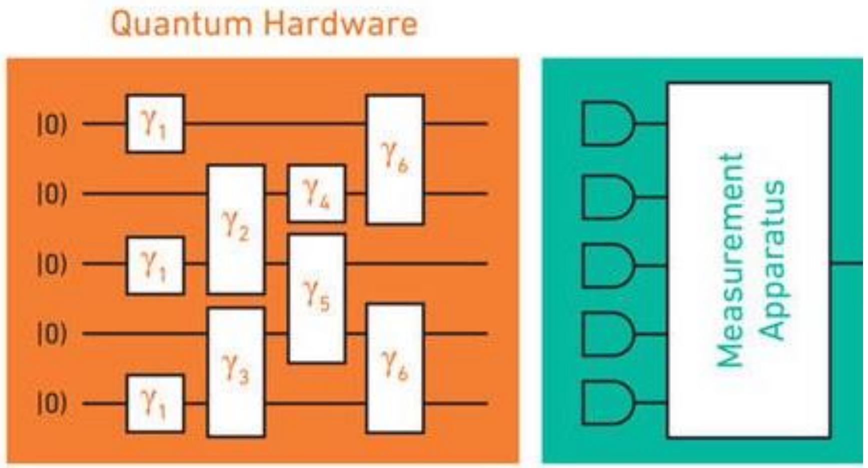
NISQ-ready algorithms for general purpose QPU

Working principle

1. Choose the parametric circuit you want to use (Variational Ansatz)

2. Implement Variational Ansatz on the QPU

$|\Psi(\vec{\theta})>$



3. Measure the qubits and calculate the cost function

$$E_{\vec{\theta}} = < \Psi(\vec{\theta})|\mathbf{H}|\Psi(\vec{\theta})>$$

NISQ-ready algorithms for general purpose QPU

Working principle

1. Choose the parametric circuit you want to use (Variational Ansatz)

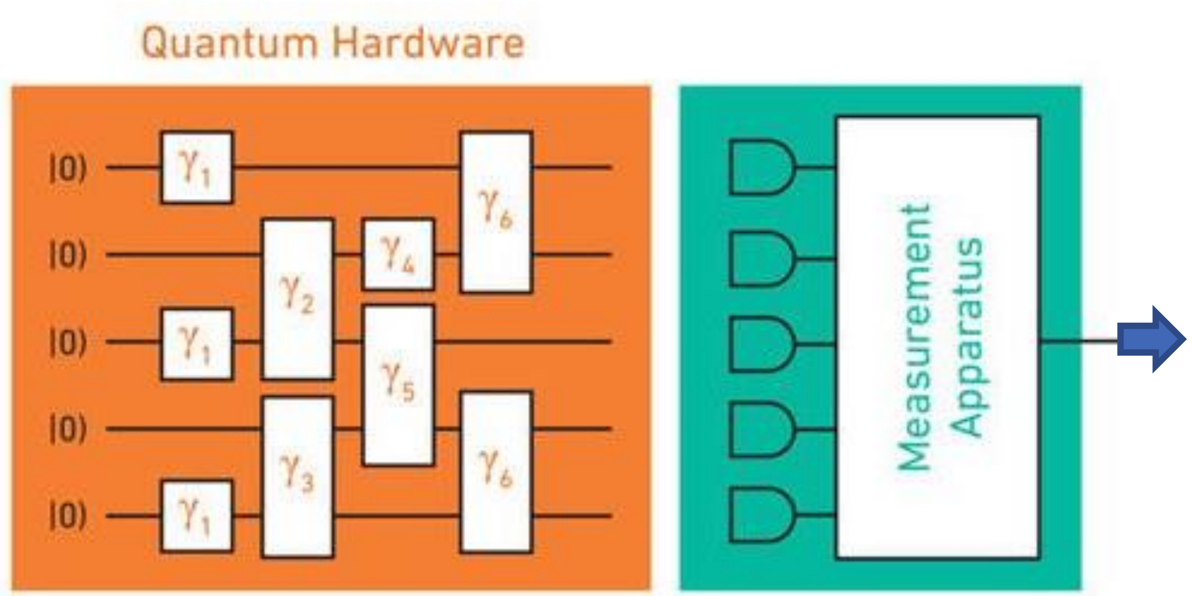
2. Implement Variational Ansatz on the QPU

3. Measure the qubits and calculate the cost function

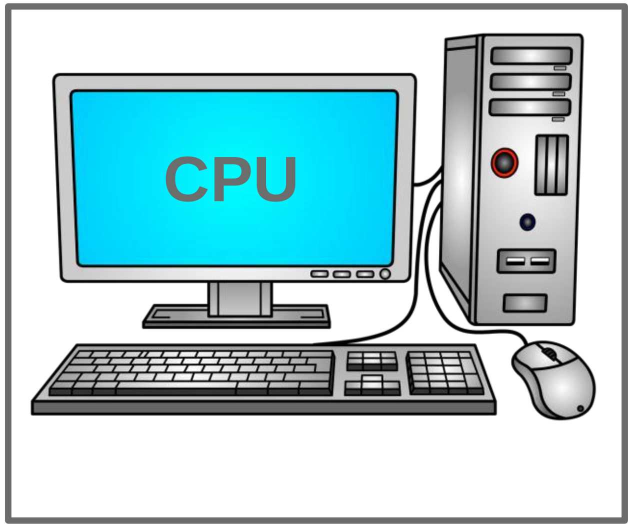
4. Use a classic computer to optimize the circuit parameters

The **optimization** of the set of parameters could be **gradient-based** or **gradient-free** (BFGS, COBYLA, L-B, SPSA, Bayesian Opt.) Depending on the type of cost function being evaluated

NISQ-ready algorithms for general purpose QPU



4. Classical optimization of parameters gamma



NISQ-ready algorithms for general purpose QPU

Working principle

1. **Choose the parametric circuit you want to use (Variational Ansatz)**

2. **Implement Variational Ansatz on the QPU**

3. **Measure the qubits and calculate the cost function**

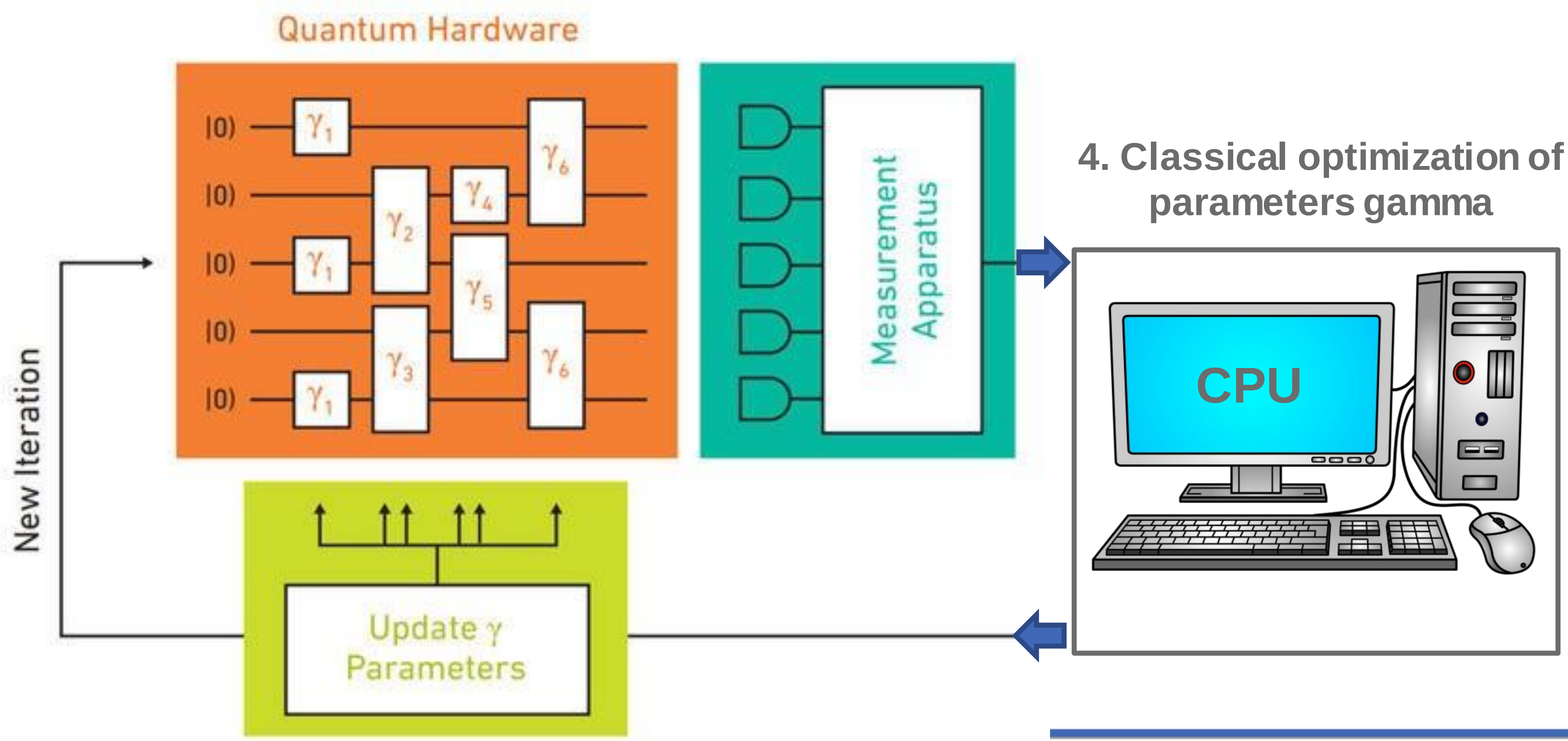
4. **Use a classic computer to optimize the circuit parameters**

This **cycle is repeated until convergence**. The final state gives us an **approximation of the solution**

Heuristic Algorithm

Quantum algorithms for NISQ Devices

NISQ-ready algorithms for general purpose QPU



NISQ-ready algorithms for general purpose QPU

The **scientific community** believes that **NISQ technology** could **outperform traditional classical computers** for **specific applications**

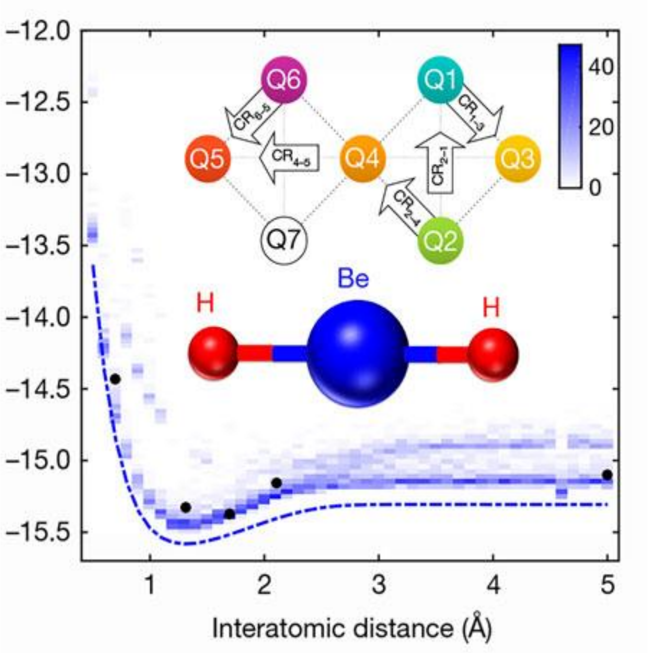
It means both getting the **same classic solutions faster**, and/or finding **new, better solutions**

- **Quantum Chemistry**
- **Quantum Optimization**
- **Quantum AI/Machine Learning**

Quantum algorithms for NISQ Devices

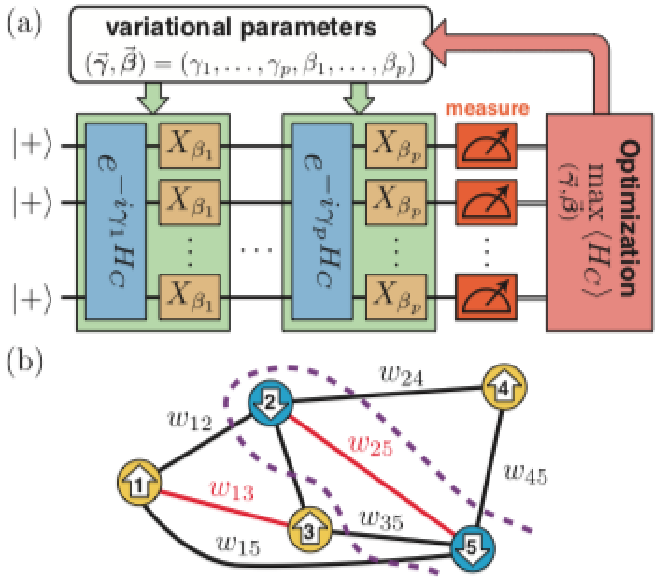
NISQ-ready algorithms for general purpose QPU

VQE



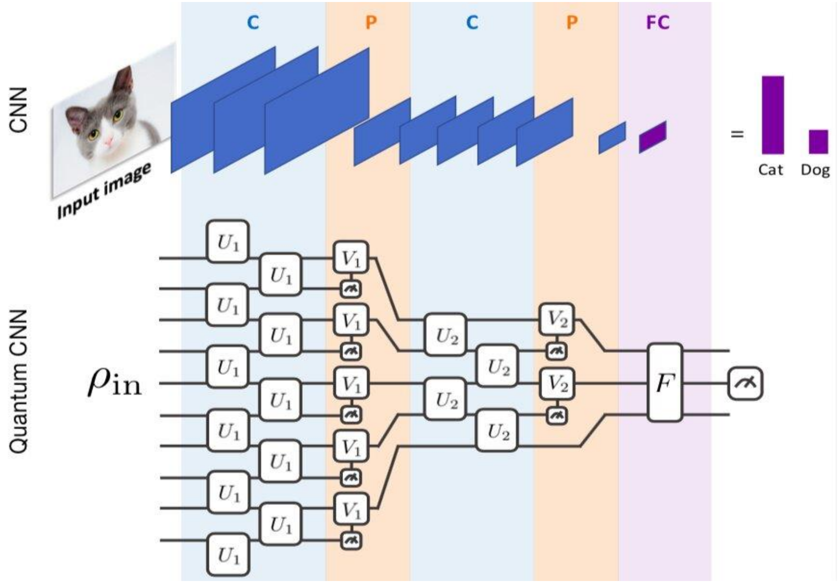
Quantum Chemistry

QAOA



Quantum Optimization

QNN



Quantum Machine Learning

Quantum algorithms for NISQ Devices

NISQ-ready algorithms for general purpose QPU

VQE

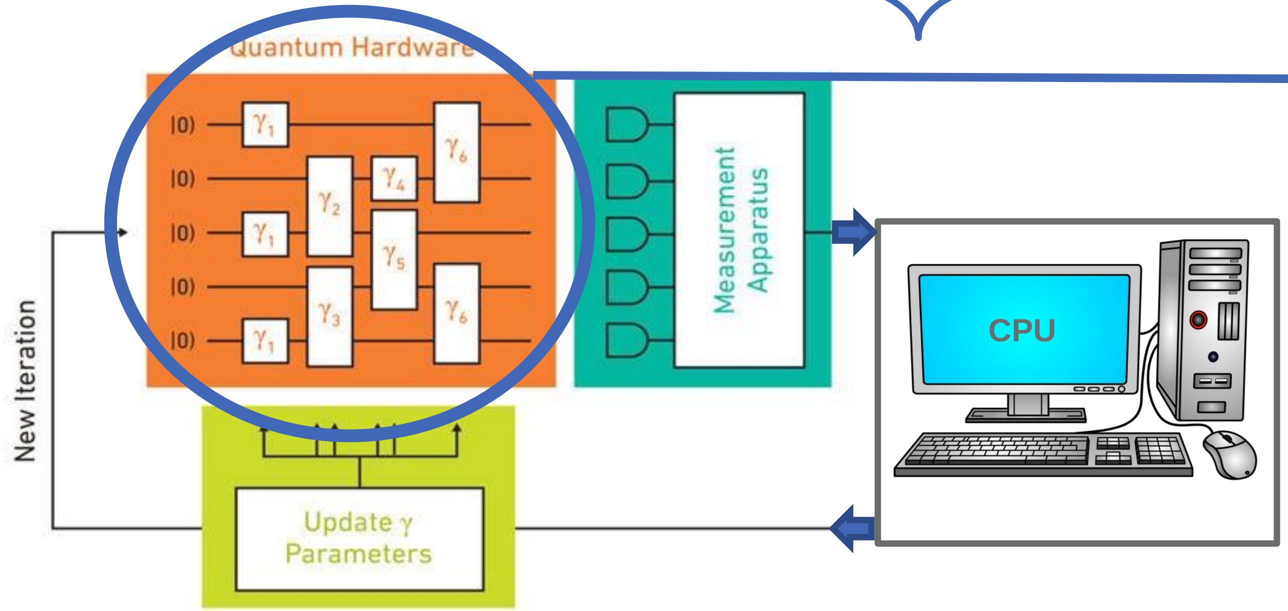
Quantum Chemistry

QAOA

Quantum Optimization

QNN

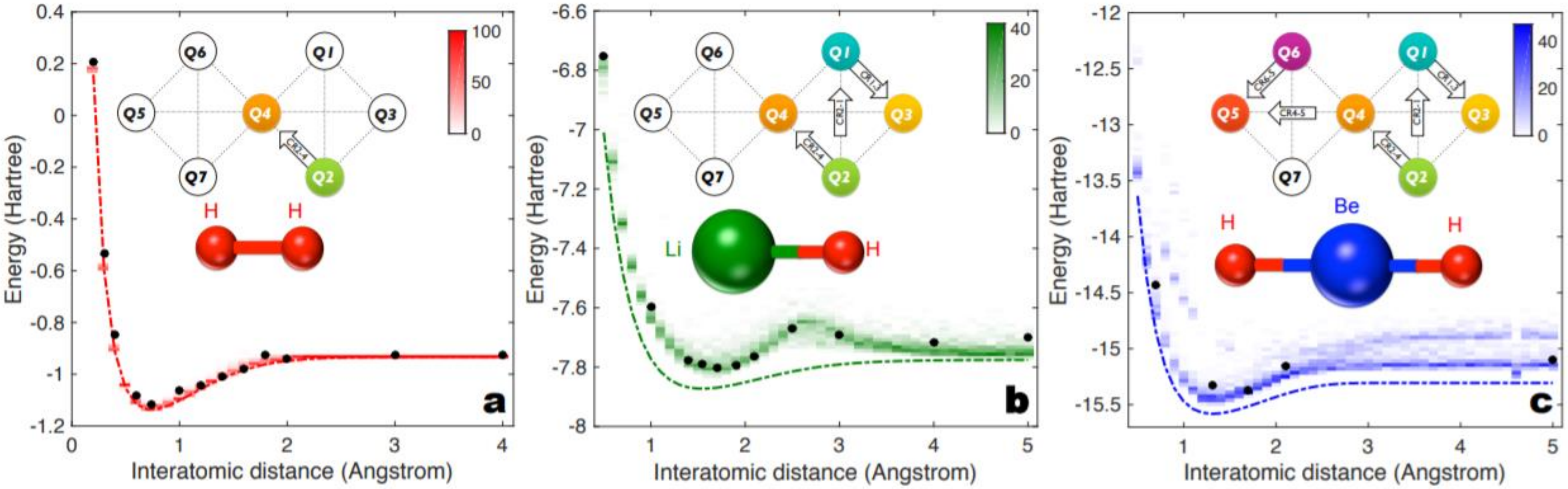
Quantum Machine Learning



The **main difference** between **VQE, QAOA** and **QNN** concerns the **choice of the parametric quantum circuit** (Variational Ansatz)

NISQ-ready algorithms for general purpose QPU

Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY



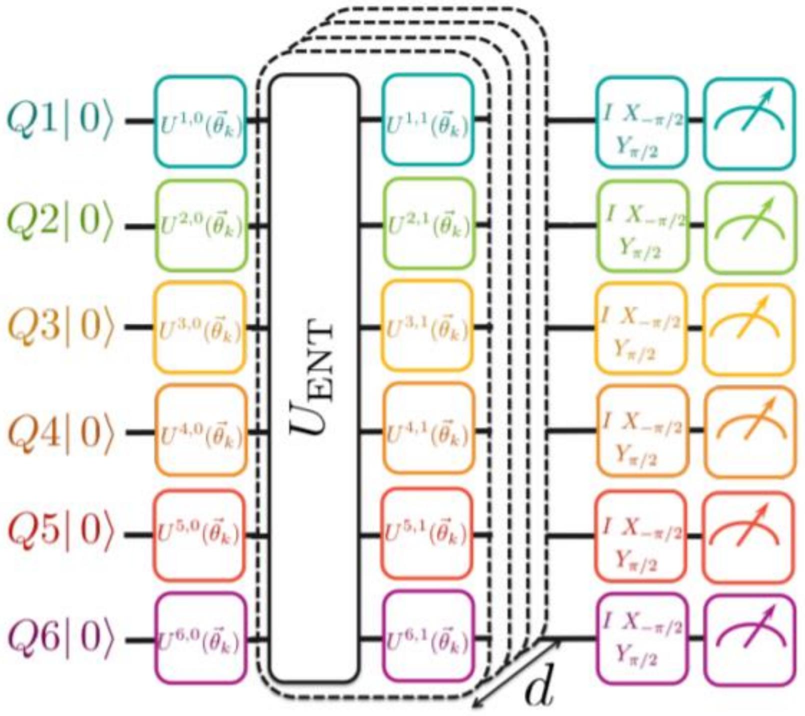
https://arxiv.org/abs/1704.05018

NISQ-ready algorithms for general purpose QPU

Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY

Objective: to **calculate the ground state of molecules** (we want to go **beyond** the approximation of the **mean field**, which is **classically very expensive** in terms of resources)

Method: the VQE uses **chemical-inspired Ansatz**, such as the Unitary Coupled Cluster (**UCC**) method **or** a **"hardware-efficient" ansatz**



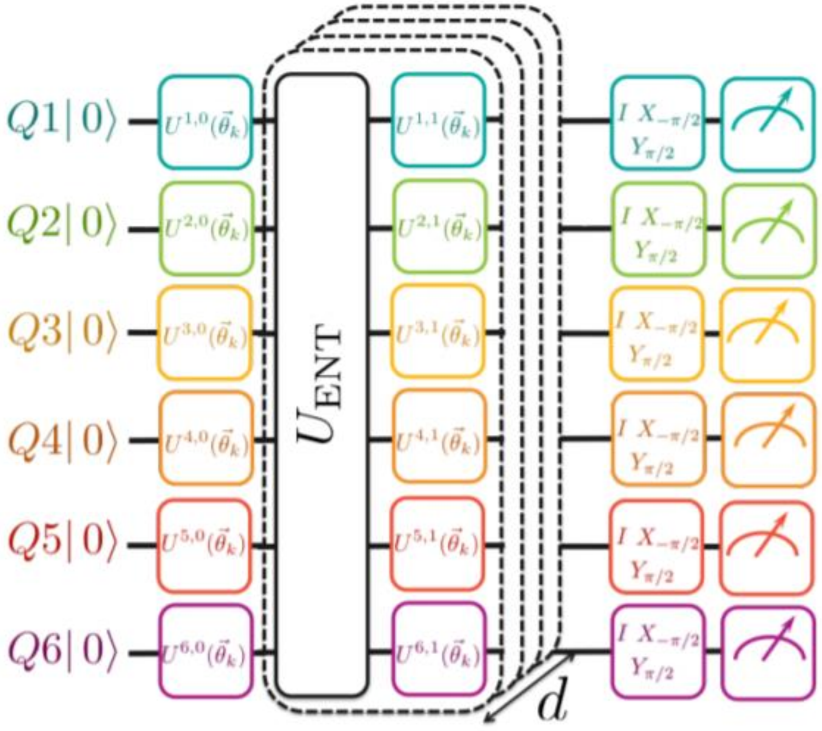
https://arxiv.org/abs/1704.05018

NISQ-ready algorithms for general purpose QPU

Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY

- **Ansatz** is a **provisional molecular ground state**
- The **classic optimizer evaluates the suitability of candidate solution** based on its **energy**.

This holds the promise to study large molecules with unprecedented accuracy



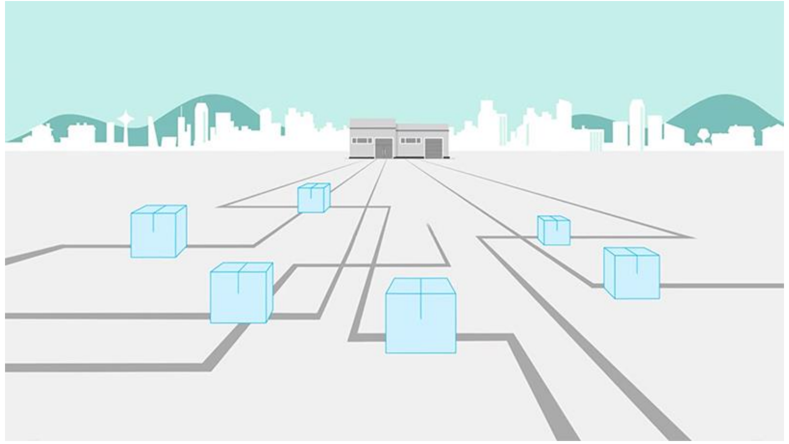
https://arxiv.org/abs/1704.05018

Quantum algorithms for NISQ Devices

NISQ-ready algorithms for general purpose QPU

Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

Optimization Problems



Routing

Scheduling

Portfolio Optimization

NISQ-ready algorithms for general purpose QPU

Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

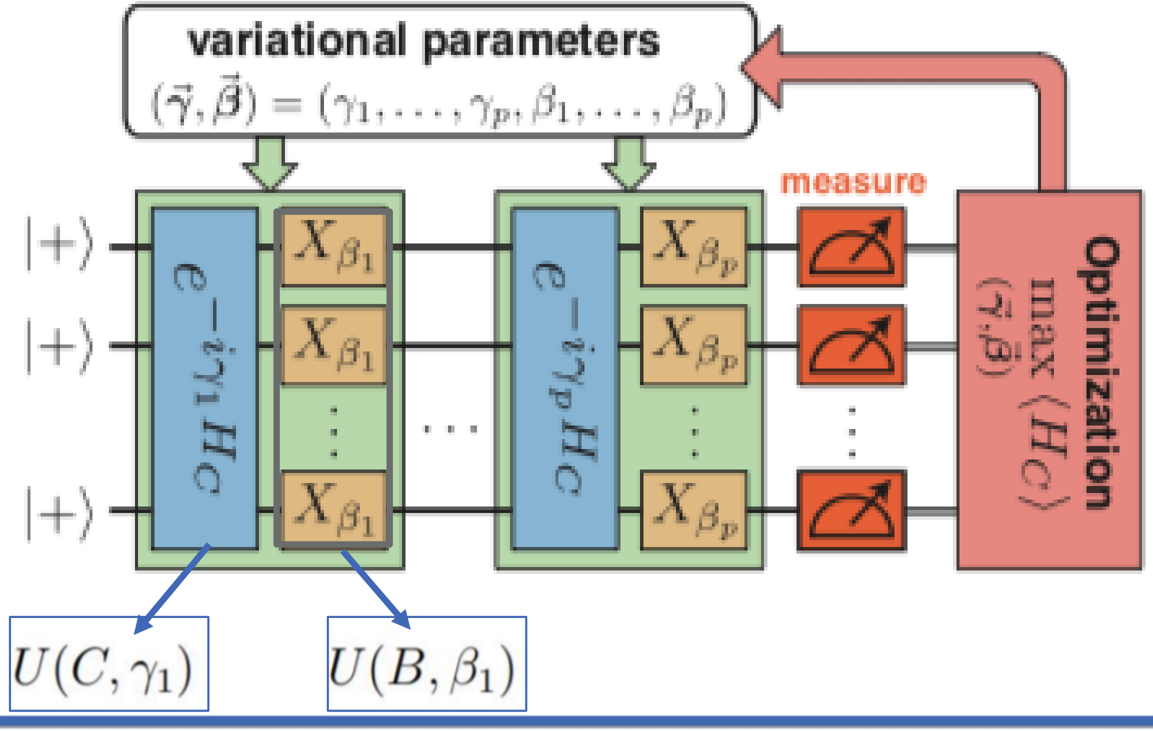
<u>Objective</u>: to **solve a combinatorial optimization problem**

<u>Method</u>: **Ansatz encodes two alternating circuits, U(C) and U(B)**, each **parameterized** by a number, **γ and β**.

Ideally, the circuit provides the **solution |γ,β>** to a **combinatorial problem implicit in** the definition of **U(C)**.

https://arxiv.org/abs/1411.4028

$$|\gamma, \beta\rangle = U(B,\beta_p)\,U(C,\gamma_p)\cdots U(B,\beta_1)\,U(C,\gamma_1)\;|s\rangle$$



NISQ-ready algorithms for general purpose QPU

Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

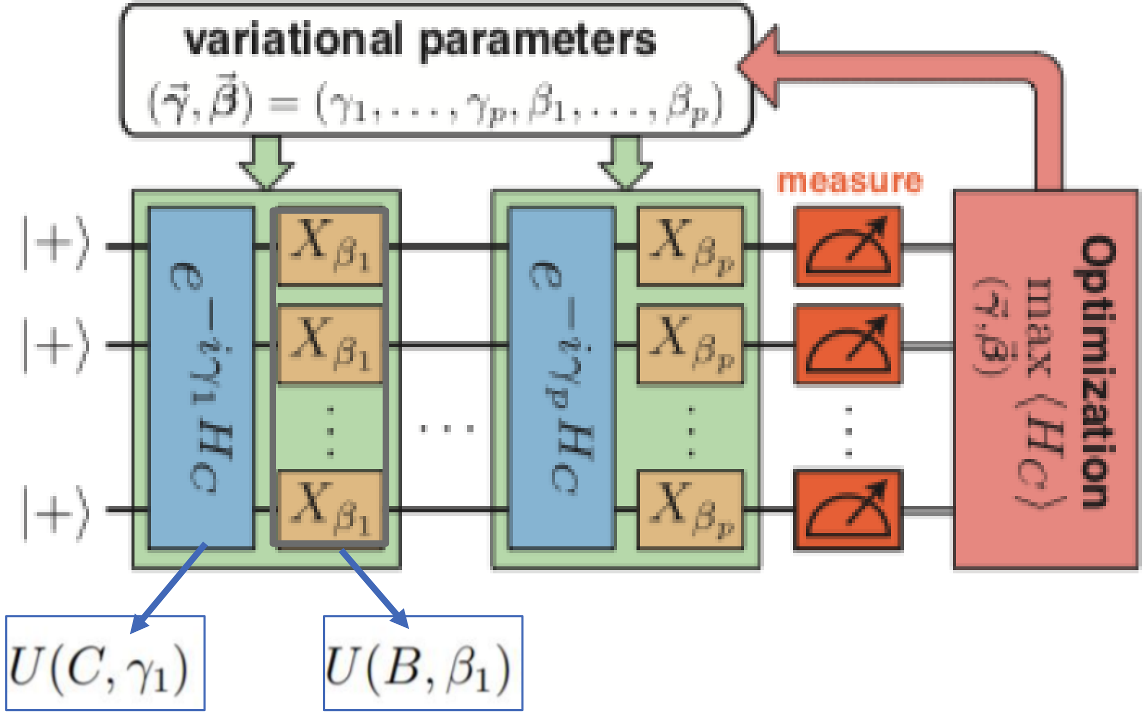
It is a heuristic optimization algorithm

$$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle = U(B, \beta_p)\, U(C, \gamma_p) \cdots U(B, \beta_1)\, U(C, \gamma_1)\ |s\rangle$$

solution — Circuit (alternating circuits) — initial state

$$U(C, \gamma) = e^{-i\gamma C} = \prod_{\alpha=1}^{m} e^{-i\gamma C_\alpha}$$

$$U(B, \beta) = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta \sigma_j^x}$$



https://arxiv.org/abs/1411.4028

NISQ-ready algorithms for general purpose QPU

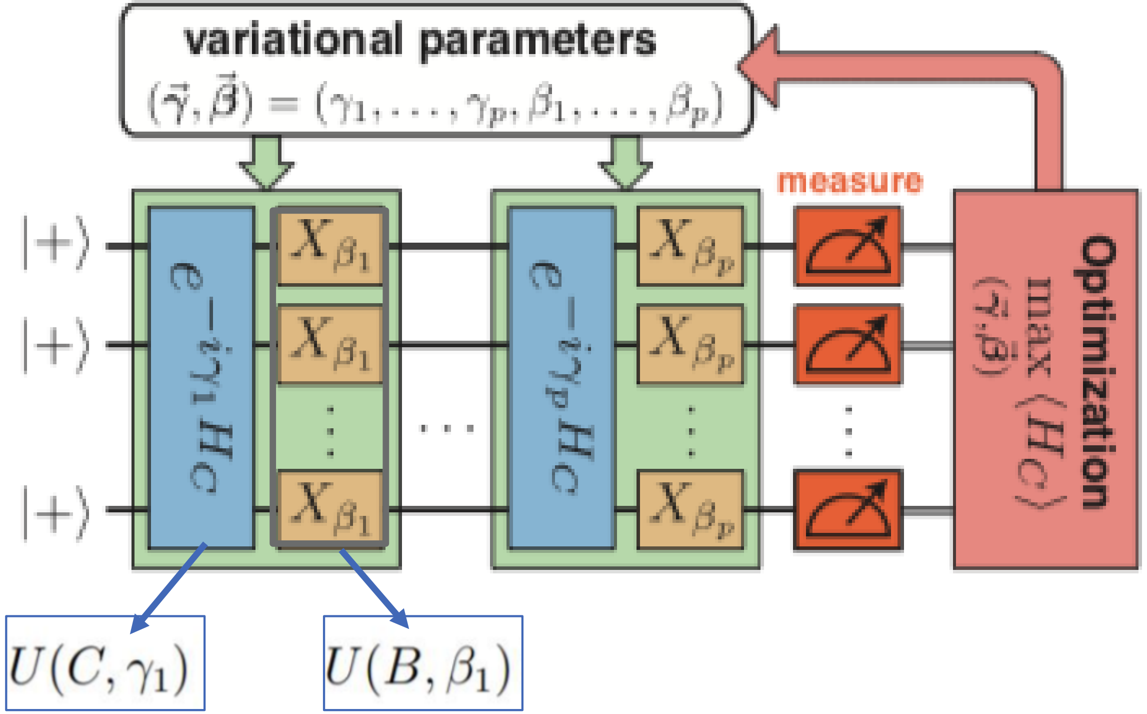
Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

$$U(C,\gamma) = e^{-i\gamma C} = \prod_{\alpha=1}^{m} e^{-i\gamma C_\alpha}$$

Encodes the optimization problem to solve (e.g. C could be some Qubo problem)

$$U(B,\beta) = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta \sigma_j^x}$$

Allow the **exploration of the solution space**

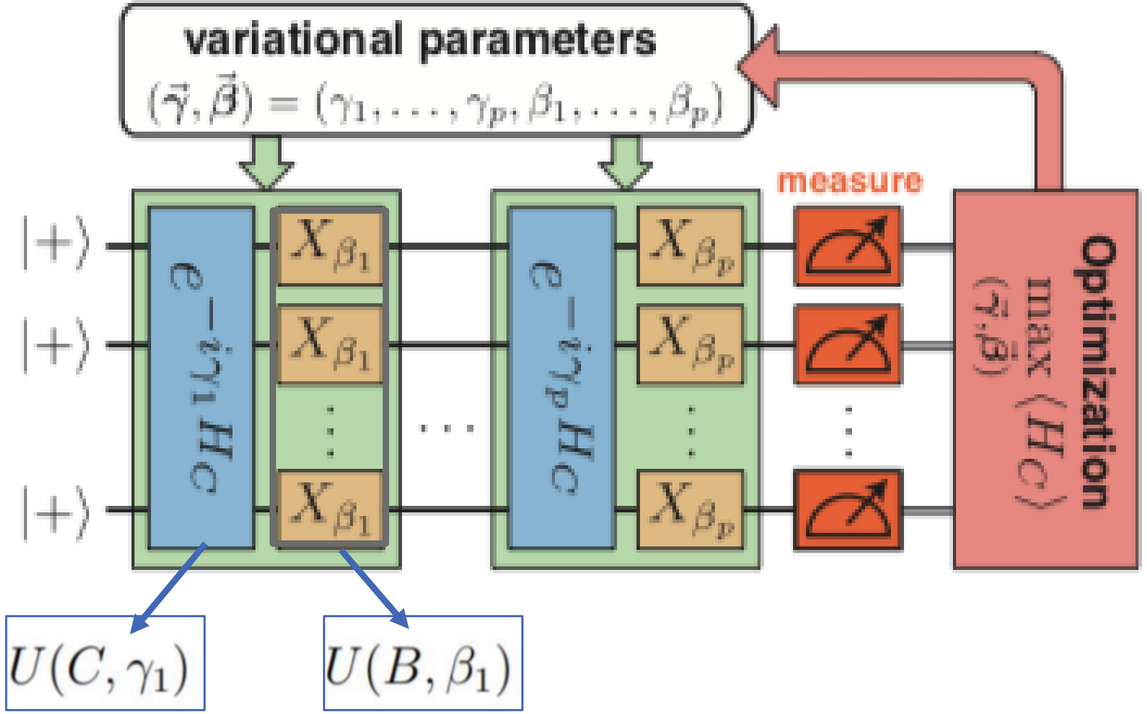


https://arxiv.org/abs/1411.4028

NISQ-ready algorithms for general purpose QPU

Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

<u>Challenge</u>: find a class of problems for which QAOA is strictly better than the best classical algorithms.



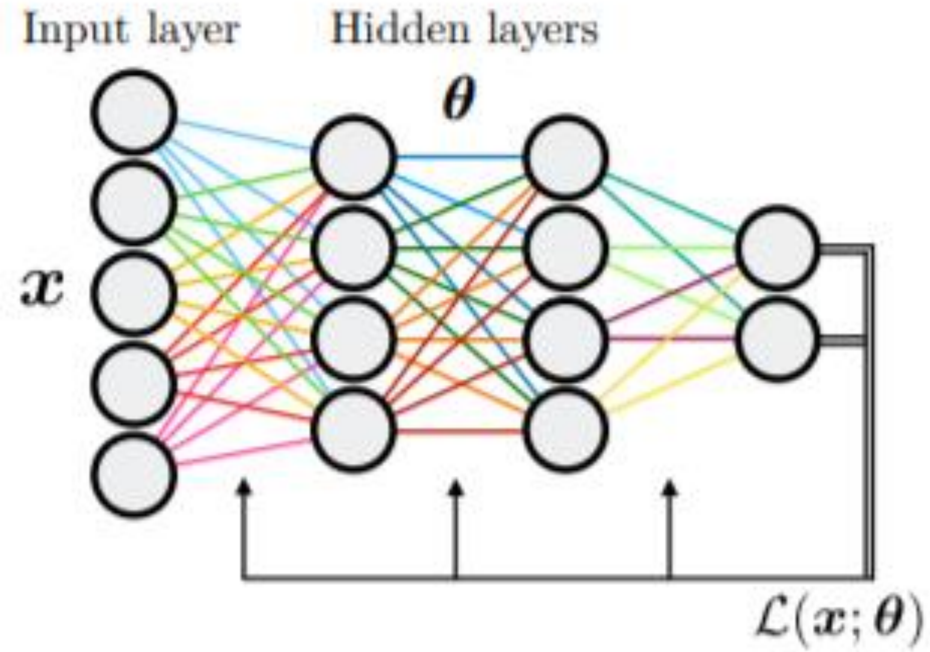
https://arxiv.org/abs/1411.4028

NISQ-ready algorithms for general purpose QPU

Quantum Neural Networks (QNN) – QUANTUM MACHINE LEARNING

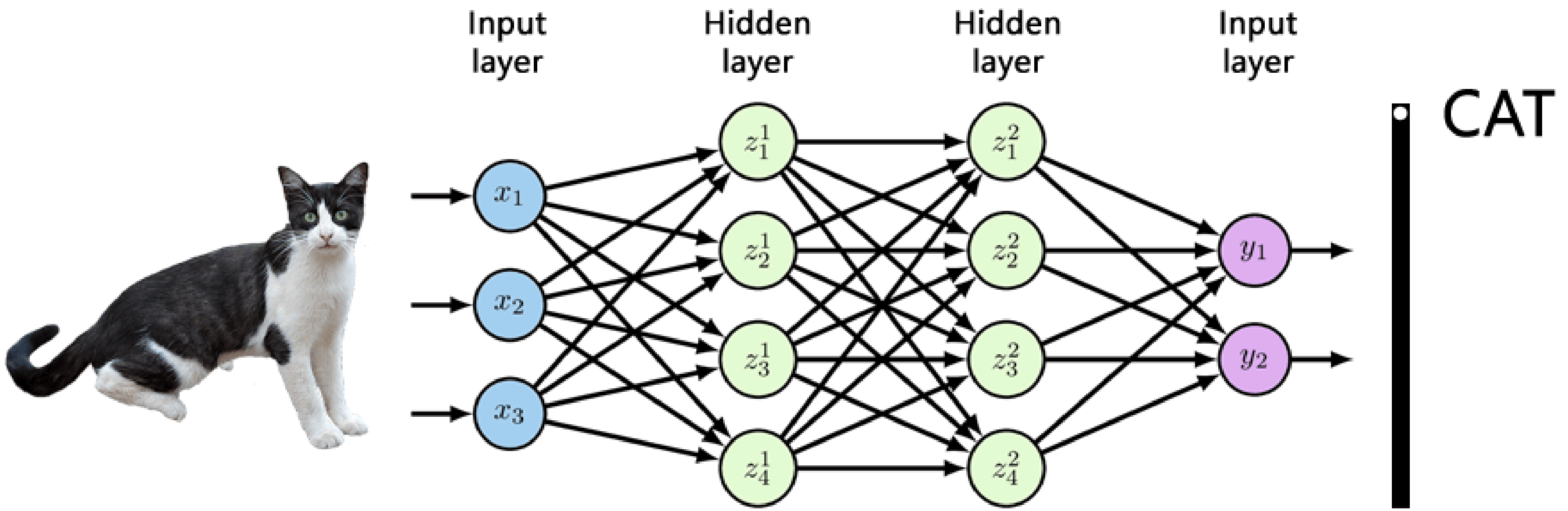
Supervised learning: the algorithm is asked to **reproduce the relations** between some inputs and outputs.

If properly trained, the **NN is able to classify new data**, i.e. data that was not used during training



NISQ-ready algorithms for general purpose QPU

Quantum Neural Networks (QNN) – QUANTUM MACHINE LEARNING

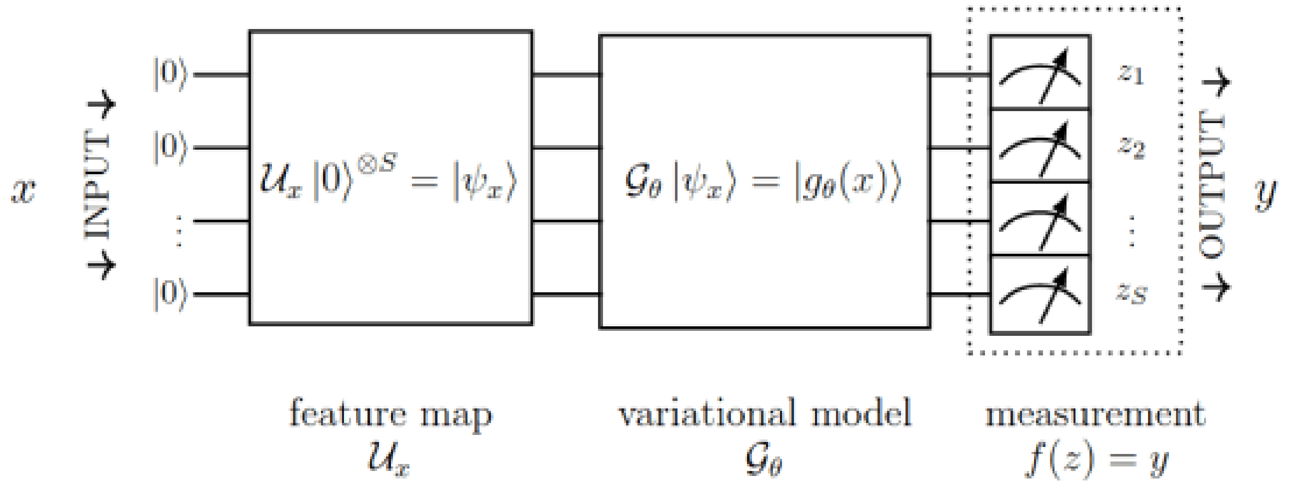


NISQ-ready algorithms for general purpose QPU

Quantum Neural Networks (QNN) – QUANTUM MACHINE LEARNING

Goal: Address a supervised machine learning problem

Method: **Ansatz** consists of a **feature map** that serves to represent classical data and a **variational part** for learning



The circuit learns to classify new inputs based on the examples seen in the training phase

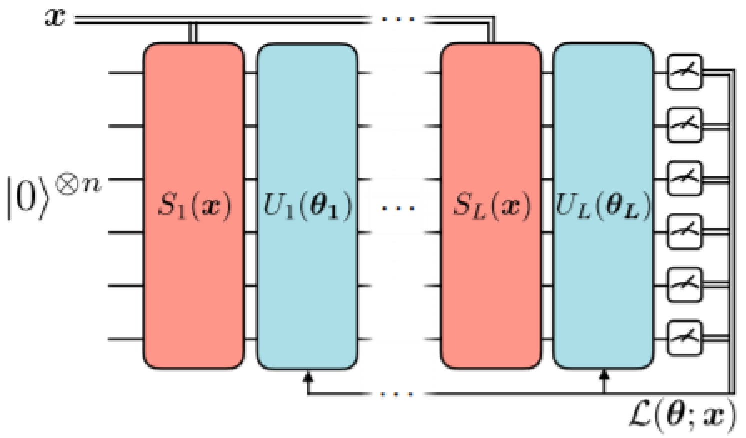
https://arxiv.org/abs/2011.00027

Quantum algorithms for NISQ Devices

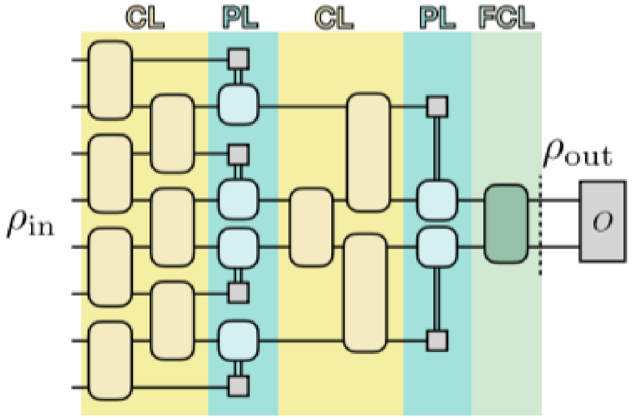
NISQ-ready algorithms for general purpose QPU

Quantum Neural Networks (QNN) – QUANTUM MACHINE LEARNING

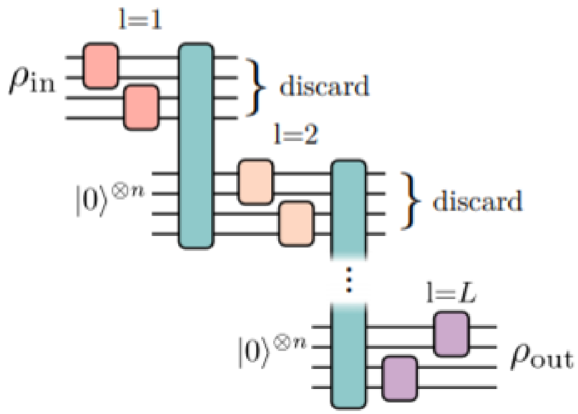
We can also have several Ansatz for QNNs



Standard QNN



Convolutional QNN



Dissipative QNN

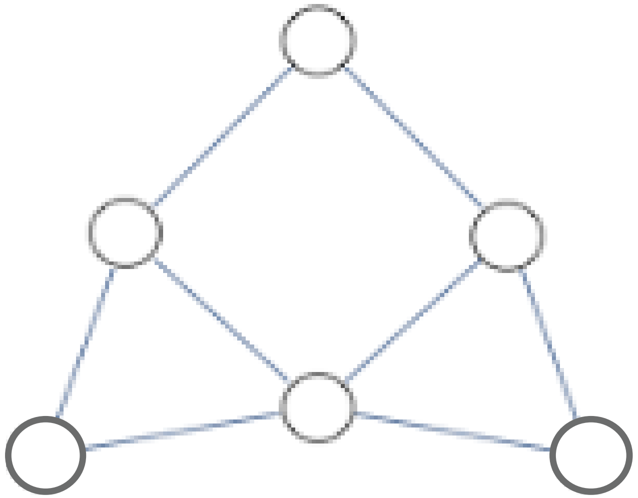
https://arxiv.org/abs/2102.03879

Application: QAOA & MIS problem

Application: QAOA & MIS problem

Maximal Independent Set (MIS) Problem

<u>Definition</u>: Given a graph, **color** the **largest number of nodes avoiding** that **nodes of the same color** are **connected** together



It is a hard **combinatorial optimization problem** (complexity class **NP-hard**)

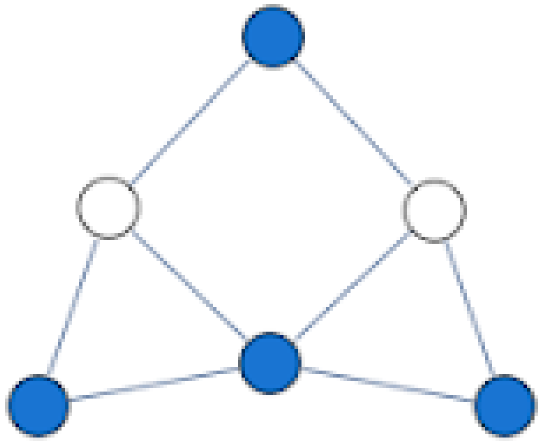
Applications:

- Modeling and Optimization in Massive Datasets
- Modeling Wireless Networks
- Matching Molecular Structures

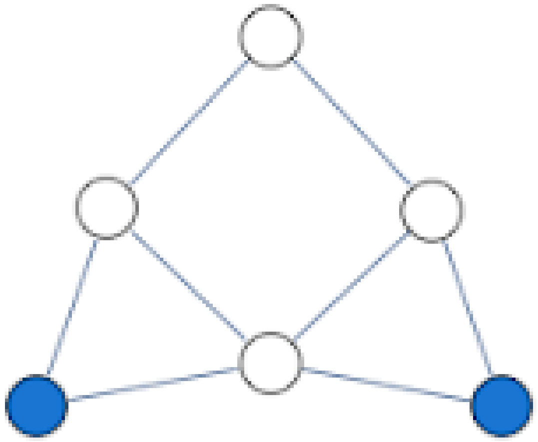
Application: QAOA & MIS problem

Maximal Independent Set (MIS) Problem

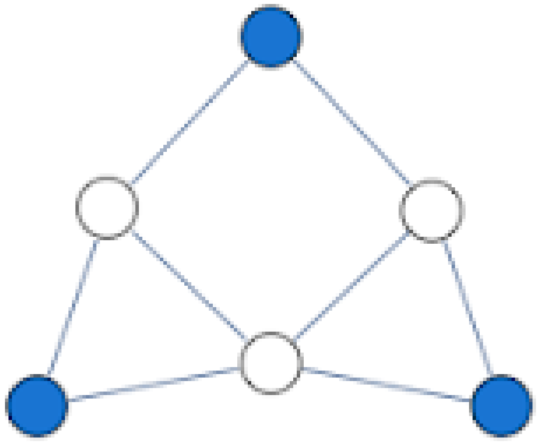
<u>Definition</u>: Given a graph, **color** the **largest number of nodes avoiding** that **nodes of the same color** are **connected** together



Not an independent set



Independent set but on maximal

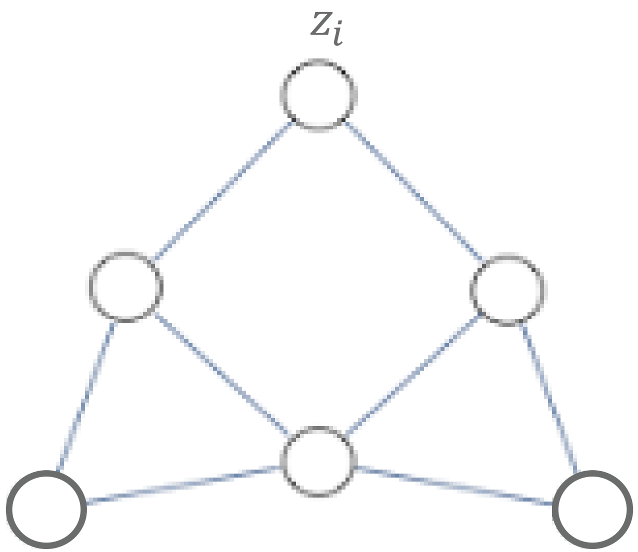


Maximal Independent Set

Application: QAOA & MIS problem

Maximal Independent Set (MIS) Problem

<u>Definition</u>: Given a graph, **color** the **largest number of nodes avoiding** that **nodes of the same color** are **connected** together



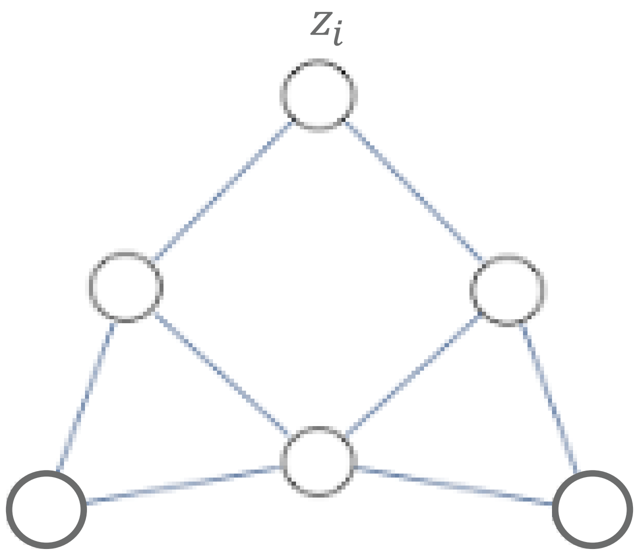
Combinatorial formulation

We can attribute a binary variable z_i to each node, where z_i = 1 if node i is colored (therefore it belongs to the independent set) and z_i = 0 otherwise.

Application: QAOA & MIS problem

Maximal Independent Set (MIS) Problem

Definition: Given a graph, **color** the **largest number of nodes avoiding** that **nodes of the same color** are **connected** together



The **Maximum Independent Set** corresponds to the **minimum** of the following **cost function:**

$$C(z_1, \ldots, z_N) = -\sum_{i=1}^{N} z_i + U \sum_{\langle i,j \rangle} z_i z_j$$

$$U \gg 1$$

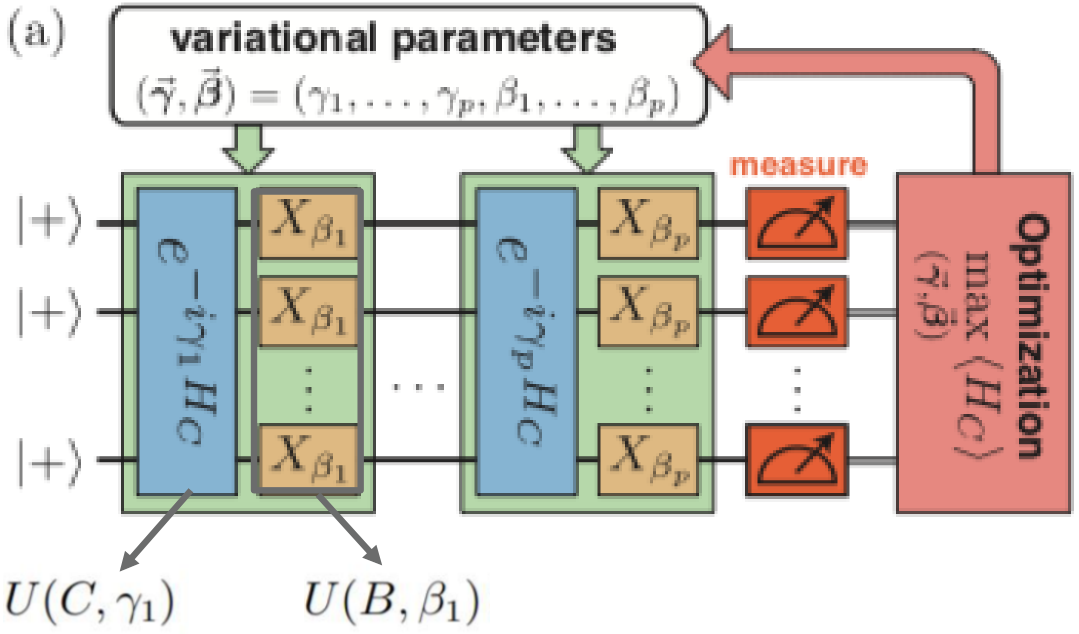
Application: QAOA & MIS problem

Maximal Independent Set (MIS) Problem

QAOA Ansatz

$$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle = U(B,\beta_p)\,U(C,\gamma_p)\cdots U(B,\beta_1)\,U(C,\gamma_1)\;|s\rangle$$

solution — Circuit (alternating circuits) — initial state



Maximal Independent Set (MIS) Problem

QAOA Ansatz

$$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle = U(B, \beta_p)\, U(C, \gamma_p) \cdots U(B, \beta_1)\, U(C, \gamma_1)\, |s\rangle$$

solution

Circuit (alternating circuits)

initial state

$$U(C, \gamma) = e^{-i\gamma C}$$

$$U(B, \beta) = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta \sigma_j^x}$$

Application: QAOA & MIS problem

Maximal Independent Set (MIS) Problem

QAOA Ansatz

$$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle = U(B, \beta_p)\, U(C, \gamma_p) \cdots U(B, \beta_1)\, U(C, \gamma_1)\, |s\rangle$$

solution — Circuit (alternating circuits) — initial state

$$U(C, \gamma) = e^{-i\gamma C}$$

$$U(B, \beta) = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta \sigma_j^x}$$

$$C(z_1, \ldots, z_N) = -\sum_{i=1}^{N} z_i + U \sum_{\langle i,j \rangle} z_i z_j$$

Application: QAOA & MIS problem

Maximal Independent Set (MIS) Problem

$$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle = U(B, \beta_p)\,U(C, \gamma_p) \cdots U(B, \beta_1)\,U(C, \gamma_1)\,|s\rangle$$

solution — Circuit (alternating circuits) — initial state

$$C(z_1, \ldots, z_N) = -\sum_{i=1}^{N} z_i + U \sum_{\langle i,j \rangle} z_i z_j$$

Pulser

$$H = \sum_{i=1}^{N} \frac{\hbar\Omega}{2}\sigma_i^x - \sum_{i=1}^{N} \frac{\hbar\delta}{2}\sigma_i^z + \sum_{j<i} \frac{C_6}{|\mathbf{r}_i - \mathbf{r}_j|^6} n_i n_j .$$

Application: QAOA & MIS problem

Maximal Independent Set (MIS) Problem

$$|\gamma, \beta\rangle = \underbrace{U(B,\beta_p)\,U(C,\gamma_p)\cdots U(B,\beta_1)\,U(C,\gamma_1)}_{\text{Circuit (alternating circuits)}}\underbrace{|s\rangle}_{\text{initial state}}$$

solution — Circuit (alternating circuits) — initial state

$$C(z_1,\ldots,z_N) = -\sum_{i=1}^{N} z_i + U\sum_{\langle i,j\rangle} z_i z_j$$

Practice Session

Pulser

$$H = \sum_{i=1}^{N}\frac{\hbar\Omega}{2}\sigma_i^x - \sum_{i=1}^{N}\frac{\hbar\delta}{2}\sigma_i^z + \sum_{j<i}\frac{C_6}{|\mathbf{r}_i - \mathbf{r}_j|^6} n_i n_j .$$

Useful Software

Pulser

Cirq

Qiskit

PENNY LANE

TensorFlow Quantum

aws

Amazon Braket

Quantum Computing @ CINECA

CINECA: Italian HPC center

CINECA Quantum Computing Lab:

- Research with Universities, Industries and QC startups
- Internship programs, Courses and Conference (HPCQC)

https://www.quantumcomputinglab.cineca.it

r.mengoni@cineca.it

QUANTUM COMPUTING LAB