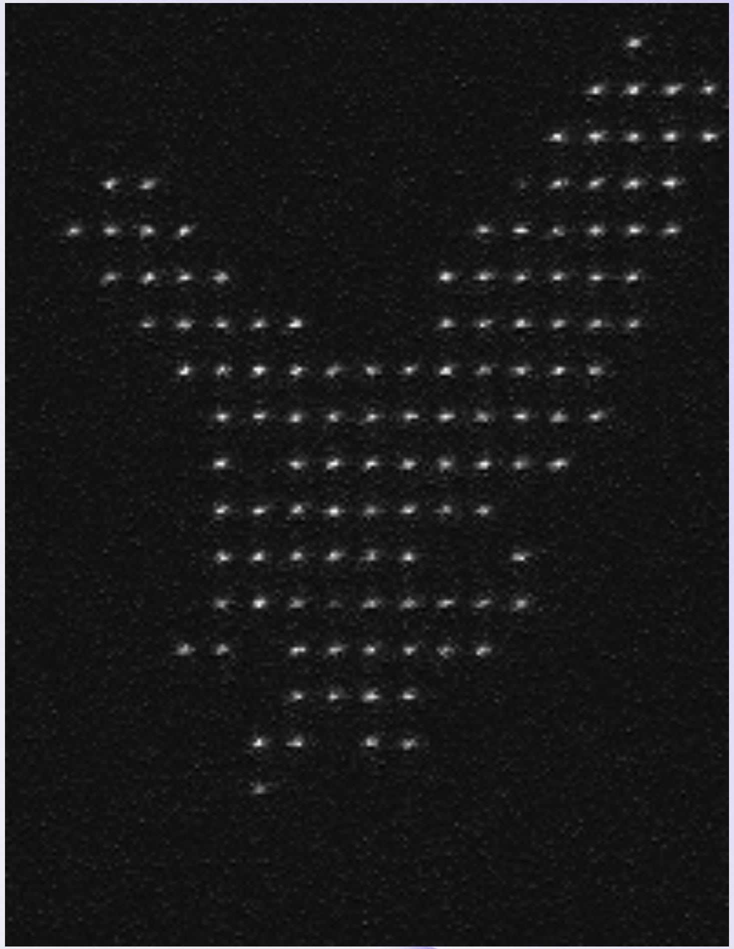
|QuEra>

Quantum computing with **neutral atoms** on the cloud



Alexei Bylinskii, Pedro Lopes, Jonathan Wurtz

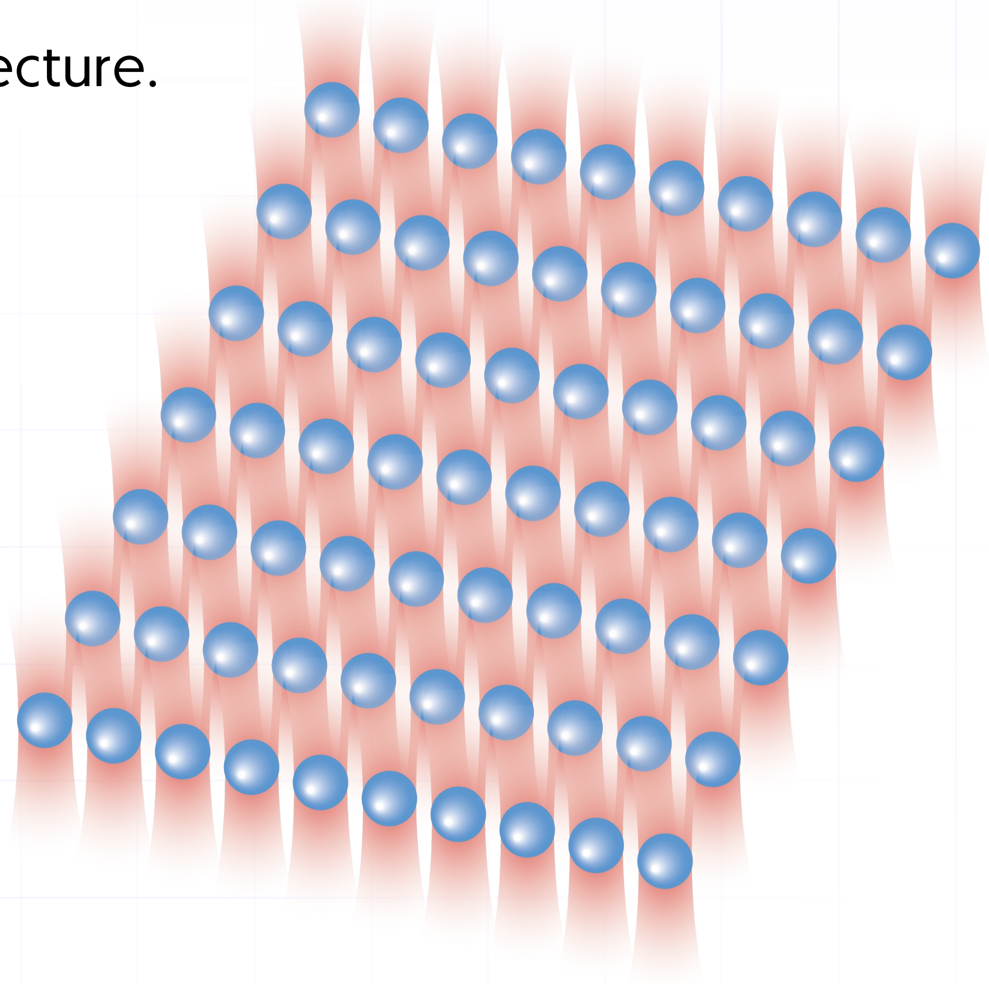
QuEra Computing Inc.

IEEE Quantum Week 2023: TUT23

Session I: Basics

Why you should care about neutral atoms.

1. Scalable to ~100,000 physical qubits with all-to-all connectivity, using current technology and physical architecture.
   - Scalable 'wiring': optical addressing + atom shuttling
   - Fidelities that don't degrade with processor size: 2-qubit 99.5%, 1-qubit 99.95%
   - Fast gates (1-100 us) and long memory times (>1 s)



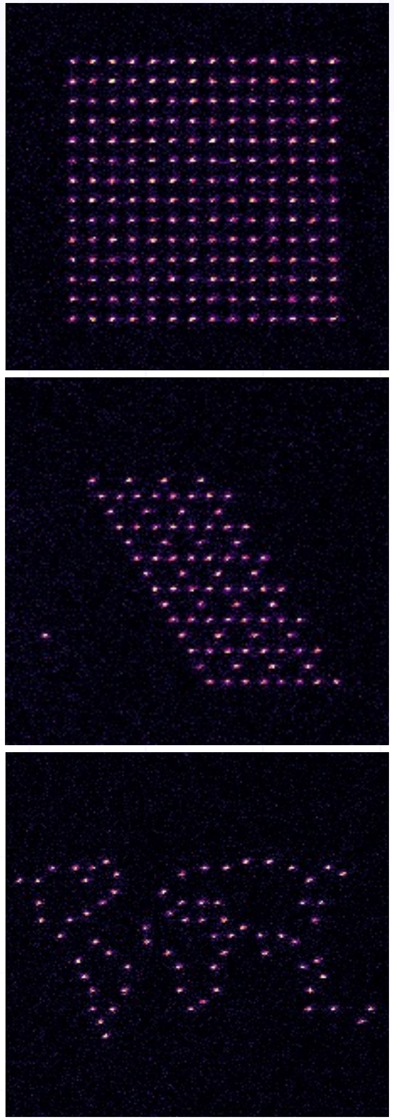
Bluvstein et al (2022), *Nature*, *604*(7906), 451–456

Evered et al (2023), arXiv:2304.05420

|QuEra>

Why you should care about neutral atoms.

2. Analog-mode machines are already driving scientific discovery and application development
   - Quantum simulations: antiferromagnets, many-body scars, spin liquids
   - Commercial applications: optimization, machine learning
   - QuEra's Aquila, with 256 qubits, is accessible on the cloud through AWS
     - FPQA™: 'Field-Programmable Qubit Array' enables problem encoding directly into atom positions
     - Defining Hamiltonian parameters as arbitrary waveforms enables a variety of quantum algorithms



Ebadi et al (2021), *Nature*, *595*(7866), 227–232

Bluvstein, (2021), *Science*, *371*(6536), 1355–1359

Semeghini (2021), *Science*, *374*(6572), 1242–1247

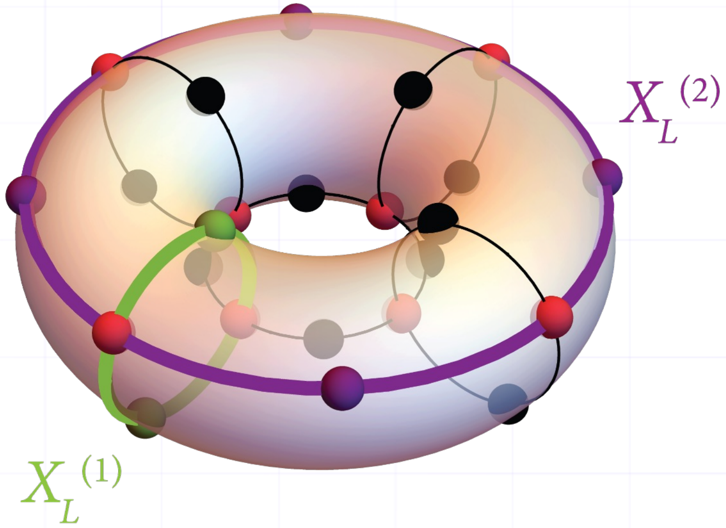
Ebadi et al (2022), *Science*, *376*(6598), 1209–1215

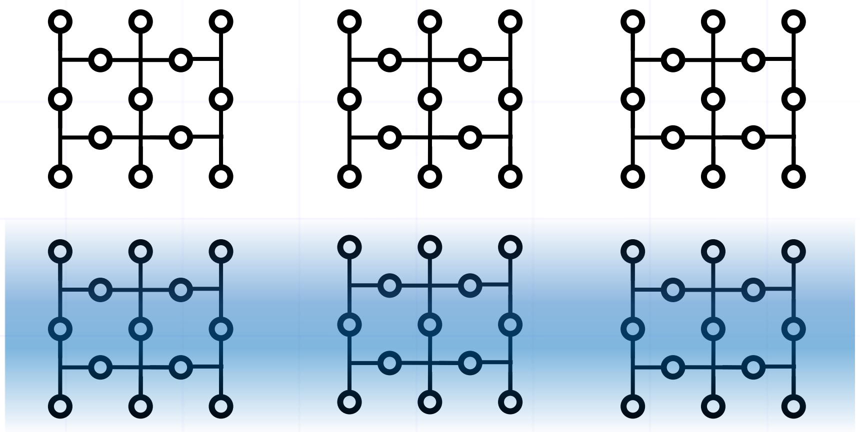
Wurtz et al (2022), arXiv:2205.08500

Wurtz et al (2023), arXiv:2306.11727

Why you should care about neutral atoms.

3. Digital-mode machines are a powerful testbed for fault-tolerant architectures:
   - FPQA™ approach enables programmable qubit layout
   - All-to-all connectivity enables novel error-correcting codes and efficient logical qubit operations
   - Mid-circuit readout and 100s of qubits coming soon





Bluvstein et al (2022), *Nature*, *604*(7906), 451-456

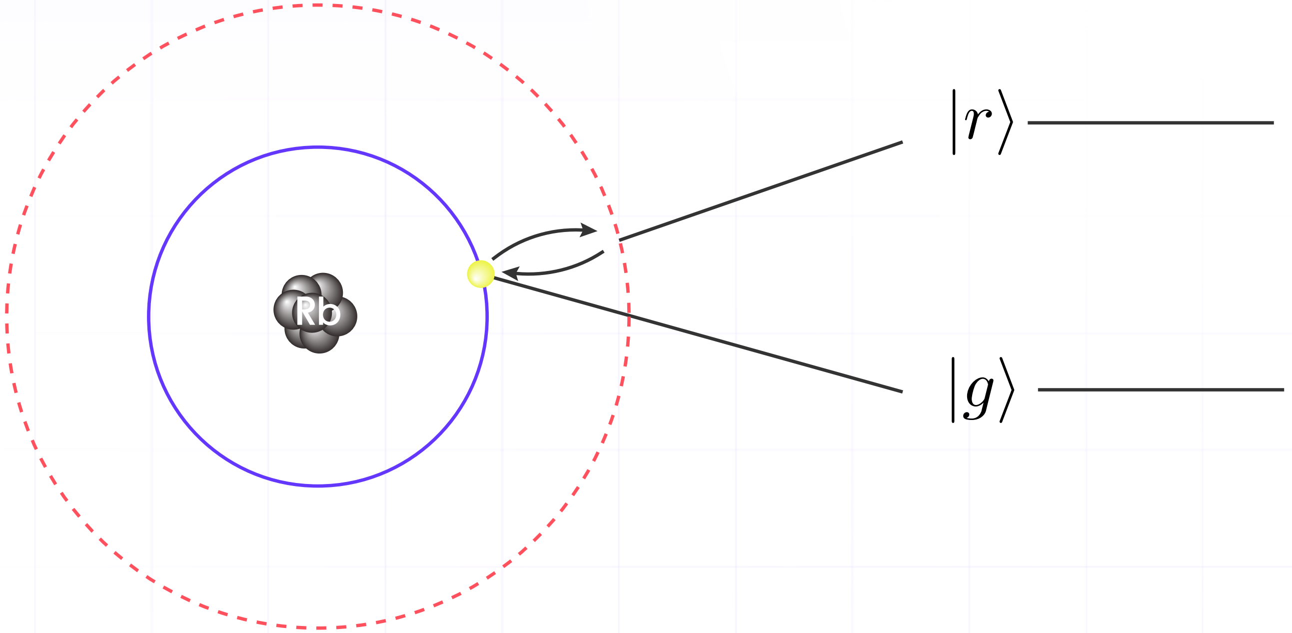
Evered et al (2023), arXiv:2304.05420

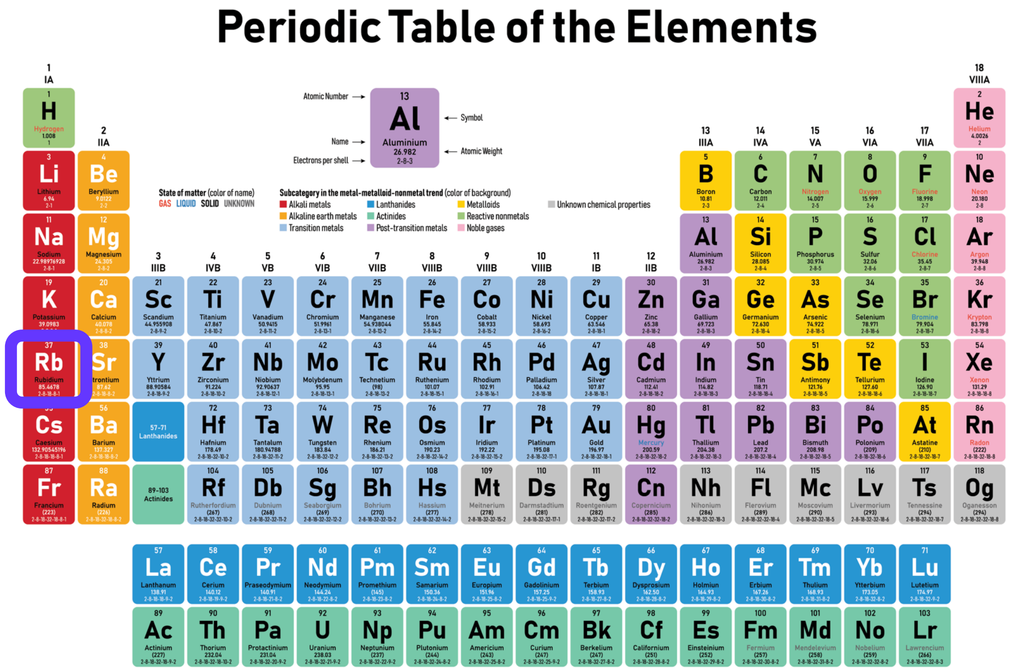
Learning objectives

By the end of the session, you will be able to:

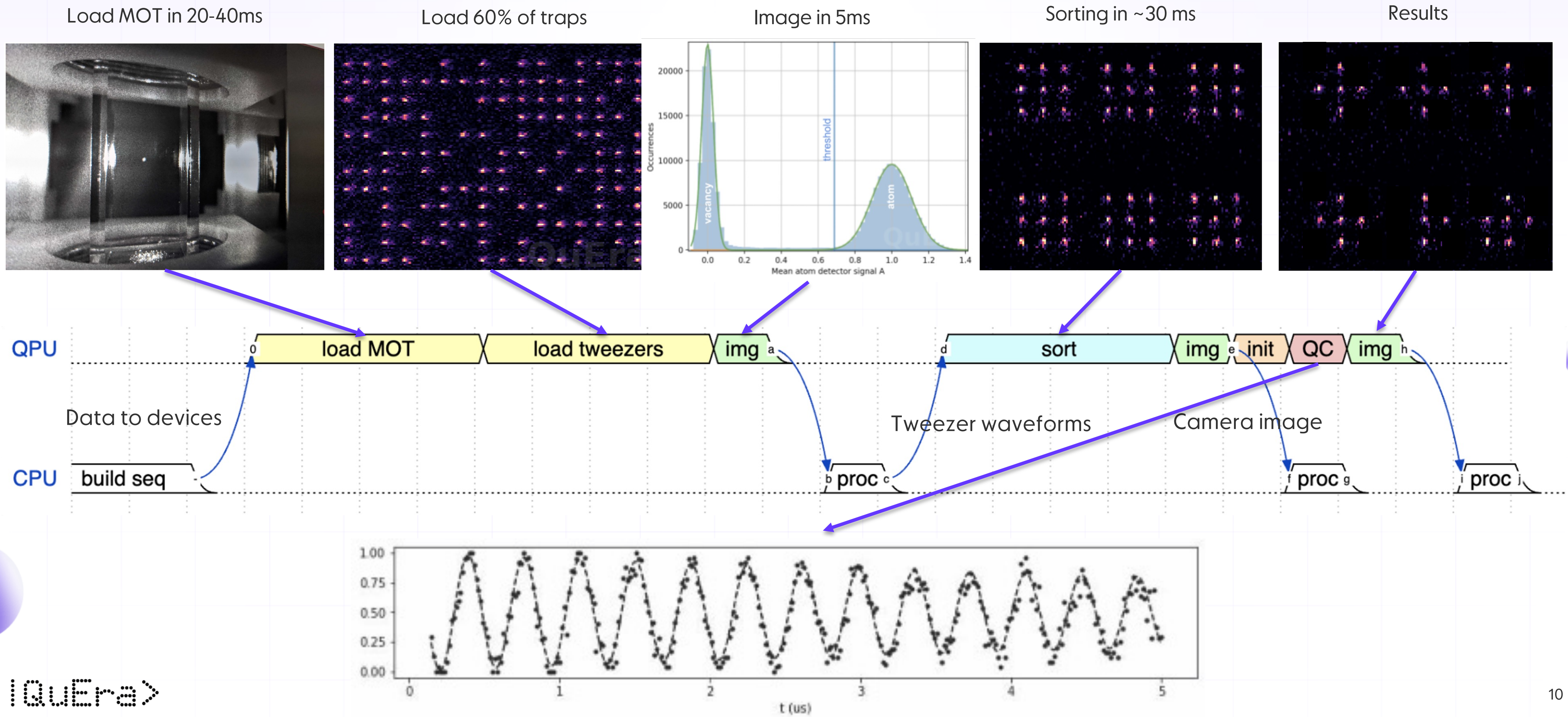
- **Explain** how **neutral atoms** can be used as a **platform** for quantum computing
- **Distinguish** **analog** and **digital** (gate-based) quantum computing
- **Describe** Aquila's **programming** range
  - its Hamiltonian and controllable parameters, limitations of service
- **Remember** the physics of **Rabi oscillations** and **describe** how to **emulate** it with QuEra's software

Qubits by puffing atoms





The routine

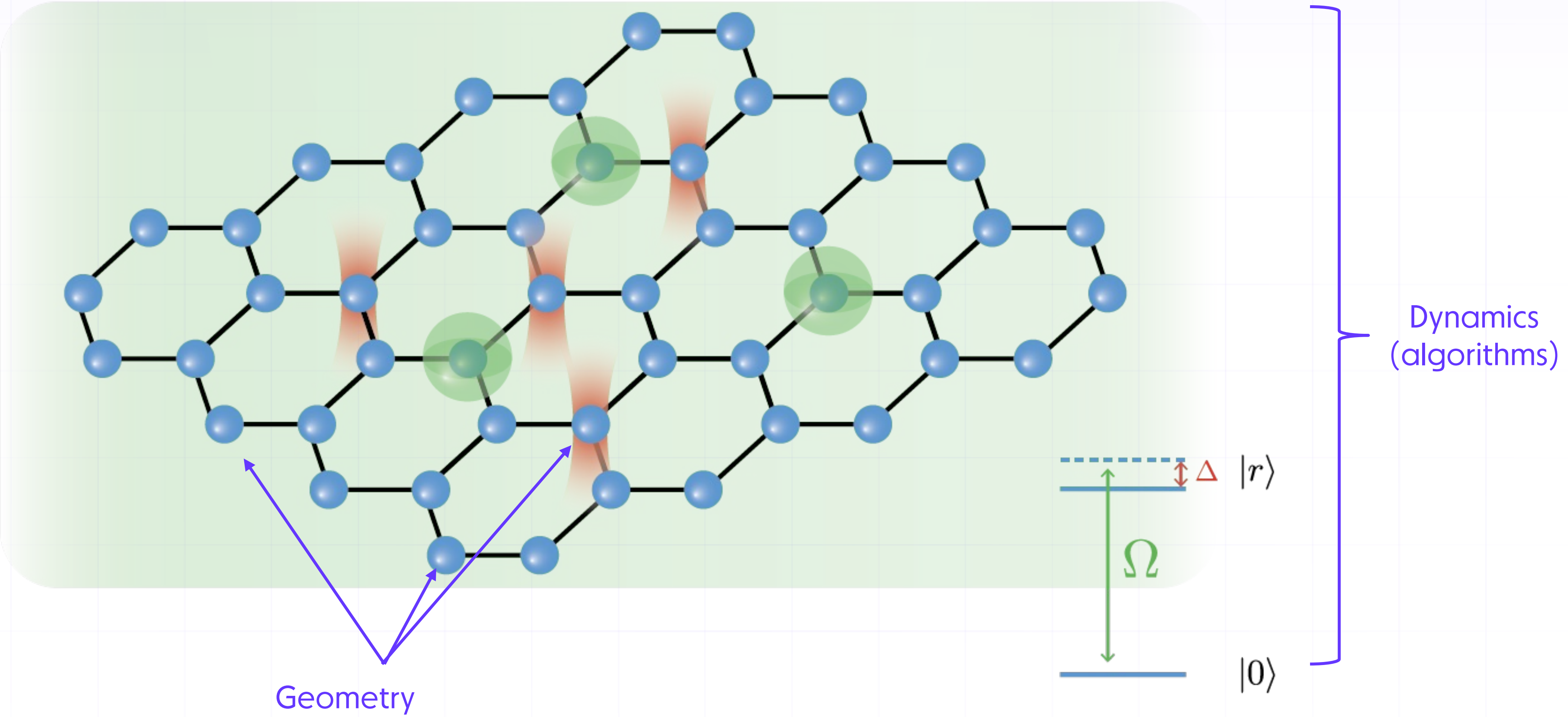


Load MOT in 20-40ms
Load 60% of traps
Image in 5ms
Sorting in ~30 ms
Results
Occurrences
vacancy
threshold
atom
Mean atom detector signal A
QPU
load MOT
load tweezers
img
sort
img
init
QC
img
CPU
build seq
proc
proc
proc
Data to devices
Tweezer waveforms
Camera image
t (us)

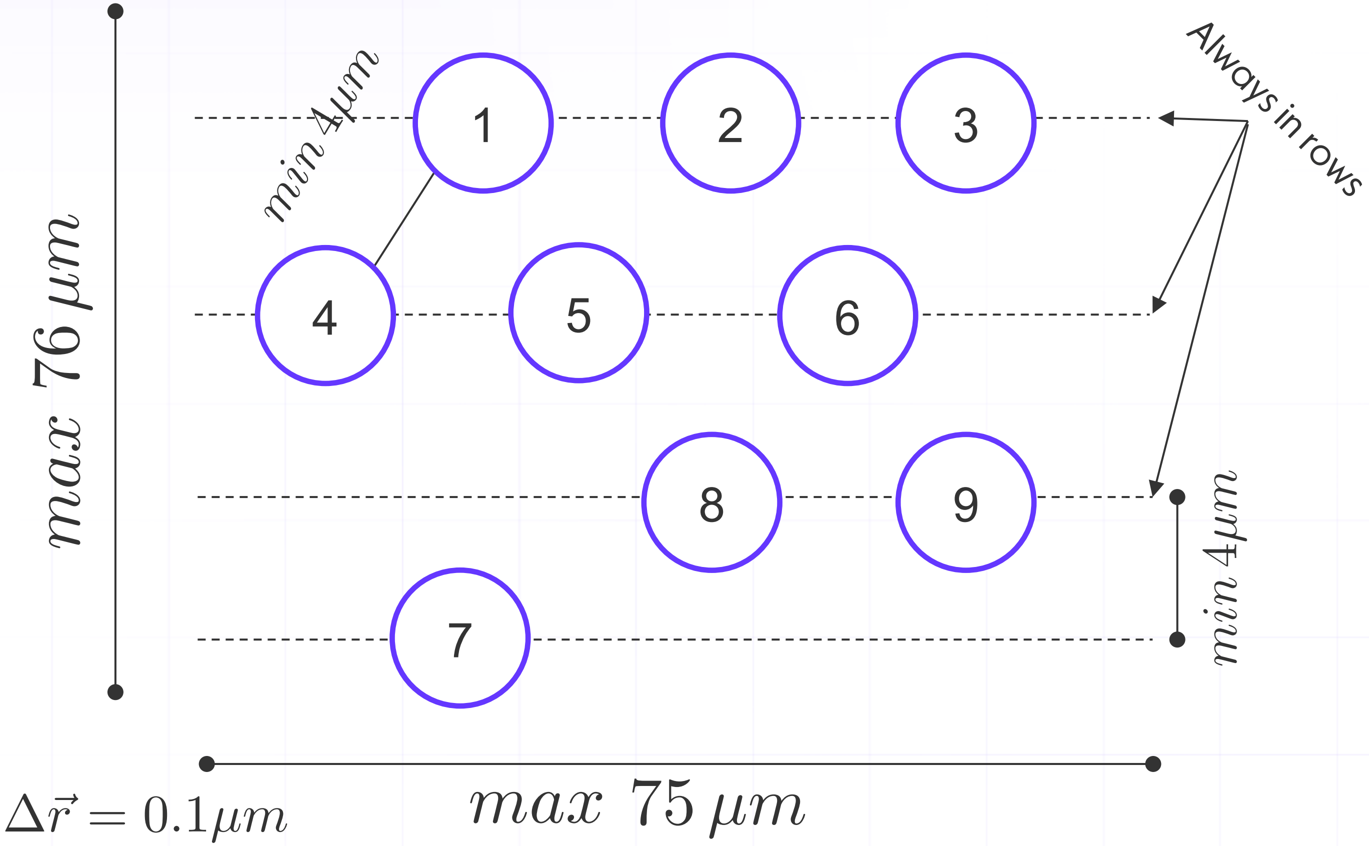
Sneak peak under the hood

Q View Demo

A neutral-atom quantum processor



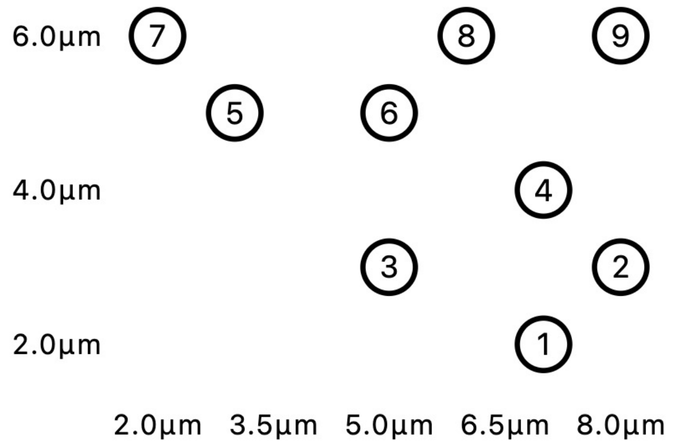
Hardware constraints: Geometry



$\Delta\vec{r} = 0.1\mu m$

BLOQADE

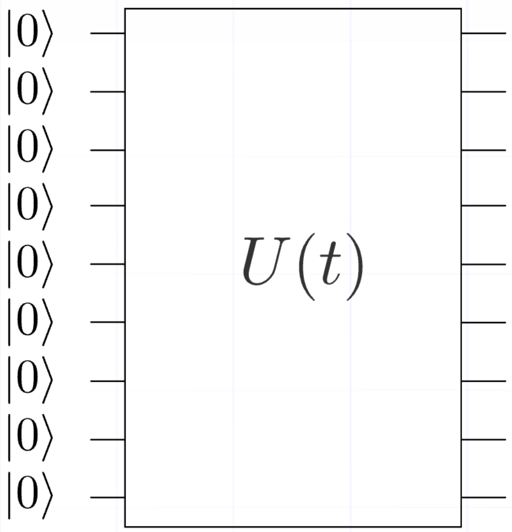
```
AtomList([(7.0, 2.0), (8.0,3.0), (5.0,3.0), (7.0, 4.0),
    (3.0,5.0), (5.0,5.0), (2.0,6.0), (6.0,6.0), (8.0,6.0)])
```



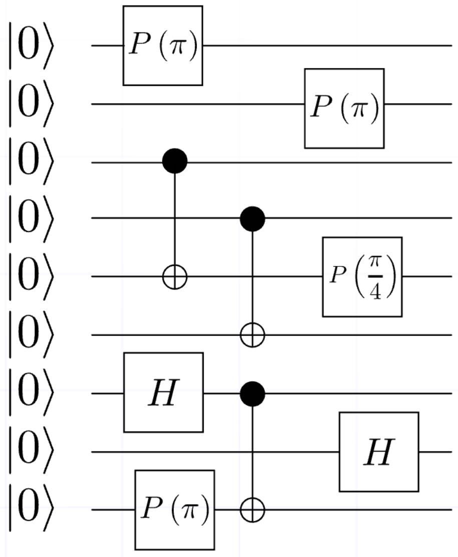
More details @ https://queracomputing.github.io/Bloqade.jl/dev/capabilities/

Information processing: Analog computing

Analog operation



Digital operation



Designed for the early stage of maturity of the quantum computing resources of today...

- Robustness to errors
- Efficient control
- Single-step large entanglement
- Universal applicability

More on analog processors:
Nature **volume 607**, p. 667–676 (2022)

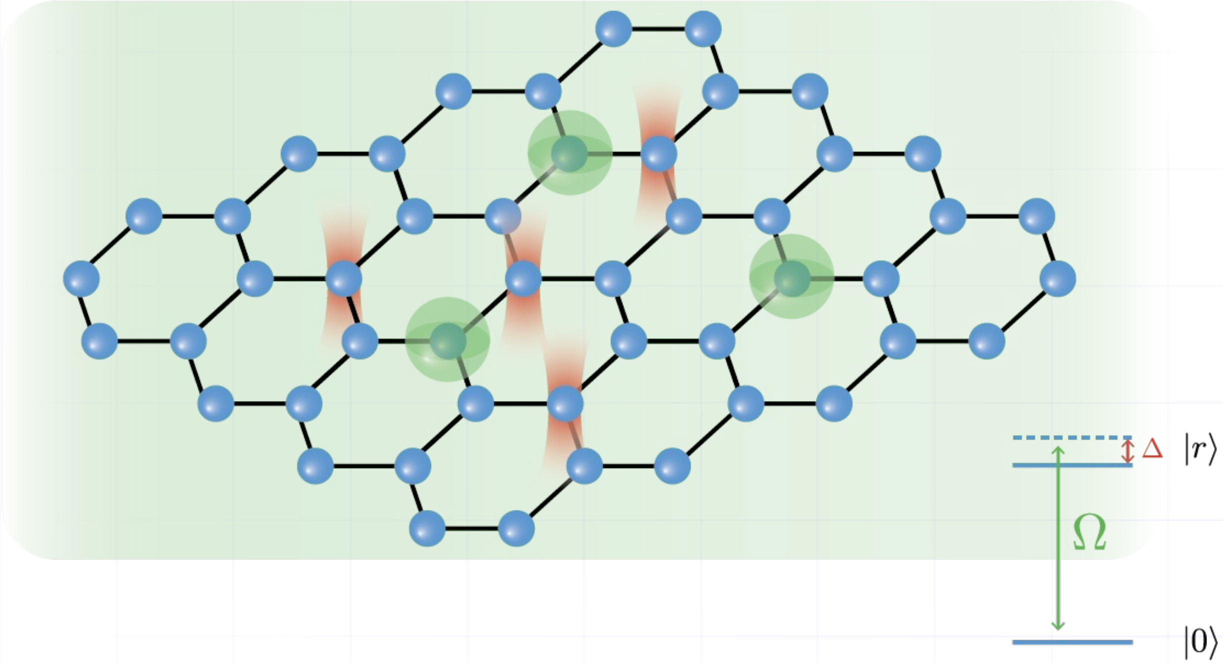
|QuEra⟩

Algorithm = time evolution

$$i\frac{\partial}{\partial t}|\psi\rangle = \boxed{H}|\psi\rangle$$

Analog quantum dynamics control

$$H = \sum_i \frac{\Omega(t)}{2}\left(e^{i\phi(t)}|g_i\rangle\langle r_i| + e^{-i\phi(t)}|r_i\rangle\langle g_i|\right) - \sum_i \Delta(t) n_i + \sum_{i<j} V_{ij} n_i n_j$$



$$n_i = 1 * |r_i\rangle\langle r_i| + 0 * |g_i\rangle\langle g_i|$$

$$V_{ij} \sim d_{ij}^{-6}$$

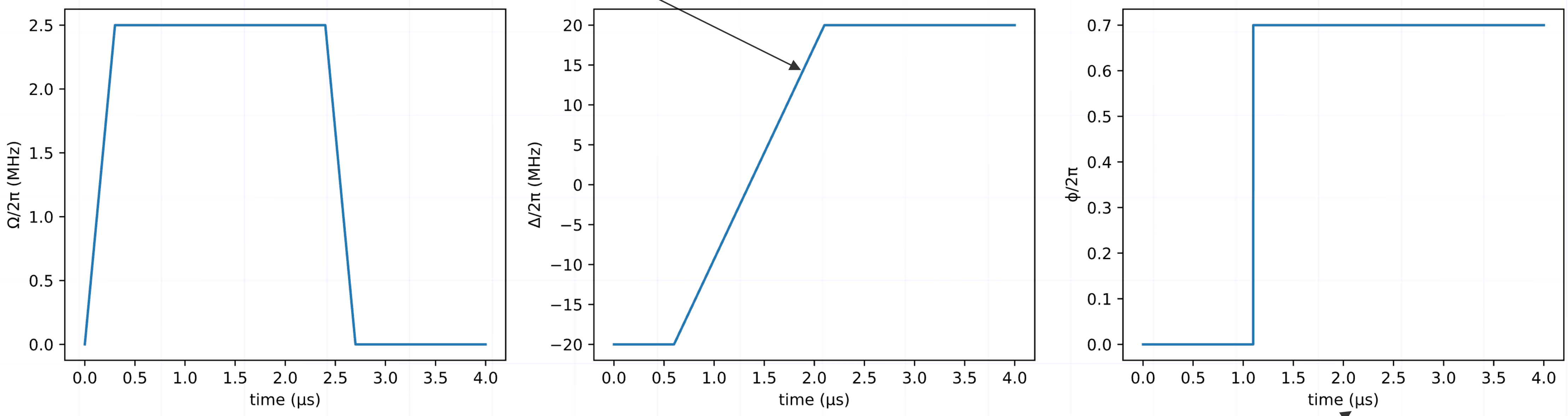
Hardware constraints: dynamics

```
Δ = piecewise_linear(clocks = [0.0, 0.1, 0.5, T_max], values = [Δ_start, Δ_start, Δ_end, Δ_end])
ϕ = piecewise_constant(clocks=[0.0,T_max], values= [0.0]);
```

BLOQADE

Piecewise-linear

max max



$$t_{max} = 4\mu s$$

Piecewise-constant

|QuEra>

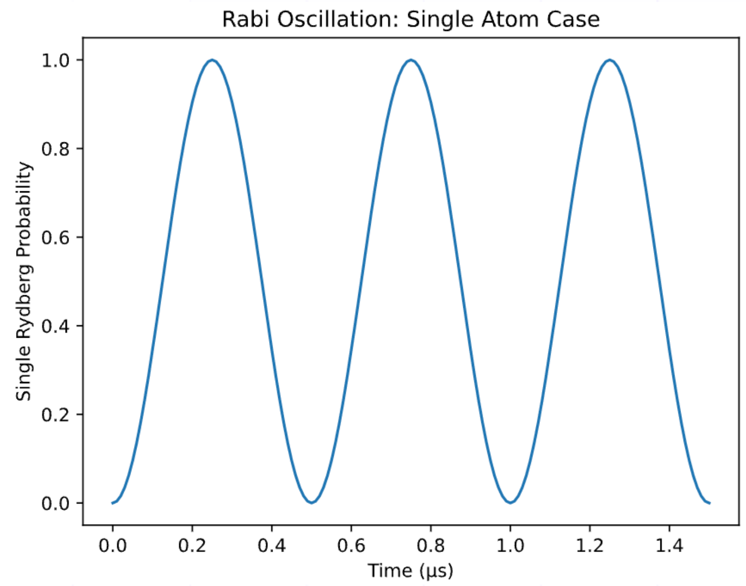
Full swing: Bloqade emulation example

- *Rabi oscillations*

$$n(t) = \langle \psi(t) | \hat{n} | \psi(t) \rangle = \sin^2 \frac{\Omega}{2} t$$

$$H = \frac{\Omega}{2} \sigma_x \qquad |\psi(0)\rangle = |g\rangle$$

$$|\psi(t)\rangle = \cos \frac{\Omega}{2} t \, |g\rangle - i \sin \frac{\Omega}{2} t \, |r\rangle$$




```
atom1 = AtomList([(0.0, 0.0)])
clocks = 0.0:1e-2:1.5
prob1 = KrylovEvolution(zero_state(1), clocks, rydberg_h(atom1; Δ = 0, Ω = 2π * 2))
```

Hamiltonian call

Evolution method

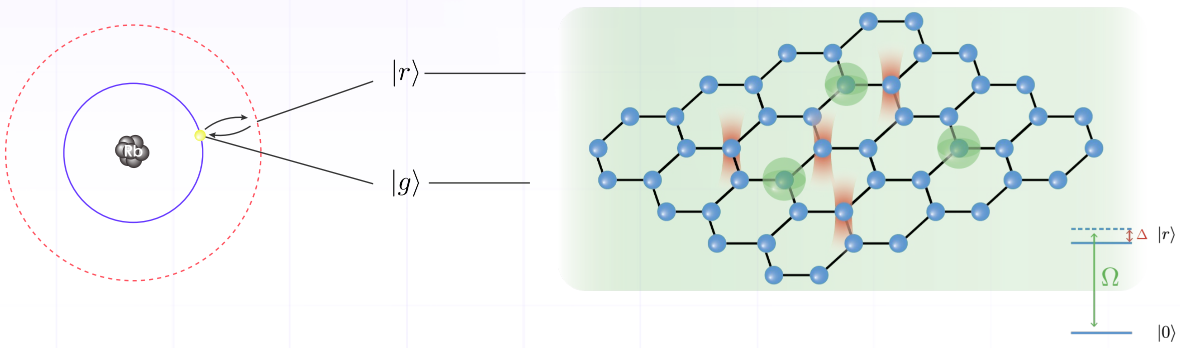
Initial register

geometry

waveforms

Summary

Architecture

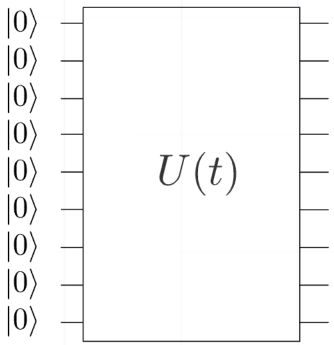


BLOQADE

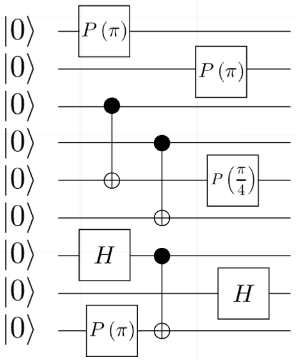
```
AtomList([(7.0, 2.0), (8.0,3.0), (5.0,3.0), (7.0, 4.0),
    (3.0,5.0), (5.0,5.0), (2.0,6.0), (6.0,6.0), (8.0,6.0)])
```

```
Δ = piecewise_linear(clocks = [0.0, 0.1, 0.5, T_max], values = [Δ_start, Δ_start, Δ_end, Δ_end])
ϕ = piecewise_constant(clocks=[0.0,T_max], values= [0.0]);
```

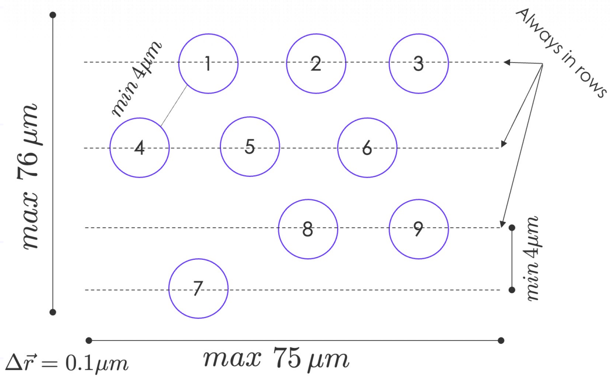
Analog operation

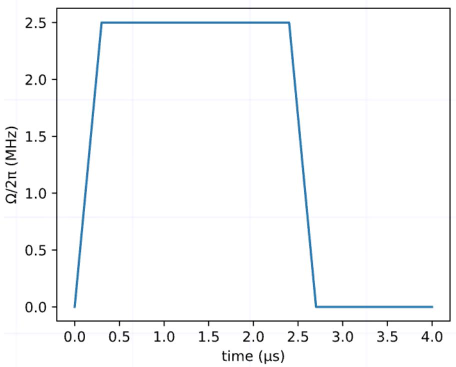


Digital operation

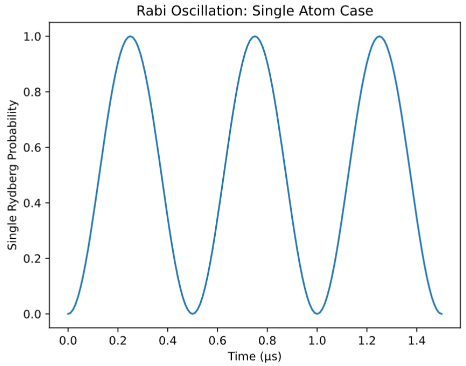


Hardware constraints





Emulation



Learning objectives

Now you are able to:

- **Explain** how **neutral atoms** can be used as a **platform** for quantum computing
- **Distinguish** **analog** and **digital** (gate-based) quantum computing
- **Describe** Aquila's **programming** range
  - its Hamiltonian and controllable parameters, limitations of service
- **Remember** the physics of **Rabi oscillations** and **describe** how to **emulate** it with QuEra's software