Introduction to Quantum computing and Callisto emulator

Viktor Radovic, *Senior Software developer*

4.12.2025. and 11.12.2025

Unique at Scale

QUANTUM SOFTWARE & EMULATOR OVERVIEW

- C12 / SW in quantum overview
- QC and Qiskit framework – quick overview
- Callisto emulator
  - Callisto - tutorials and exercises
  - Quantum adder exercise
  - GHZ state fidelity benchmark

Notes & plan:

Max points **20 points**

1. **5 points** → presence (2.5 - 4/12, 2.5 - 11/12)
2. **7 points** → Small exercises (qiskit & emulator)
3. **4 points** → Quantum adder on Callisto emulator
4. **4 points** → GHZ state

Contact: viktor @ c12qe.com

We are a leader in spin qubit quantum computing

- Founded in **2020**
- Spin-off from École Normale Supérieure Physics Lab
- **€50M** in total funding (equity + public)
- **65** employees, including **25** PhDs
- **20** nationalities, ⅓ women
- **8** patents
- Multiple world-first results in spin qubits

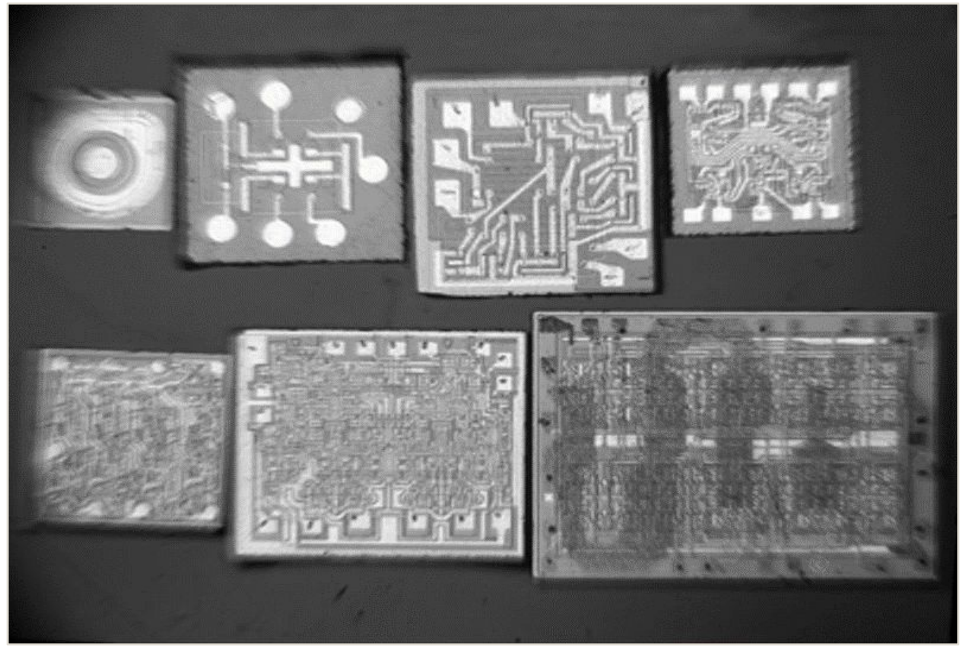
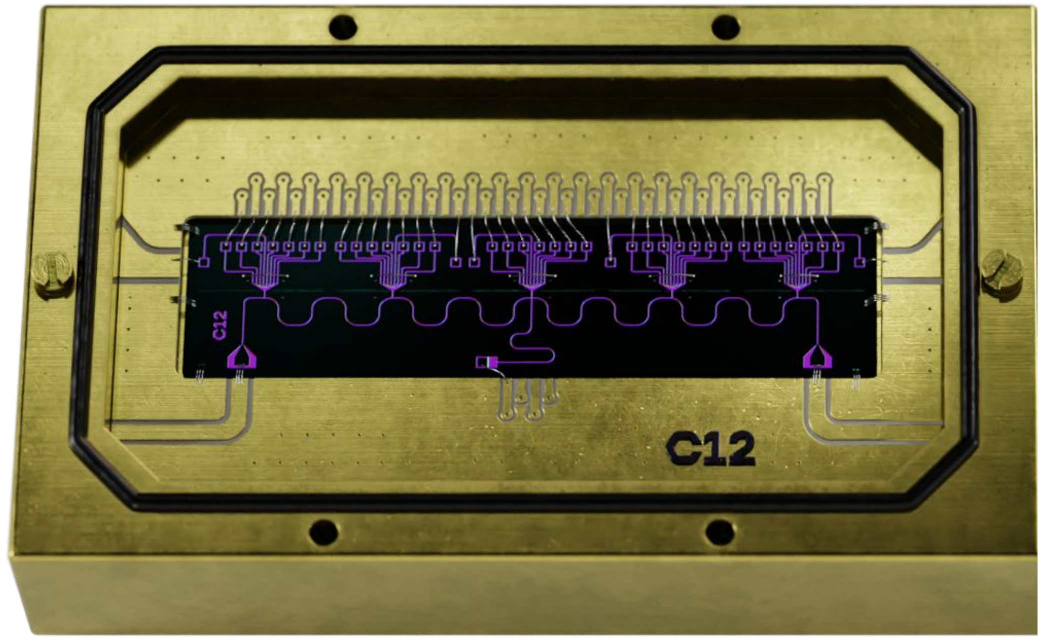
Varsity | European Innovation Council | VERVE VENTURES | bpifrance DIGITAL VENTURE | 360 CAPITAL | BNP PARIBAS | AIRBUS VENTURES | C12

Our vision

Reimagining how quantum hardware is built

to create quantum computers that are unique at scale

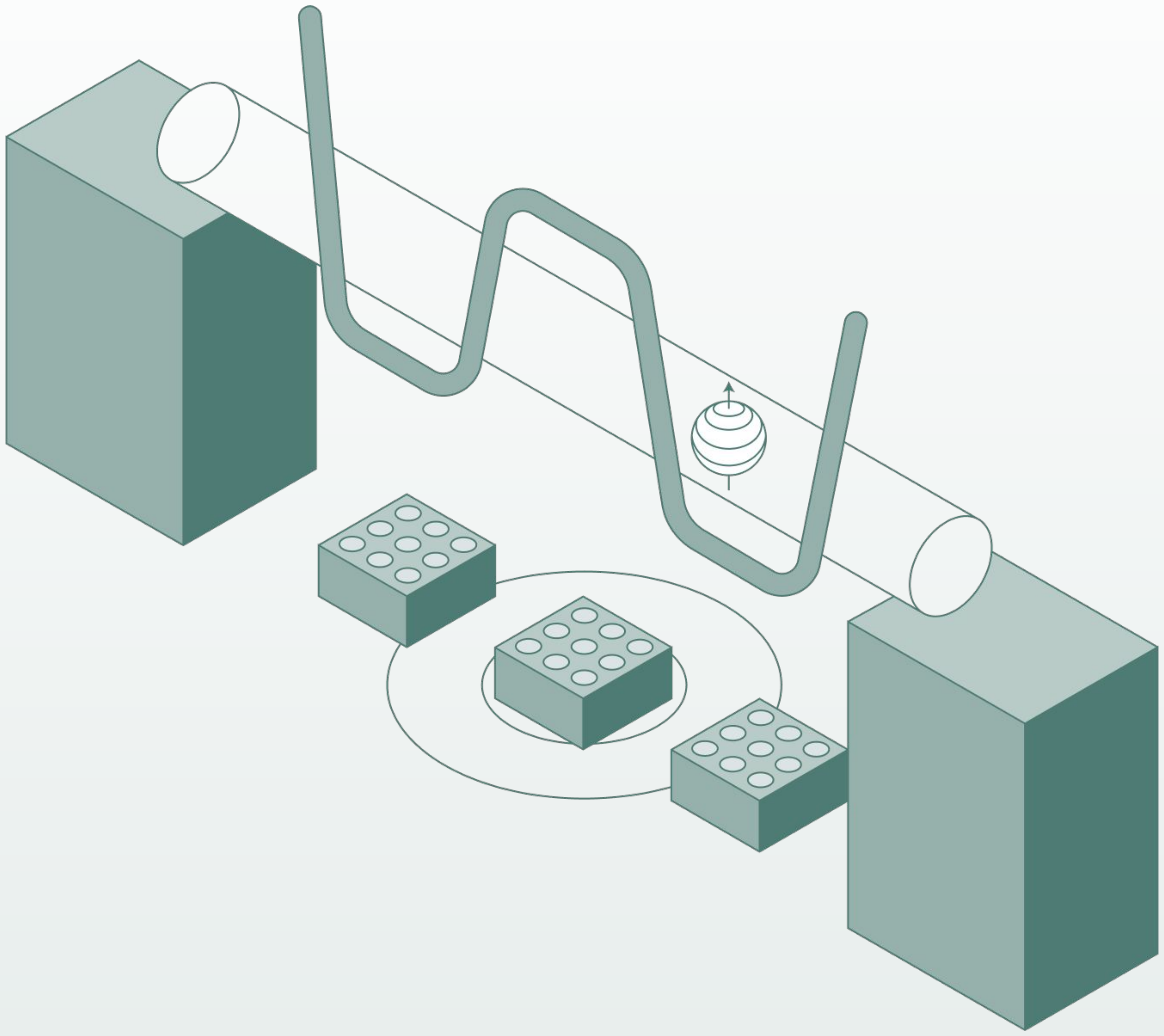
Carbon nanotubes will be what silicon was for classical transistors: the enabling material

Classical computing **[From 1960s]**	Silicon	
Quantum computing **[From 2020s]**	Carbon nanotubes	

C12

C12's qubit is a spin qubit hosted in a single carbon nanotube

- Ultra-pure carbon nanotube, connected between electrical contacts, is suspended above an array of gate electrodes
- Gate electrodes manipulate a single electron within a double quantum dot
- Local magnetic field entangles the electronic spin with the charge dipole in the double quantum dot
- This spin qubit is addressed via microwave pulses



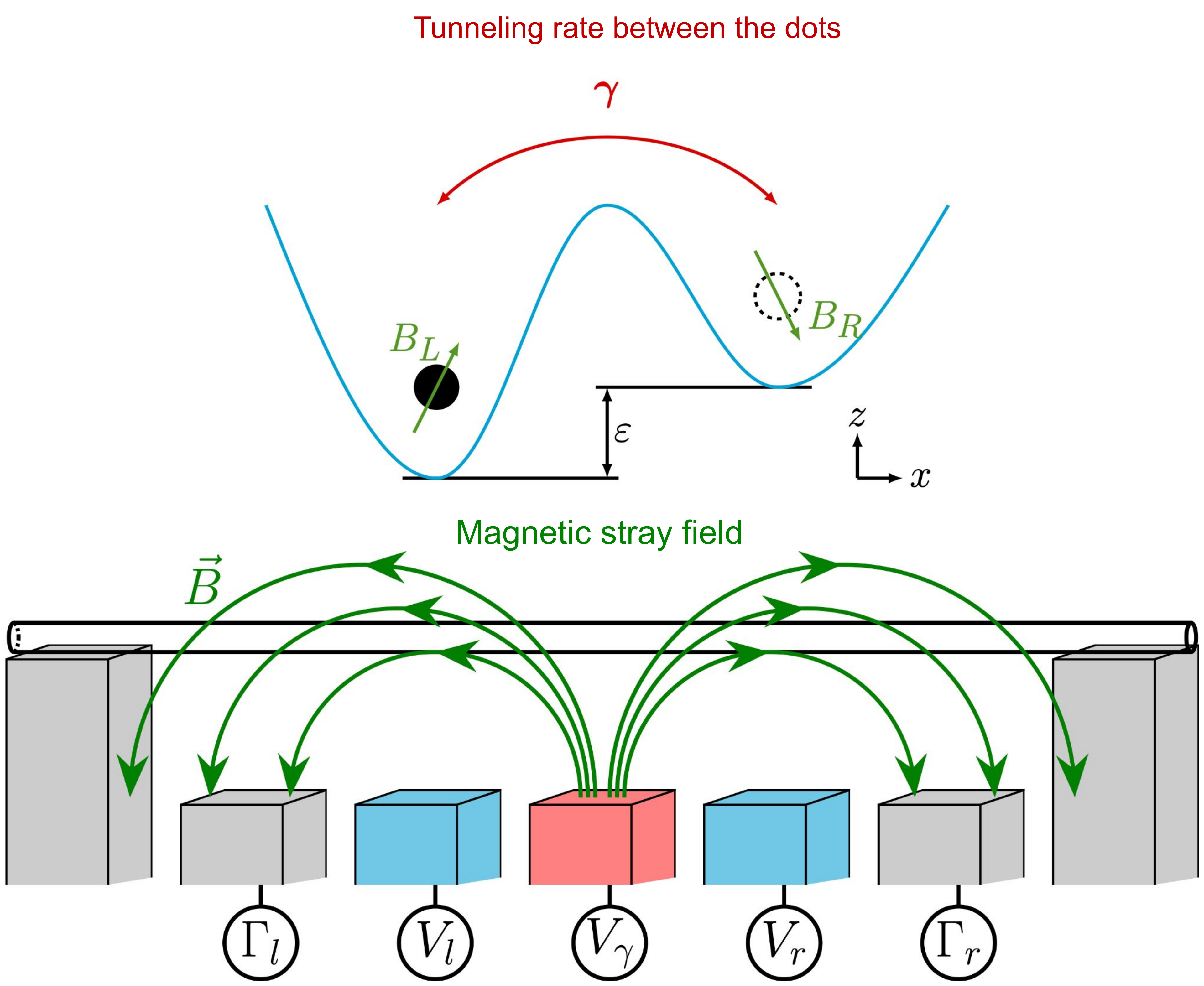
The spin is controlled through the charge

An electron is held within a double quantum dot in an inhomogeneous magnetic field

"Spin up" = ½ = | 0 >

"Spin down" = - ½ = | 1 >

So when reading the qubit, two possible values; keeping in mind the actual qubit state prior to reading is a probability



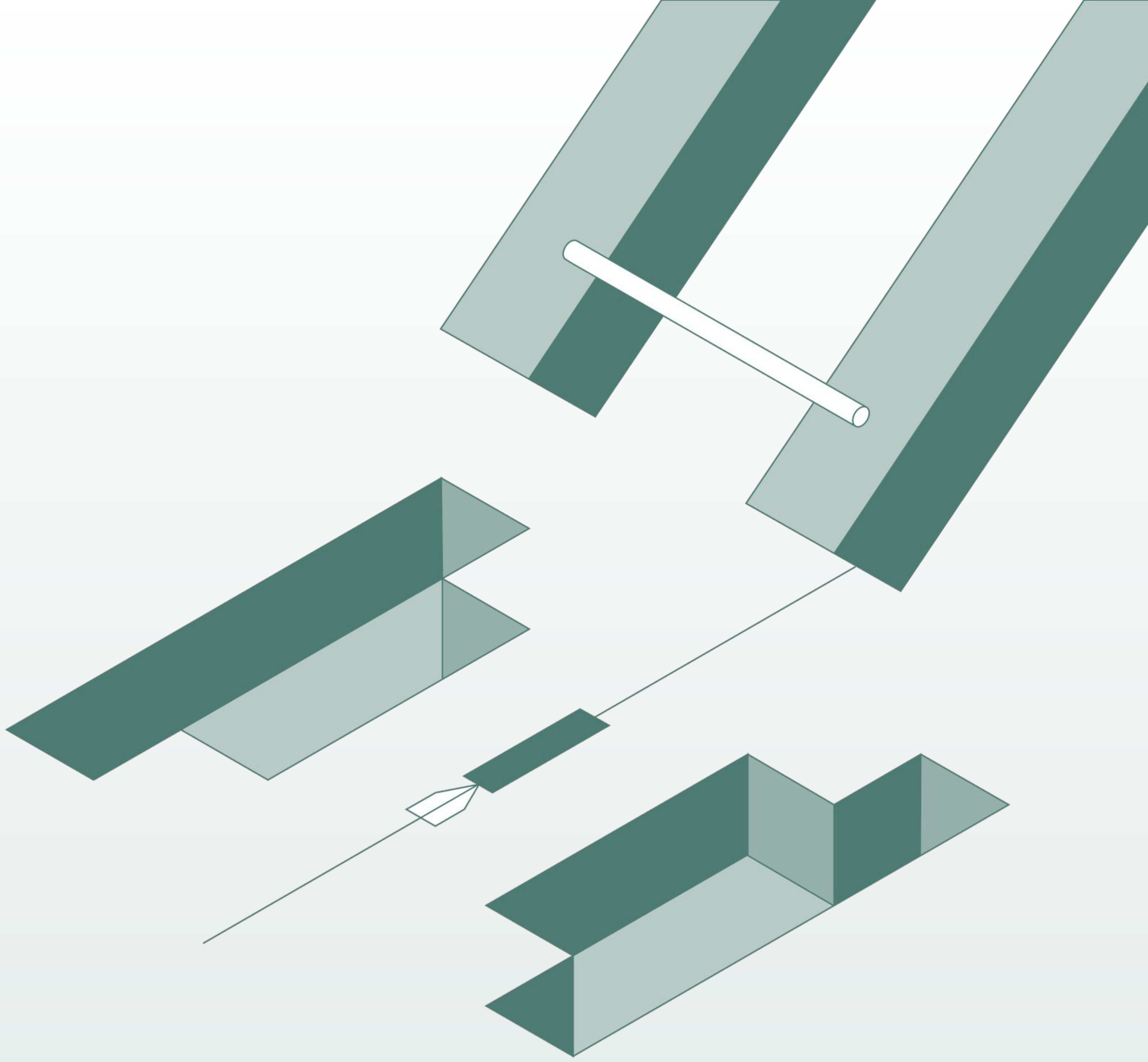
@ Q. Schaeverbeke

Groundbreaking nano-assembly process

Circuit fabrication done using mature semiconductor fabrication techniques

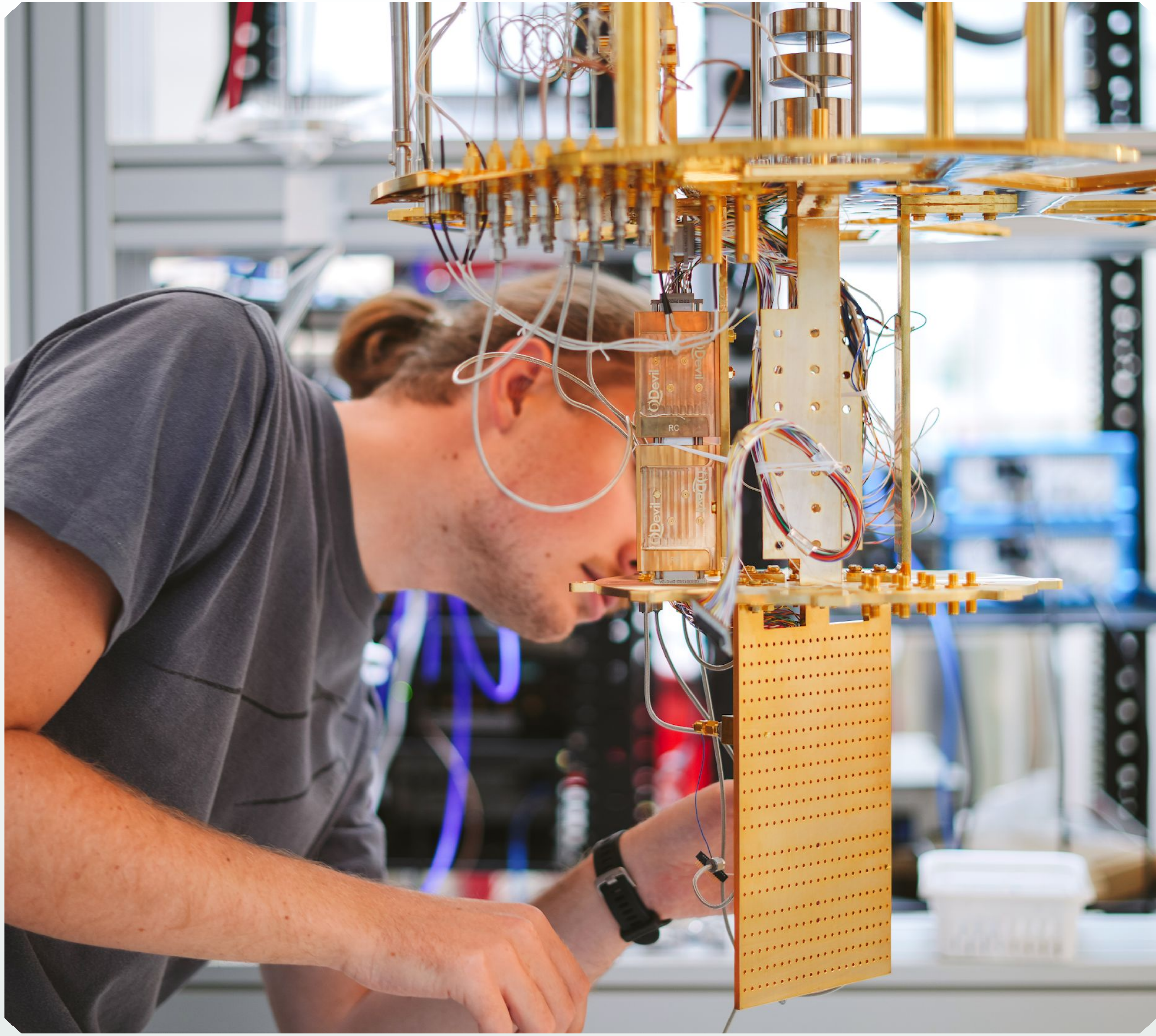
Proprietary process to grow, characterize and select nanotubes

Patented circuit integration at the end of the fabrication process to attach the nanotubes to electrodes which cut them



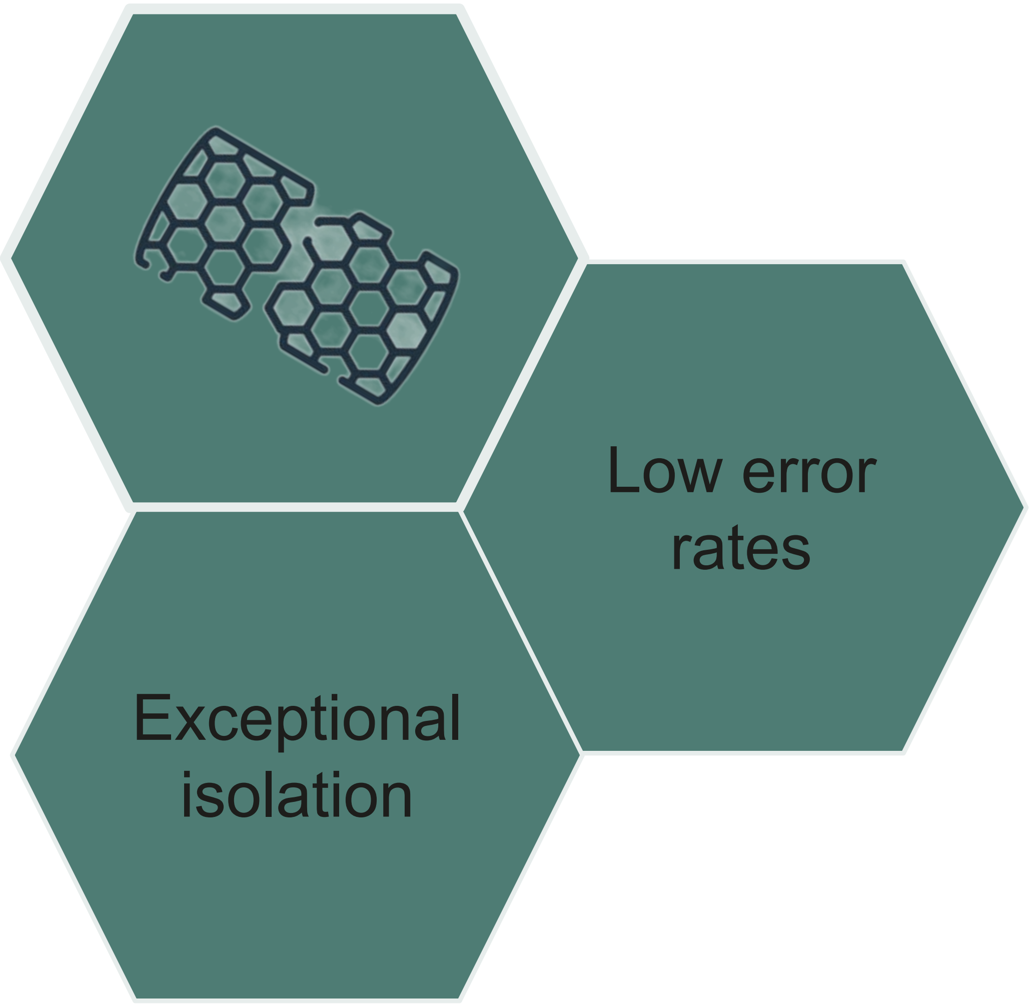
Scalable quantum computing

- Semiconductor integration leverages mature CMOS technology to enable scalability through advanced packaging, heterogeneous integration, and cryo-electronics.
- Qubit pre-selection ensures consistent quality
- High-qubit density, connectivity and quality enable an efficient error correction scheme at scale



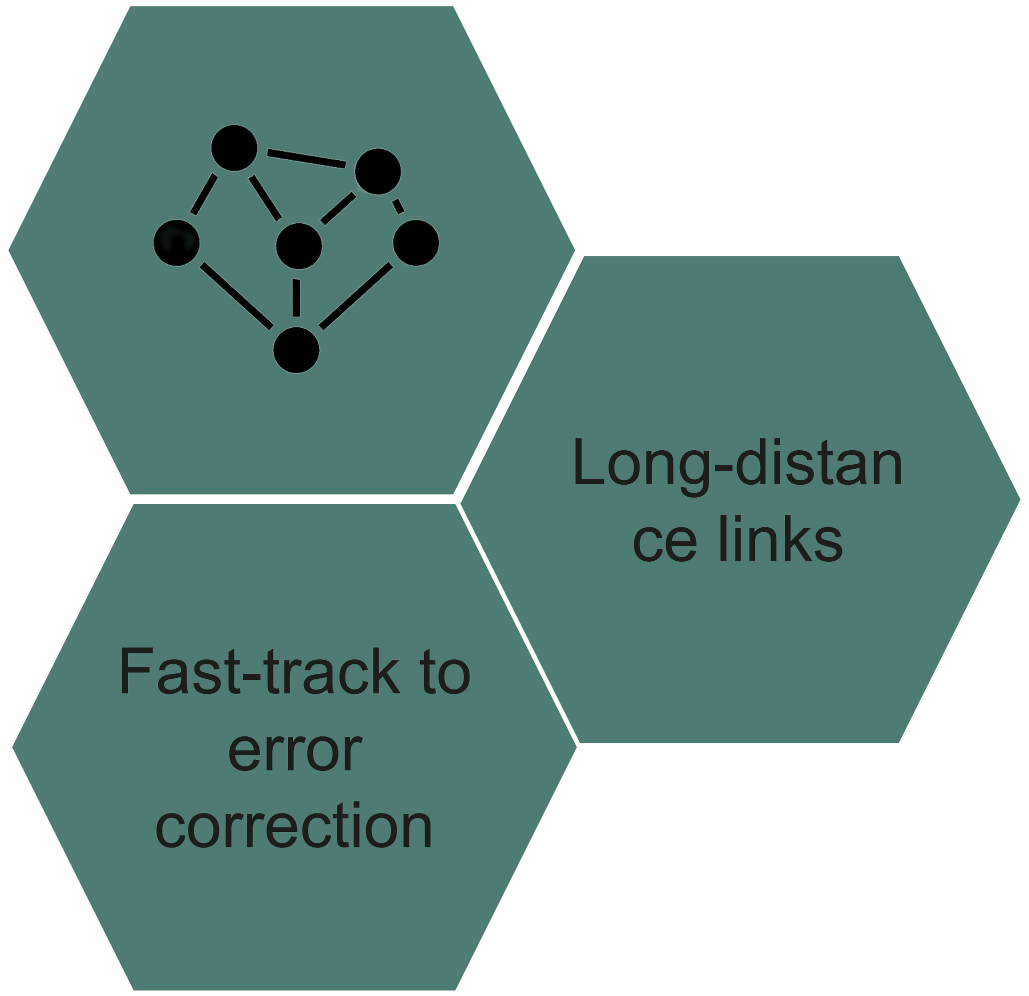
C12’s approach has a unique combination of benefits

High-fidelity qubits



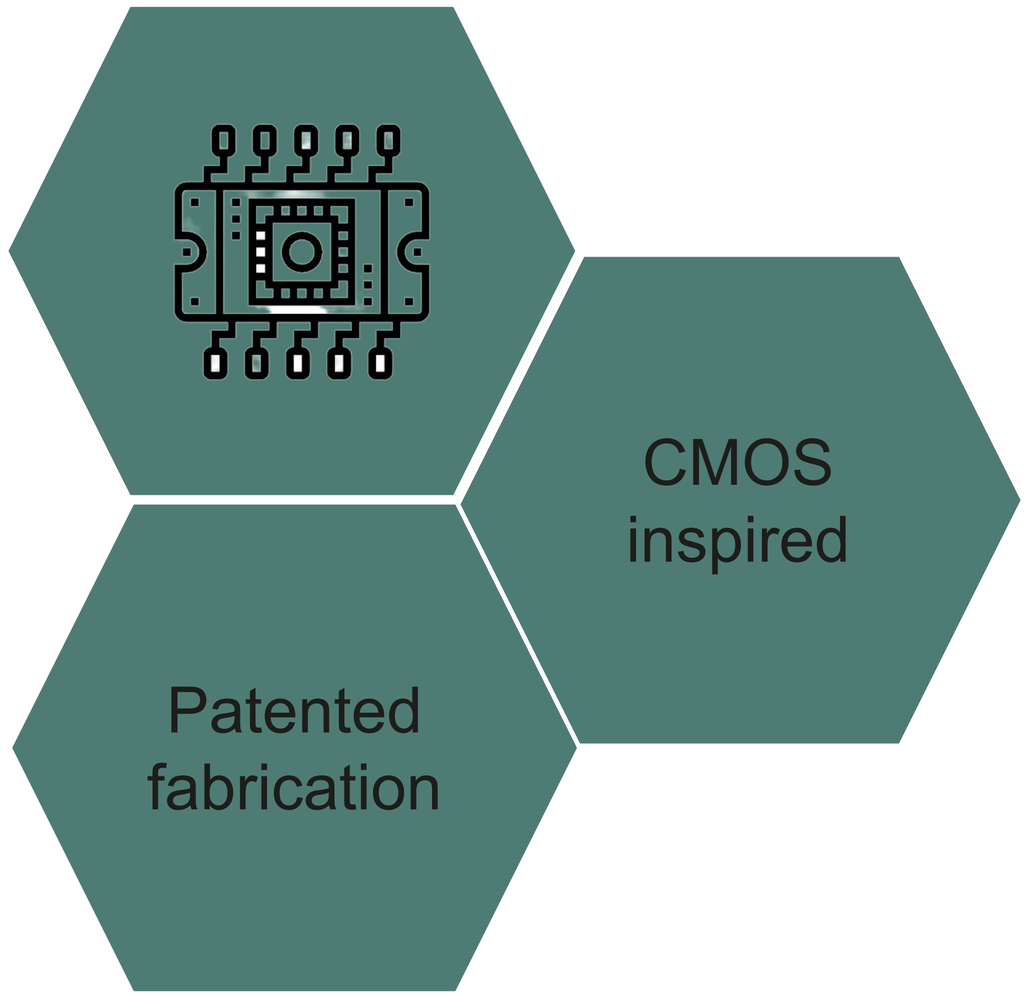
Qubits are hosted in **ultra-pure carbon nanotubes** to minimize environment-induced errors.

High connectivity



High connectivity leads to fast algorithms and very **low overhead** for error correction.

Scalable fabrication



Proprietary **high-throughput assembly** technique compatible with semiconductor **fabrication**.

Our own fab: driving 10x faster iteration cycles and full control

- 900 m² of offices & labs
- 200 m² ISO-7 cleanroom
- 3 growth & characterization labs
- Full control from material to device

Inside C12's quantum foundry

Nanofabrication

- Wafer-scale fabrication
- Magnetic gate (2nd generation)
- Patent for 3rd generation

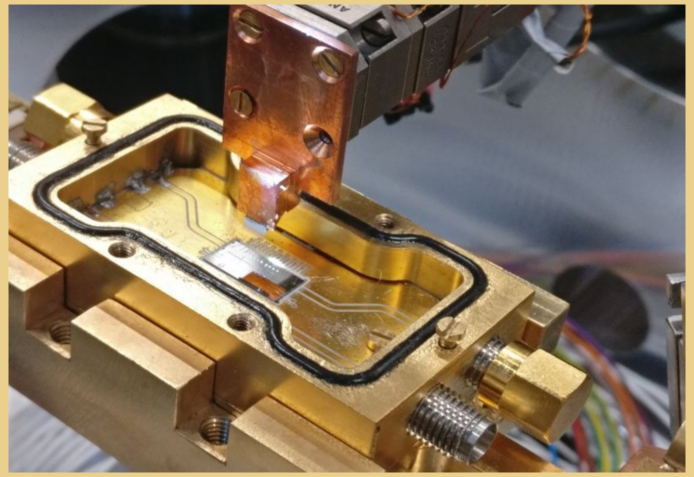
CNT Growth

- Localized deposition
- Stable growth

CNT Characterization

- Optical characterization to select single tubes vs bundles
- In situ electrical detection

CNT Nano-assembly



- High throughput
- Quality check

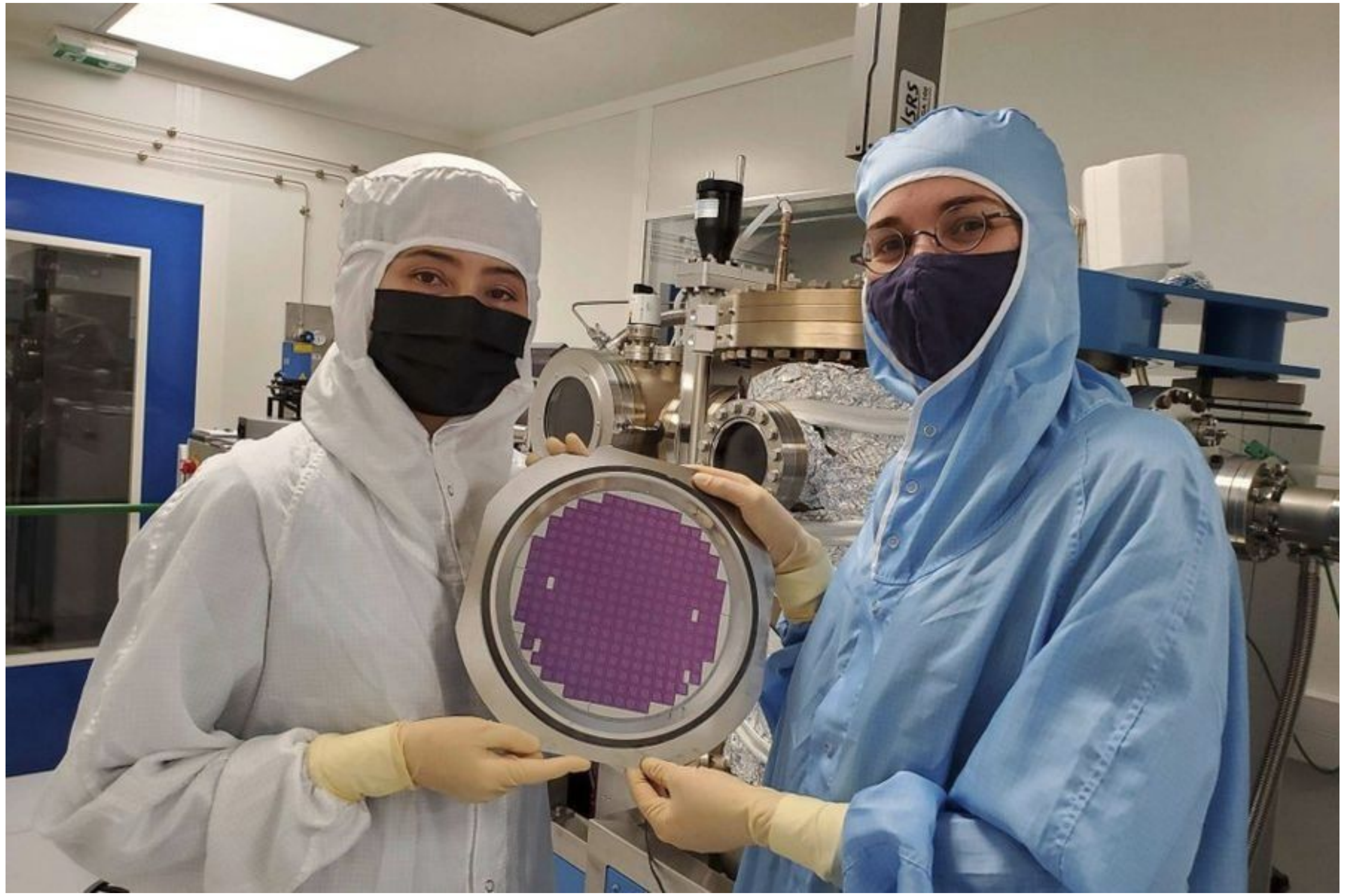
Nano-fabrication: producing integrated circuits on chips

In-house cleanroom:

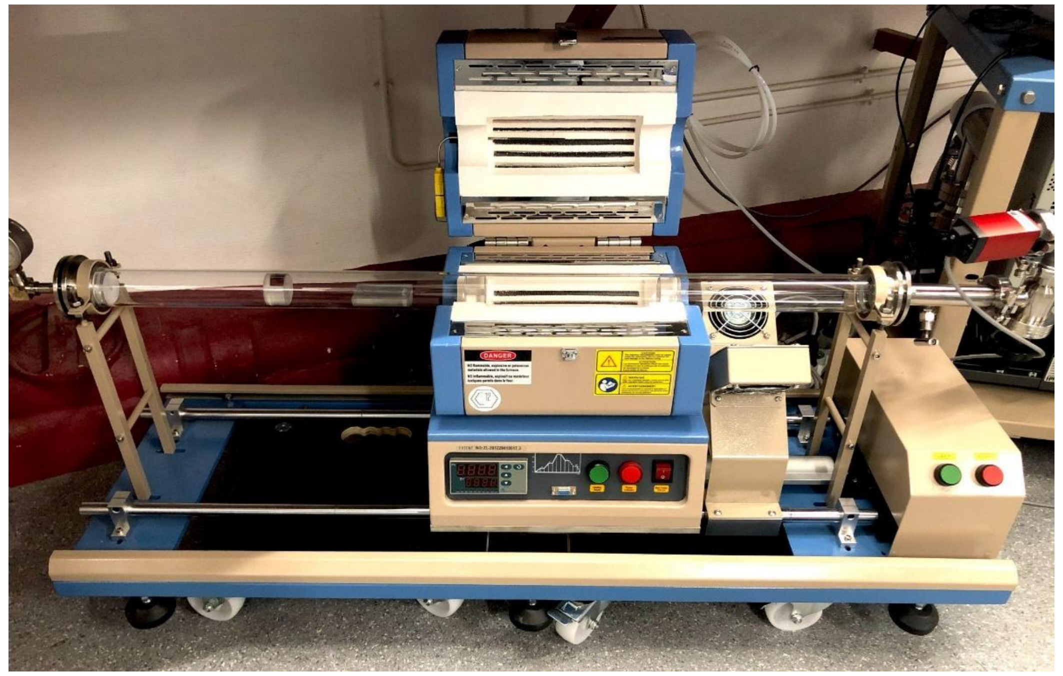
- Electron beam lithography
- Scanning electron microscopy
- Evaporation
- Reactive ion etching
- etc.

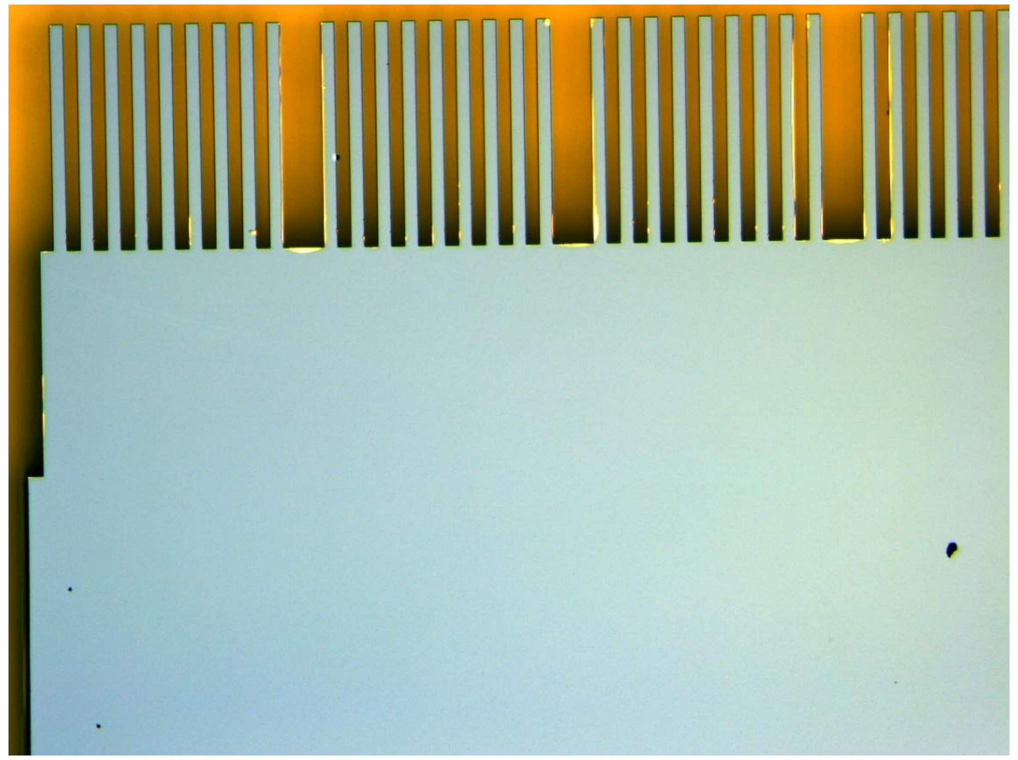
Chips from CEA-LETI (Grenoble, France)

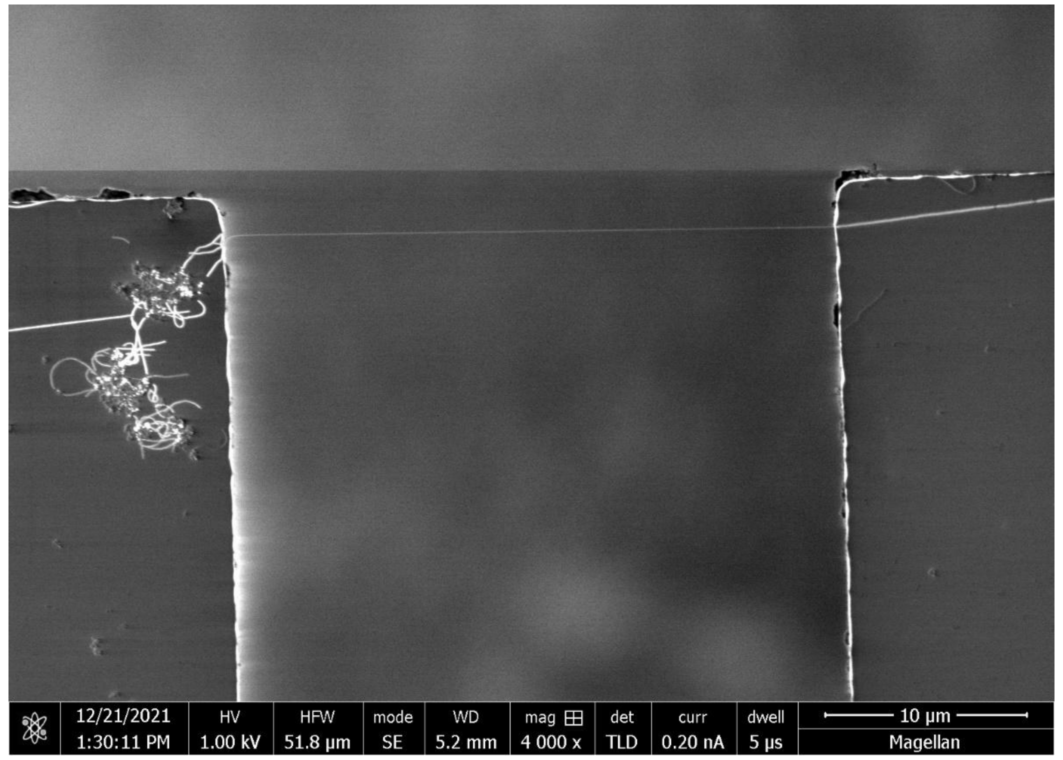
Collaboration with Institut Jean Lamour (Nancy, France)

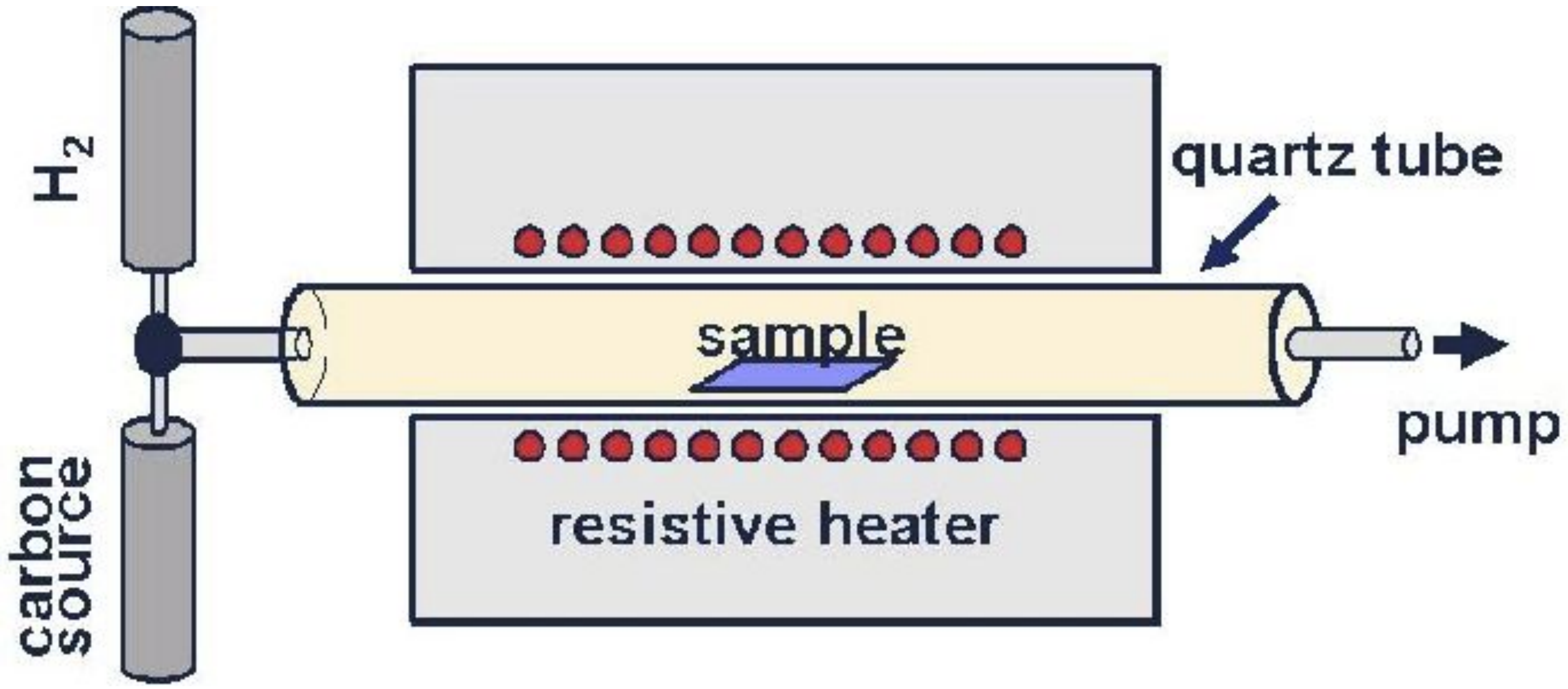


Growth: Growing carbon nanotubes on cantilever dies with chemical vapor deposition





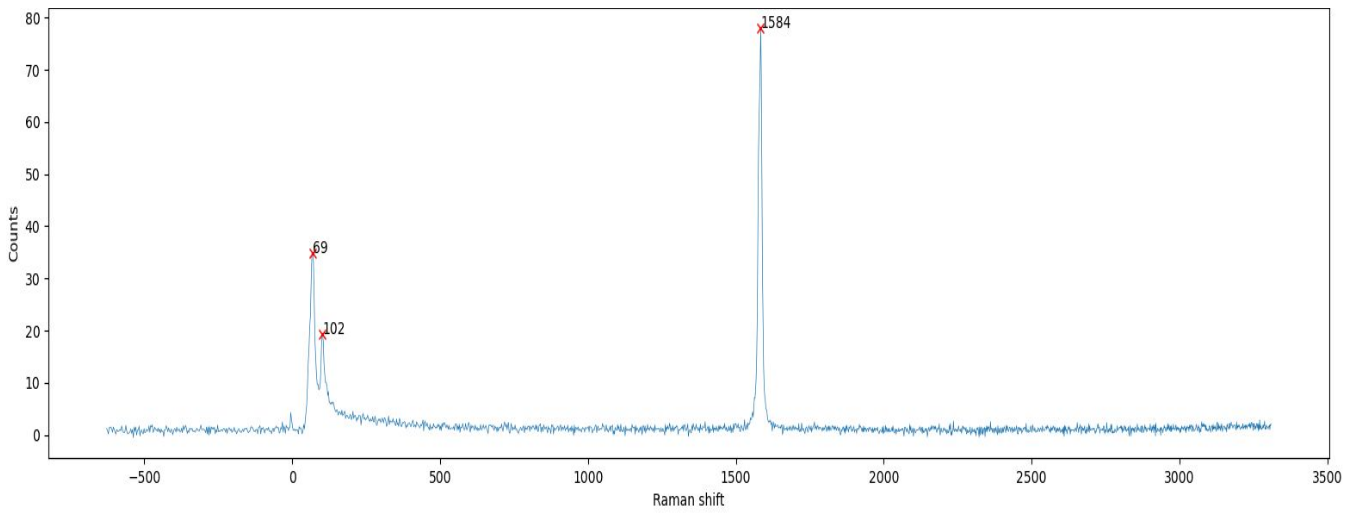


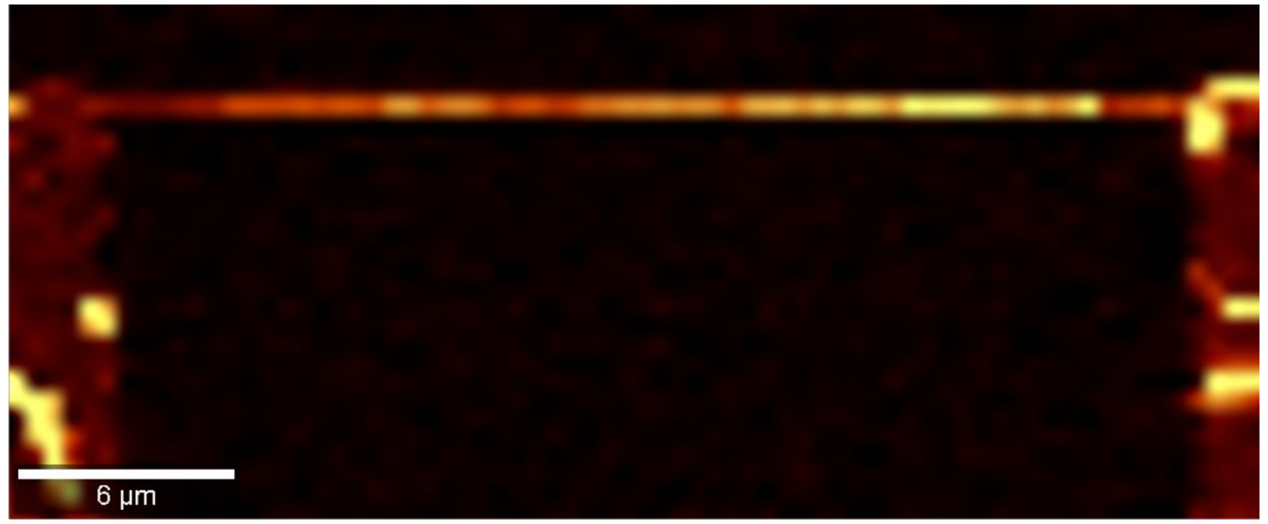


Seidel et al., 2004, Faster and Smaller with Carbon Nanotubes?

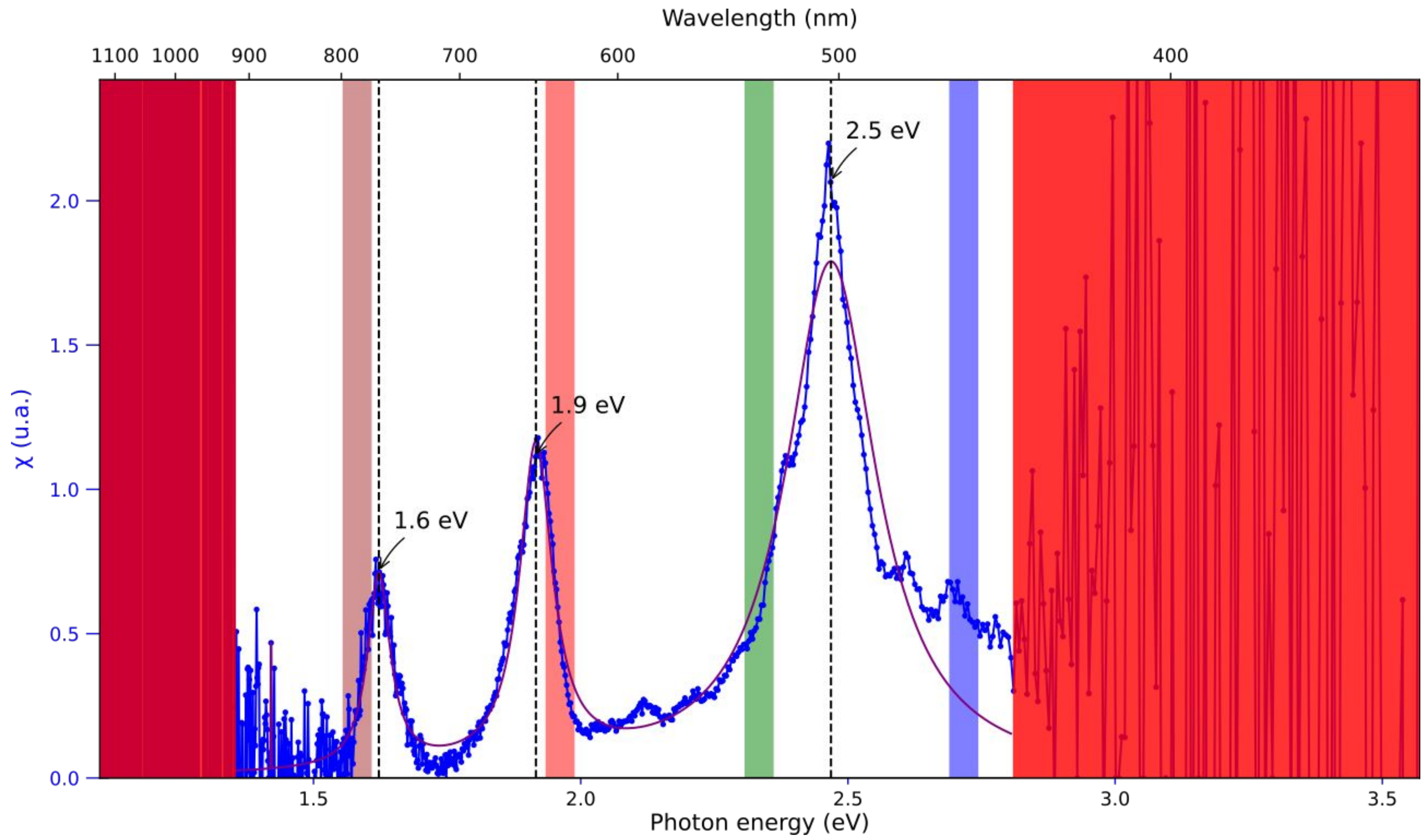
Characterization: selecting good nanotubes for qubits

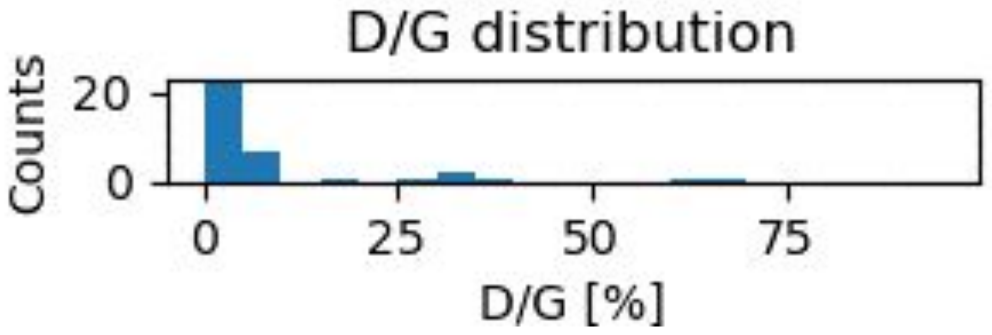
Raman: vibrational modes, defects

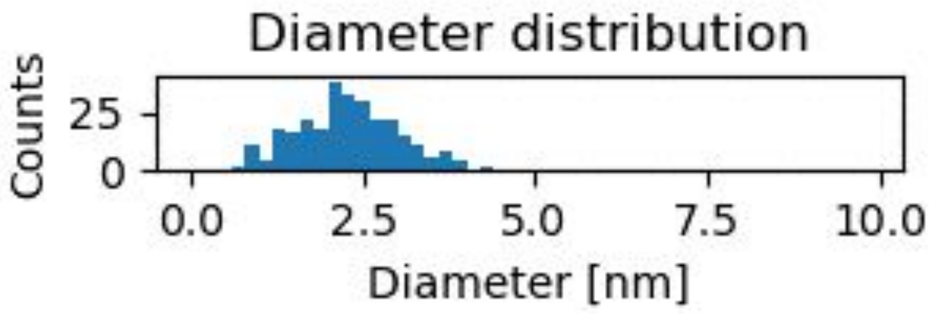


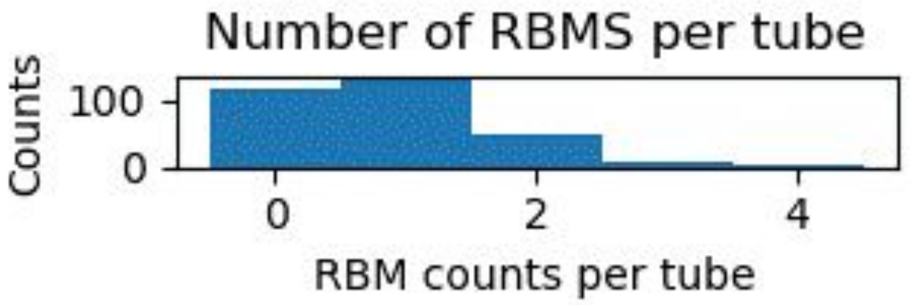


Rayleigh: size, shape, distribution





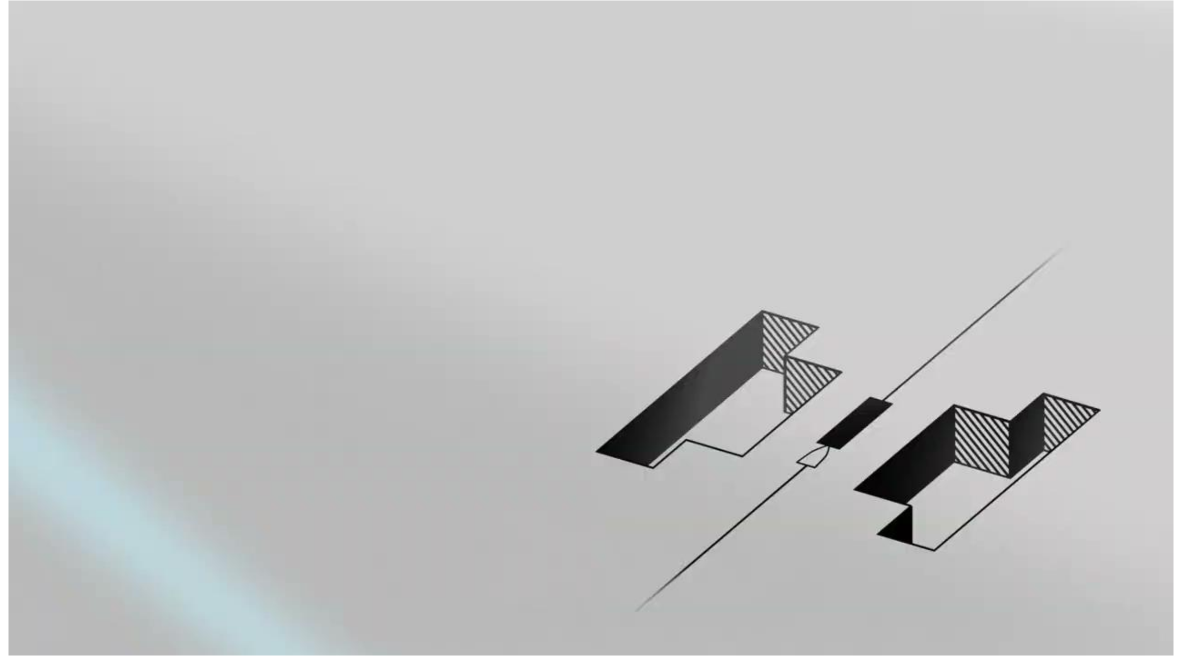


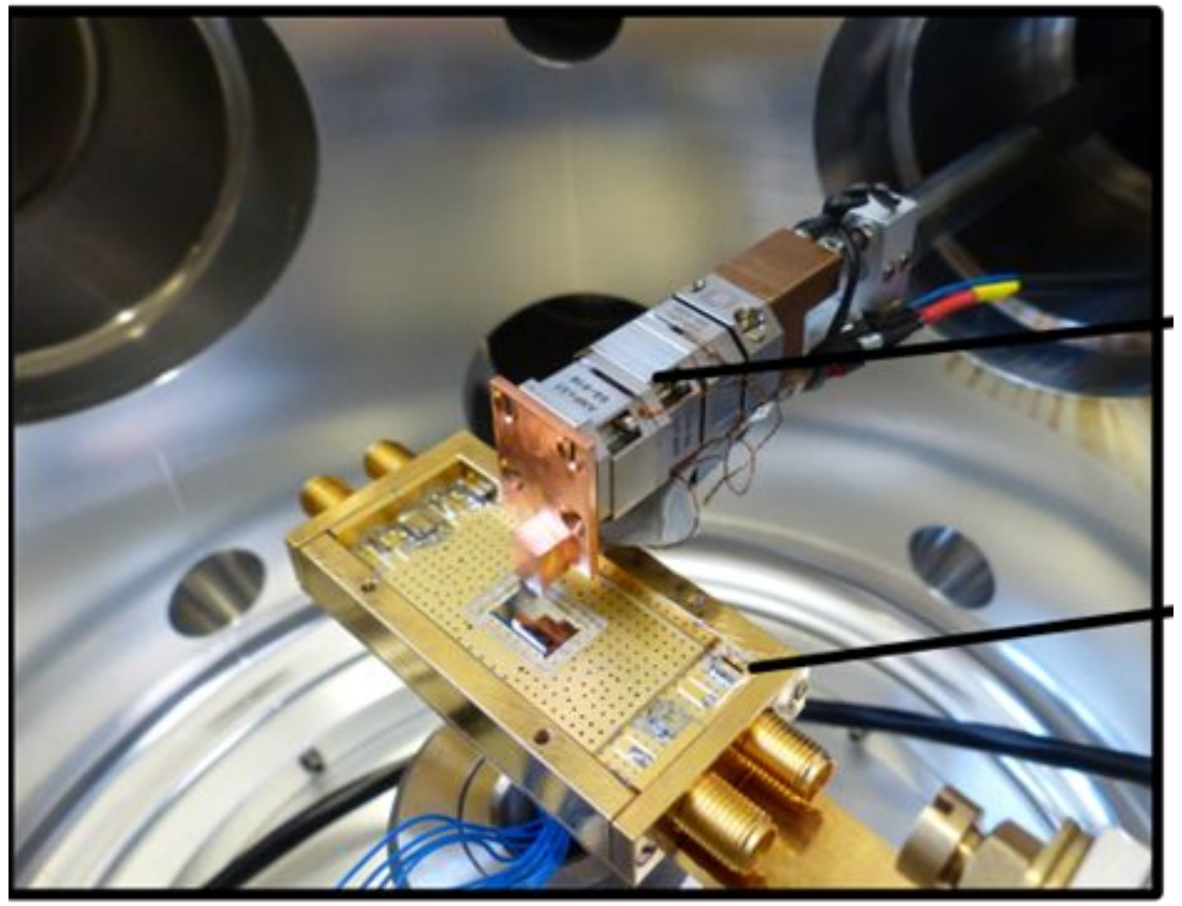


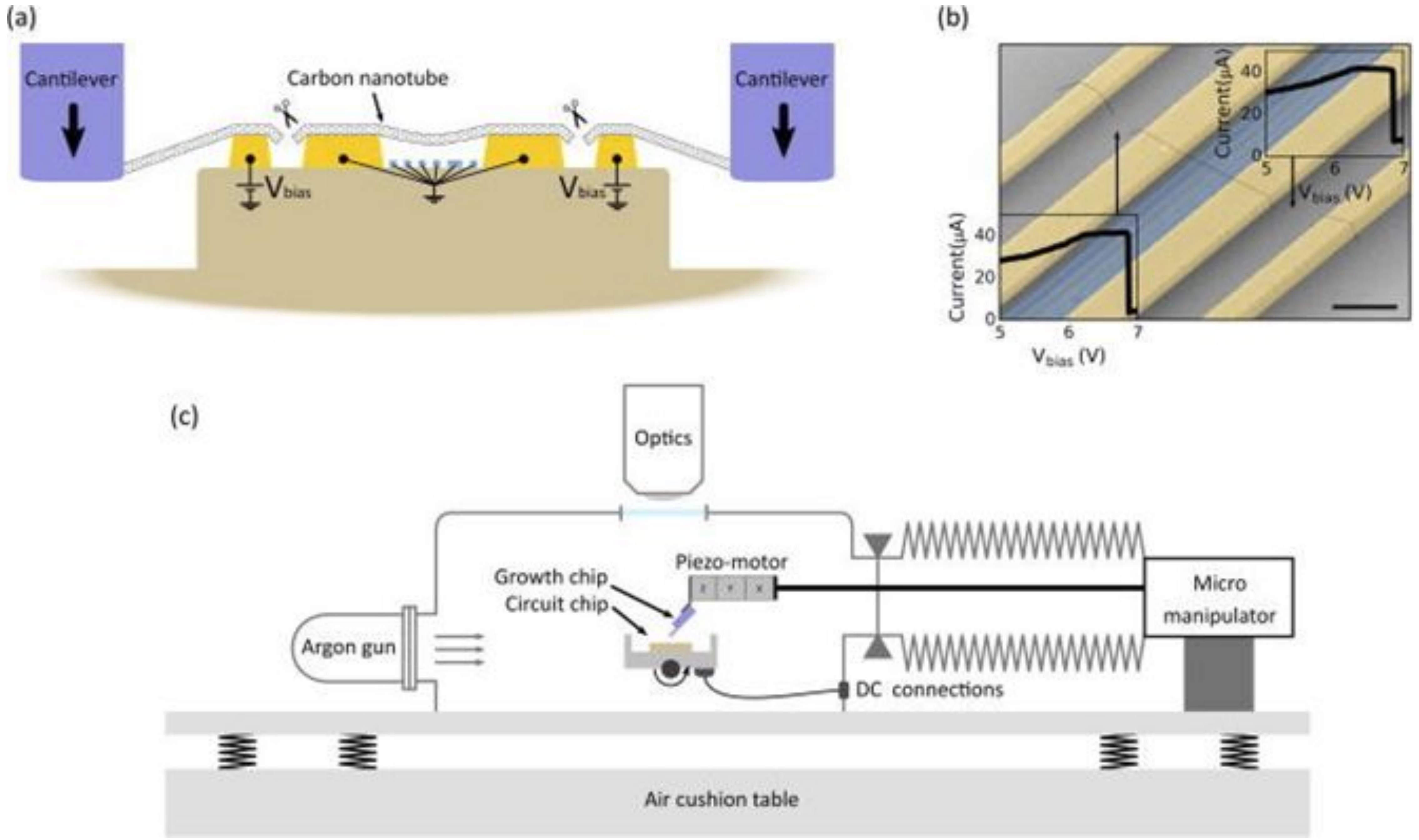
Internal measurements

C12

Nano-assembly: stapling carbon nanotubes on chips





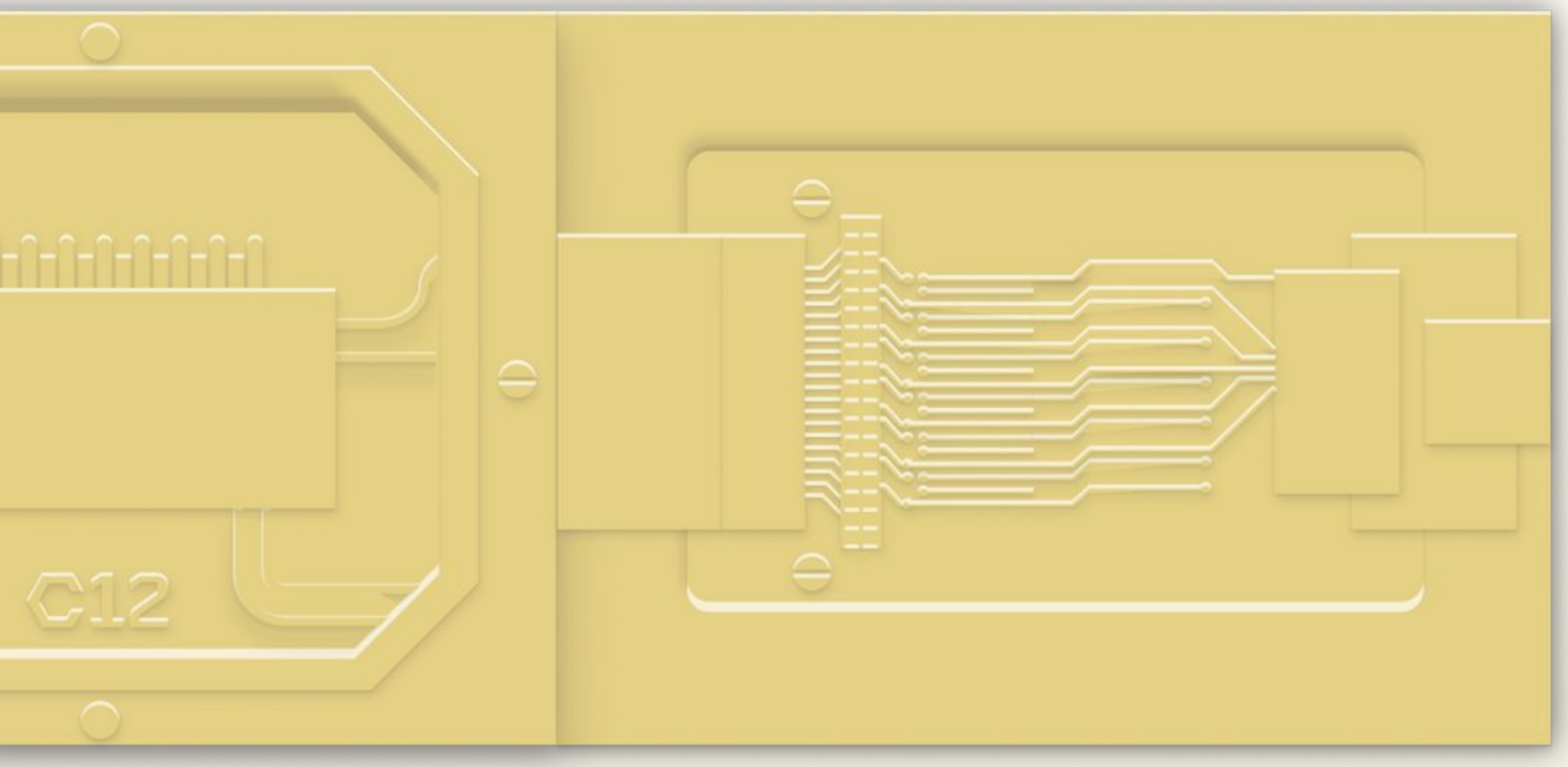


Cubaynes et al., Appl. Phys. Lett., 2020

Brief onsite virtual tour (facility inauguration, Oct 2023)

https://www.youtube.com/embed/9aWrTo1rOeU

C12, a unique approach to quantum computing



Carbon nanotube qubits

Ultra-clean, low-disorder material

Record coherence time for spin qubits with long distance connections

Unlocking reliable quantum operations

Quantum bus

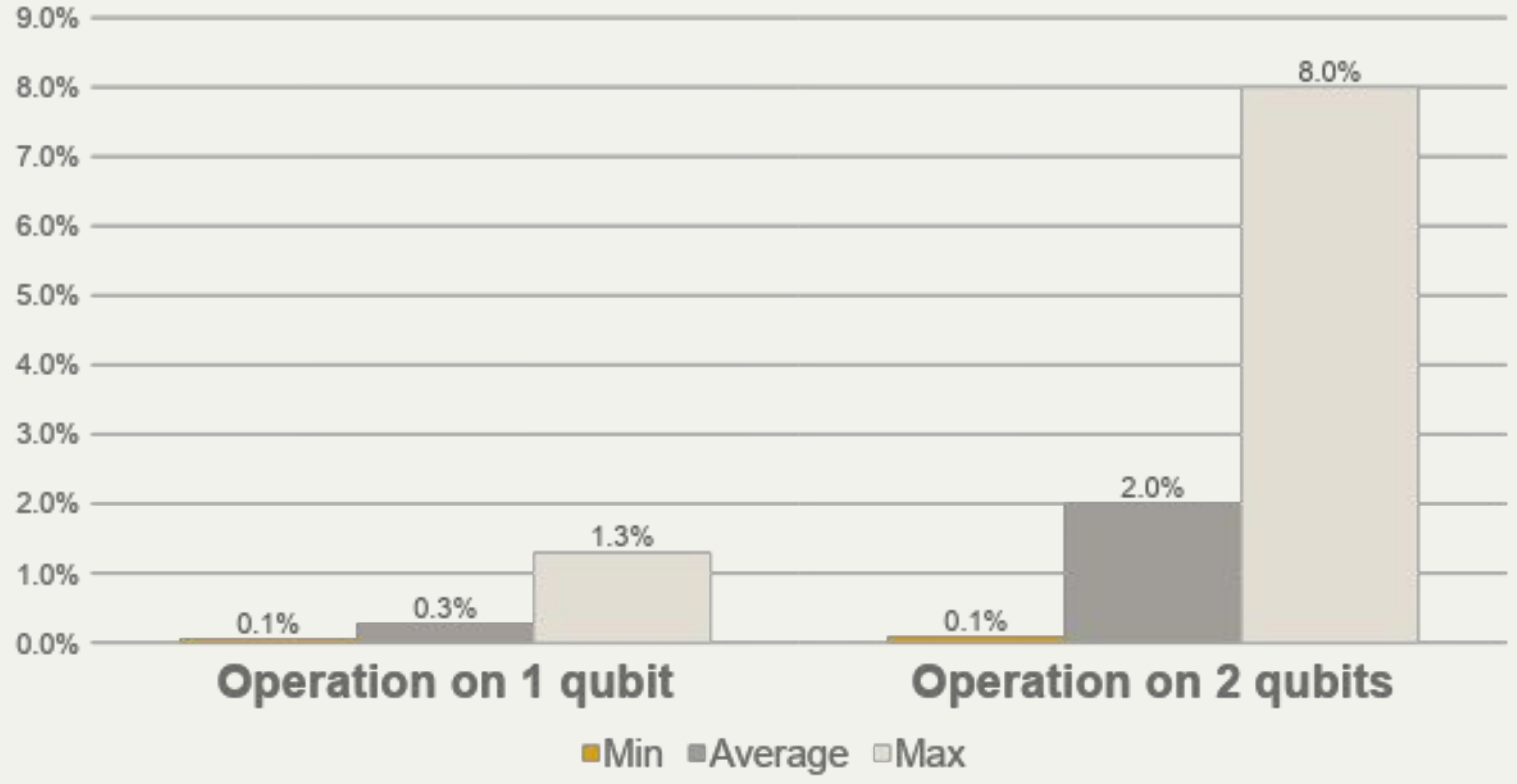
Fast, long-range 2-qubit gates with >99.9% fidelity (projected)

World-record interconnect performance

Enabling modular and high-connectivity qubit architectures

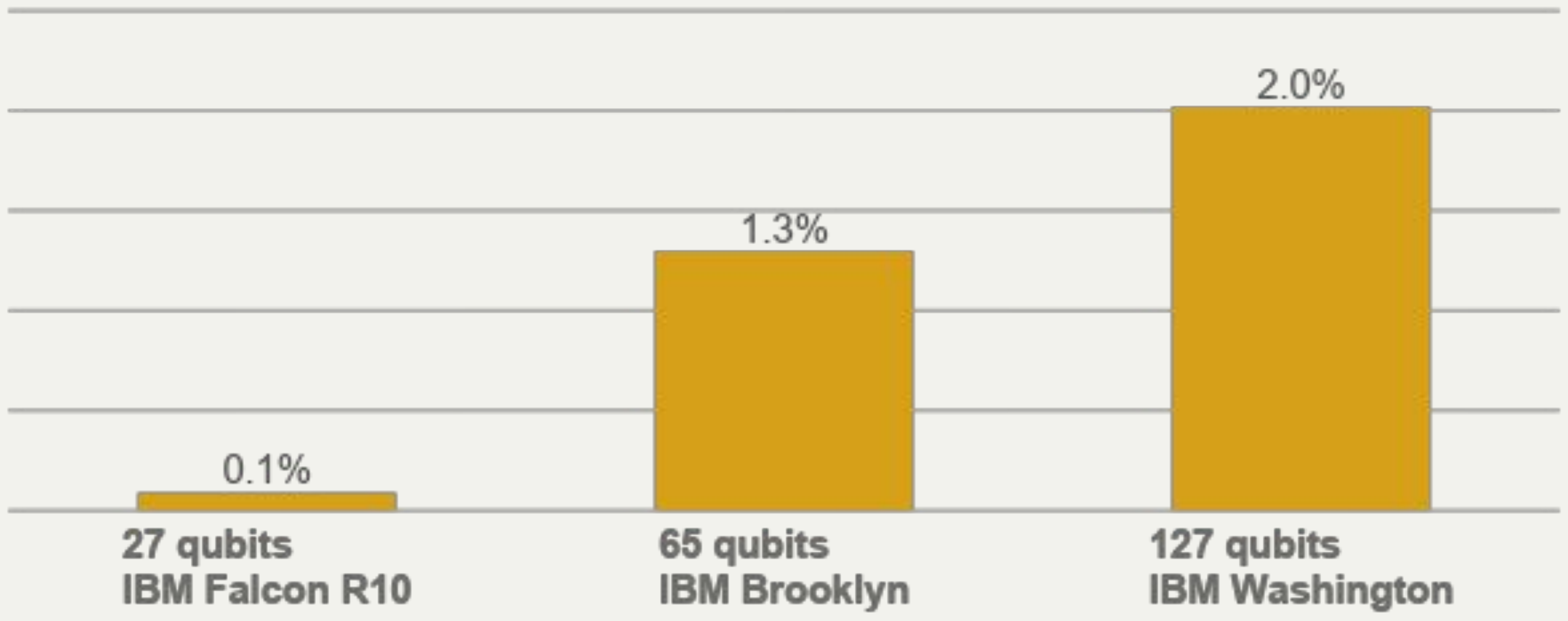
But high error rates of current approaches block the scalability, thus usability of this life-changing technology

Today's most developed chips have high error rates for operations on 1 and 2 qubits…



… and increasing the number of qubits can increase the error rate, leading to a lower number of operations achievable

IBM chips, error rate for operations on 2 qubits (%)



Max 1000 operations

Max 70 operations

Max 50 operations

Source: Olivier Ezratty, Understanding Quantum Technologies 2022

Software needs in the quantum computing

It’s not all quantum circuits!

There are many aspects to a quantum computer that require programming:

- Conducting quantum resource estimation to solve a given problem
- Hybrid classical/quantum algorithms for the actual quantum use cases
- Quantum circuit transpiling to optimize efficiency based on hardware characteristics
- Emulating quantum hardware as a test run
- Designing pulse-based algorithmic approaches to suppress quantum noise
- Designing efficient QPU calibration routines
- Accounting for calibration during operation, e.g. defective qubits, connectivity

It’s not all quantum circuits!

Not done yet! There’s also:

- Arbitrary GHz RF pulse generation to control qubit gates
- Interfacing with dilution refrigerator software for temperature control and monitoring
- Real-time, high-bandwidth quantum error correction control and syndrome decoding
- Precisely coordinating control and readout signal multiplexing
- Integrating with cloud compute for hybrid algorithms
- Supporting quantum hardware job scheduling, reporting via a cloud platform
- Post-processing computation outputs

It’s not all quantum circuits!

Each technology involves its own unique manufacturing processes that must be automated. In the C12 case we must:

- Script the complex nanofabrication processes
- Automate optical and electrical measurements during nanotube characterization and assembly
- Program sequences to mechanically control nanotube manipulation and deposition
- Design comprehensive factory test sequences beyond daily calibration

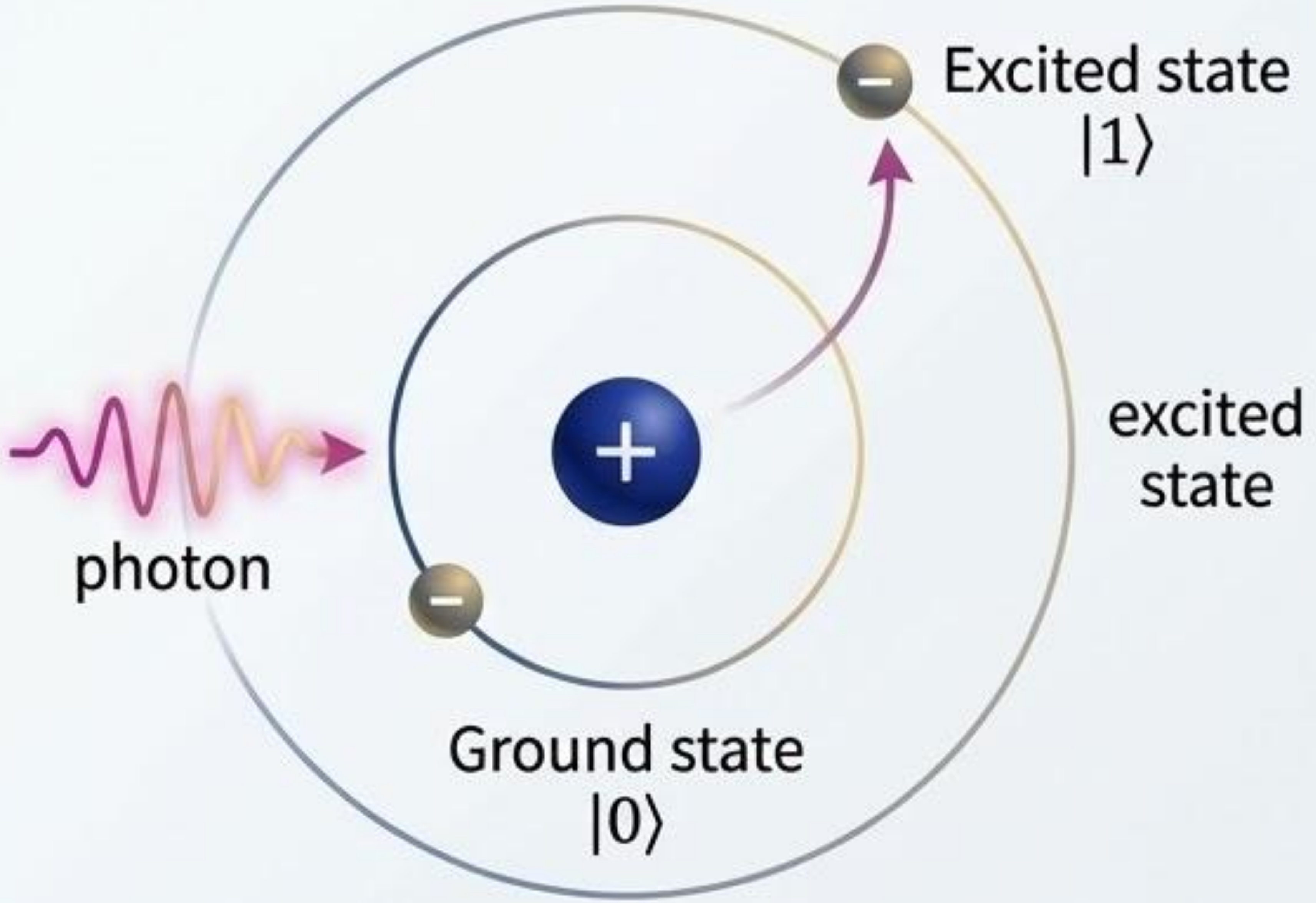
For other qubit types, you might be looking at extensive laser control, holographic optical trap calculation, acousto-optic deflector control...

Quantum computing overview

C12

Qubit (1)

- Qubits are the basic units of a quantum computation. They are quantum analogs of bits.
- **Hydrogen atom** example:
  - Ground state = $|0\rangle$
  - Excited state = $|1\rangle$
- The state of the electron in the hydrogen atom is described by the **wave function** χ.



Qubit (2)

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}^{\mathrm{T}}, \quad |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}^{\mathrm{T}}$$

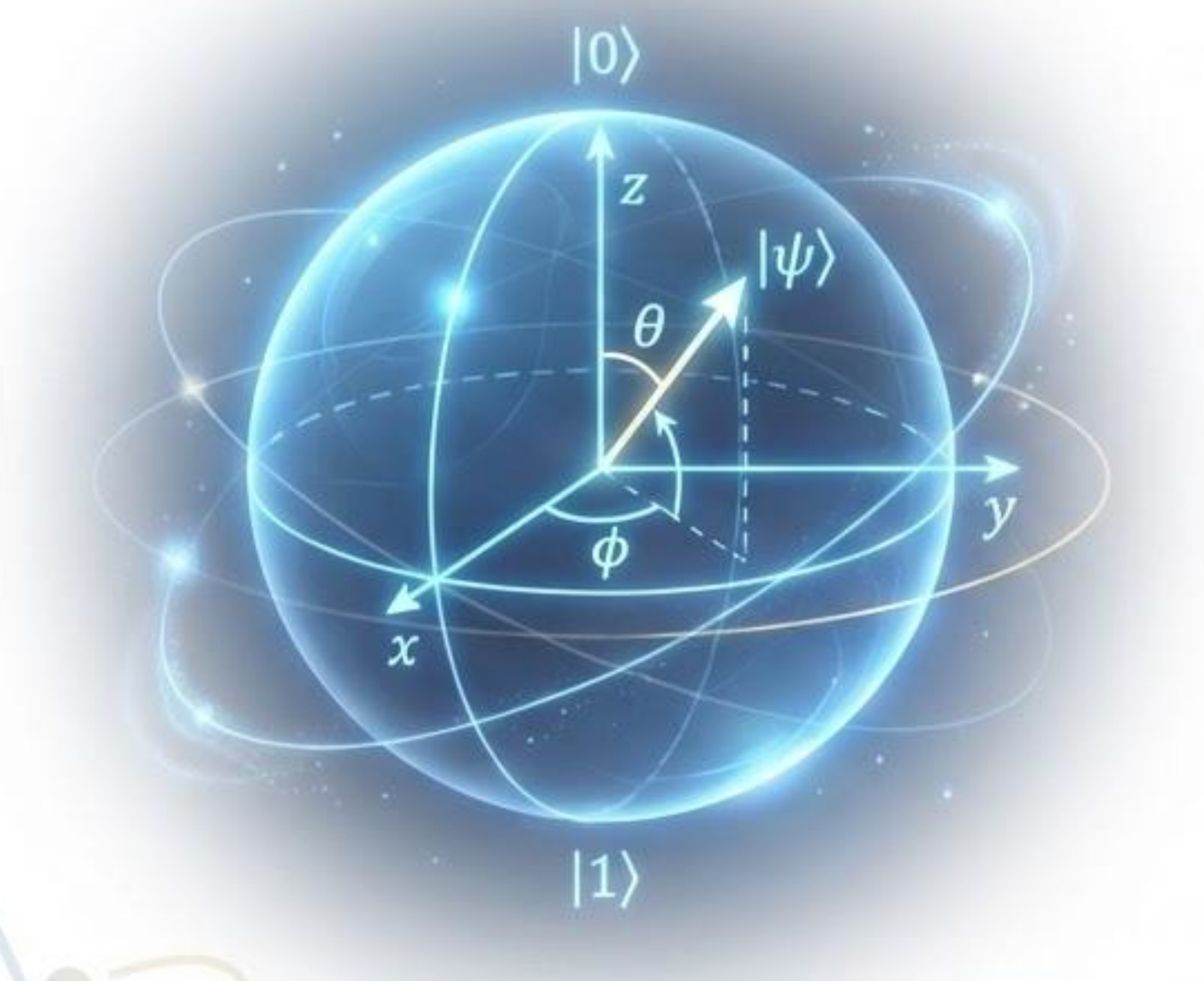
- Its quantum state is represented as a linear superposition of two orthonormal basis states $|0\rangle$ (ground) and $|1\rangle$ (excited state).
- α and β are the complex numbers, representing the probability that the electron is in the corresponding state. That's why:

$$|\alpha|^2 + |\beta|^2 = 1$$

$|\alpha|^2$ is the probability that the qubit is in the state $|0\rangle$

⚠ **Important:** Measurement disturbs the state! After the measurement, the state of the electron is changed!

Qubit representation(s)



- The **Bloch sphere** is a **unit** sphere where the state of a qubit is represented as a **point** on the surface of the sphere.

Sign basis: $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),\ |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

Can we have more than one qubit?

$$|0\rangle = |00\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}$$

$$|1\rangle = |01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

$$|2\rangle = |10\rangle = |1\rangle \otimes |0\rangle = \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}$$

$$|3\rangle = |11\rangle = |1\rangle \otimes |1\rangle = \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

$$\boxed{|v\rangle \otimes |w\rangle = \begin{pmatrix}\alpha\\\beta\end{pmatrix} \otimes \begin{pmatrix}\delta\\\gamma\end{pmatrix} = \begin{pmatrix}\alpha\delta\\\alpha\gamma\\\beta\delta\\\beta\gamma\end{pmatrix}}$$

- Generalize to a two qubit system using tensor product:

$$|\psi\rangle = \sum_{i=0}^{k-1} \sum_{j=0}^{l-1} \gamma_{ij} |ij\rangle = |i\rangle \otimes |k\rangle$$

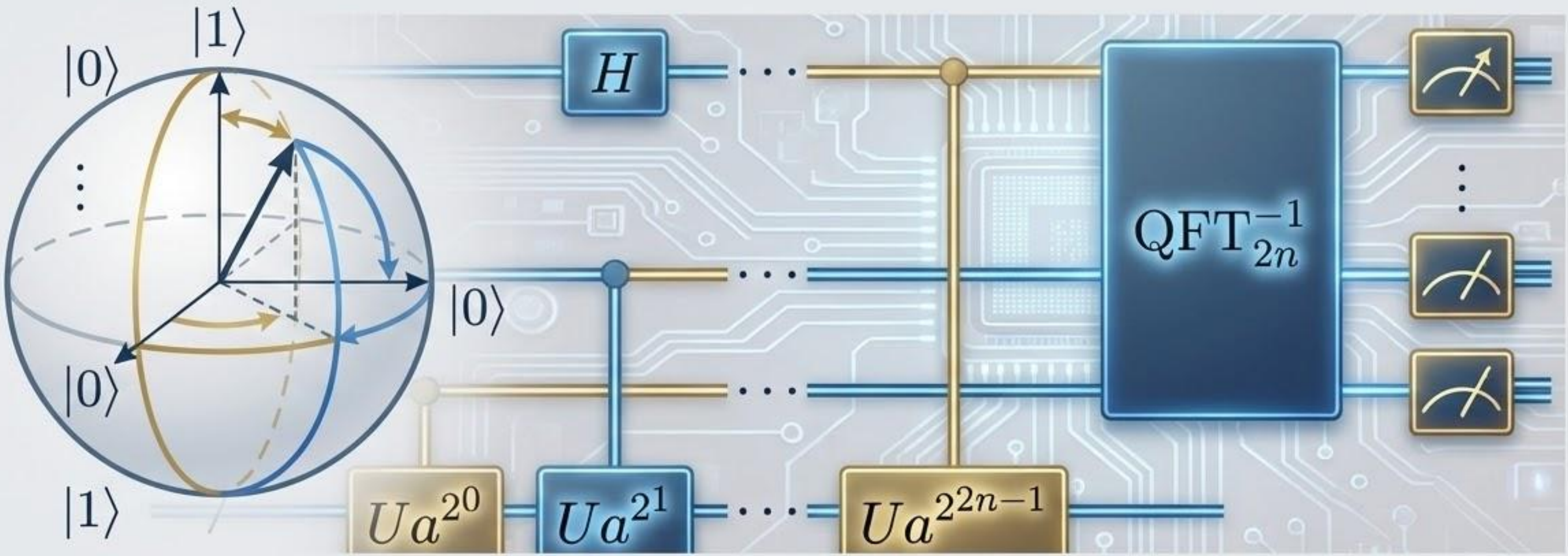
$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

- Entangled qubits?

C12

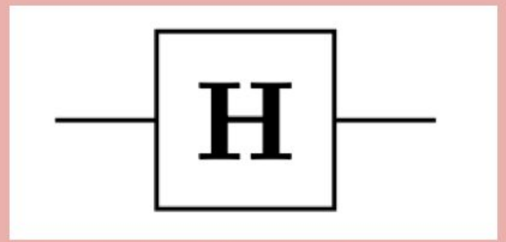
Quantum gates and circuits

- Quantum gates are used to manipulate quantum information => they operate on the qubits
- A qubit can be put in a desired superposition by applying quantum operations, representing rotations on the Bloch sphere.
- Quantum gates are combined to produce quantum circuits, and quantum algorithms



Hadamard

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

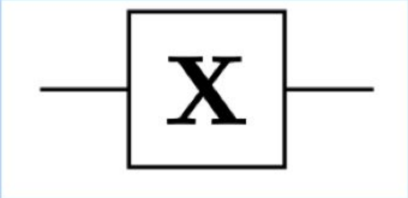


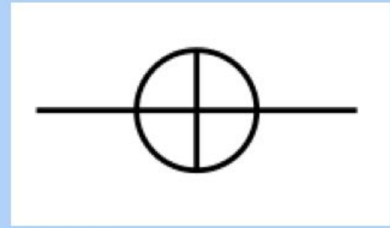
Rotation by π around $\pi/8$ axis in complex plane

Maps $|0\rangle, |1\rangle$ to $|+\rangle, |-\rangle$ basis

NOT/Pauli-X

$$NOT = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

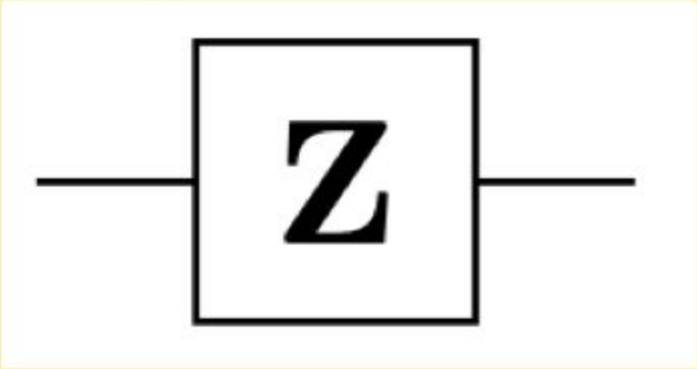




Real and symmetric → $H = H^\dagger$

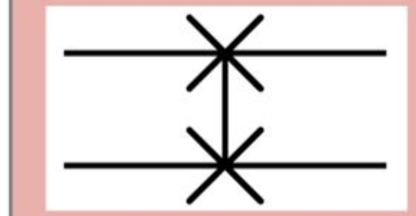
Phase Flip (Z-gate)

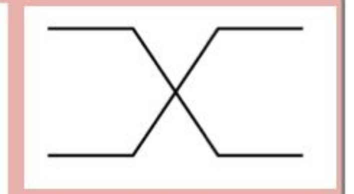
$$Z = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$



SWAP

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

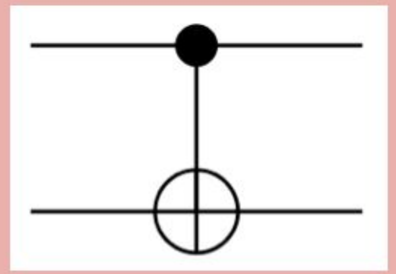




Swaps two qubits

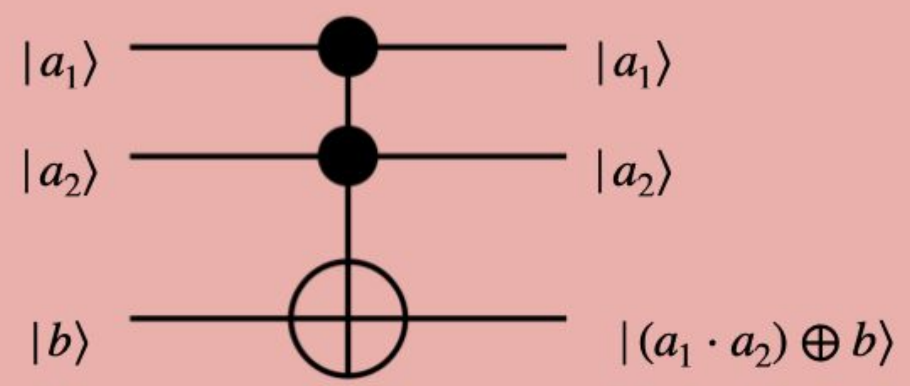
CNOT

$$CNOT = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$



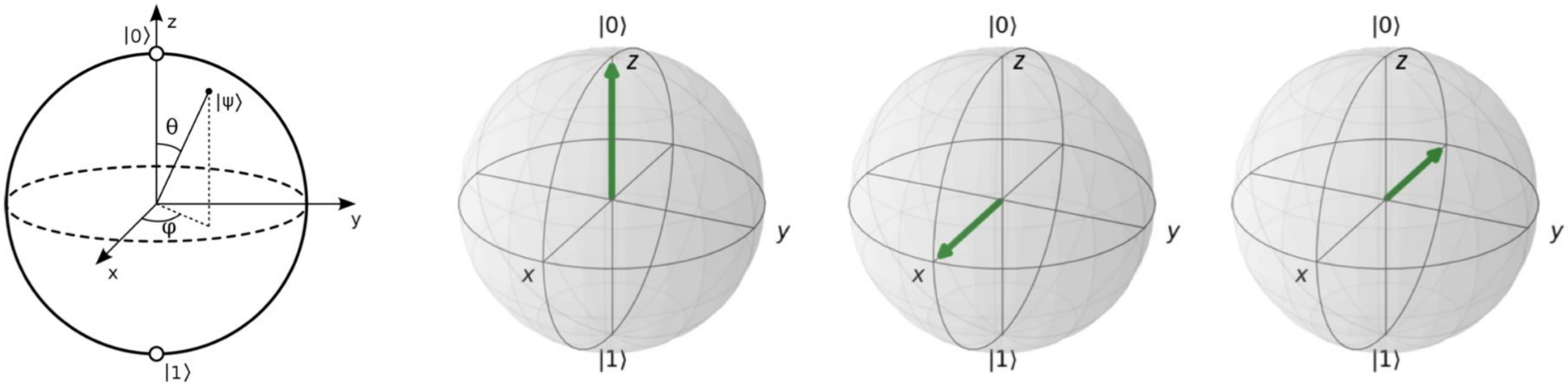
Flips 2nd (target) qubit iff the first qubit is $|1\rangle$

Toffoli or CCNOT

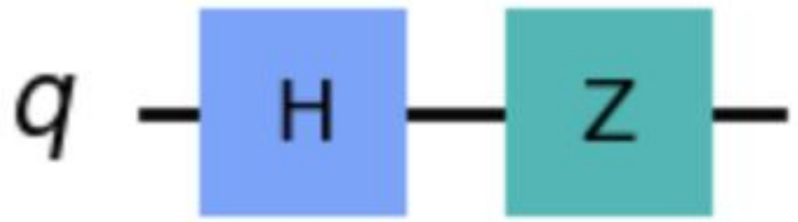


If the first two bits are 1, the third bit is inverted

Can be generalized to an n-bit version



Initially, the qubit is in the ground state. Then, it first gets manipulated by an *H* gate in an equal superposition state, then by a R_z gate.



Universal gate set

The Challenge

- To build a programmable quantum computer, we need it to perform *any* series of quantum operations.
- However, we cannot physically implement *all* possible quantum gates directly.

The Solution: A Universal Gate Set

- Define a small, specific subset of fundamental quantum gates.
- Any other arbitrary quantum operation can be efficiently approximated using only gates from this basic set.

C12 basis gate set in matrix representation

$$RX(\theta) = \exp\left(-i\frac{\theta}{2}X\right) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

$$RY(\theta) = \exp\left(-i\frac{\theta}{2}Y\right) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

$$RZ(\lambda) = \exp\left(-i\frac{\lambda}{2}Z\right) = \begin{pmatrix} e^{-i\frac{\lambda}{2}} & 0 \\ 0 & e^{i\frac{\lambda}{2}} \end{pmatrix}$$

$$\mathrm{iSWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

C12 basis gate set in circuit symbology

Rx

```
       ┌───────┐
q_0: ┤ Rx(θ) ├
       └───────┘
```

Ry

```
       ┌───────┐
q_0: ┤ Ry(θ) ├
       └───────┘
```

Rz

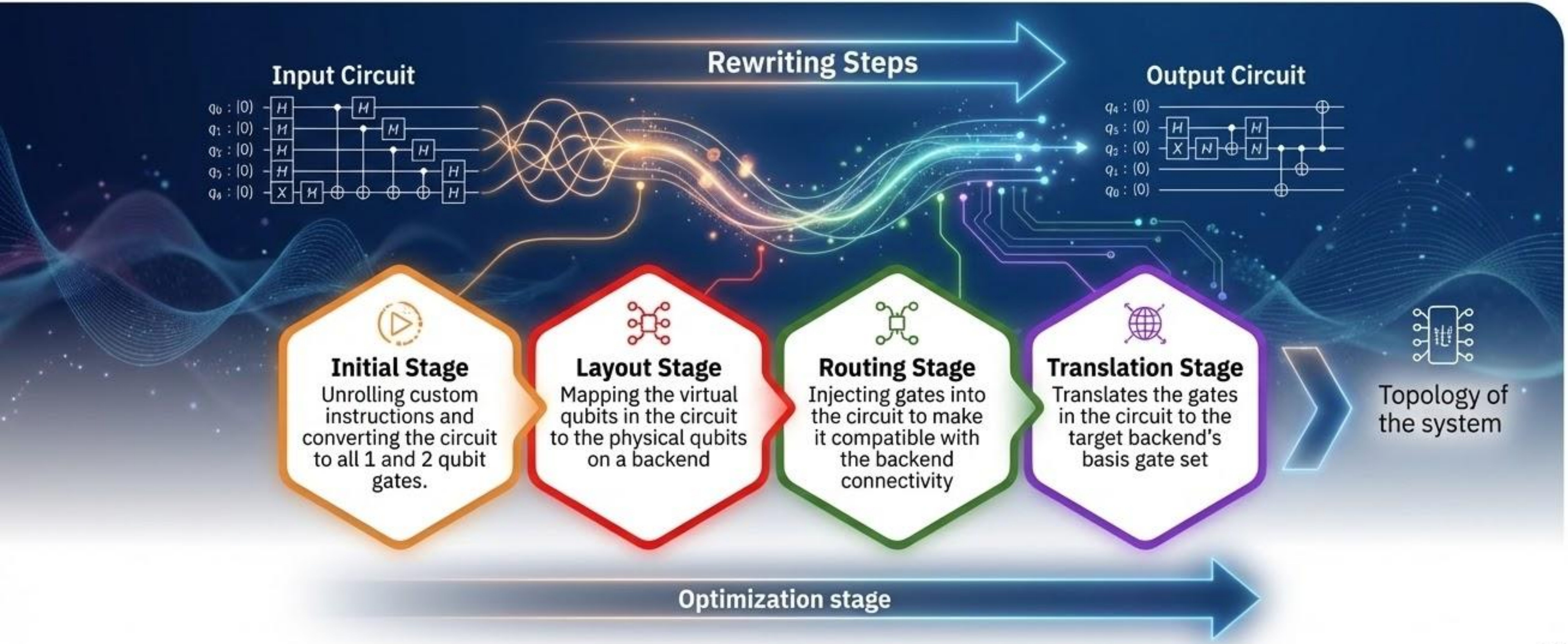
```
       ┌───────┐
q_0: ┤ Rz(λ) ├
       └───────┘
```

```
q_0: ─⊗─
       │
q_1: ─⊗─
```

iSWAP

Transpilation

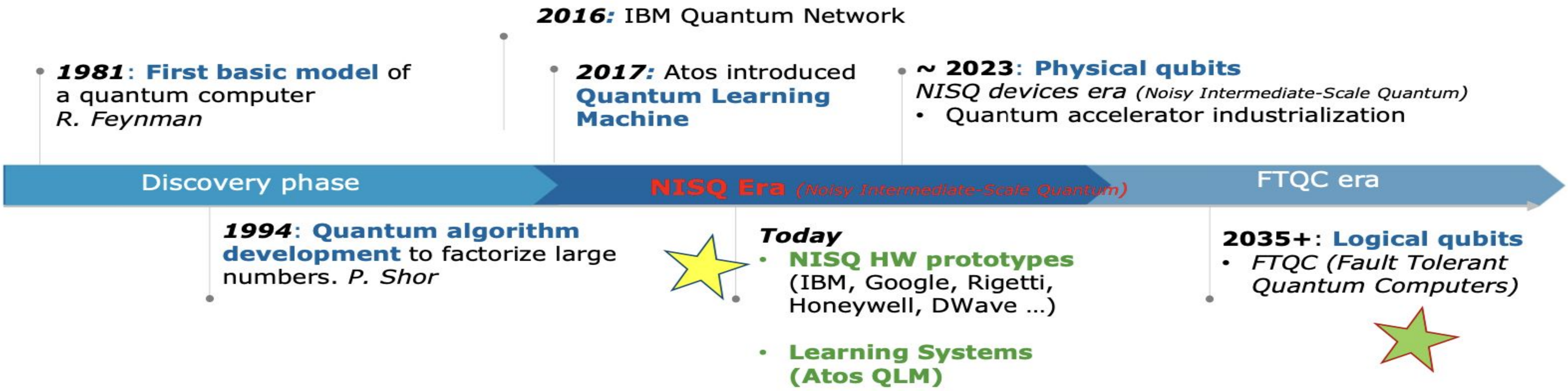
- Transpilation is the process of rewriting a given input circuit to match the topology of a specific quantum device, and/or to optimize the circuit for execution on present-day noisy quantum systems.



C12

Current state

- **Quantum advantage** - it consists in demonstrating that a quantum device can solve a problem faster than classical computers.
- **Quantum supremacy** - to demonstrate that a programmable quantum device can solve a problem that a classical computer can not due to its complexity



... and for (many) other reasons increasing the number of qubits and keeping the "quantum effects" (or coherence) is a key challenge:
➔ **NISQ Era** ("Noisy Intermediate Scale Computing")

... **and as a consequence far from demonstrating a "Quantum Advantage"**

Credit: https://www.roadef.org/jfro/static/files/slides_330.pdf

Qiskit

https://github.com/c12qe/epita/blob/master/EPITA_Introduction_to_Qiskit.ipynb

Exercises

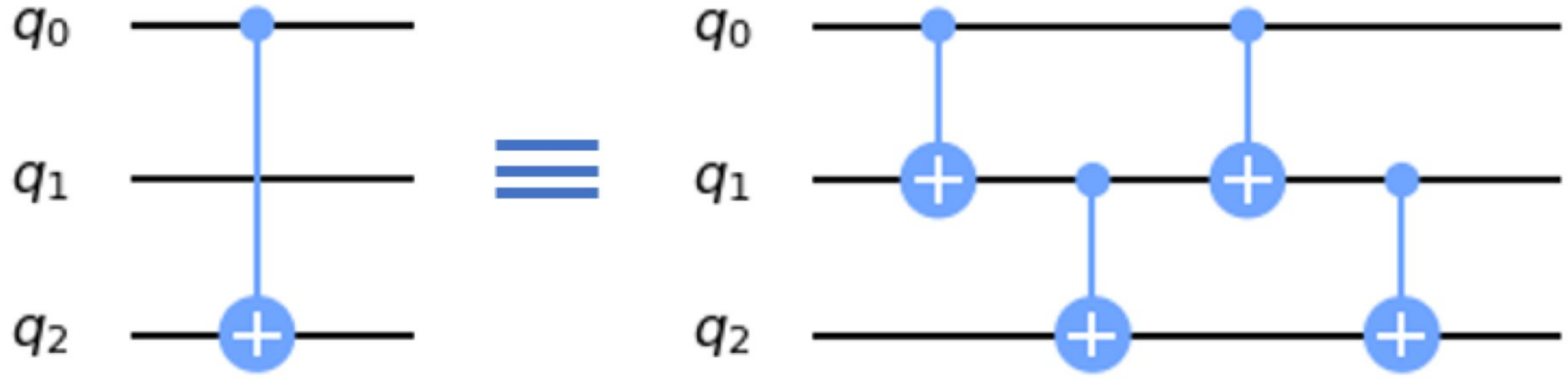
1. Create a one qubit circuit and apply Hadamard gate to create |+> and |-> states. After that use statevector to print the state, and to print the probabilities. Than, use Aer statevector simulator to get the statistics for 10000 shots. Compare the results obtained for |+> and |->. Plot the obtained statistics.

2. Create a circuit **H-X-H**, and perform the same analysis as in the previous exercise. What do you think the probability is going to be ?

3. Implement the QFT for 3 qubits using H gates, controlled phase torations and SWAP gate. Apply this QFT circuit to the input state |001>. Use the statevector and Aer simulator the print the resulting amplitudes. Run the inverse QFT and recover the initial input state.

Exercises

4. Show that the following two circuits are equivalent (on paper as well as using Qiskit). Multiple solutions are possible!



C12 quantum emulator

CALLISTO

Includes:

- Charge noise
- Phononic noise
- Relaxation through the resonator

DGA THALES DASSAULT AVIATION Air Liquide

Simulating the real world

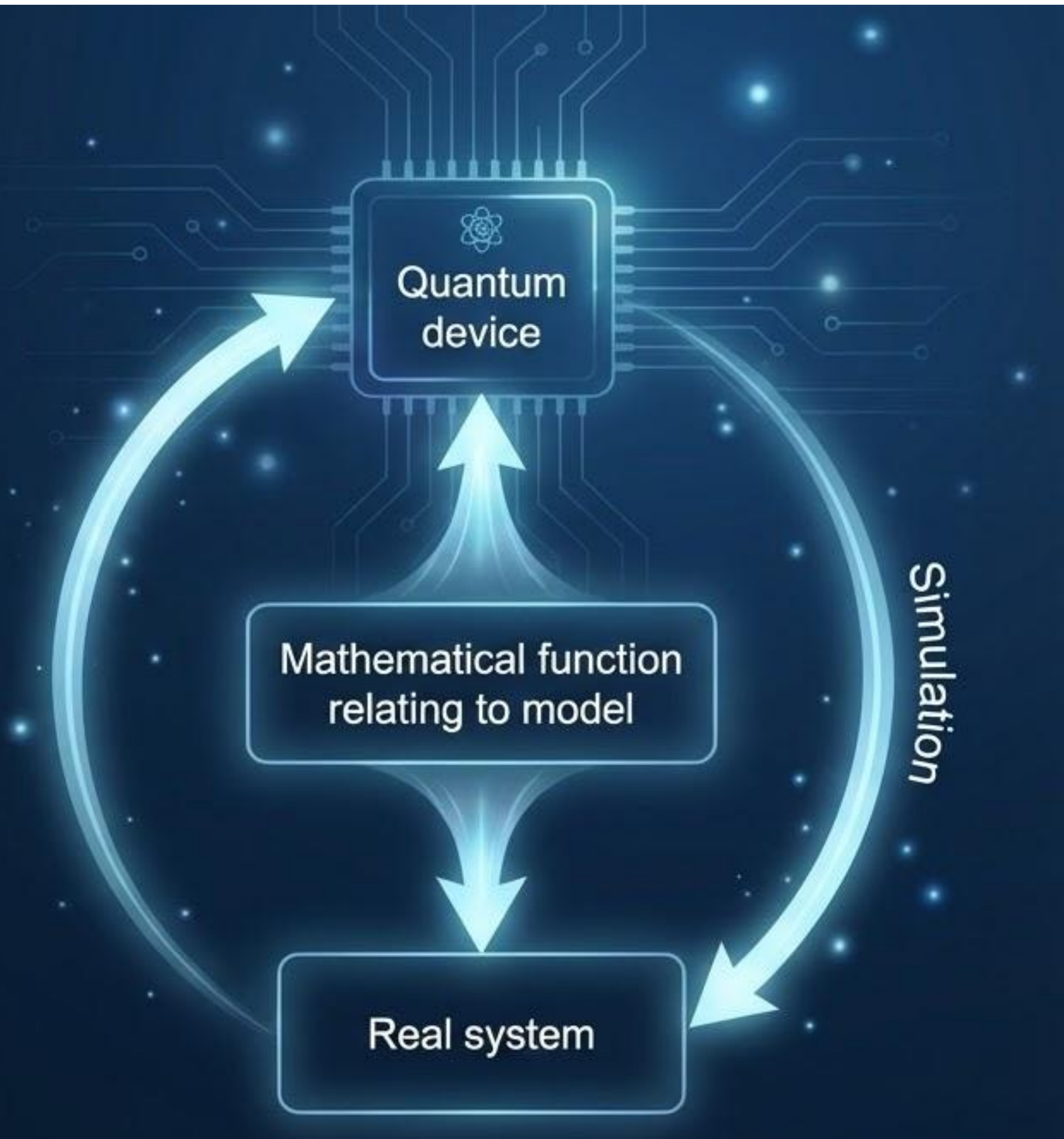
The simulator reveals the information about the mathematical model that describes the physical system of interest.

What is the purpose of the simulators

- to reveal the information about the real system
- to reveal the information about a model and compare it to the system of interest

Simulators vs. emulators ?

Do we need simulators?



Credit: Joshnson et al. (2014)

Quantum emulators

- Are they possible?
  - A quantum computer does not violate the Turing thesis (or Church-Turing thesis).
- Difficulties
  - The huge amount of memory required to store the explicit state of the quantum system.
  - Quantum states are described by a number of parameters that grows exponentially with the system size, and also the number of necessary operations increase with the system size
  - N qubit system -> 2^N x 2^N matrix in the memory
  - 16PB of memory for 50 qubit system!

Source:
https://thequantuminsider.com/2022/05/27/quantum-computing-tools/

1. PROJECTQ

2. CIRQ

3. Q-CTRL PYTHON OPEN CONTROLS

4. QUANTIFY

5. INTEL QUANTUM SIMULATOR

Qiskit

Atos QLM Atos

C12

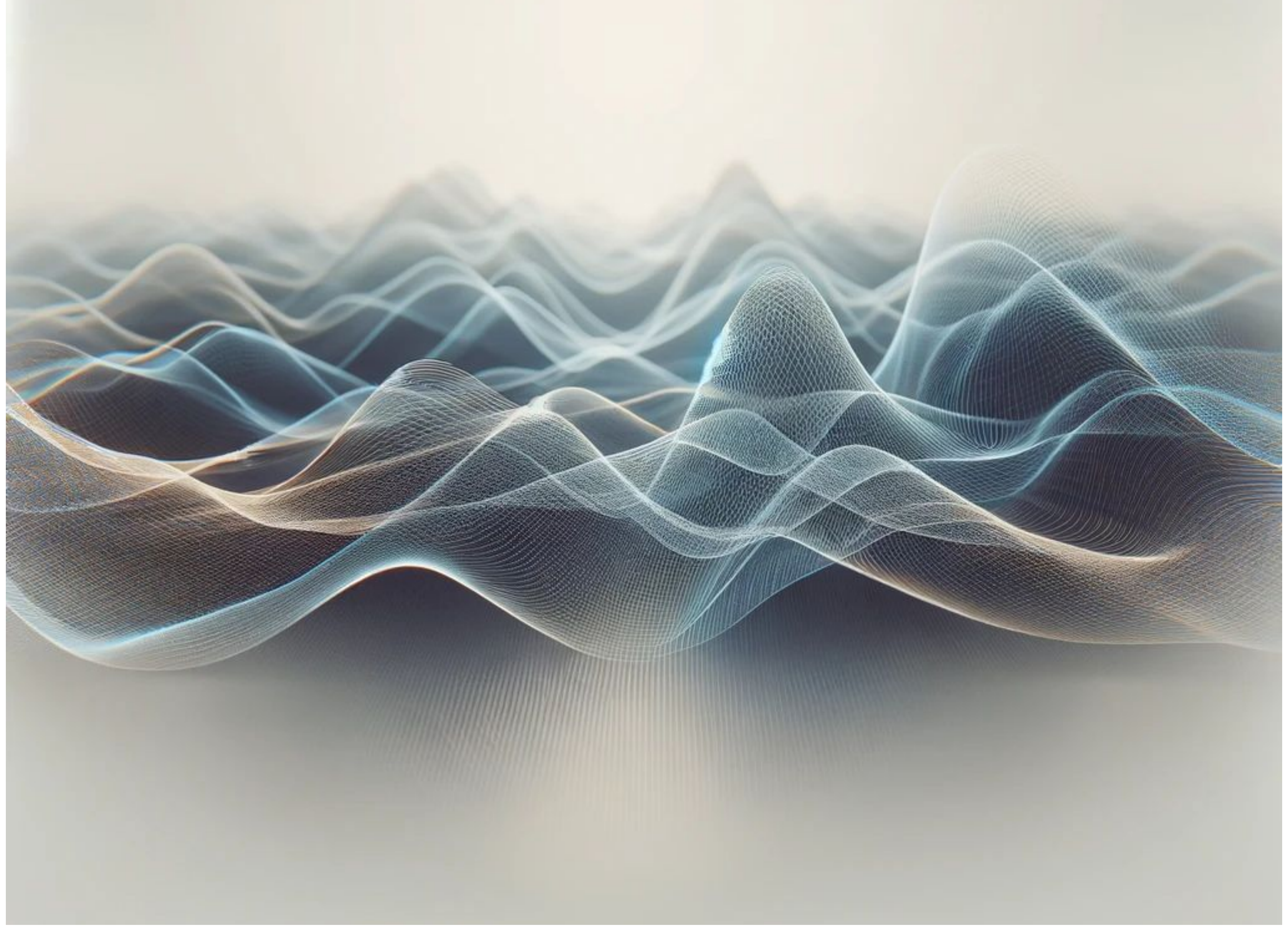
Our emulator mimics our next-generation quantum computers…

All quantum phenomena in the emulator

Direct link between the real quantum chip & algorithmic runs

Compatible with Qiskit, TKET and OpenQASM

…testing them in a realistic environment with the exact noise model



- Charge noise
- Phonon noise
- Relaxation through the resonator

It provides tools to create and run quantum circuits and to compare both ideal and realistic outcomes

Version 1

- 13 qubits
- High-fidelity gates
  - **RX**: 99.9989%
  - **RY**: 99.9989%
  - **RZ**: 100.0%
  - **iSWAP**: 99.6446%
- Mid-circuit measurement
- Noisy initialization

Version 2

More qubits, more advanced features to come…

<u>How to make a quantum simulator?</u>

Time evolution of a **closed** quantum
system is described by Schrodinger e

$$i\hbar \frac{d|\psi\rangle}{dt} = H|\psi\rangle$$

H is Hermitian operator - Hamiltonian of the closed system.

Time evolution of the closed system is then represented by the unitary operator U:

$$U(t) = e^{-iHt}.$$

$$|\psi'\rangle = U|\psi\rangle .$$

Time evolution of a **open** quantum system is described by Lindblad eq.

$$\frac{d\rho}{dt} = -\frac{i}{\hbar}[H,\rho] + \sum_j \left[2L_j\rho L_j^\dagger - \{L_j^\dagger L_j, \rho\}\right]$$

Density matrix

$$\rho \equiv \sum_i p_i |\psi_i\rangle\langle\psi_i|.$$

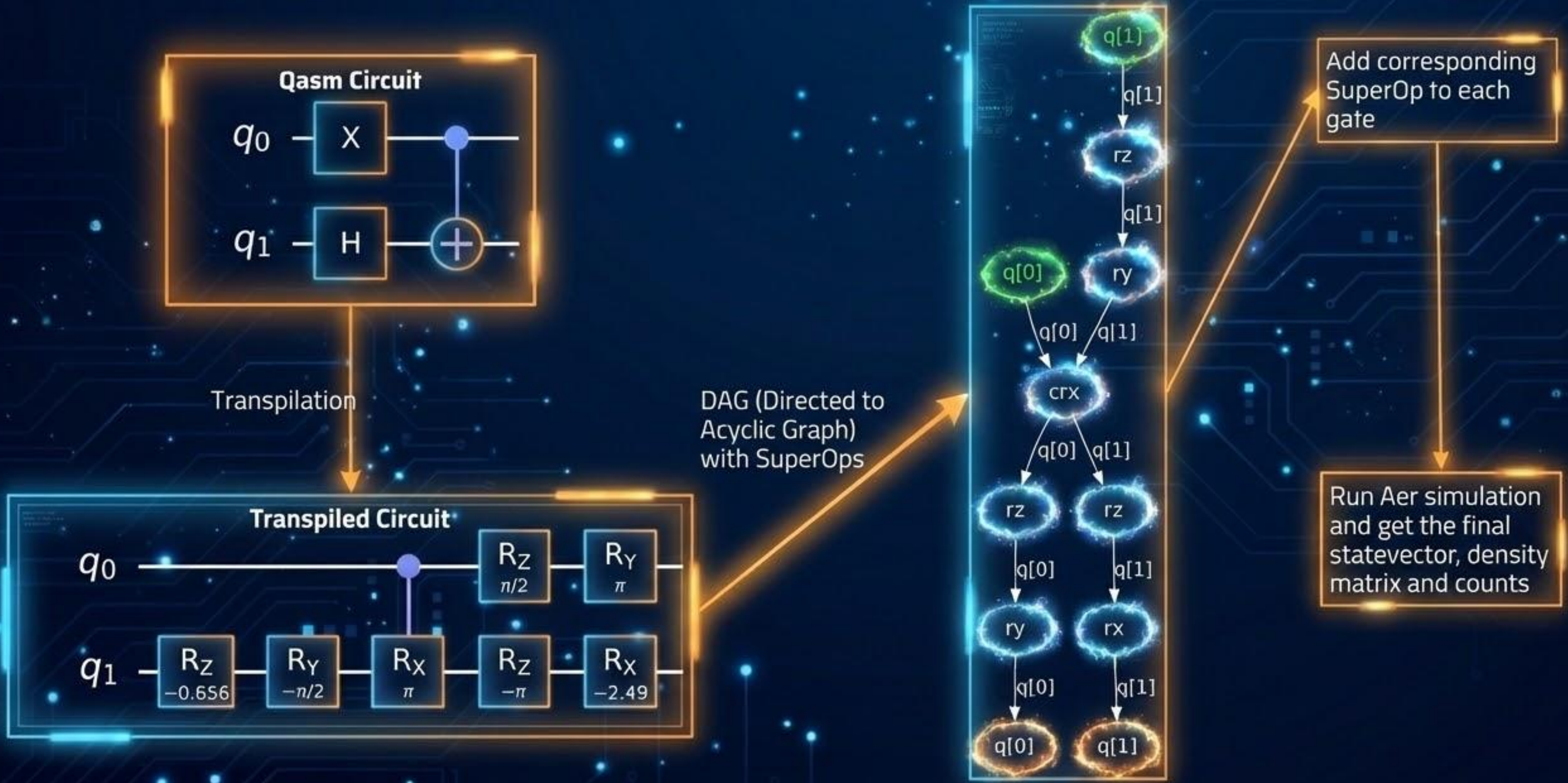
Different tools to describe the time evolution of a quantum systems:

- Superoperators (We use them in Qiskit)
- Choi matrices
- Kraus operators

C12


OpenQASM representation of the circuit
Circuit transpilation
Transformed circuit to basis gate set (to be executed on the QC)
Change each gate by adding the corresponding superop
C12 NOISE MODEL
Noisy circuit
Running the Qiskit's Aer simulator to get the results
Results

Example for adding a noise for the one-qubit gate



Open QASM

Open QASM

- Open Quantum Assembly Language (https://openqasm.com/)
- The language was first described in 2017 by IBM (Qiskit).

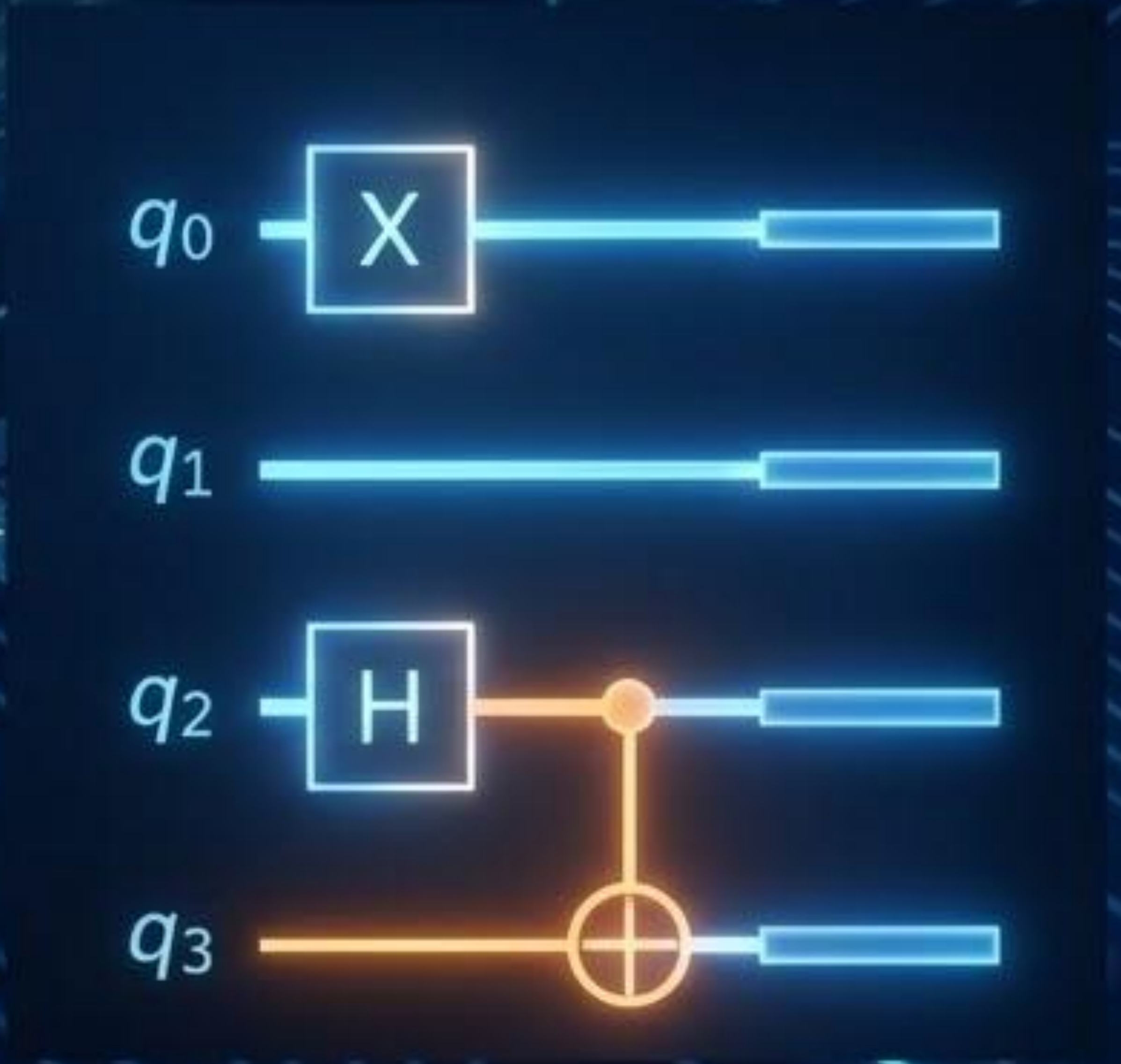
- A low-level language that allows quantum algorithms to be expressed in a way that is independent of the specific hardware being used.

Version & File with gate specifications

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[4];
h q[2];
cx q[2],q[3];
x q[0];
```

4 Quantum registers

Circuit



Add noise (add superops)

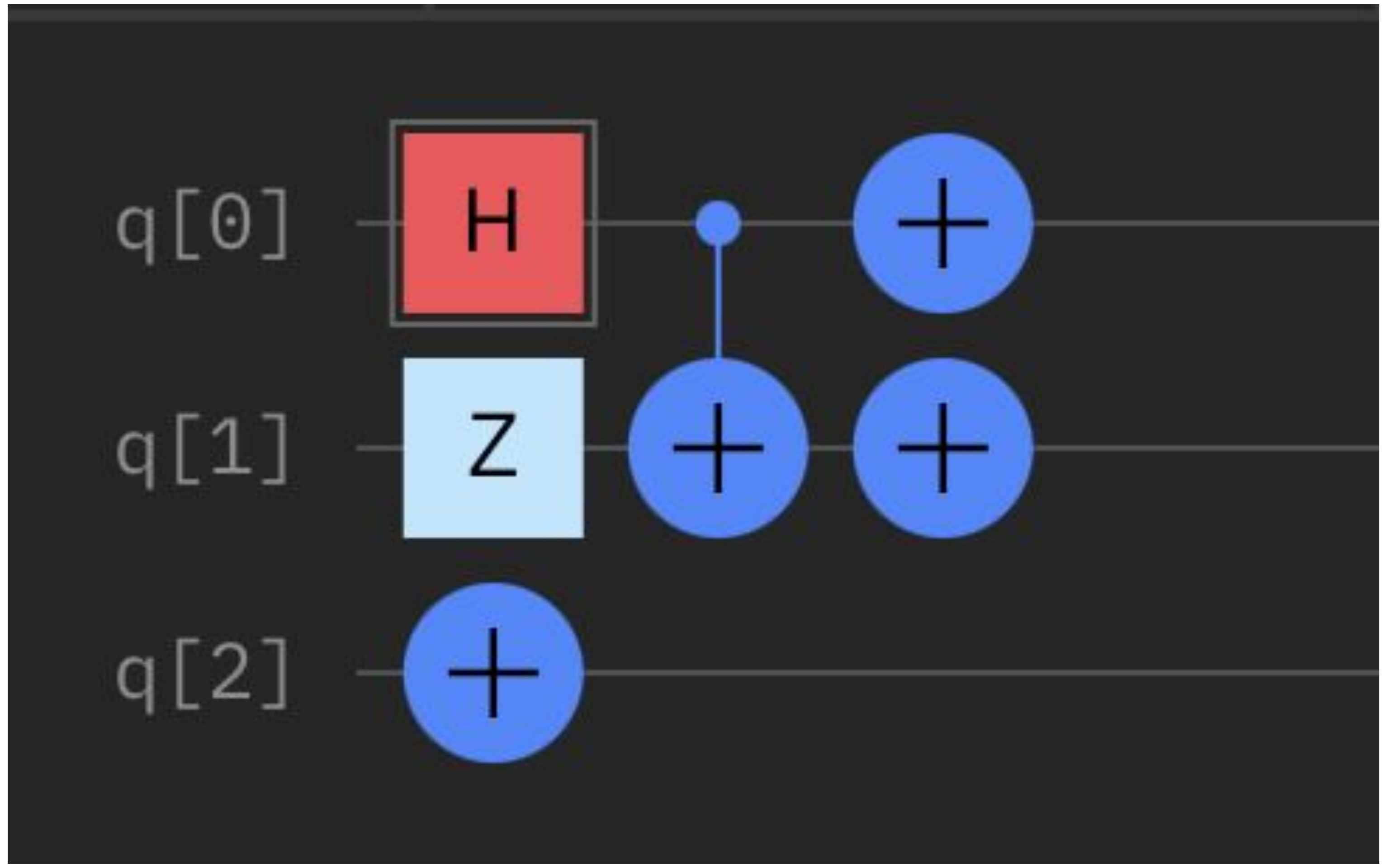
- Qiskit passes are used to go through each gate inside the circuit and depending on the gate type the corresponding calculated superop will be returned instead of the default one (without the noise)

```
def _get_noisy_circuit(self, circuit: QuantumCircuit, transpile=True) -> QuantumCircuit:
    assert circuit is not None

    if self.noisy:
        transpiled_circuit = compiler.transpile(circuit, basis_gates=self.basis_gates) if transpil
        dag = circuit_to_dag(transpiled_circuit)
        dag = self.local_noise_pass.run(dag)
        noisy_circuit = dag_to_circuit(dag)
        return noisy_circuit
    else:
        return circuit
```

Exercises

1. Create a custom TrnasformationPass that will replace all X gates with H-Z-H “cutstom” gate. Test it on the circuit given below.



C12 simulator architecture

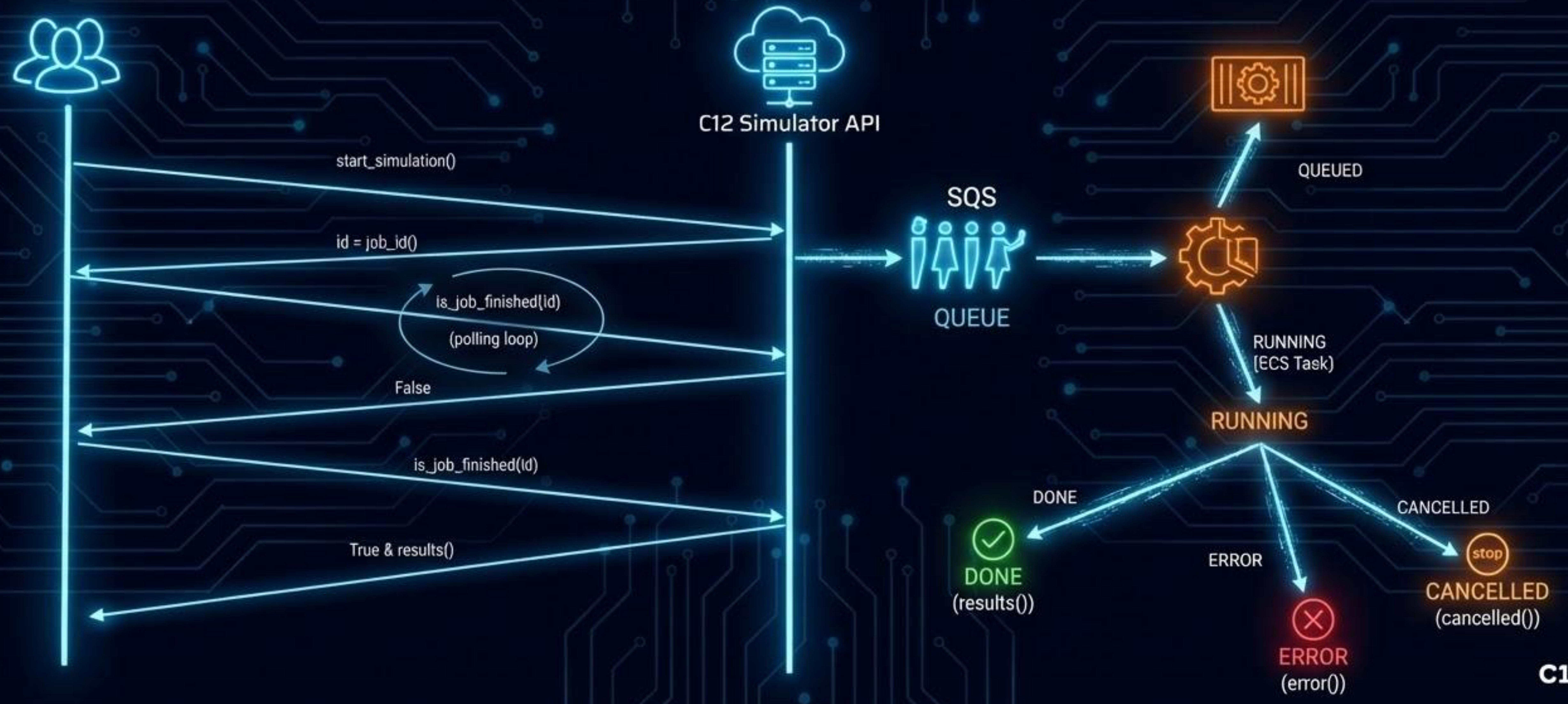
OVHcloud

C12 SYSTEM ARCHITECTURE

- Memory usage -> Powerful instances -> Larger cost -> **ECS** (Elastic Container Service)
- Number of users that need to be served -> **Queue** (First In First Out) -> **SQS** (Simple Queue Service)
- C12 simulator API
- Terraform is used as a language to describe the system organization and structure (IaC)
- **AWS CodePipeline** (CI/CD)

- C12 simulator API
- Client - server architecture

- **Python:** Fast API, Celery, Pydantic, Qiskit, pyTket
- **IaC:** Terraform, Ansible
- **CI/CD:** AWS CodePipeline



C12

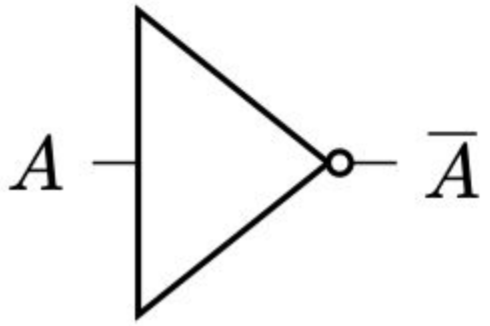

CALLISTO

Quantum adder algorithm example

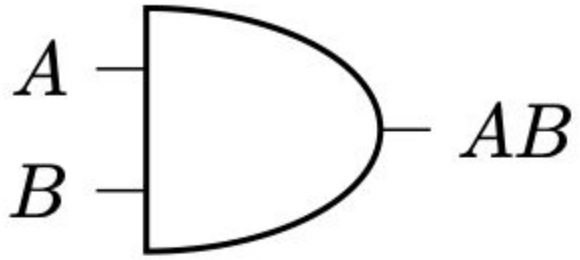
- Create a small example that will do bit addition using quantum states

  Exercise: Create a classical & quantum adder and try to test it using the C12 emulator Callisto.
  Compare the results obtained from the previous exercises and the ones obtained on the Callisto.

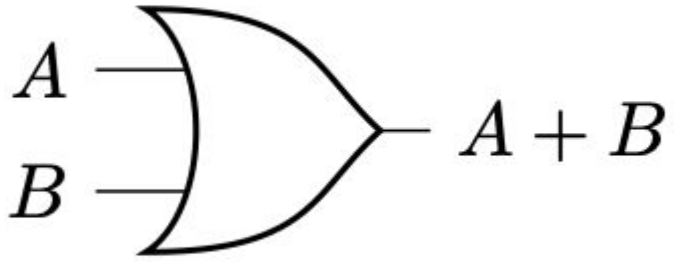
Logic gates



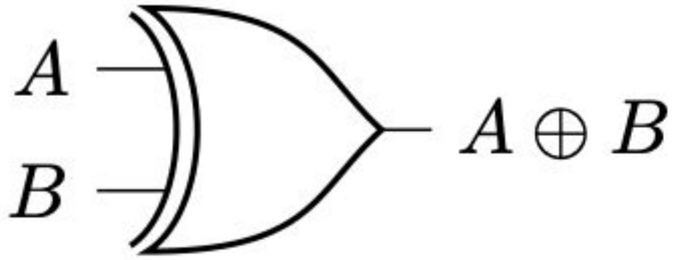
A	$\overline{A}$
0	1
1	0



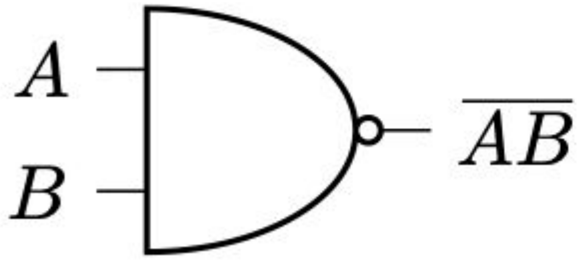
A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1



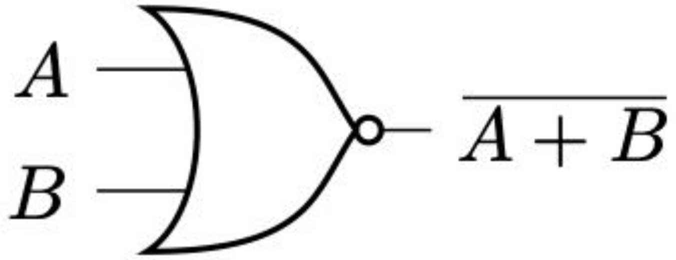
A	B	$A+B$
0	0	0
0	1	1
1	0	1
1	1	1



A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

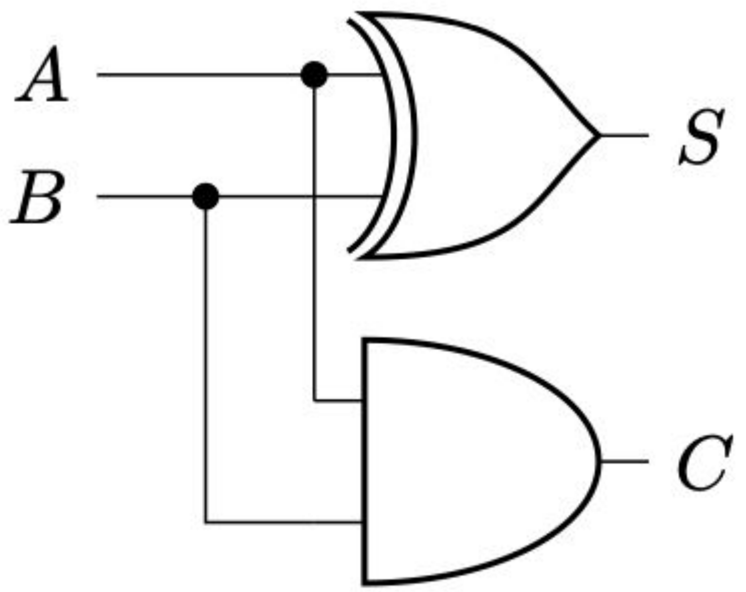


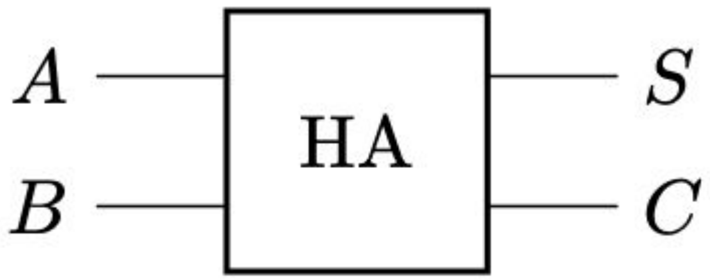
A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0



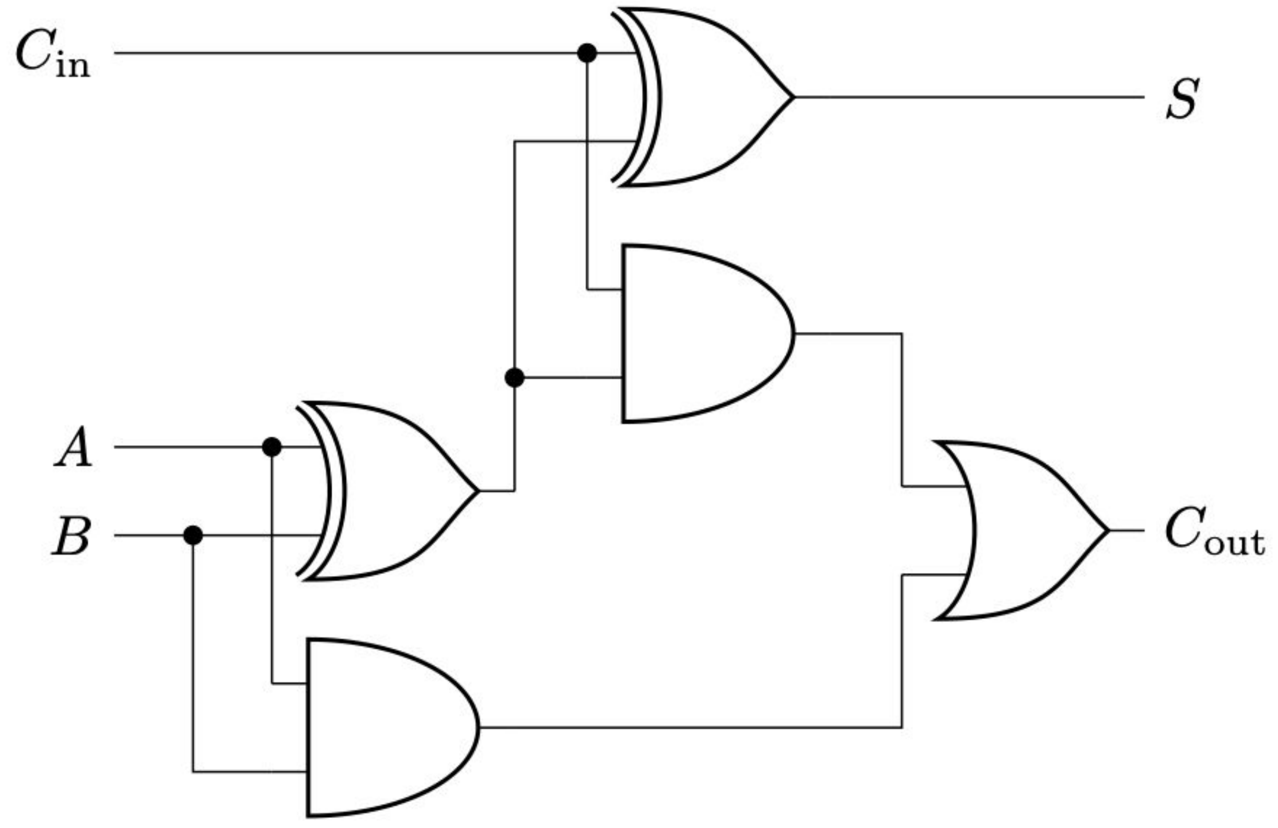
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

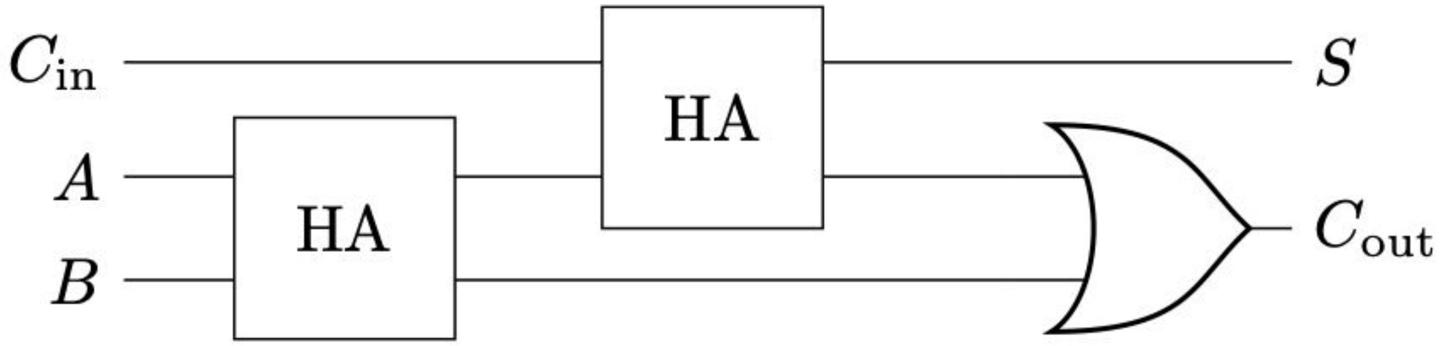
Adding binary numbers

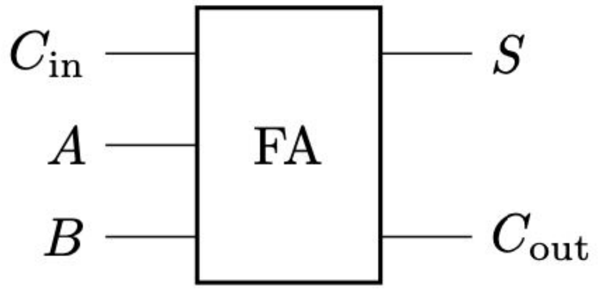




Half-adder

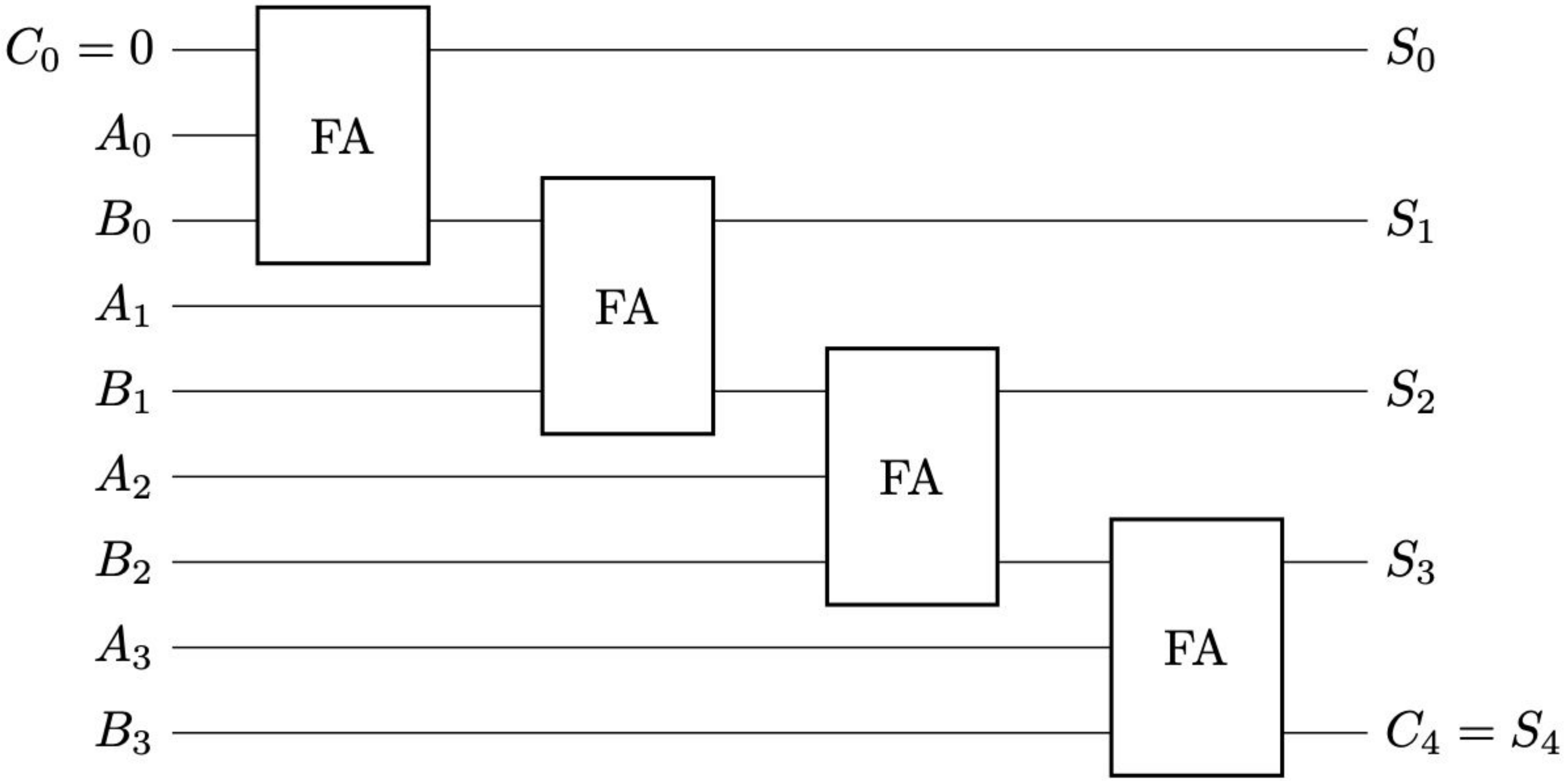






Full-adder

Adding binary numbers



Ripple-carry adder

Classical		Reversible/Quantum	
NOT	$A \rightarrow \overline{A}$	X-Gate	$A \rightarrow \boxed{X} \rightarrow \overline{A}$
AND	$A, B \rightarrow AB$	Toffoli	$A, B, 0 \rightarrow A, B, AB$
OR	$A, B \rightarrow A + B$	anti-Toffoli	$A, B, 1 \rightarrow A, B, A + B$
XOR	$A, B \rightarrow A \oplus B$	CNOTs	$A, B, 0 \rightarrow A, B, A \oplus B$
NAND	$A, B \rightarrow \overline{AB}$	Toffoli	$A, B, 1 \rightarrow A, B, \overline{AB}$
NOR	$A, B \rightarrow \overline{A + B}$	anti-Toffoli	$A, B, 0 \rightarrow A, B, \overline{A + B}$

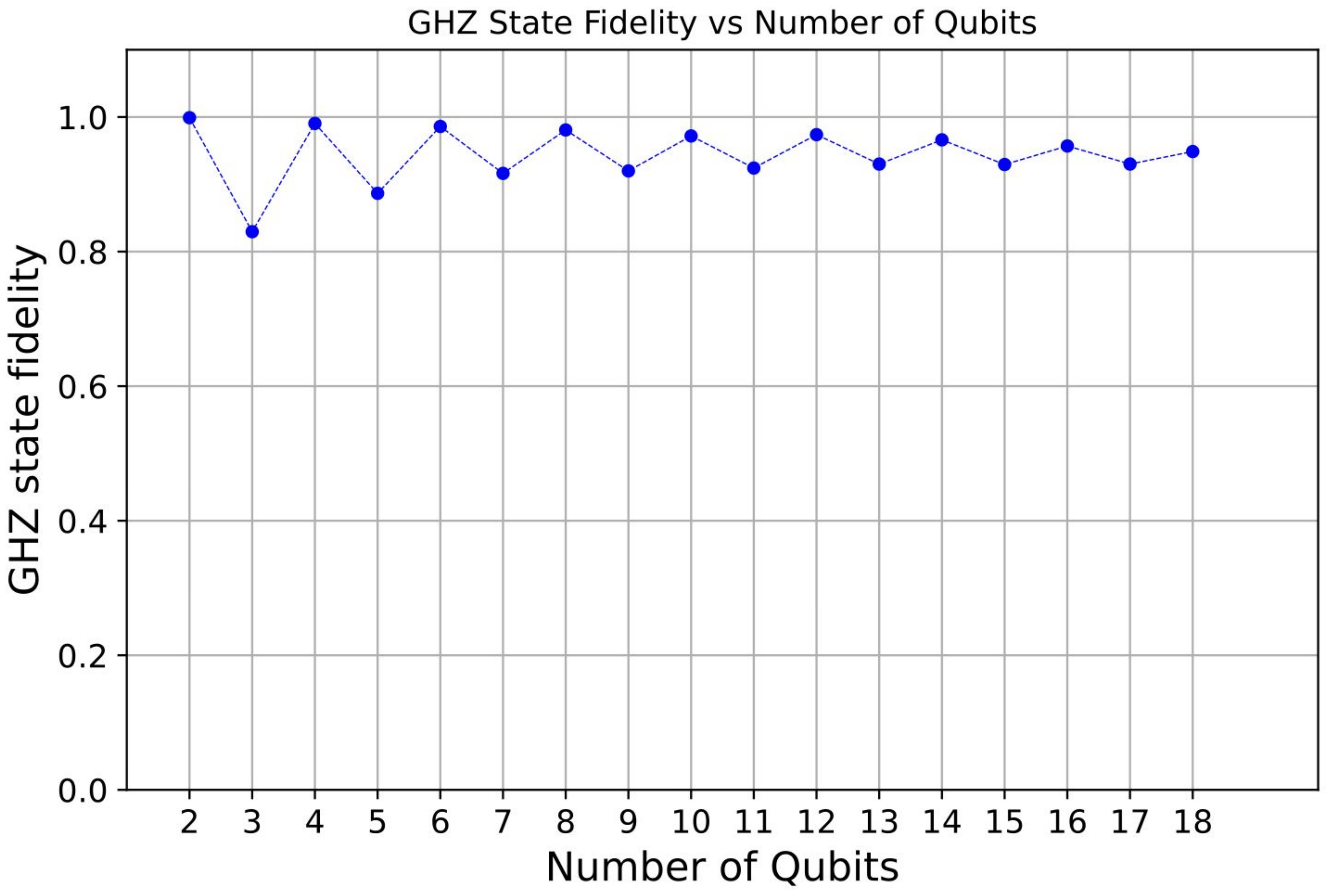
GHZ State Fidelity

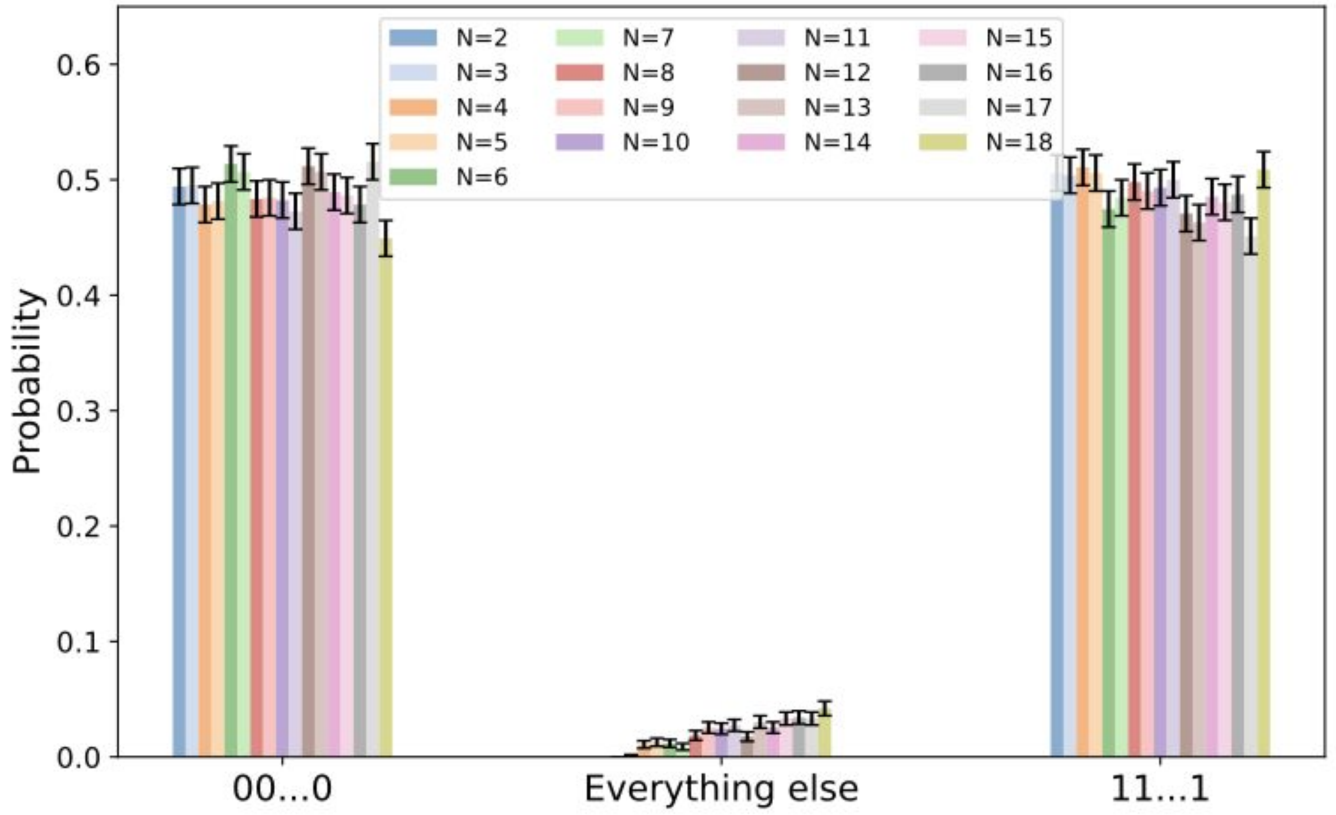
- GHZ state fidelity: The Greenberger-Horne-Zeilinger (GHZ) state creation is a popular approach to entanglement verification. Measuring the GHZ state fidelity is a useful test for measuring the quality and performance of the quantum device or emulators.

Exercise: Create one small benchmarking using GHZ fidelity & Aer noiseless simulator & Callisto emulator.

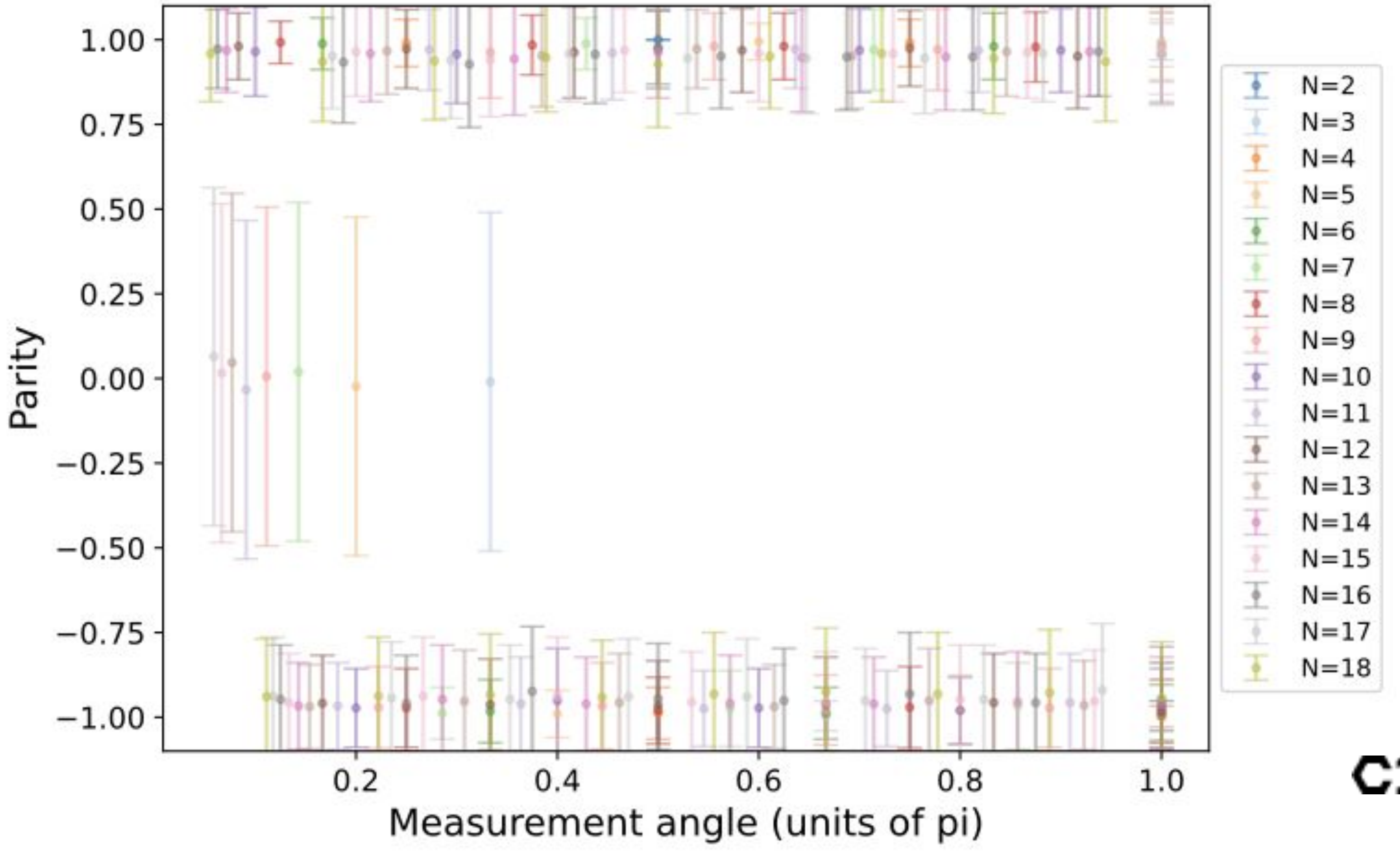
- Create GHZ state for $|GHZ_N\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes N} + |1\rangle^{\otimes N}\right),$

  Advanced:Use logarithmic ghz circuit, see *arxiv.1807.05572*
- Estimate GHZ state fidelity *F (from arxiv.0706.2432).*





(A) Results of GHZ-populations.



C12

Unique at scale

viktor@c12qe.com