Calcul Québec

Getting started with MonarQ (MNQ-101)

Juliette Geoffrion and Kim Renaud - February 25 2025

Members and Partners

Université Laval | McGill | Université de Sherbrooke | Université de Montréal

Bishop's University | HEC Montréal | Université Concordia University | Polytechnique Montréal

UQAC Université du Québec à Chicoutimi | UQTR Savoir. Surprendre. | UQÀM | ÉTS

INRS Institut national de la recherche scientifique | UQAT Université du Québec en Abitibi-Témiscamingue | Université TÉLUQ | Ouranos

Connectivity Partners

risq Le réseau au service du savoir | canarie

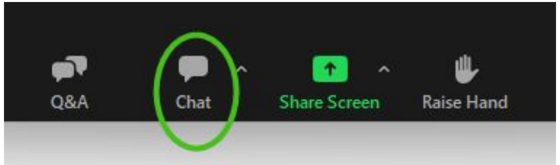
Financial Partners

A partner of the Digital Research Alliance of Canada

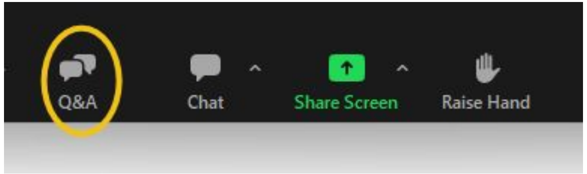
Québec

Zoom

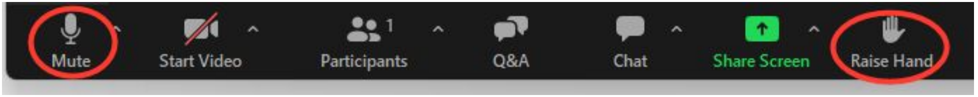
- If you experience technical problems, mention them in the *Chat*. An assistant will help you.



- If you have a question related to the course content, use the *Q&A*.

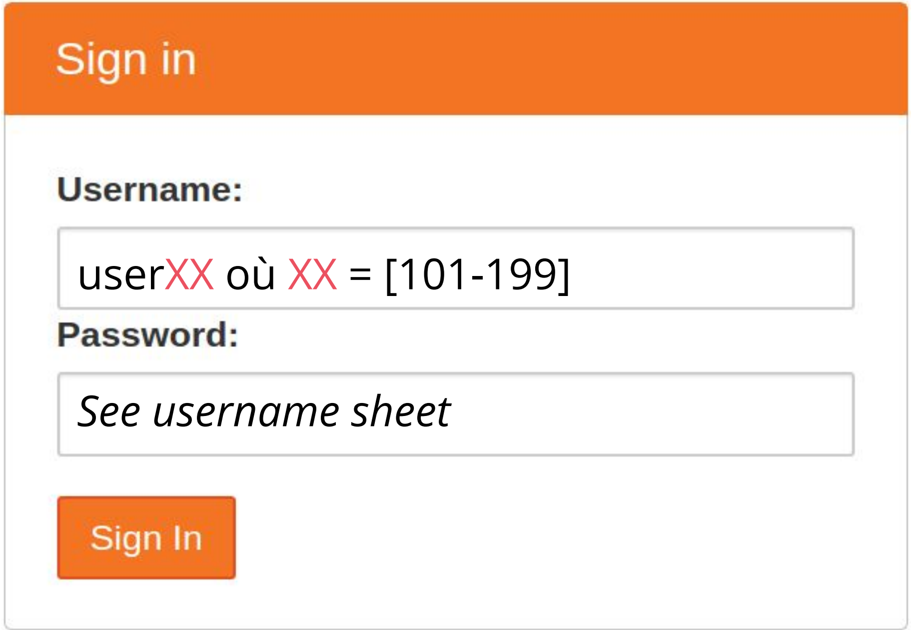


- You can also raise your hand and activate your microphone to ask a question.



Connect to JupyterLab

https://mnq101.calculquebec.cloud/



Video Recording

For whom and why?

- ➔ For registered participants to refer to the recording
- ➔ For Calcul Québec staff to improve the workshop

For how long?

- ➔ The video will be stored and available for 6 months.

Prefer not to be identified?

- ➔ Mute yourself and only ask your questions in the chat

Getting started with MonarQ (MNQ-101)

- Calcul Québec and MonarQ
- Quantum information fundamentals overview
  - Notebook 1 (optional) : Build a quantum simulator
- Quantum computing in NISQ era
  - Capabilities and limitations of MonarQ
  - Literature overview
- How do you program a quantum computer?
  - Notebook 2 : PennyLane basics
- Case study (time permitting) :
  - Notebook 3 : Double slit experiment
- Questions

Learning goals

1. **Review core concepts in quantum information**

   Core concepts : Qubits, Dirac notation, superposition, quantum operations, Bloch sphere, measurement

2. **Explain the challenges and limitations of NISQ era quantum computing**

   Core concepts : NISQ, FTQC, MonarQ performances and constraints

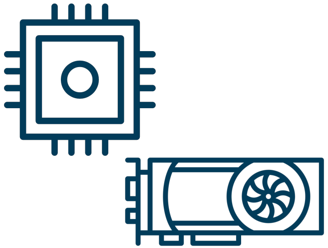
3. **Write quantum circuits in PennyLane**

   Core concepts : quantum circuits, devices, qnodes, quantum gates and measurements

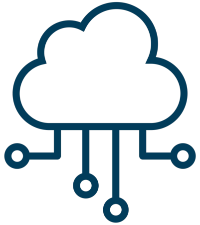
Calcul Québec

A few words about who we are

Our services



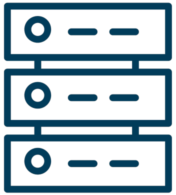
Computing clusters



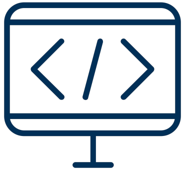
Cloud computing

Quantum computing

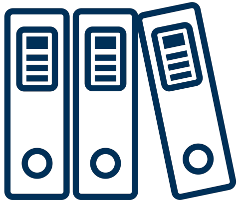
Personalized support



Data Storage



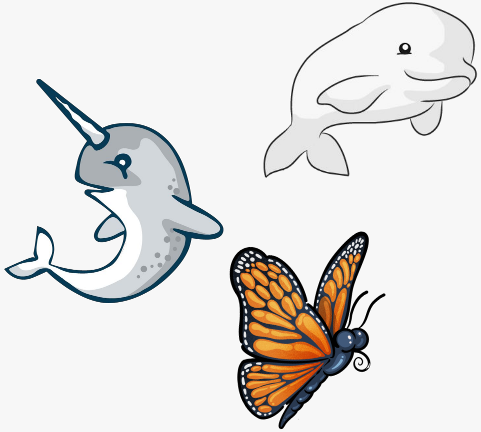
Interactive Computing (notebooks)

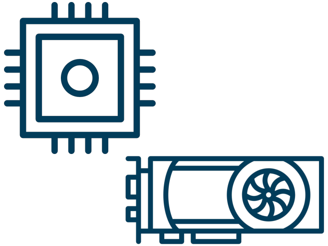


Documentation

Training Workshops

Calcul Québec's Infrastructure





Computing clusters

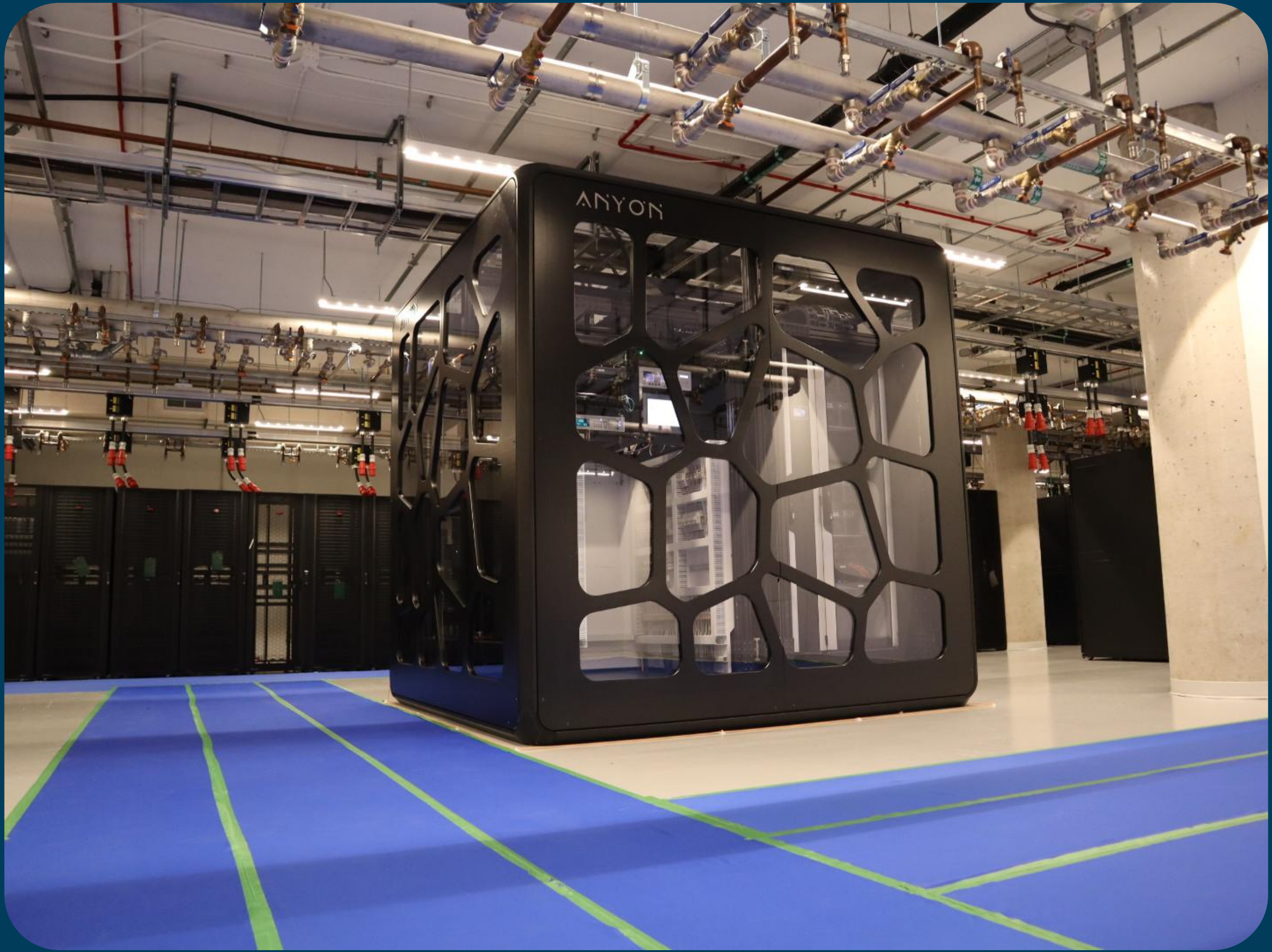
Classical Computing clusters

- Béluga and Narval
- 140,000 CPU
- 1,300 GPU

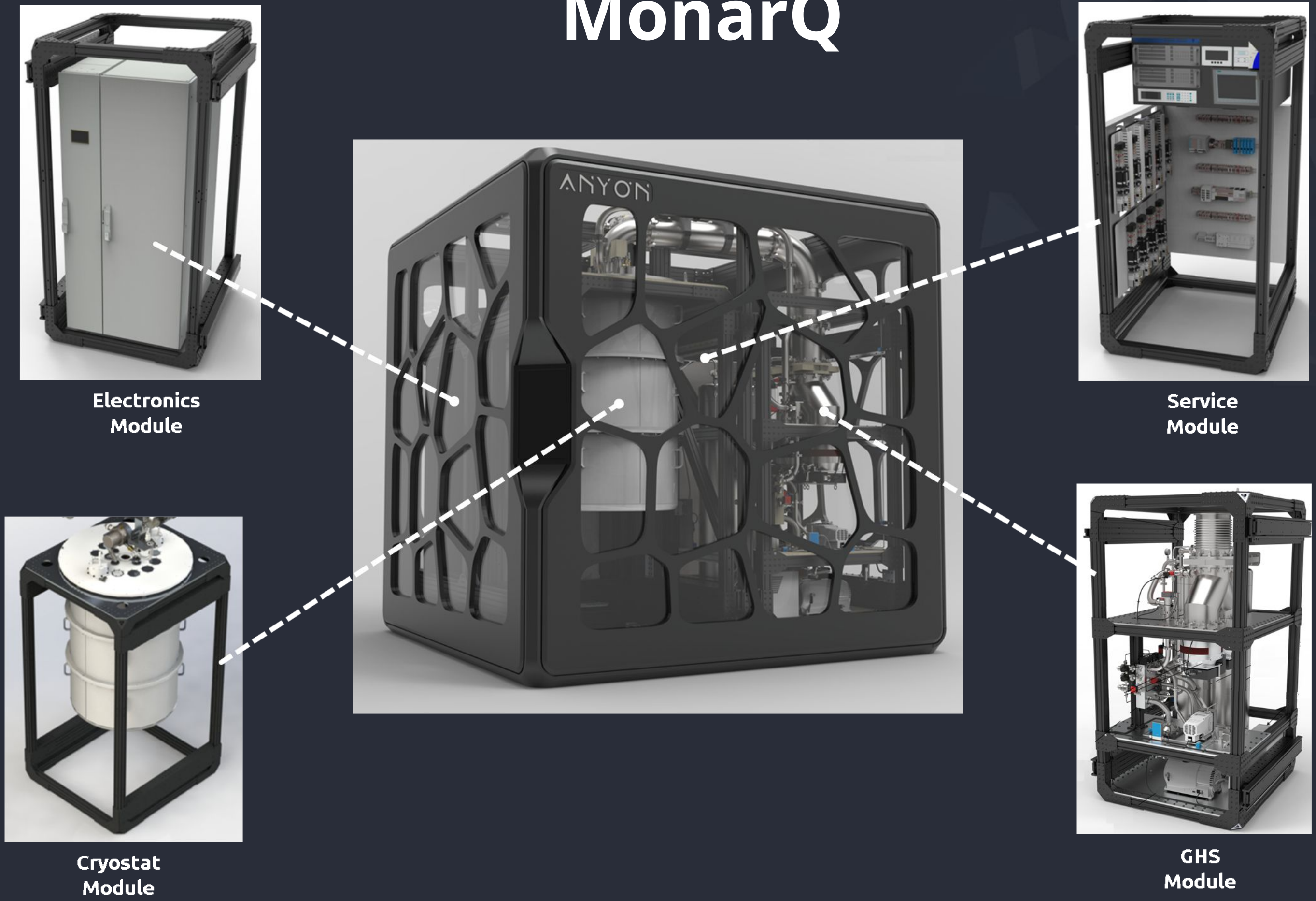
Quantum computing

Universal quantum computer

- MonarQ
- 24 superconducting qubits

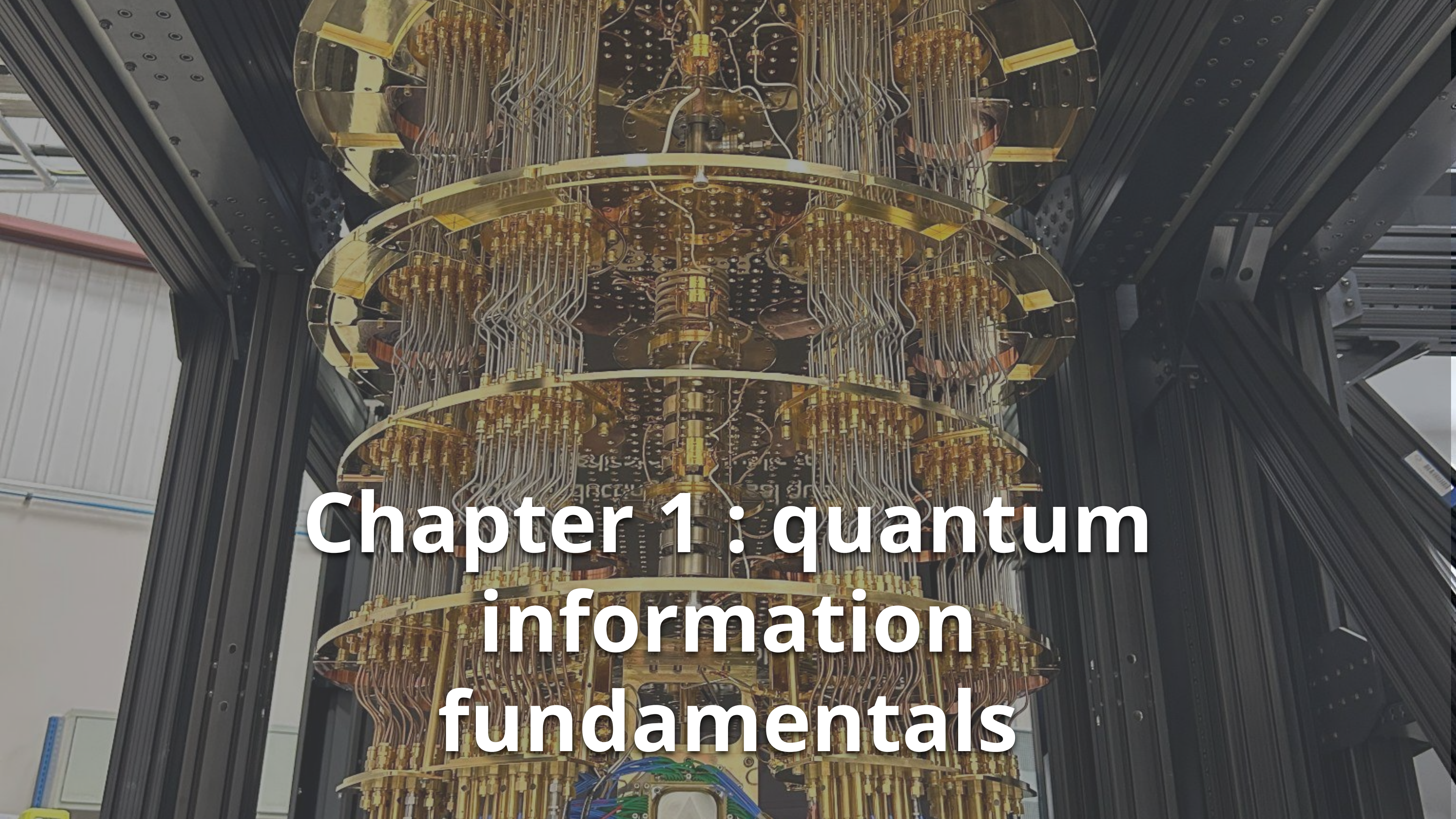

ANYON

MonarQ


ANYON
Electronics Module
Service Module
Cryostat Module
GHS Module

Cryostat Module (dilution refrigerator)

- QPU is kept around 15 mK

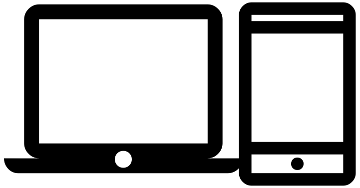


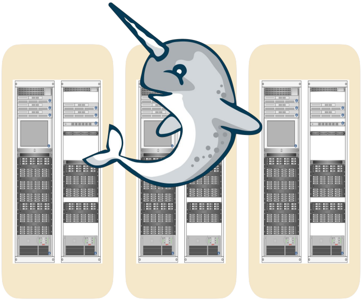
Chapter 1 : quantum information fundamentals

Overview

Classical computing

- Uses binary states or **bits** (0 and 1)
- Deterministic computation
- Laptops, phones, computing clusters etc



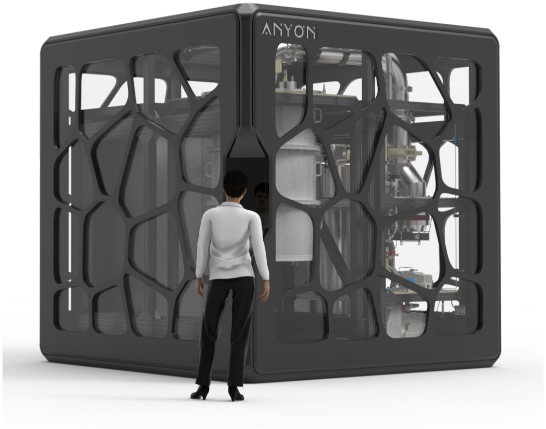


Quantum computing

- Uses quantum bits or **qubits**

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

- Superposition, entanglement and interference
- Probabilistic computation
- MonarQ



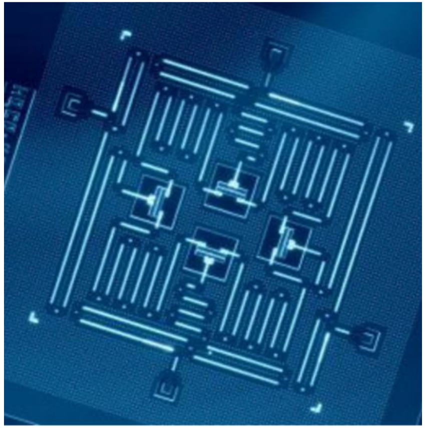
Qubits

- Fundamental element of computation exhibiting the properties of quantum mechanics.

- The Boolean states *0* and *1* are represented by a pair of normalised and orthogonal quantum states $|0\rangle$ and $|1\rangle$

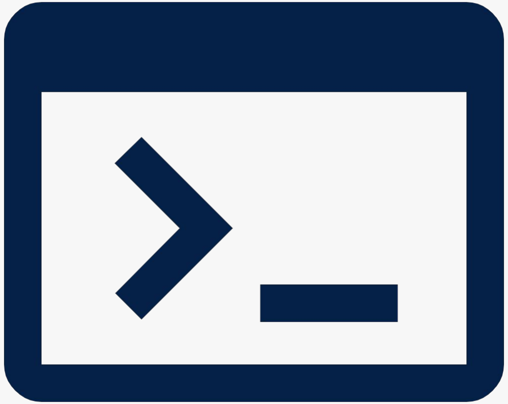
- In practice, a qubit is typically a microscopic system such as an atom, an electron, a spin or a polarised photon.

MonarQ's qubits are **transmons** made from superconducting material operating at extremely low temperatures and manipulated by microwave pulses.



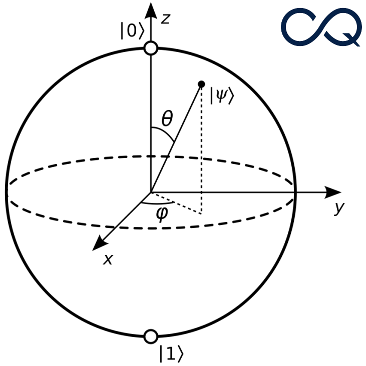
https://www.nature.com/articles/s41534-016-0004-0

Coding break



Notebook 1 :
Building a quantum simulator

Notebook 1 : Cheat Sheet



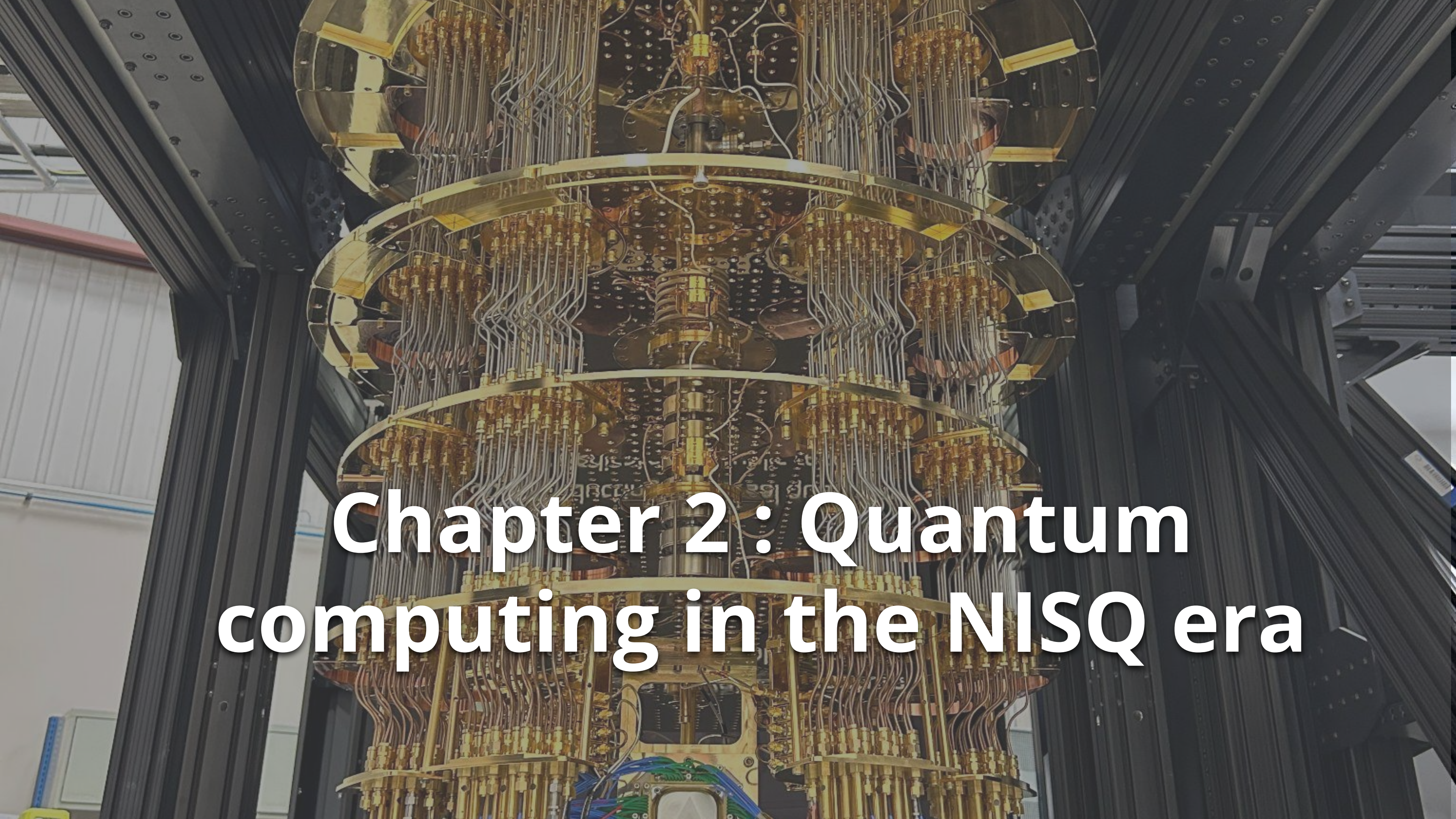
1. **Qubits** can be in a **superposition state**
   - $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ with $|\alpha|^2 + |\beta|^2 = 1$
   - Represented by a vector in a complex vector space

2. **Dirac** notation
   - A **ket** $|\psi\rangle$ is a column vector and its associated **bra** $\langle\psi|$ is a row vector obtained by taking the complex conjugate transpose.

3. **Bloch sphere**
   - **Visual representation of a qubit** as a vector on a sphere with θ and ϕ as coordinates for the state : $|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle$

4. Quantum **gates** are **unitary matrices** and act on qubits like **rotations** on the Bloch sphere.

5. A **quantum circuit** is a sequence of unitary transformations (gates) applied to an initial state.
   - $|\psi_{\text{final}}\rangle = U_n \cdots U_2 U_1 |\psi_{\text{initial}}\rangle$
   - **Measurement** gives **probabilistic** results based on the final state of the qubit



Chapter 2 : Quantum computing in the NISQ era

Quantum computers are noisy

- Qubits are highly susceptible to quantum noise
- Circuit depth (number of circuit layers) makes runtime longer, and thus adds and propagates noise
- Circuits must be as short as possible
- Circuit must be executable on a given machine

https://www.vecteezy.com/vector-art/23929483-noise-pollution-harmful-sounds-causing-disturbances-affecting-health-and-environment

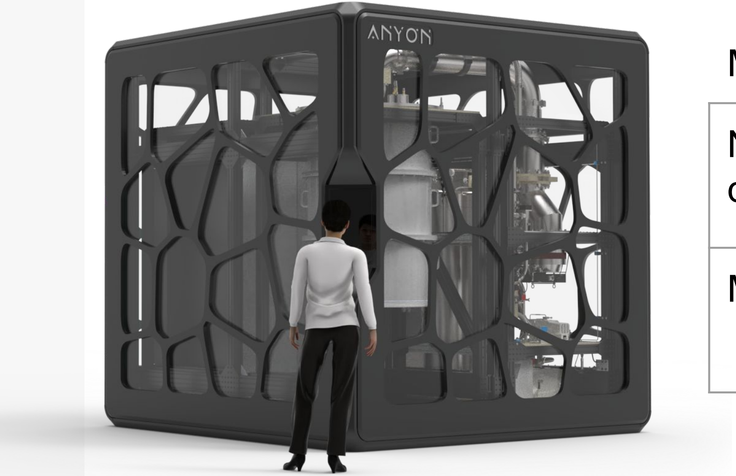
NISQ Era

Noisy
Intermediate
Scale Quantum
Computers

Currently available quantum processors

High Noise levels

- ➔ Limits qubit counts
- ➔ Limits circuit depth

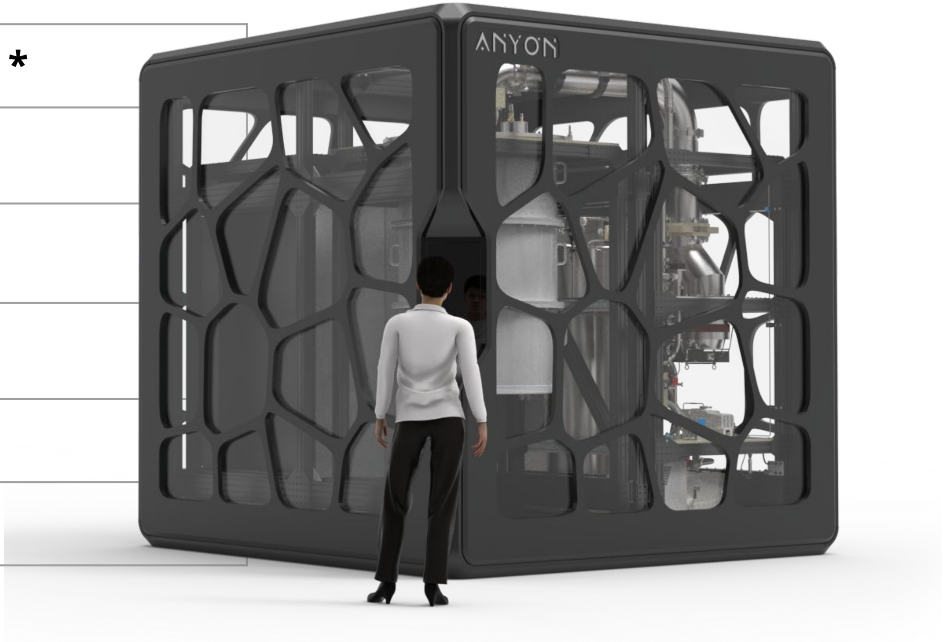


MonarQ

Number of qubits	24 qubits
Max depth	~350

MonarQ performance metric

Metric	Median value *
T_1	9.4 μs
$T_{2,Ramsey}$	1.3 μs
$T_{2,Echo}$	3.3 μs
Single-qubit gate fidelity	99.75%
CZ gate fidelity	96.2%



*Based on data from Dec. 4, 2024 to Dec. 18, 2024

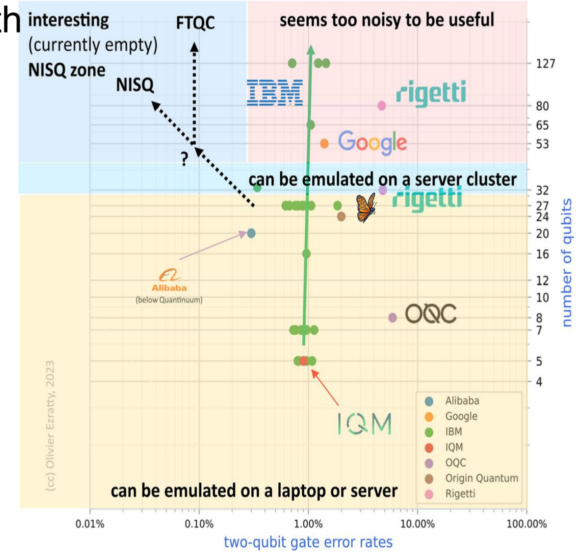
Towards FTQC

Fault Tolerant Quantum Computers

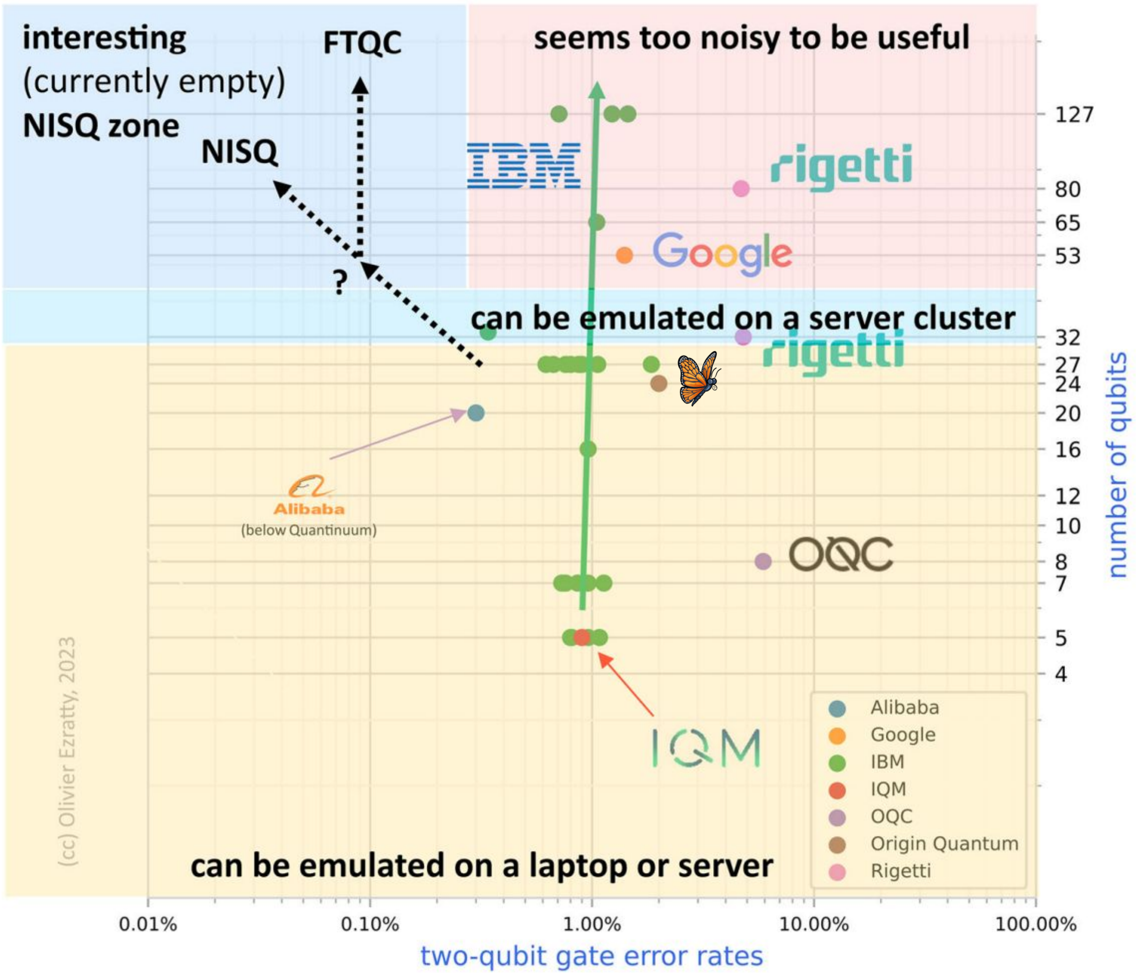
Long term goal of quantum computing

Error-corrected quantum computers

➔ Larger number of qubits needed to make **logical** qubits (1000:1)
➔ Logical qubit : robust to errors (as opposed to a **physical qubit**)
➔ Increased circuit depth



Source : Eur. Phys. J. A (2023) 59:94 https://doi.org/10.1140/epja/s10050-023-01006-7



Common misconception

Misconception : All problems benefit from quantum speedup.

Reality : Quantum speedup depends on the problem. Many tasks see no advantage, and the overhead of quantum state preparation and error mitigation can outweigh the benefits on current devices.

When should I use a quantum computer?

Laptop will suffice

NISQ friendly problems

Fault-tolerant QC required

- Arithmetic
- Checking emails

- Chemistry
- Machine learning
- Optimization

… and other fields of research

- Shor’s algorithm
- Quantum fourier transform
- Grover’s algorithm

Slide adapted from https://learning.quantum.ibm.com/course/quantum-computing-in-practice/what-problems-are-quantum-computers-good-for

NISQ Case Study

Chemistry

Greene-Diniz et al. *EPJ Quantum Technology* (2022) 9:37
https://doi.org/10.1140/epjqt/s40507-022-00155-w

EPJ.org

EPJ Quantum Technology
a SpringerOpen Journal

COMMENT Open Access

Check for updates

Modelling carbon capture on metal-organic frameworks with quantum computing

Gabriel Greene-Diniz[1], David Zsolt Manrique[1], Wassil Sennane[2], Yann Magnin[3], Elvira Shishenina[4,2], Philippe Cordier[2], Philip Llewellyn[3], Michal Krompiec[1], Marko J. Rančić[2*] and David Muñoz Ramo[1]

*Correspondence:
marko.rancic@totalenergies.com
[2]TotalEnergies, OneTech, One R&D, 8 Boulevard Thomas Gobert, 91120 Palaiseau, France
Full list of author information is available at the end of the article

Abstract

Despite the recent progress in quantum computational algorithms for chemistry, there is a dearth of quantum computational simulations focused on material science applications, especially for the energy sector, where next generation sorbing materials are urgently needed to battle climate change. To drive their development, quantum computing is applied to the problem of CO_2 adsorption in Al-fumarate Metal-Organic Frameworks. Fragmentation strategies based on Density Matrix Embedding Theory are applied, using a variational quantum algorithm as a fragment solver, along with active space selection to minimise qubit number. By investigating different fragmentation strategies and solvers, we propose a methodology to apply quantum computing to Al-fumarate interacting with a CO_2 molecule, demonstrating the feasibility of treating a complex porous system as a concrete application of quantum computing. We also present emulated hardware calculations and report the impact of device noise on calculations of chemical dissociation, and how the choice of error mitigation scheme can impact this type of calculation in different ways. Our work paves the way for the use of quantum computing techniques in the quest of sorbents optimisation for more efficient carbon capture and conversion applications.

Keywords: Quantum computing; NISQ; Carbon capture; Climate change; Quantum algorithms

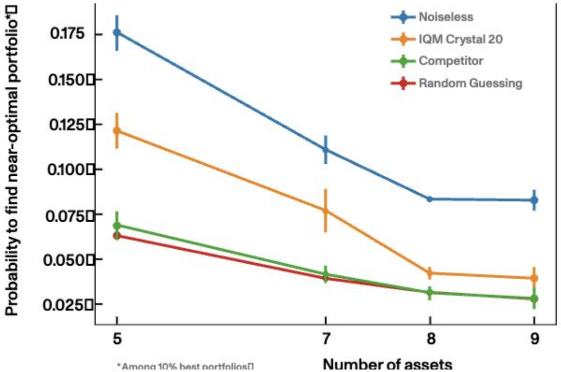
Source : https://epjquantumtechnology.springeropen.com/articles/10.1140/epjqt/s40507-022-00155-w

NISQ Case Study

Optimization

IQM and DATEV advance quantum solutions for portfolio optimization

In our collaboration with DATEV, we developed a specific quantum algorithm tailored to the company's product portfolio optimization challenge. By using the data of DATEV and applying advanced transpilation techniques, we were able to execute the algorithm on our quantum hardware, leveraging a quantum processing unit (QPU) with 20 qubits.



Source : https://www.meetiqm.com/lp/the-future-of-portfolio-optimization-is-in-quantum

Machine Learning

Applying Quantum Autoencoders for Time Series Anomaly Detection

Robin Frehner and Kurt Stockinger

Zurich University of Applied Sciences, Switzerland.

*Corresponding author(s). E-mail(s): frehner.robin@gmail.com; kurt.stockinger@zhaw.ch;

Abstract

Anomaly detection is an important problem with applications in various domains such as fraud detection, pattern recognition or medical diagnosis. Several algorithms have been introduced using classical computing approaches. However, using quantum computing for solving anomaly detection problems in time series data is a widely unexplored research field.

This paper explores the application of quantum autoencoders to time series anomaly detection.

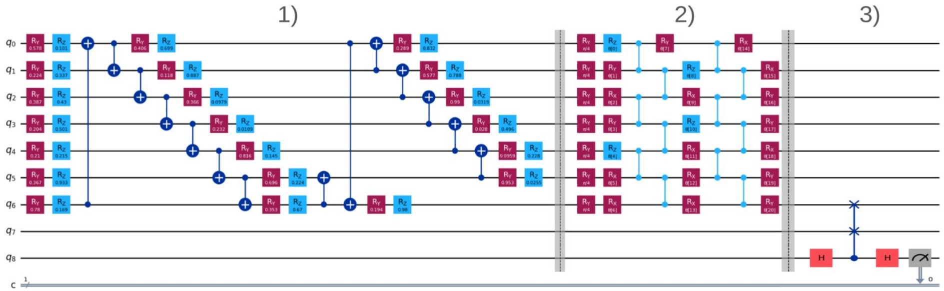


Fig. 8: Illustration of a circuit optimized for execution on real quantum hardware before transpilation.

Source : https://arxiv.org/abs/2410.04154

NISQ Case Study

and other active fields of research

Material science

Quantum Algorithm for Vibronic Dynamics: Case Study on Singlet Fission Solar Cell Design

Danial Motlagh,[1] Robert A. Lang,[1] Jorge A. Campos-Gonzalez-Angulo,[2,3] Tao Zeng,[4] Alan Aspuru-Guzik,[2,3,5,6] and Juan Miguel Arrazola[1]

[1] Xanadu, Toronto, ON, M5G 2C8, Canada
[2] University of Toronto, Toronto, ON, M5S 3H6, Canada
[3] Vector Institute for Artificial Intelligence, Toronto, ON, M5G 1M1, Canada
[4] Department of Chemistry, York University, Toronto, ON, M3J 1P3, Canada
[5] Canadian Institute for Advanced Research (CIFAR), Toronto, ON, M5G 1M1, Canada
[6] Acceleration Consortium, Toronto, ON, M

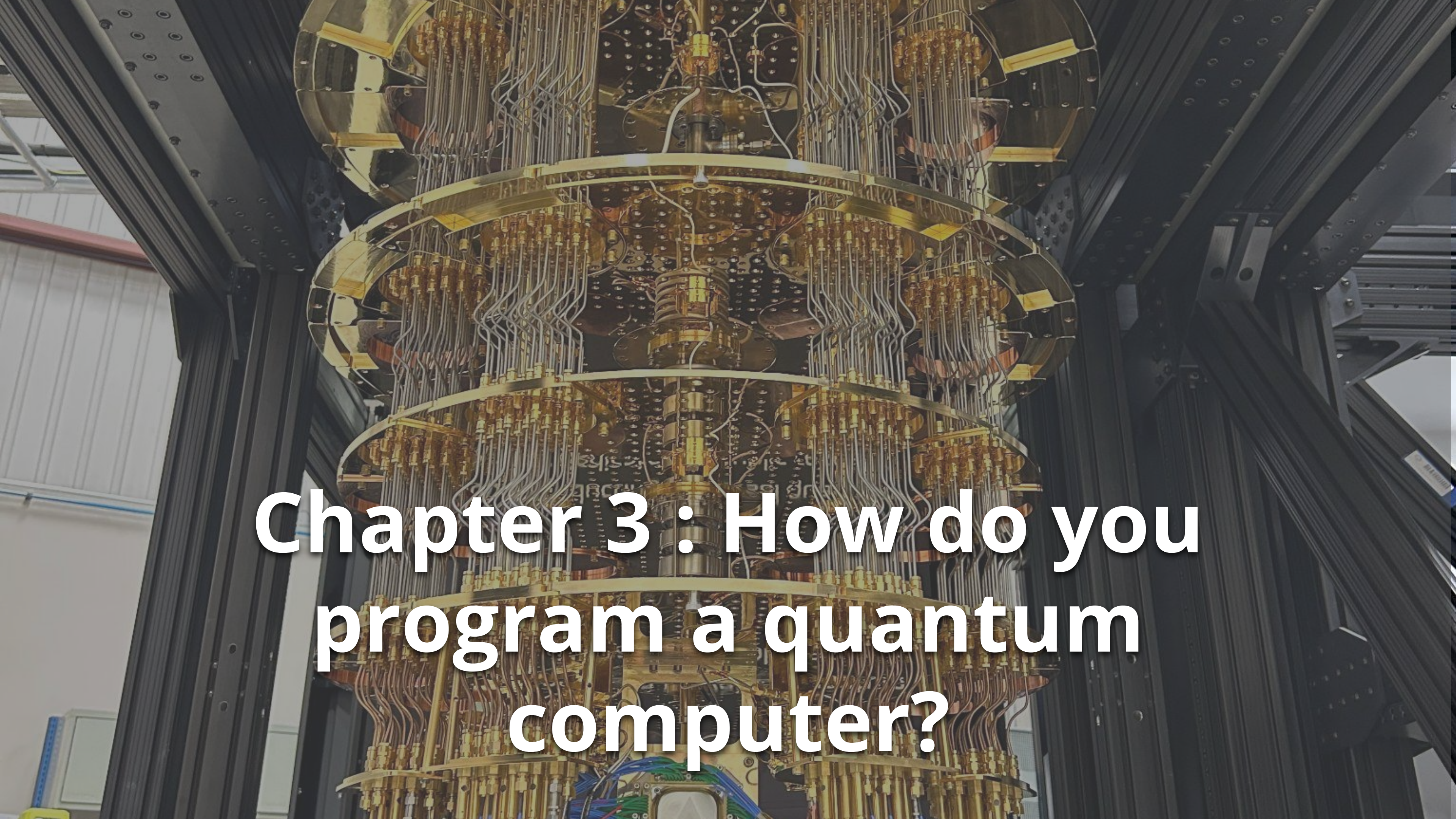
Vibronic interactions between nuclear motion and electronic modeling of photochemistry. However, accurate simulations of f ics are often prohibitively expensive for classical methods beyo present a quantum algorithm based on product formulas for sim eral vibronic Hamiltonian in real space, capable of handling an a and vibrational modes. We develop the first trotterization schen

Biology

A biological sequence comparison algorithm using quantum computers

Büsra Kösoglu-Kind[1], Robert Loredo[2,3], Michele Grossi[4], Christian Bernecker[5], Jody M. Burks[2] & Rüdiger Buchkremer[1]✉

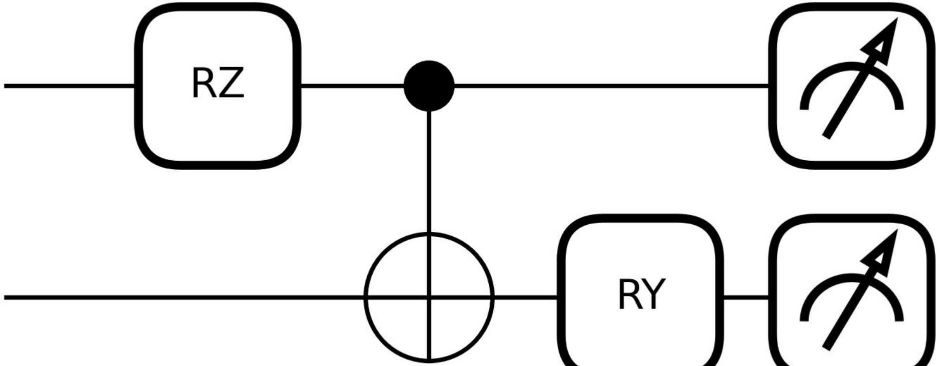
Genetic information is encoded as linear sequences of nucleotides, represented by letters ranging from thousands to billions. Differences between sequences are identified through comparative approaches like sequence analysis, where variations can occur at the individual nucleotide level or collectively due to various phenomena such as recombination or deletion. Detecting these sequence differences is vital for understanding biology and medicine, but the complexity and size of genomic



Chapter 3 : How do you program a quantum computer?

Quantum circuits

```
dev = qml.device('default.qubit', wires = 2)
@qml.qnode(dev) # qnode decorator
def quantum_function(x, y):
    qml.RZ(x, wires=0)
    qml.CNOT(wires=[0,1])
    qml.RY(y, wires=1)
    return qml.state()
```



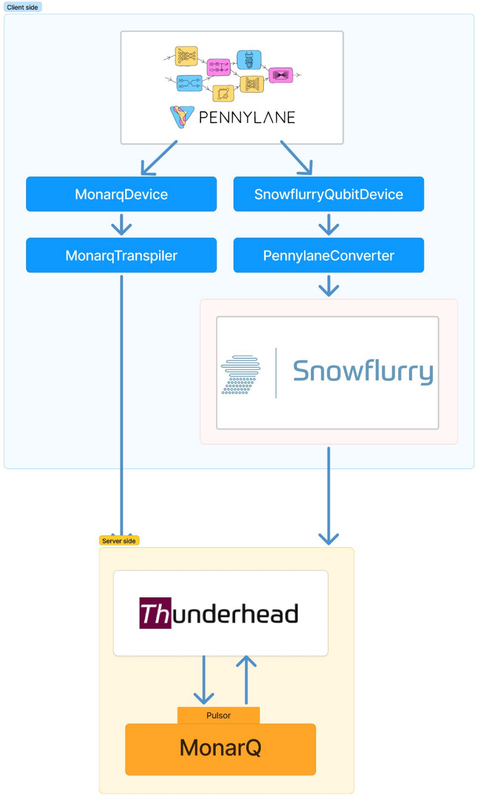
Quantum software development kits

- **PennyLane**
  - Developed in Python by Xanadu
  - Interface directly with MonarQ

- **Snowflurry**
  - Developed in Julia by Anyon Systems
  - Interface directly with MonarQ

- **Qiskit**
  - Developed in Python by IBM
  - Qiskit-calculquebec plugin in development

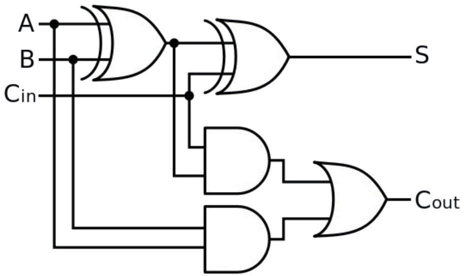
Write circuits and more !



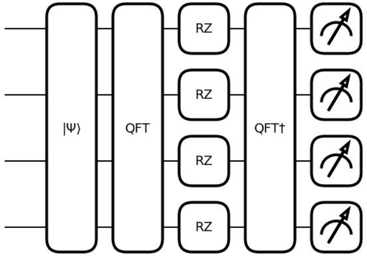
Quantum circuits

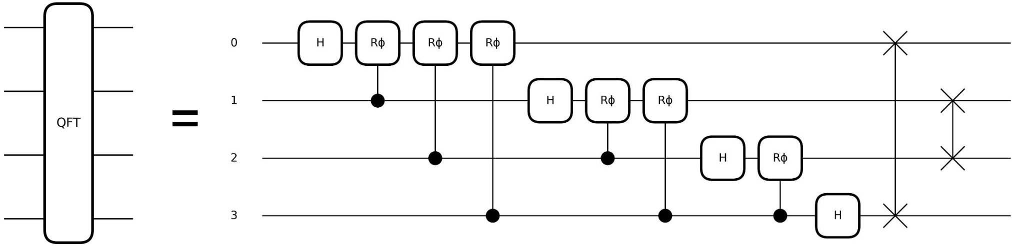
Classical computers also use circuits, but we do not generally think about these explicitly.

- **Classical** boolean circuit for addition



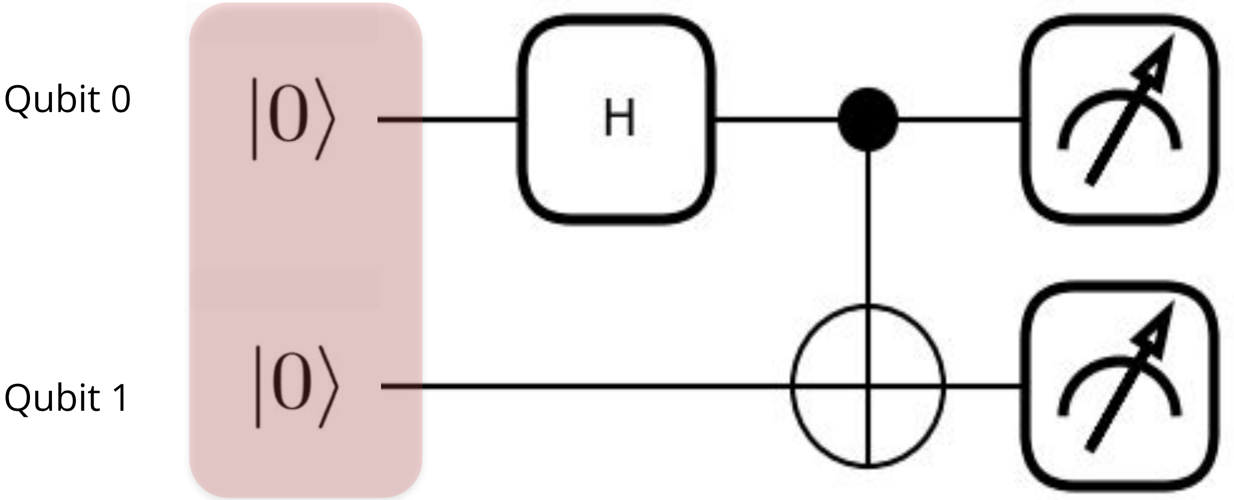
- **Quantum** circuit for addition





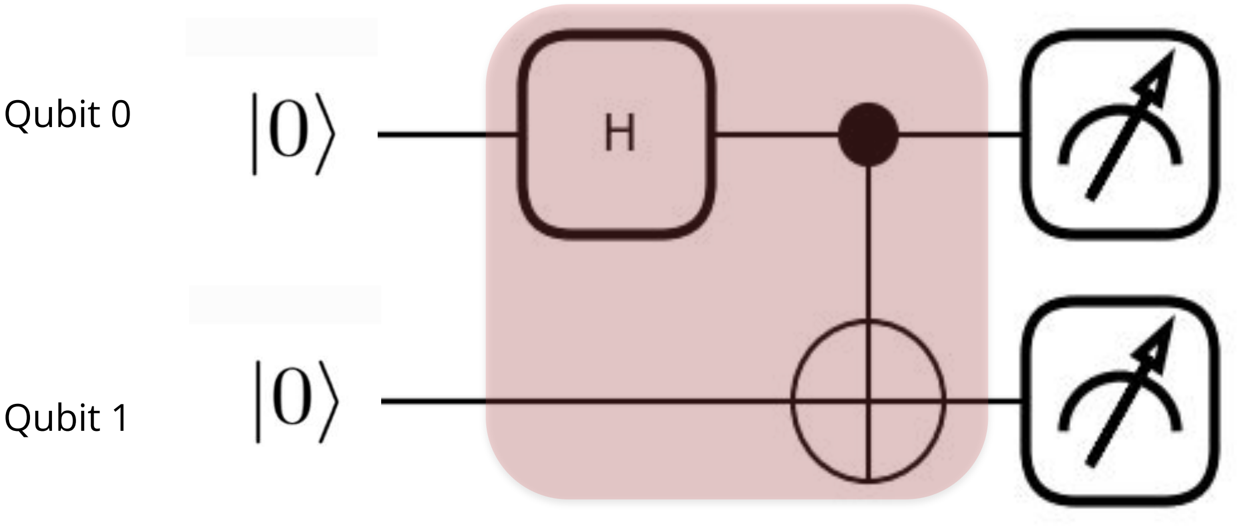
Key components of a quantum circuit

Each qubit is initialized
to state 0 by default



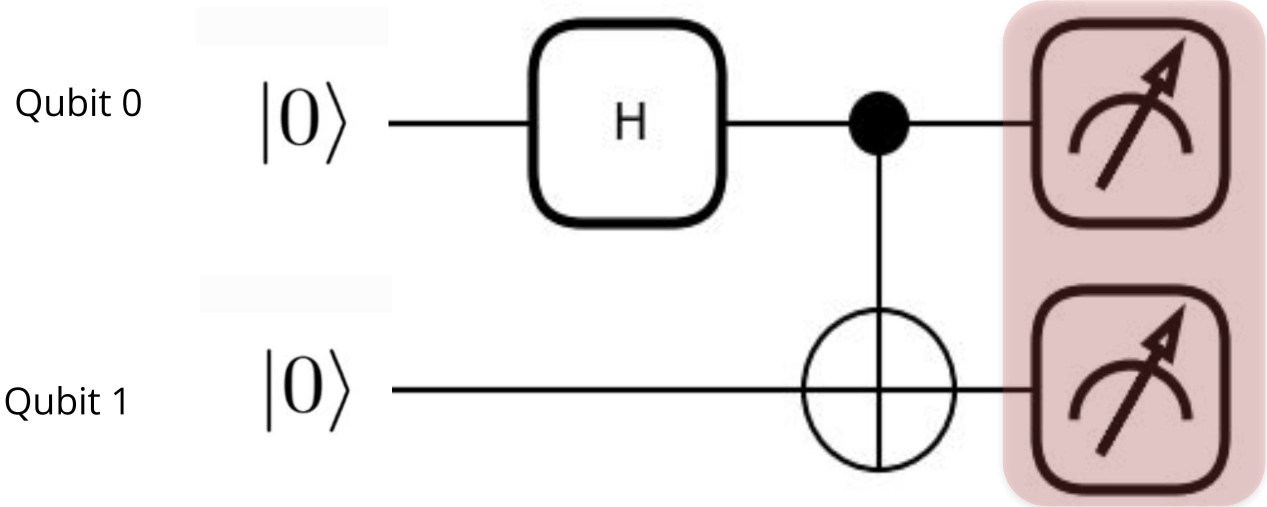
Key components of a quantum circuit

Apply quantum gates



Key components of a quantum circuit

Measure the state of one or more qubit



Coding break

Notebook 2 : PennyLane Basics

Notebook 2 cheat sheet

1. **Quantum circuits are composed of gates**

Gate	Circuit Element	Matrix Representation	Action on Basis States
Hadamard Gate H	$\boxed{H}$	$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$	$H\|0\rangle = \frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)$ $H\|1\rangle = \frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)$
Pauli-X Gate X	$\boxed{X}$	$\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$	$X\|0\rangle = \|1\rangle$ $X\|1\rangle = \|0\rangle$
Pauli-Y Gate Y	$\boxed{Y}$	$\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}$	$Y\|0\rangle = i\|1\rangle$ $Y\|1\rangle = -i\|0\rangle$
Pauli-Z Gate Z	$\boxed{Z}$	$\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$	$Z\|0\rangle = \|0\rangle$ $Z\|1\rangle = -\|1\rangle$
CNOT Gate	$\bullet$ $\oplus$	$\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}$	$CNOT\|00\rangle = \|00\rangle$ $CNOT\|01\rangle = \|01\rangle$ $CNOT\|10\rangle = \|11\rangle$ $CNOT\|11\rangle = \|10\rangle$

2. **Quantum circuits must return a measurement**
(`qml.state`, `qml.expval`, `qml.probs`, `qml.counts`)

3. **Link a circuit and a device together with a Qnode**

Qnode

Quantum function

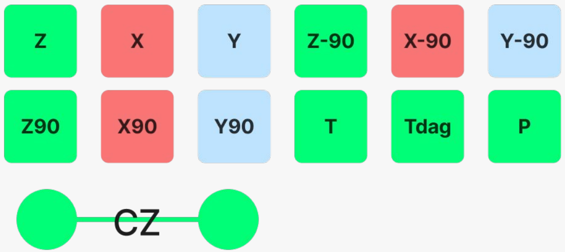
```
def circuit():
  …
 return qml.counts(0)
```

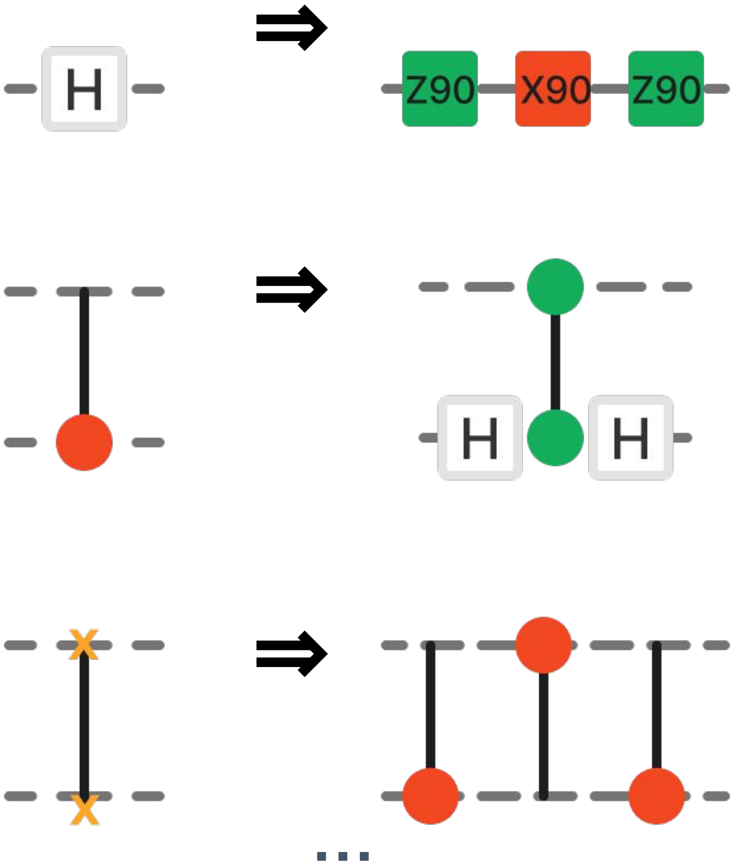
Device

```
qml.device(...)
```

About MonarQ

- MonarQ has a native set of 13 gates
- Any operation can be translated to that gate set (it is universal)
- Operations include : Z, X, Y, RZ90, X90, Y90, ZM90, XM90, YM90, T, adjoint(T), Phase shift and CZ

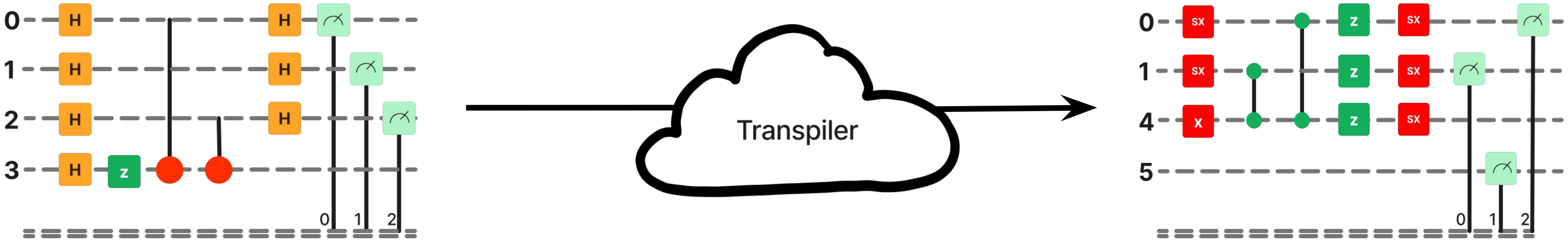




`pennylane-cq` plugin : A built-in transpiler for MonarQ

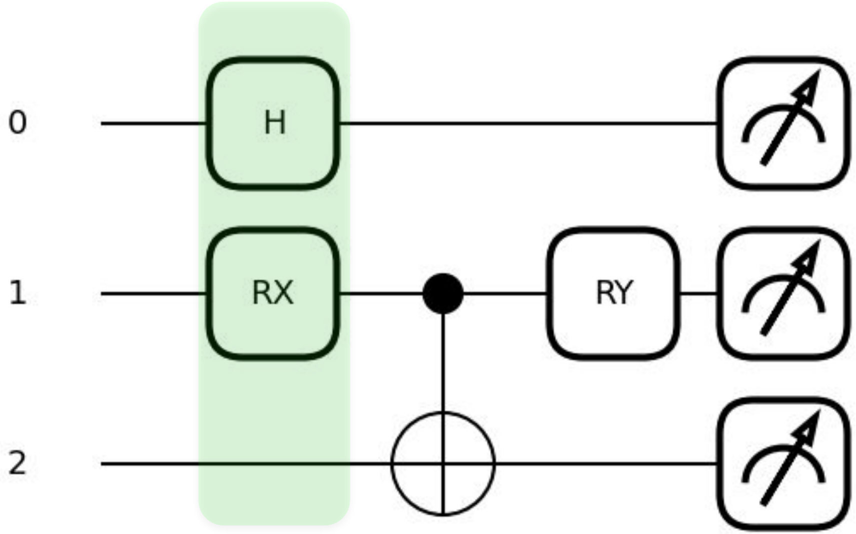
- A transpiler makes sure a quantum circuit can run on a machine by
  - **Mapping** the circuit's wires to qubits in the machine
  - **Routing** 2+ qubit operations
  - **Converting** non-native gates*

- A transpiler optimizes the circuit (making it as compact as possible) by
  - **Optimizing** placement and routing techniques
  - **Cancelling** trivial operations and inverses
  - **Finding** equivalences that shorten the circuit's size

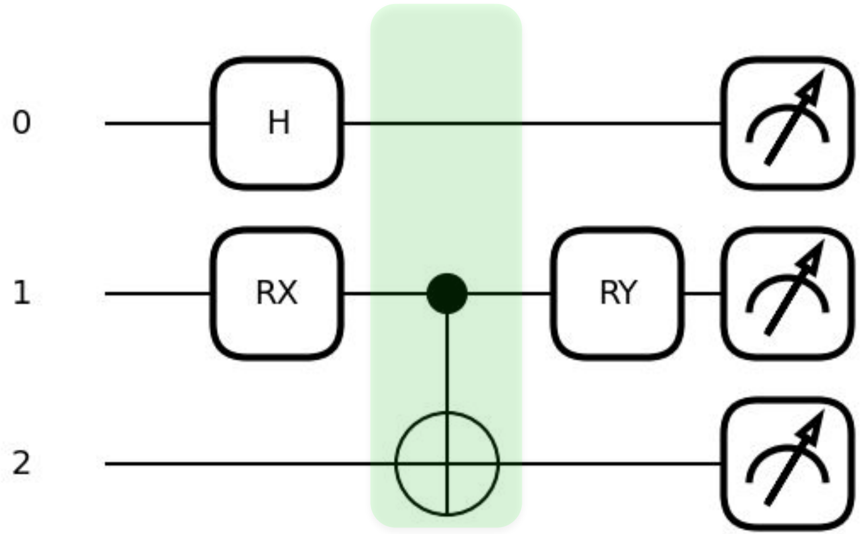


* to gates that are implemented natively on the machine

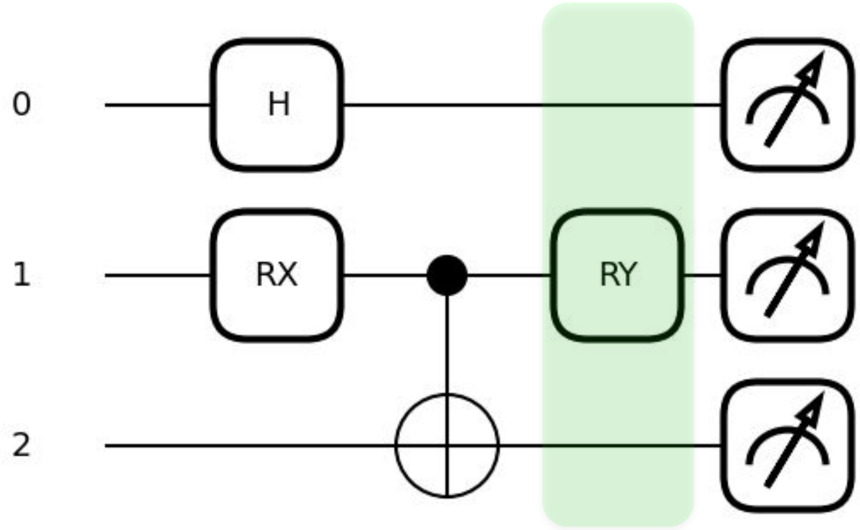
Circuit depth



Circuit depth

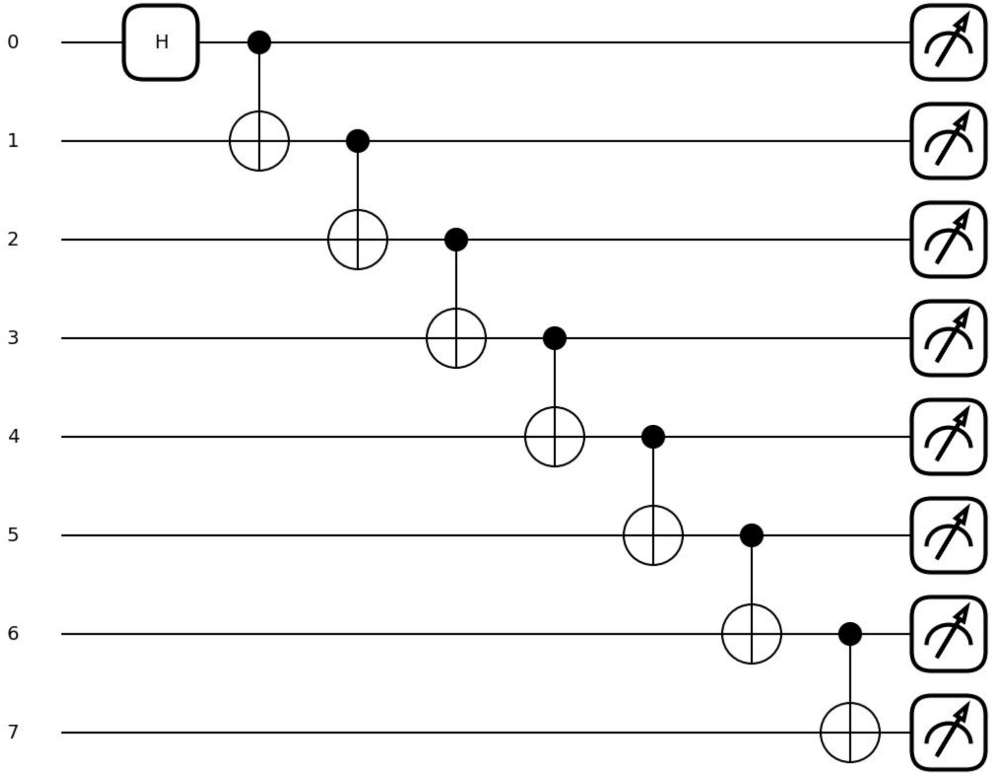


Circuit depth



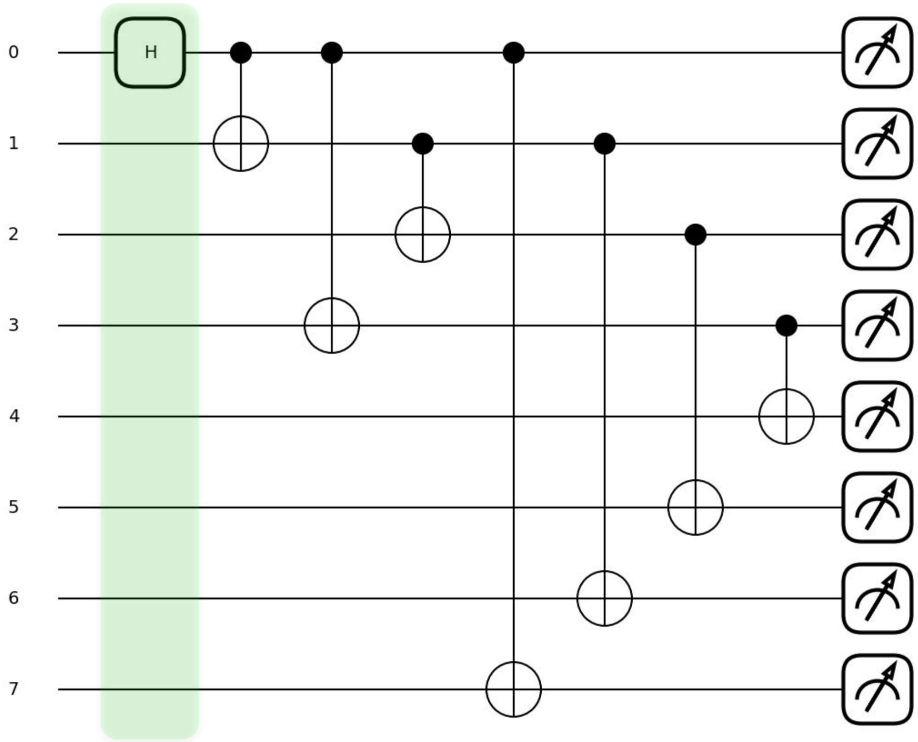
The depth of this circuit is 3.

Circuit depth

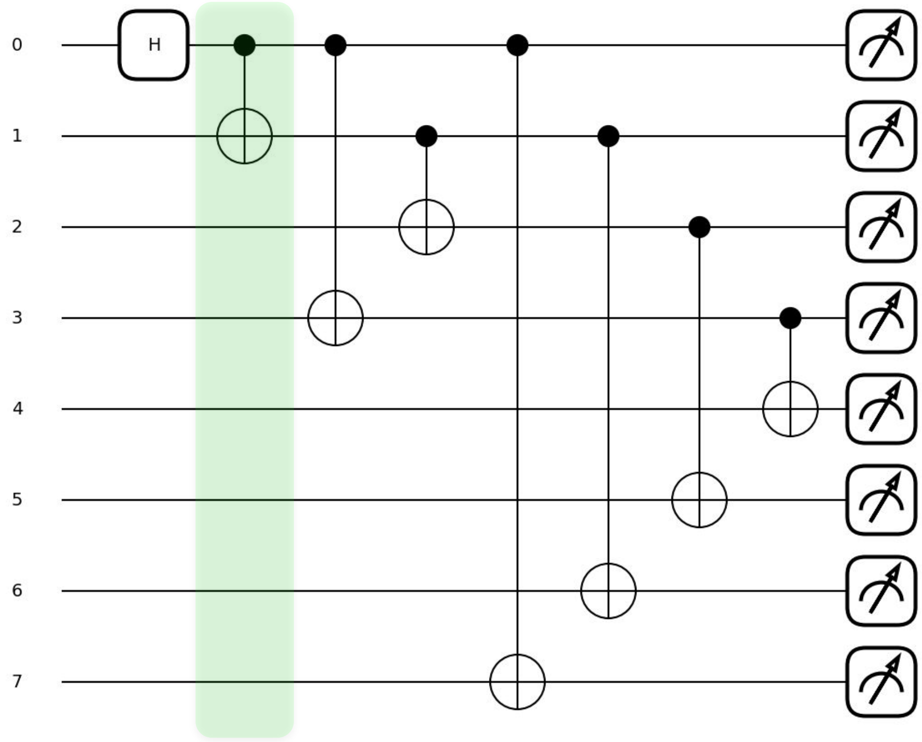


What is the depth of this circuit?

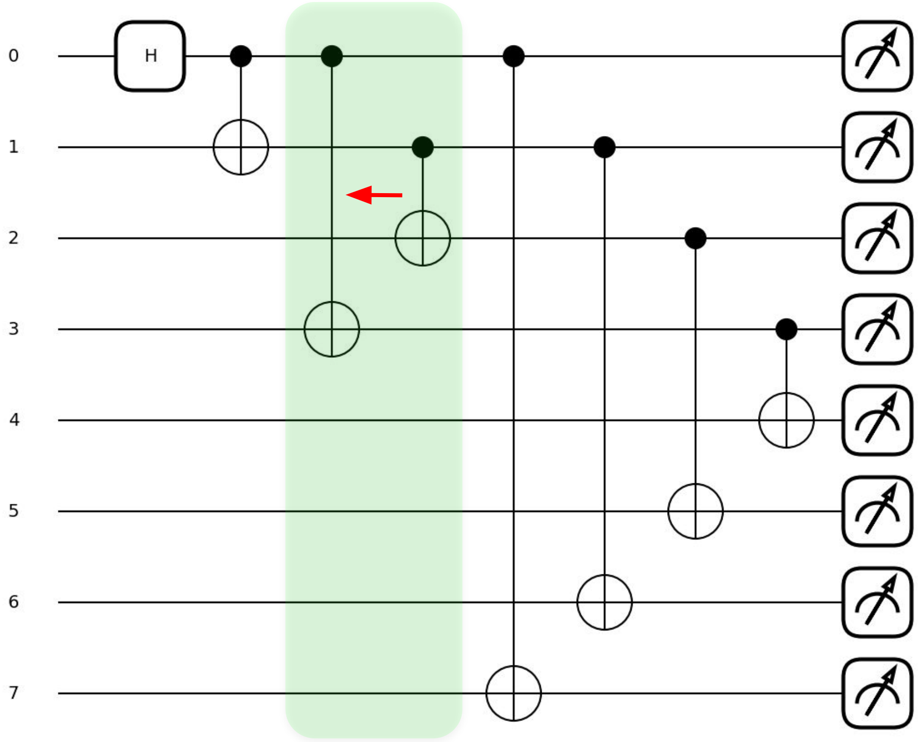
Circuit depth



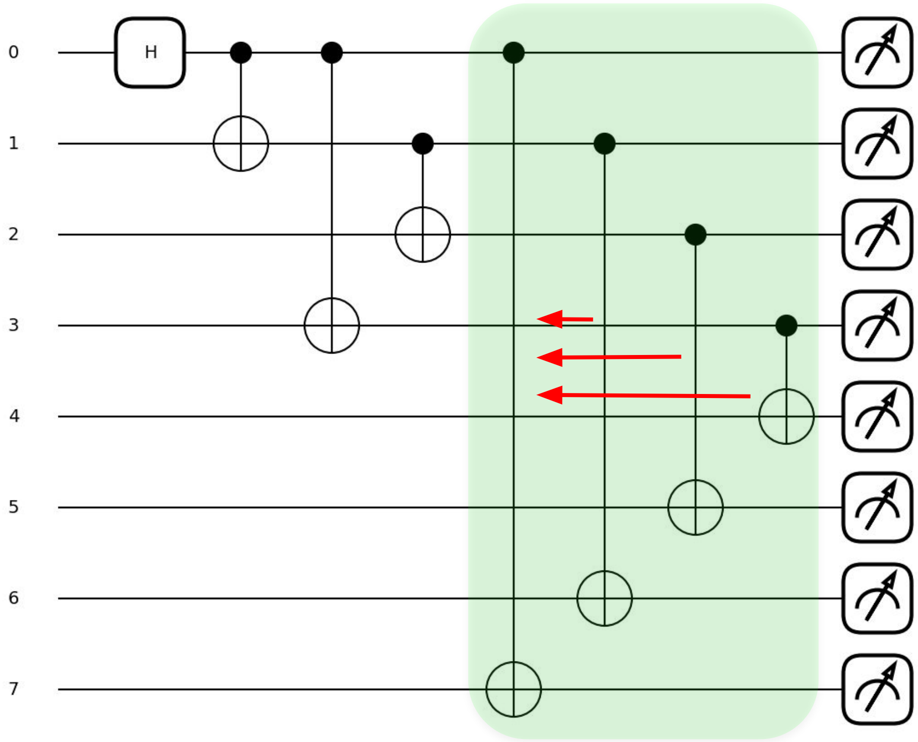
Circuit depth



Circuit depth

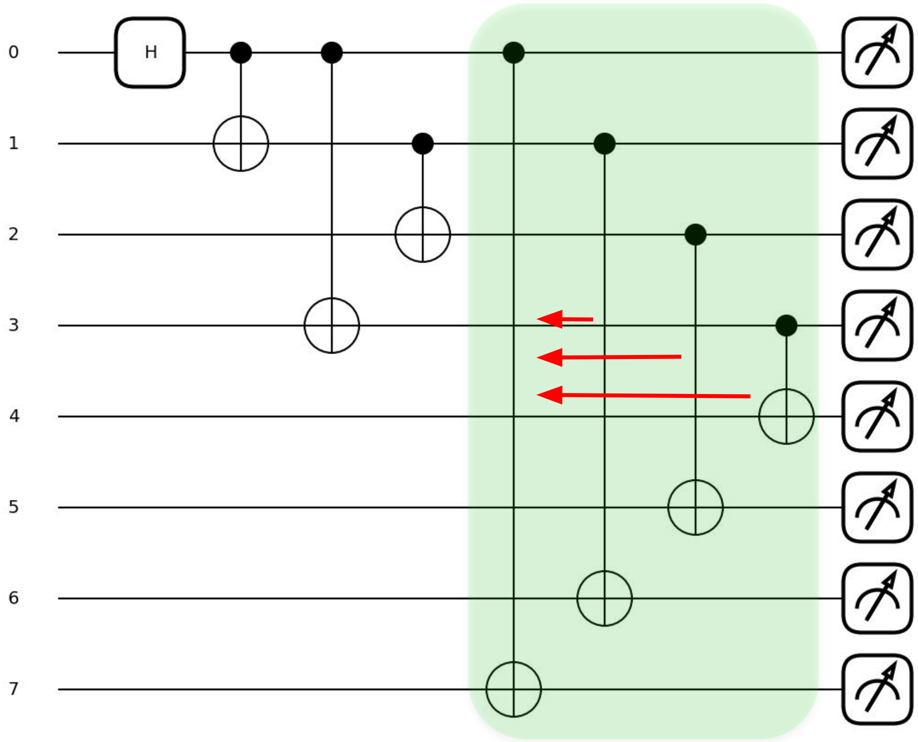


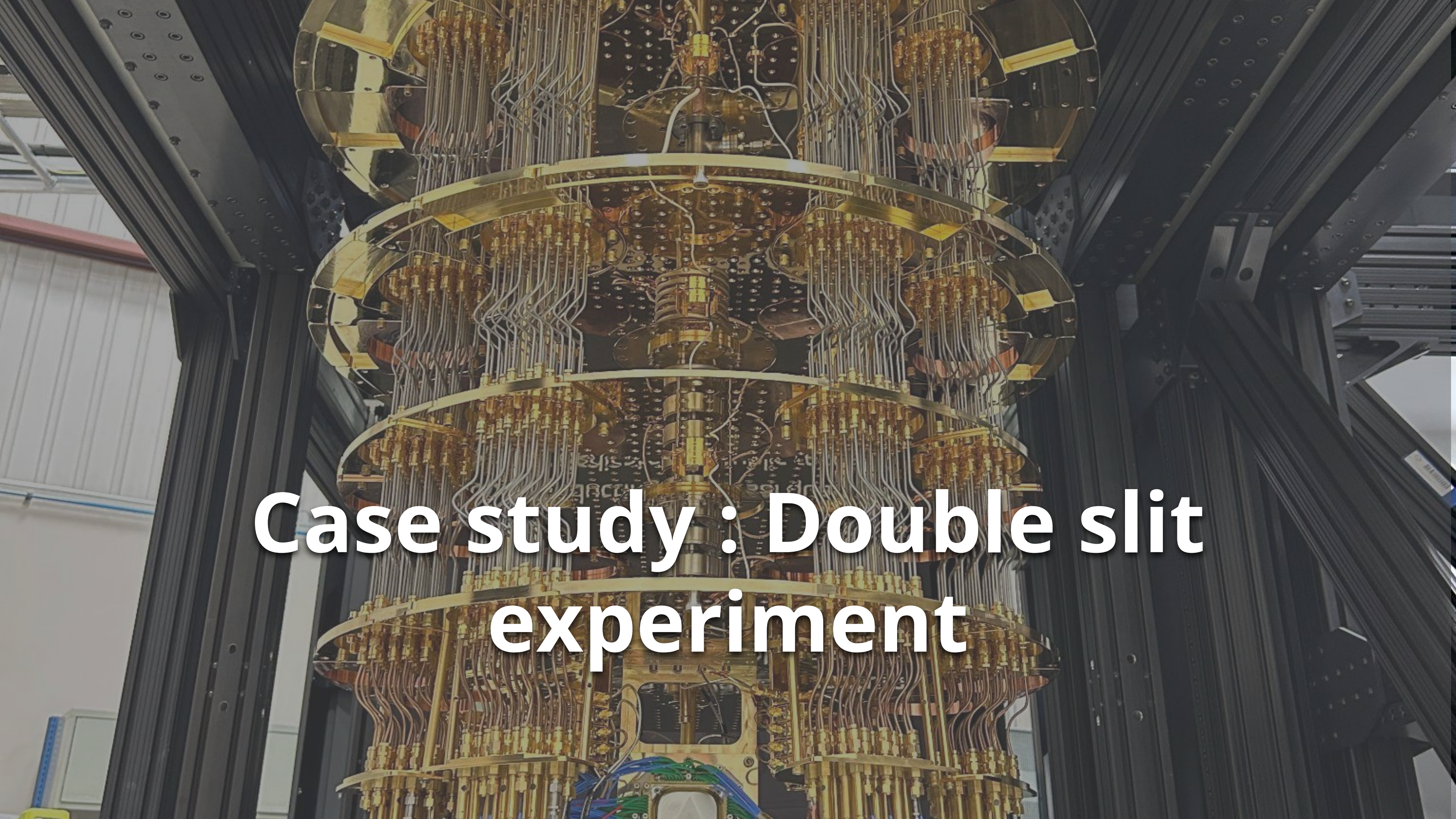
Circuit depth



The depth of this circuit is 4.

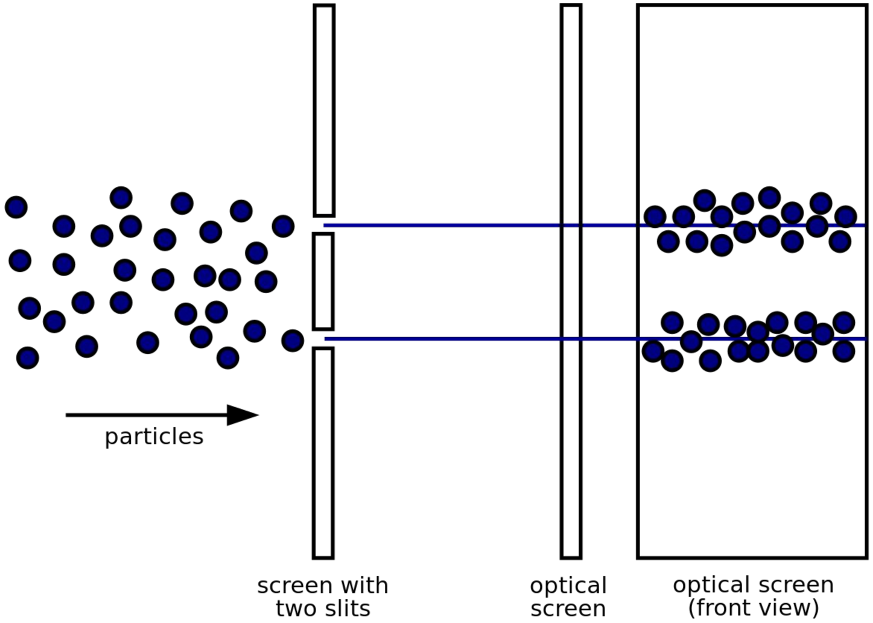
Circuit depth



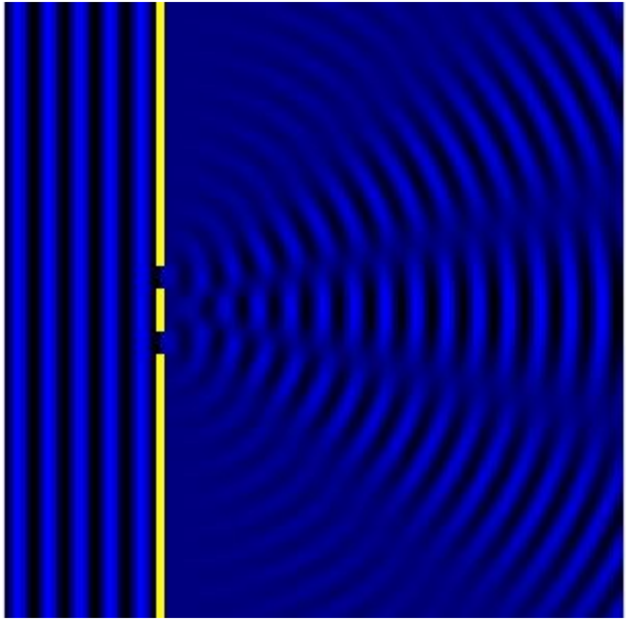


Case study : Double slit experiment

Classical particles



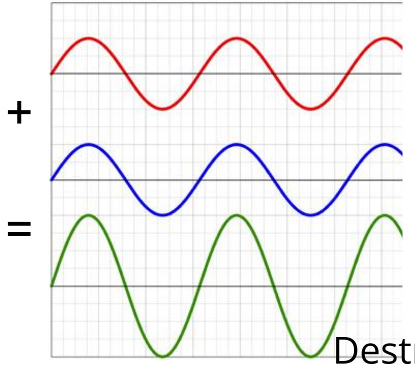
Double slit experiment : constructive and destructive interference



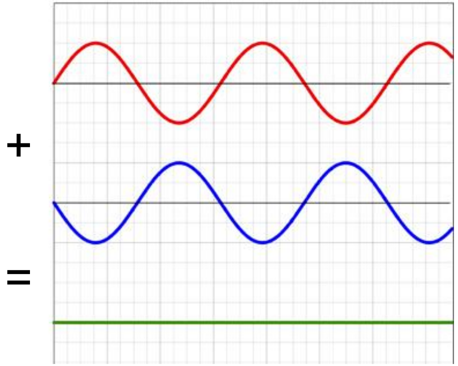
Evolution of the wavefunction as it crosses the two slits

Probability distribution on the screen (measurements)

Constructive

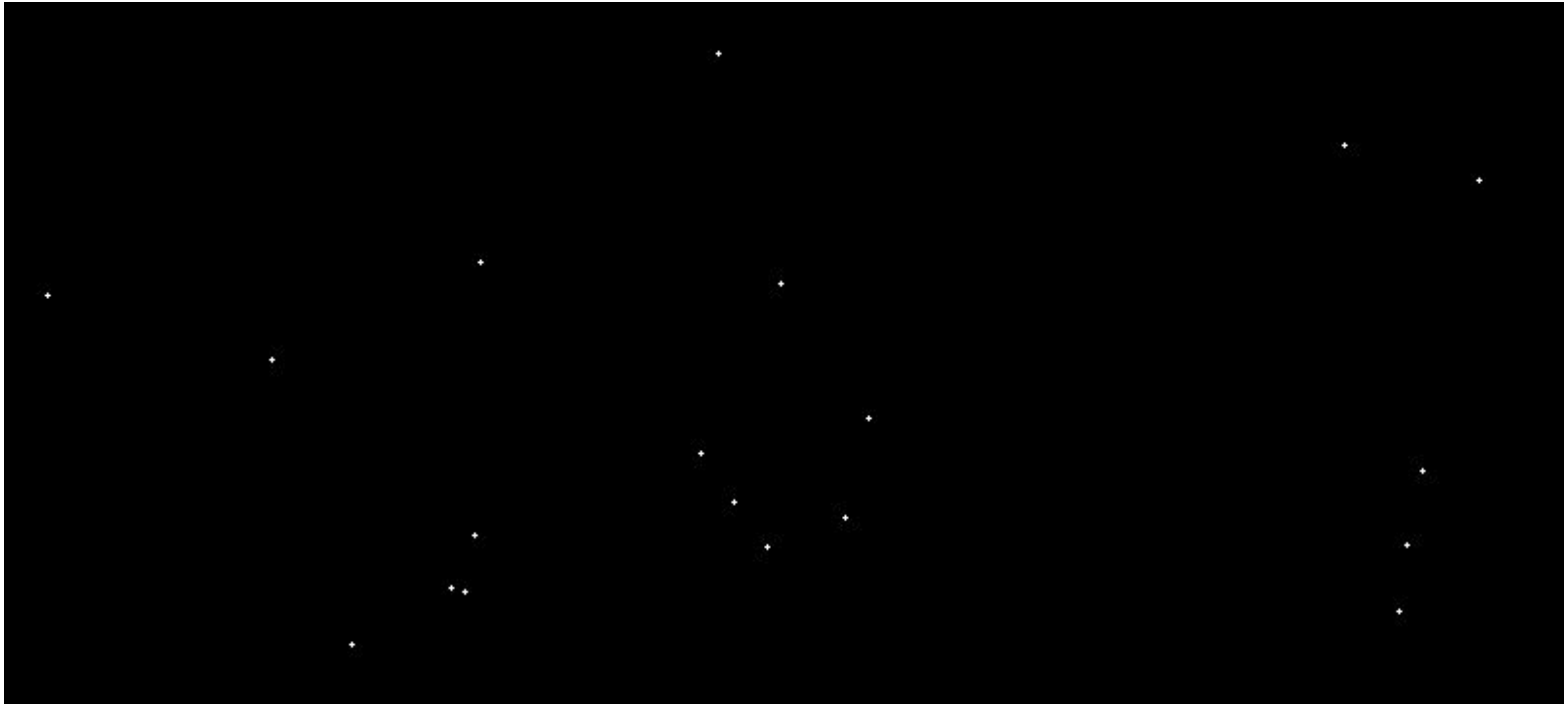


Destructive



Classically, $I(\theta) \propto 1 + \cos(\phi)$

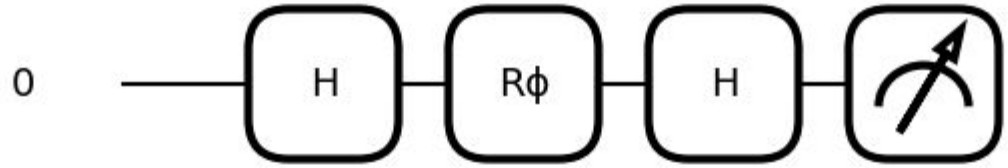
Double slit experiment with electrons : interference pattern



source : https://en.wikipedia.org/wiki/File:Wave-particle_duality.gif

Mapping the double slit experiment to a quantum computer

$$R_\phi(\phi) = e^{i\phi/2} R_z(\phi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix}.$$



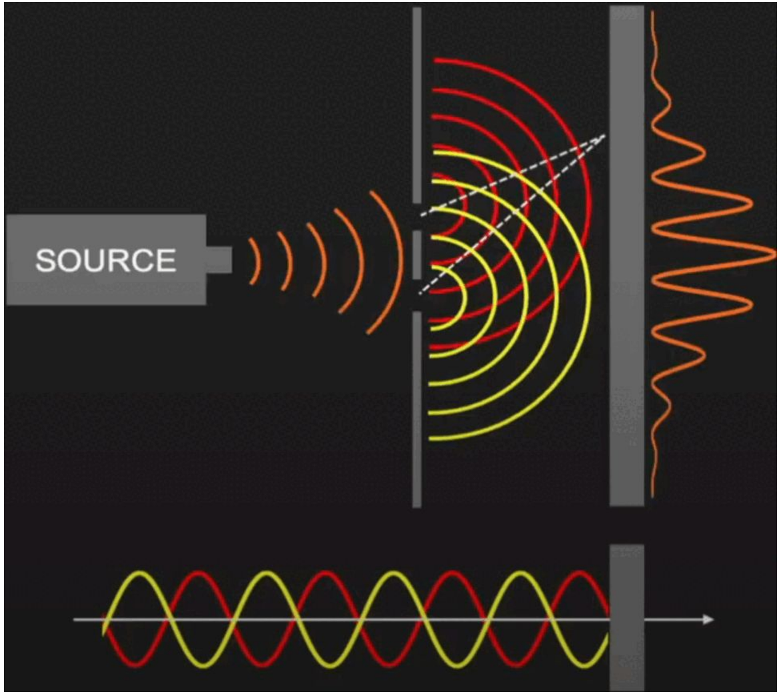
- Hadamard gate : puts qubit into **superposition** i.e the particle passes through both slits simultaneously

- Phase shift gate : Applies a phase shift of phi to state |1>, simulating a **relative dephasing** between the paths taken i.e slit 1 or 2

- Hadamard gate : Returns to computational basis so that we can measure |0> and |1>.

Dephasing

- Phase shift gate : Applies a phase shift of phi to state |1>, simulating a **relative dephasing** between the paths taken i.e slit 1 or 2



Source : https://www.youtube.com/watch?v=8CCgJhsEhFQ

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \longleftarrow \textbf{superposition}$$

$$R_\phi(\phi)\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\phi}|1\rangle\right) \longleftarrow \textbf{Relative dephasing}$$

$$H\left(\frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\phi}|1\rangle\right)\right) \longleftarrow \textbf{Interference}$$

$$H\left(\frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\phi}|1\rangle\right)\right) = \frac{1}{2}\left[(|0\rangle + |1\rangle) + e^{i\phi}(|0\rangle - |1\rangle)\right]$$

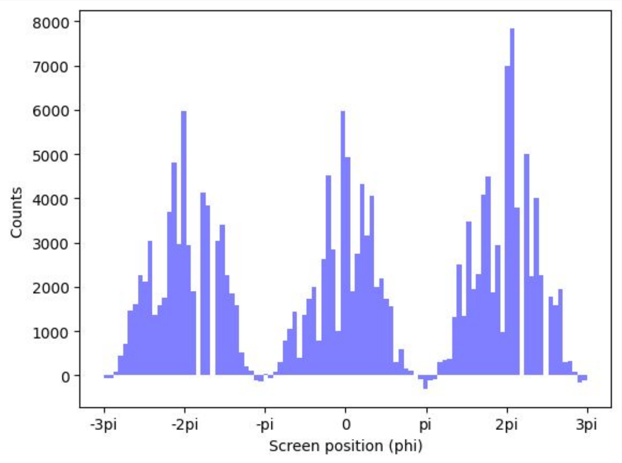
$$= \frac{1}{2}\left[\left(1 + e^{i\phi}\right)|0\rangle + \left(1 - e^{i\phi}\right)|1\rangle\right]$$

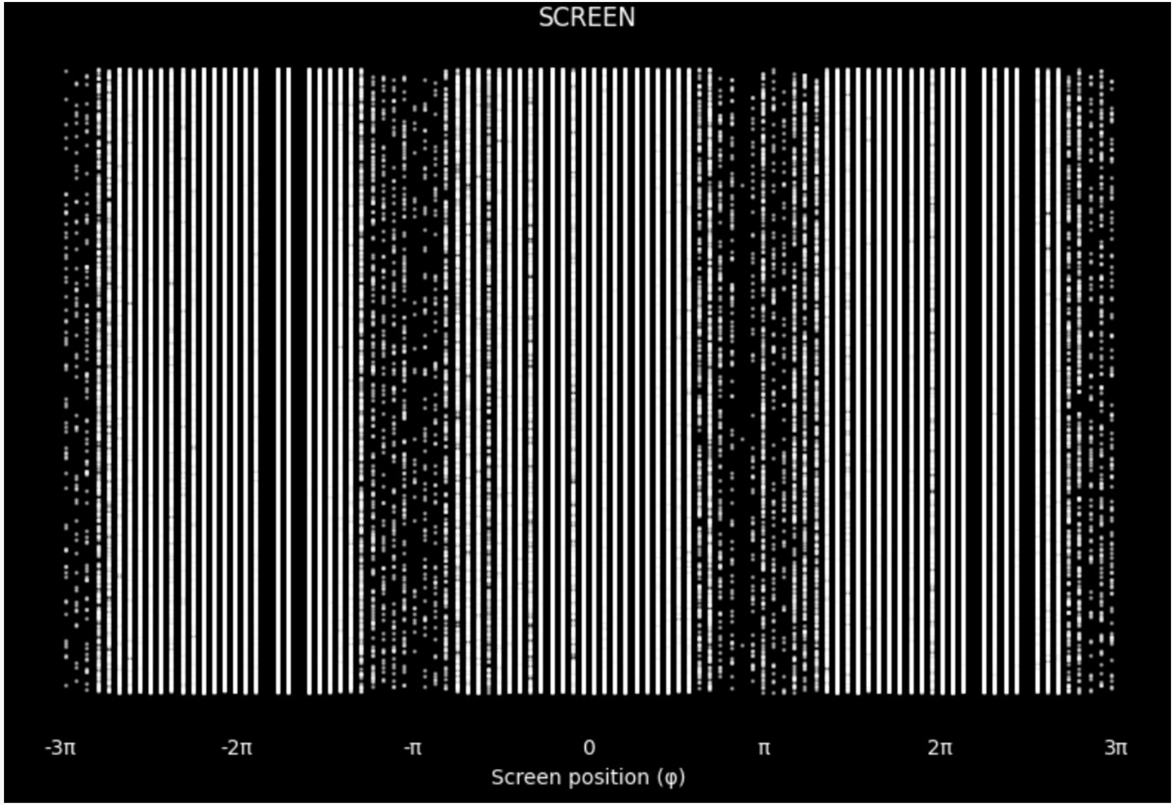
The probability of measuring |0> ie measuring particle at some point on the screen is given by

$$P(0) = \left|\frac{1 + e^{i\phi}}{2}\right|^2 = \frac{1 + \cos(\phi)}{2}$$

with $\phi = 0, P(0) = 1$ constructive interference and $\phi = \pi, P(\pi) = 0$ destructive interference

Double Slit Experiment on MonarQ





Coding break

Notebook 3 :
Mapping the double slit experiment to MonarQ

To go further

- Online documentation for MonarQ https://docs.alliancecan.ca/wiki/MonarQ/en
- More practice in PennyLane https://pennylane.ai/codebook
- Ready to use MonarQ for a research project?
  - Plan an initial consultation and kick off you quantum project by filling out this form
  - General questions? quantum@calculquebec.ca

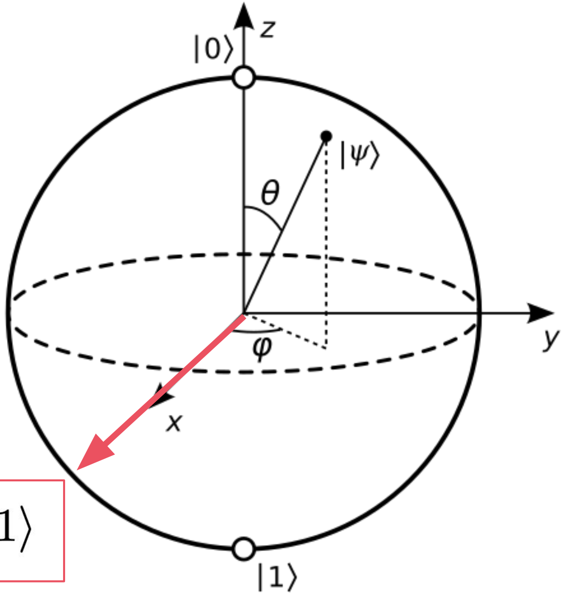
Satisfaction survey

Please take a few minutes to fill out the satisfaction survey and share your thoughts with us!

Questions?

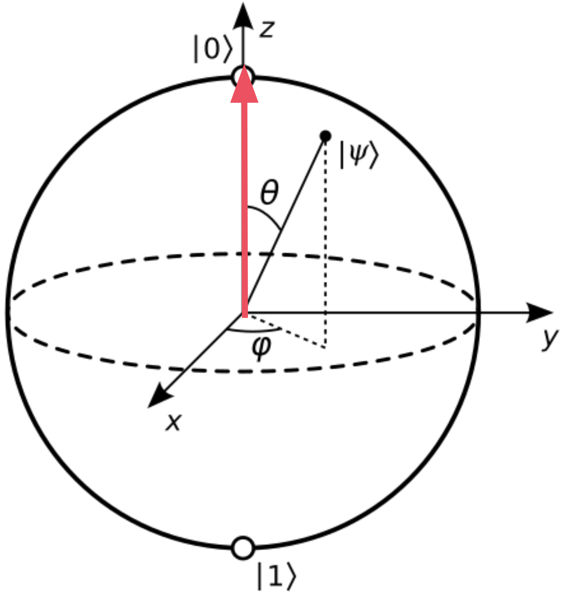
quantum@calculquebec.ca

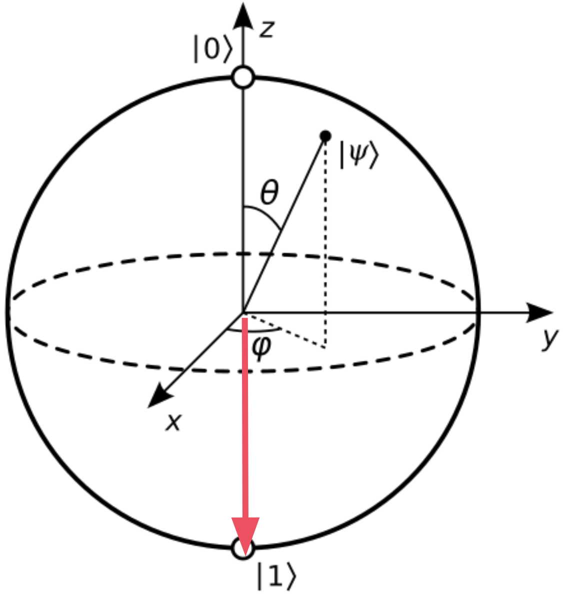
Calcul Québec



$$\boxed{|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + |1\rangle}$$

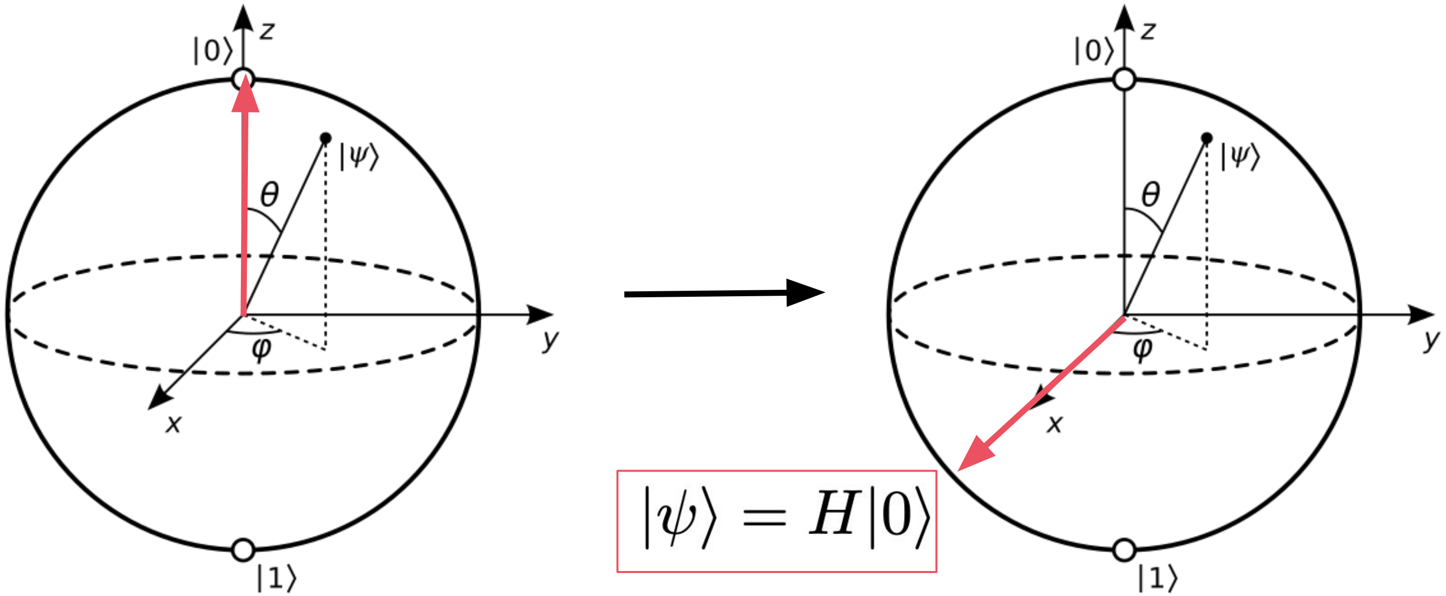
$$|\psi\rangle = |0\rangle$$





$$|\psi\rangle = |1\rangle$$

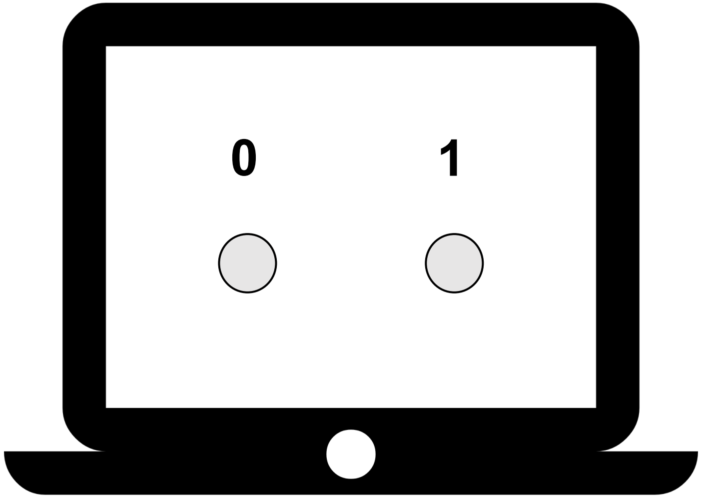
$$|\psi\rangle = |0\rangle$$



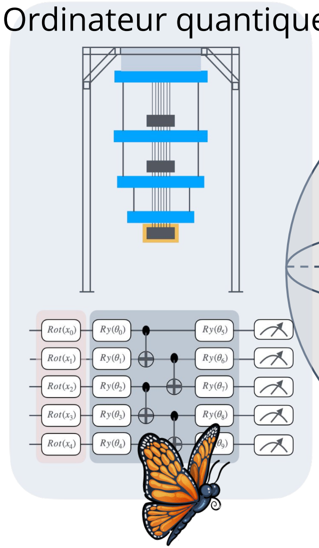
$$|\psi\rangle = H|0\rangle$$

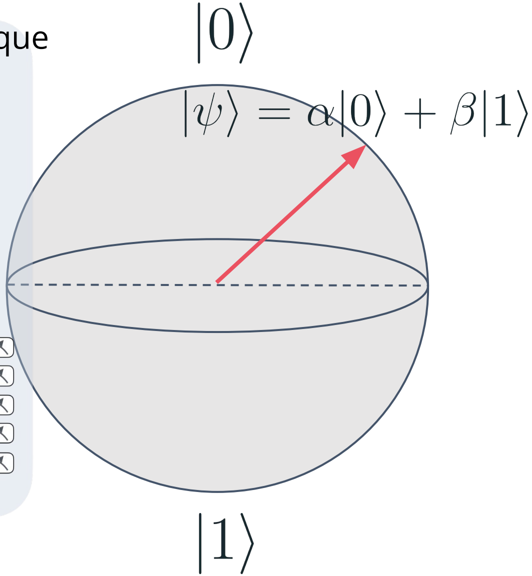
Bits vs qubits

Just like bits, qubits are abstractions.



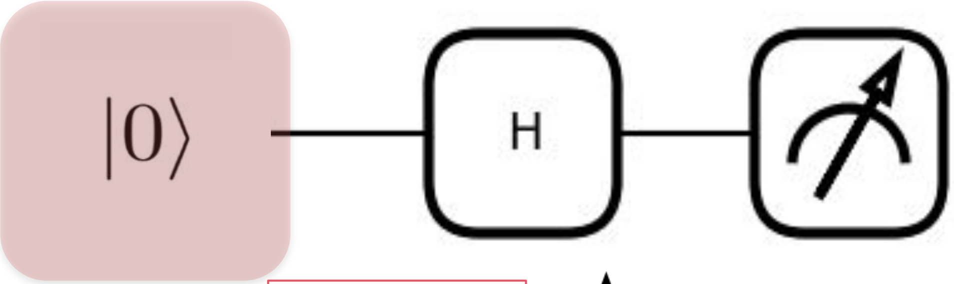
Classical bit



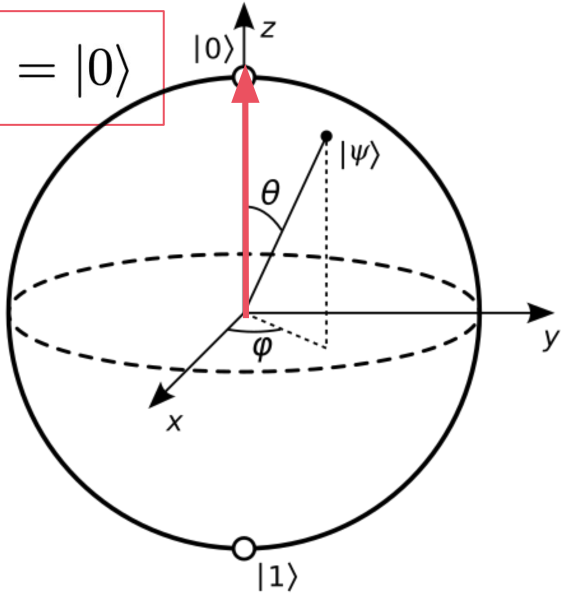


Quantum bit (qubit)

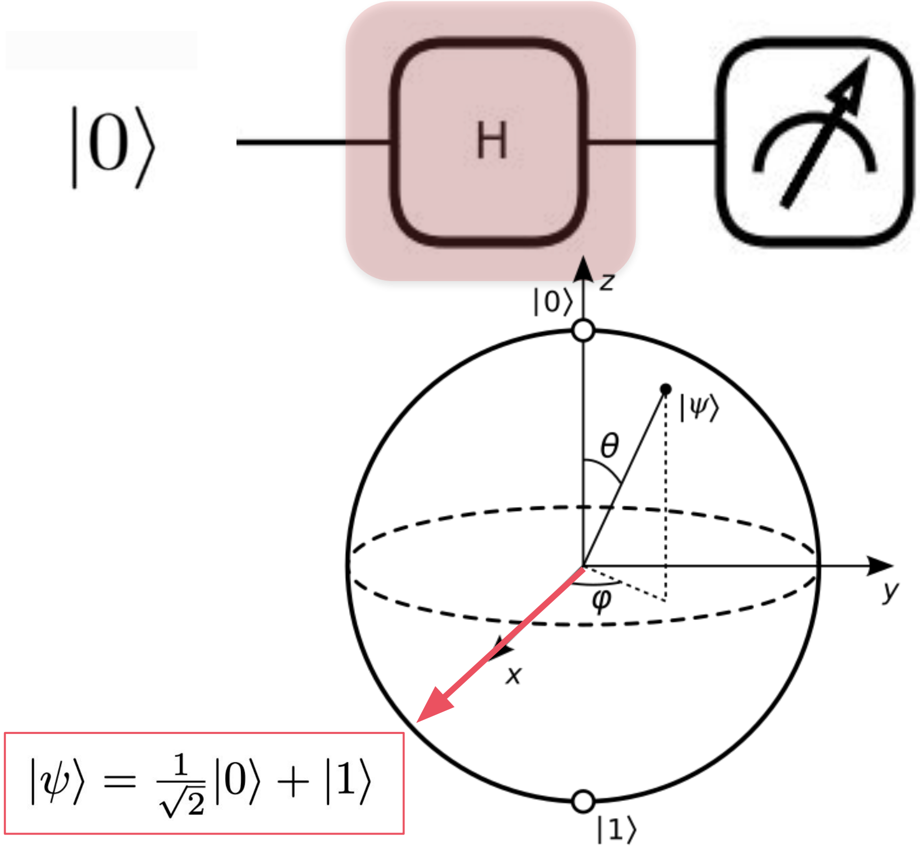
1. Initialize the qubit



$$|\psi\rangle = |0\rangle$$

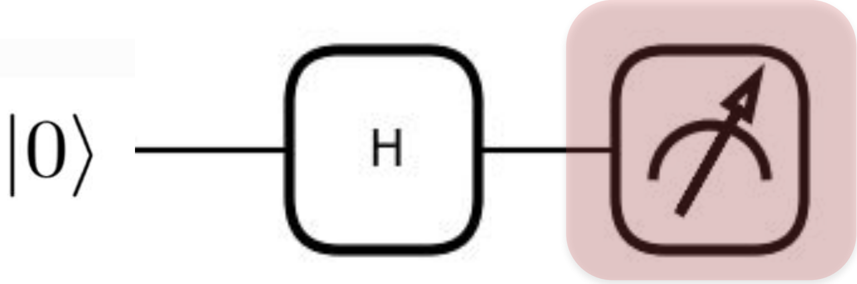


2. Modify the state of the qubit



$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + |1\rangle$$

3. Measure the qubits



Sampling from the
probability distribution,

Measure 0 (50% probability)
Measure 1 (50% probability)