Quantum computers in theory and in practice

Andrea Mari

Unitary Fund

https://unitary.fund

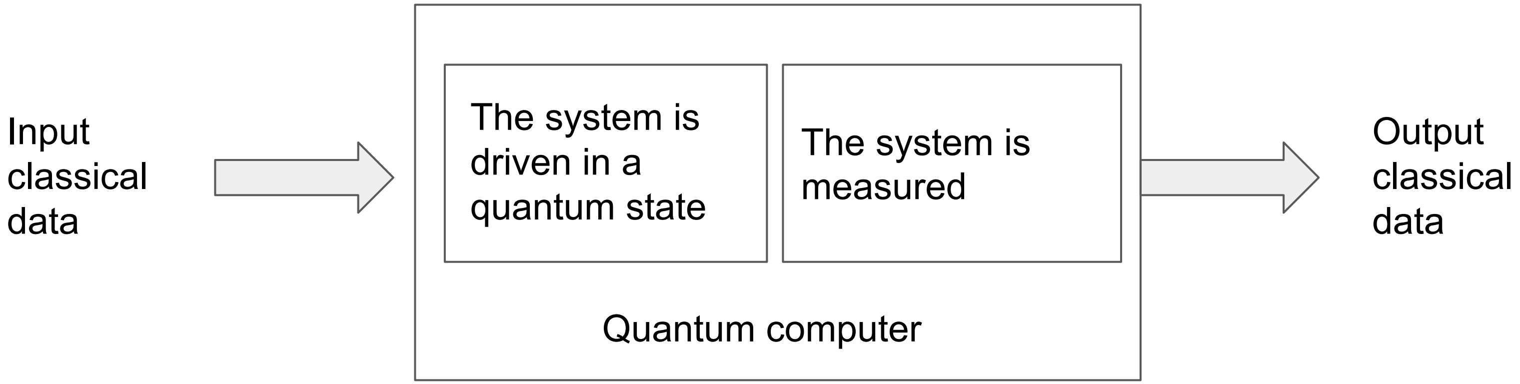
June 7th, 2022 – De Componendis Cifris – PostQuantumCifris

Outline

- What is a quantum computer *in theory*?
- Implications of quantum computers for cryptography
- What is a quantum computer *in practice*?
- What problems can be solved by near-term quantum computers?

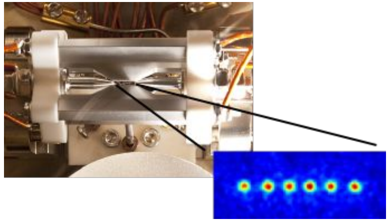
What is a quantum computer?

*A quantum computer is a **physical** machine that **intentionally** uses the **laws of quantum mechanics** to perform computations.*

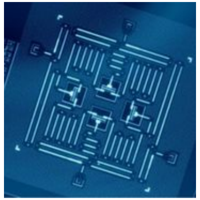


Examples of physical systems that could be used as quantum computers

- Atoms



- Electrons

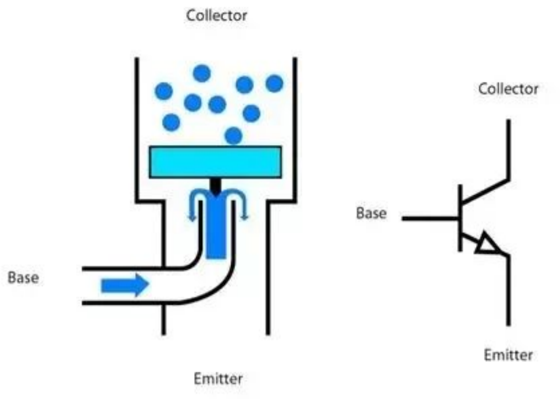


- Photons

Quantum physics is also used in classical computers. But…

Some components of classical computers are based on quantum effects (e.g. transistors).
However, this is irrelevant for the **abstract computational model**.

In principle, the transistors of a classical computer could be replaced by classical water valves!



In a quantum computer instead, even the abstract computational model is based on quantum theory.

The computational model of a quantum computer

Classical computer	Quantum computer	
	$|\alpha|^2 + |\beta|^2 = 1$	
Bit: **0** or **1**	Qubit: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$	[Superposition principle]
Operations map bitstrings to bitstrings	Operations are unitary transformations on qubits.	[Shrödinger equation]
Readout is deterministic.	Readout is binary and non-deterministic: **0** with probability $|\alpha|^2$ **1** with probability $|\beta|^2$	[Born rule]
State is unchanged by measurements	State changes after measurements	[Collapse of the wavefunction]

Mathematical representation of a probabilistic classical computer

Computational state = array of 2^n probabilities

$$\vec{p} = [p_{00...0}, p_{00...1}, \ldots, p_{11...1}]^\top$$

$$\boxed{\|\vec{p}\|_1 = \sum_z p_z = 1}$$

State evolution

$$\boxed{\vec{p'} = T\vec{p}}$$

T is a stochastic matrix: positive elements and preserves || . ||_1

Measurement outcome is a bitstring ***z*** sampled from $\boxed{P(z) = p'_z}$

Mathematical representation of a quantum computer

$$|\psi\rangle = \psi_{00...0}|00...0\rangle + \psi_{00...1}|00...1\rangle + \ldots \psi_{11...1}|11...1\rangle$$

$$\psi_z \in \mathbb{C}$$

Can be represented as a vector of **complex amplitudes** (Hilbert space)

$$\vec{\psi} = \left[\psi_{00...0}, \psi_{00...1}, \ldots, \psi_{11...1}\right]^\top$$

$$\boxed{\left\|\vec{\psi}\right\|_2 = \sum_z |\psi_z|^2 = 1}$$

State evolution

$$\boxed{\vec{\psi}' = U\vec{\psi}}$$

U is a unitary matrix. *U* has complex elements and preserves || . ||_2

Measurement outcome is a bitstring ***z*** sampled from

$$\boxed{P(z) = |\psi'_z|^2}$$

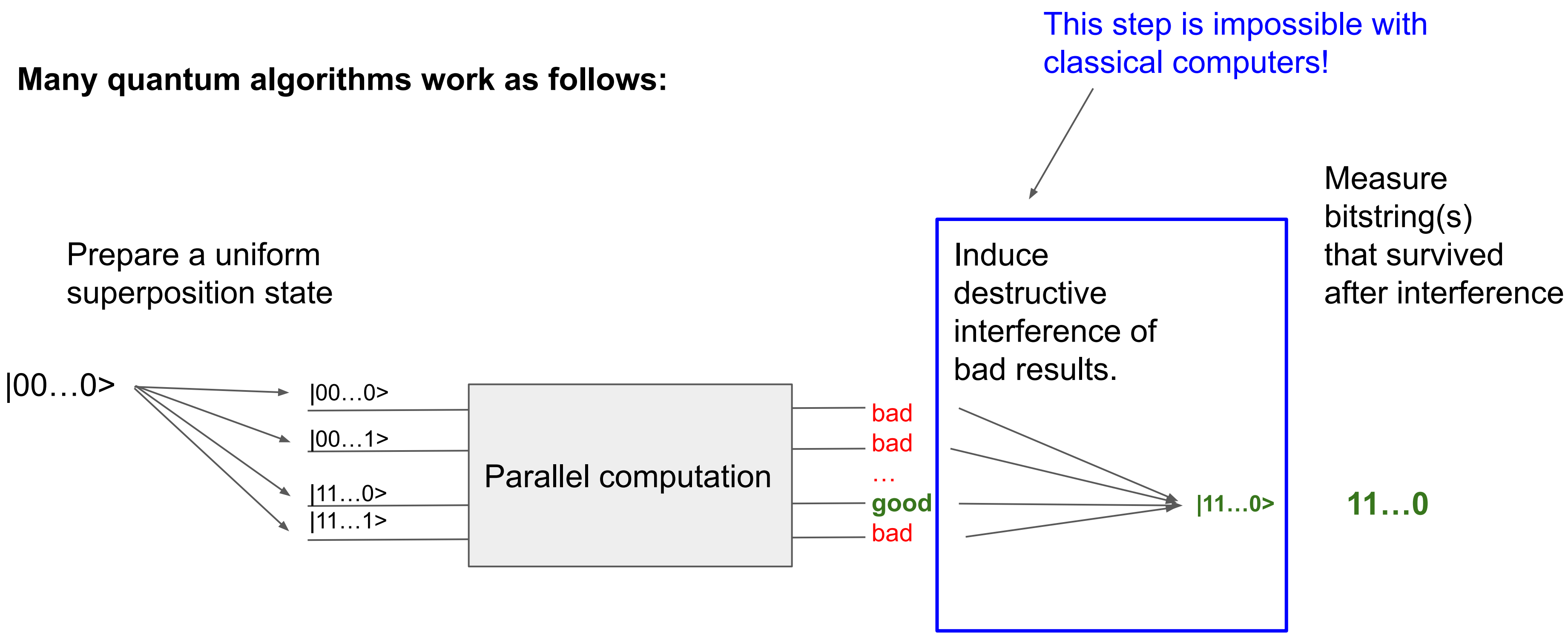
Why complex amplitudes are more powerful than probabilities?

Short answer: because amplitudes can generate **interference**!

Why complex amplitudes are more powerful than probabilities?

Short answer: because amplitudes can generate **interference**!

Many quantum algorithms work as follows:



Outline

- What is a quantum computer *in theory*?
- Implications of quantum computers for cryptography
- What is a quantum computer *in practice*?
- What problems can be solved by near-term quantum computers?

The most famous algorithm: Shor's algorithm

Integer factoring problem

Given an integer number ***N = p q*** which is the product of two prime factors (*p* and *q*), find ***p*** and ***q***.

Computational complexity

Let ***n*** be the number of bits to represent ***N***

Best known classical algorithm scales **exponentially** with ***n.***
Given a solution (*q*, *p)*, one can efficiently verify it.

complexity = NP

At the basis of many cryptographic algorithms e.g. RSA.

Quantum Shor's algorithm scales **polynomially** in ***n*** !

Exponential quantum advantage!

The *second* most famous algorithm: Grover's algorithm

The problem

Given a balck-box function f(z) such that $\begin{cases} f(z) = 1 & \text{if } z = z' \\ f(z) = 0 & \text{if } z \neq z' \end{cases}$, find z'.

Computational complexity

Let N be the number of all possible inputs for f(z). $N = 2^n$ for n-bit inputs.

Best classical algorithm (brute-force research) requires $O(N)$ calls to f(z).

Grover's algorithm requires $O(\sqrt{N})$ calls to f(z) → **Polynomial quantum advantage.**

It can be shown that a better scaling is impossible. ☹

Implications of quantum computers for cryptography

Shor's algorithm

Can break most existing **public-key** cryptographic algorithms!
In particular those based on:

- integer factorization
- discrete logarithm
- elliptic-curves

Retroactive risk!

Encrypted data can be stored **today,** to be decrypted **tomorrow**.

Countermeasures:

Quantum key distribution

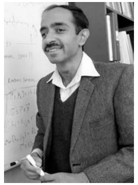
- Information is carried by quantum systems (e.g. photons)
- Hard to implement (requires quantum links)
- Security is based on law of physics.

Post-quantum cryptography

- Information transmitted over conventional classical channels
- Easy to implement
- Security is based on theoretical assumptions

Implications of quantum computers for cryptography

Grover's algorithm



(1996)

Can *weakly* undermine **symmetric** cryptographic algorithms.

Square-root speedup in:

- Brute-force exhaustive algorithms
- Collision attacks
- Function inversion problems

Retroactive risk!

Encrypted data can be stored **today,** to be decrypted **tomorrow**.

Countermeasure: Doubling the number ***n*** of bits, for all types of secret keys.

Quantum cost = $\sqrt{N_{\text{long}}} = \sqrt{2^{2n}} = 2^n = N_{\text{short}}$ = Classical cost

Relationship between quantum computers and cryptography

Beyond the obvious **competition** aspect, there are actually interesting **cooperation** possibilities.

A few examples:

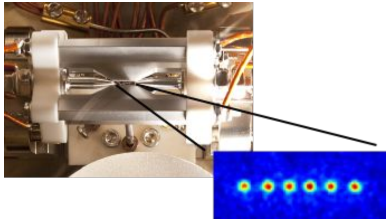
- Using **true quantum randomness** for classical cryptography
- Classical cryptography can be used to encrypt data in quantum computations.
  This is known as **homomorphic encryption.**
- The theory of **classical codes** is at the basis of **quantum error correction**.
- Classical cryptographic algorithms can be used to **verify and test quantum computers**

Outline

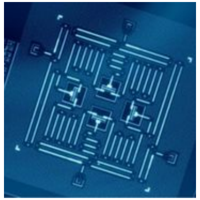
- What is a quantum computer *in theory*?
- Implications of quantum computers for cryptography
- What is a quantum computer *in practice*?
- What problems can be solved by near-term quantum computers?

Small-scale and noisy quantum computers already exist!

- Atoms



- Superconducting circuits

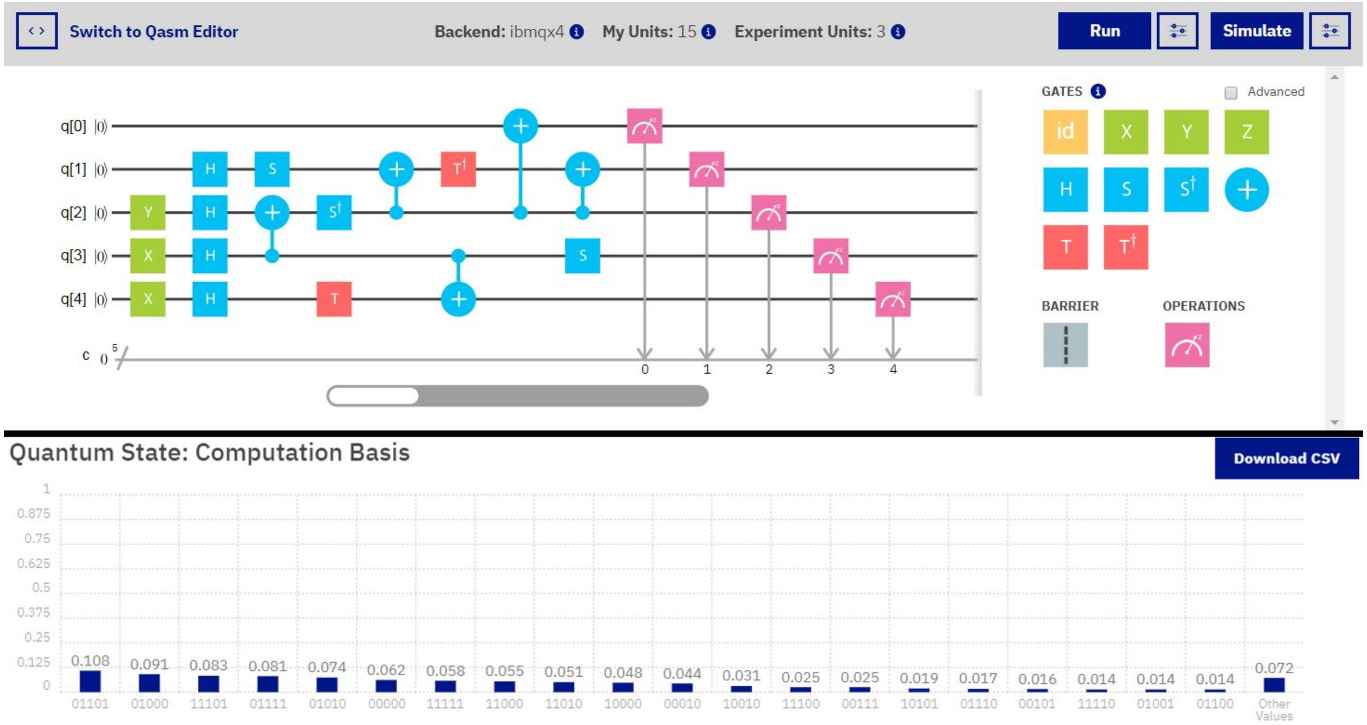


- Photons

PASQAL
IONQ
Honeywell
AQT
|QuEra⟩ COMPUTING INC.
ColdQuanta
QUANTUM BRILLIANCE
IBM Q
Google Quantum AI
QuTech
rigetti
D-Wave The Quantum Computing Company™
AQT
IQM
OQC
XANADU
ORCA Computing
1958 University of Science and Technology of China
PsiQuantum
QUANDELA

Quantum computers can be easily programmed by users

Example: IBM quantum experience.



Quantum software

In practice, one can program quantum computers via:

- **Software libraries** for conventional languages
  - Mostly Python based
  - Mostly **open source**

Cirq	Qiskit	pyQuil	Braket	PennyLane

- **Quantum-specific languages**
  - Mostly **open source**

(non-exhaustive lists)

A quantum open source ecosystem

qosf

A large quantum open source ecosystem is rapidly growing. See e.g. https://qosf.org/project_list/

Unitary Fund non-profit helping create a quantum technology ecosystem that benefits the most people.

- Microgrant program (4k $ per grant)
- Community events like ***unitaryhack***
- mitiq , a quantum error mitigation library
- metriq, a web platform for community-driven quantum benchmarks.

unitaryhack.dev

For more details, please have a look at our website https://unitary.fund

Outline

- What is a quantum computer *in theory*?
- Implications of quantum computers for cryptography
- What is a quantum computer *in practice*?
- What problems can be solved by near-term quantum computers?

In practice, existing quantum computers are still very **small** and **noisy**.

As figures of merit,

- Most current circuit-based quantum computers have less than ~**150 qubits**.
- **Beyond 10 qubits** NISQ computations are **dominated by noise**.

They have been named **NISQ** devices. (**N**oisy-**I**ntermediate **S**cale **Q**uantum devices)

John Preskill, Quantum 2, 79 (2018)

Two key questions:

- What is the computational power of NISQ devices?
- How far is a fault-tolerant quantum computer?

What problems can NISQ computers solve?

Example of problems:

- Quantum chemistry problems (e.g. energy spectrum of molecules)
- Simulation of quantum dynamics (e.g. simulation of high-energy physics)
- Optimization problems (e.g. MaxCut)
- ~~Shor's algorithm~~
- ~~Grover's algorithm~~ (they need to many clean qubits)

☹ Current computers are still too small to compete with classical methods.

☺ **But they could become competitive within the next ~5 years!**

⚠ It is not theoretically clear if a NISQ quantum advantage is possible at all for "useful problems"

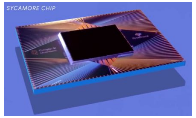
Artificial problems and quantum supremacy

Example of artificial “useless” problems:

- Sampling from the output state of a random circuit
- Sampling from the output state of a random Gaussian photonic circuit

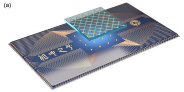
Quantum theoretically and experimentally proved!

- 2019 US, Google Sycamore, 53 qubits

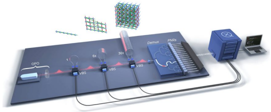


- 2020 China, Jiuzhang, 50 photonic modes

- 2021 China, Zuchongzhi, 56 superconducting qubits



- 2022 Canada, Xanadu, Borealis, 216 photonic modes



What about Shor’s and Grover’s algorithms?

☹ Impossible to run on near-term NISQ computers.

- Shor/Gover’s algorithms require **millions** of qubits
- Shor/Gover’s algorithms require **fault-tolerant** quantum computers.

Very hard to make predictions for a fault-tolerant QC

- It could take 30, 50 or even 100 years.
- Soon or later it’s going to happen.

So, is classical cryptography secure?

- For many years it will be ok.
- However, remember the retroactive risk! ⚠

(A classical computer in 1953)

Conclusions

Unitary Fund

https://unitary.fund

- *In theory*, quantum computers are very powerful.
    - So powerful to break classical cryptography
- *In practice*, NISQ quantum computers quite powerful too.
    - So powerful to achieve quantum supremacy
    - But not enough to break classical cryptography

Keep in mind the retroactive risk!

Two open questions

- Can new NISQ algorithms be dangerous for classical cryptography?
- Beyond competing: Can cryptography and quantum computing cooperate?