

A Practical Introduction to Quantum Computing: From Qubits to Quantum Machine Learning and Beyond

Elías F. Combarro
combarro@gmail.com

CERN openlab (Geneva, Switzerland) - University of Oviedo (Oviedo, Spain)

CERN - November/December 2020



Universidad de Oviedo

Part I

Introduction: quantum computing...
the end of the world as we know it?

I, for one, welcome our new quantum overlords

NEWS

QUANTUM PHYSICS

Google officially lays claim to quantum supremacy

A quantum computer reportedly beat the most powerful supercomputers at one type of calculation

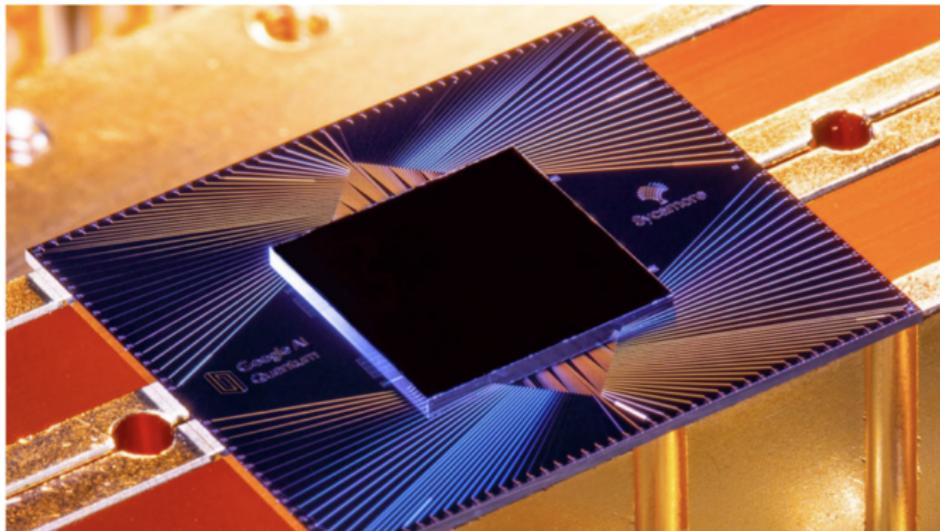


Image credits: sciencenews.org

Philosophy of the course

If you can't
explain it to a
computer
you don't
understand it
yourself.

ALBERT EINSTEIN

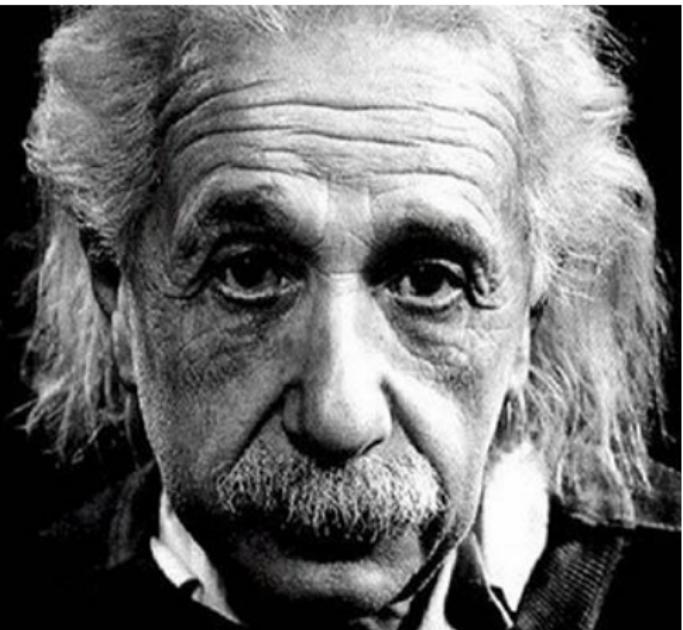


Image credits: Modified from an Instagram image by Bob MacGuffie

Tools and resources

- Jupyter Notebooks
 - Web application to create and execute notebooks that include code, images, text and formulas
 - They can be used locally (Anaconda) or in the cloud (mybinder.org, Google Colab...)
- IBM Quantum Experience
 - Free online access to quantum simulators (up to 32 qubits) and **actual quantum computers** (1, 5 and 15 qubits) with different topologies
 - Programmable with a visual interface and via different languages (python, qasm, Jupyter Notebooks)
 - Launched in May 2016
 - <https://quantum-computing.ibm.com/>

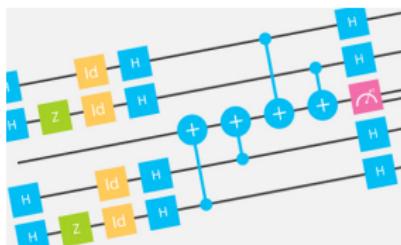


Image credits: IBM

Tools and resources (2)

- Quirk
 - Online simulator (up to 16 qubits)
 - Lots of different gates and visualization options
 - <http://algassert.com/quirk>
- D-Wave Leap
 - Access to D-Wave quantum computers
 - Ocean: python library for quantum annealing
 - Problem specific (QUBO, Ising model...)
 - <https://www.dwavesys.com/take-leap>



The shape of things to come



Image credits: Created with wordclouds.com

What is quantum computing?

Quantum computing

Quantum computing is a computing paradigm that exploits quantum mechanical properties (superposition, entanglement, interference...) of matter in order to do calculations

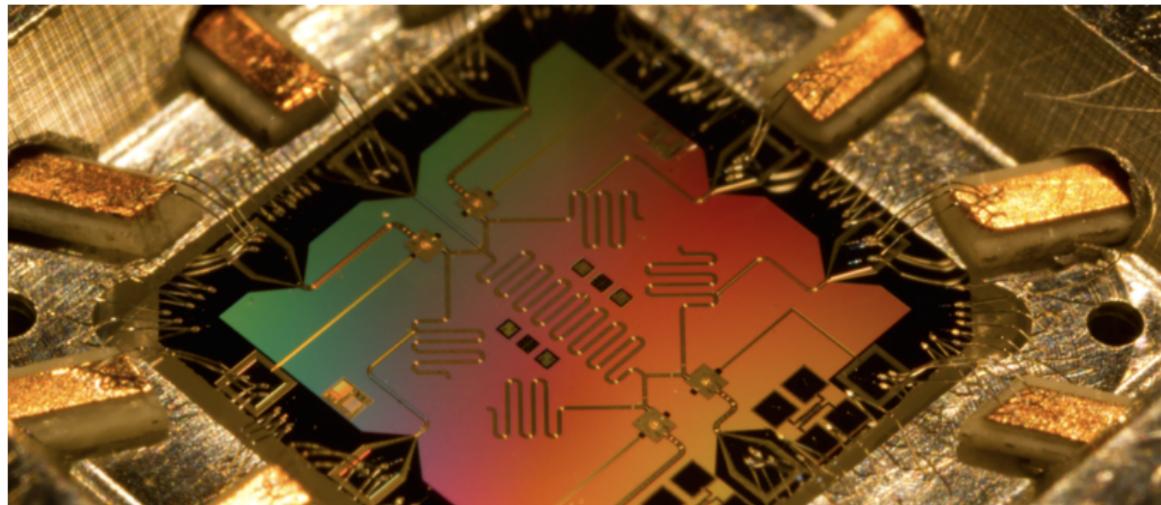


Image credits: Erik Lucero

Models of quantum computing

- There are several models of quantum computing (they're all equivalent)
 - Quantum Turing machines
 - **Quantum circuits**
 - Measurement based quantum computing (MBQC)
 - Adiabatic quantum computing
 - Topological quantum computing
- Regarding their **computational capabilities**, they are equivalent to classical models (Turing machines)

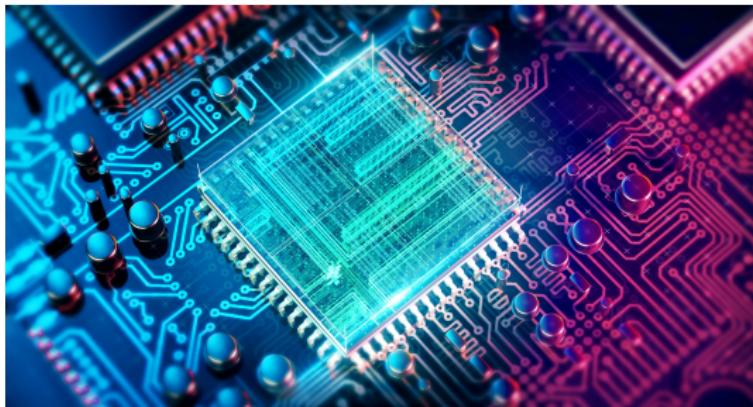


Image credits: Getty Images

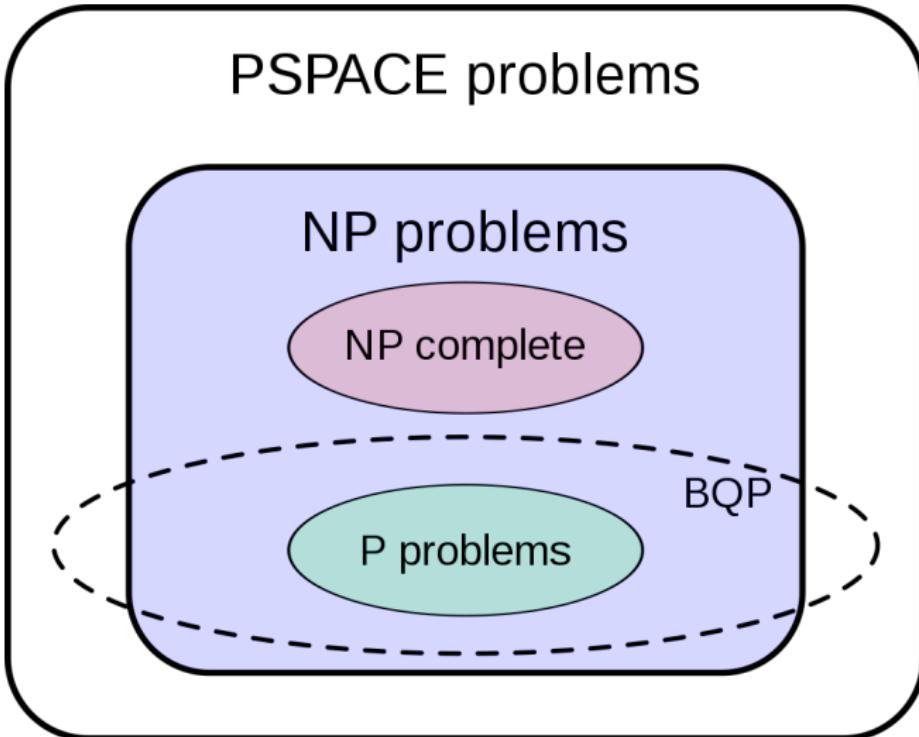
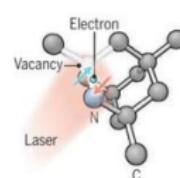
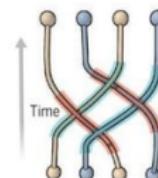
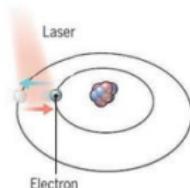
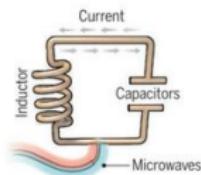


Image credits: wikipedia.org

What technologies are used to build quantum computers?



Superconducting loops

Company support

Google, IBM, Quantum Circuits

Pros

Fast working. Build on existing semiconductor industry.

Cons

Collapse easily and must be kept cold.

Trapped ions

ionQ
Very stable. Highest achieved gate fidelities.

Silicon quantum dots

Intel
Stable. Build on existing semiconductor industry.

Topological qubits

Microsoft, Bell Labs
Greatly reduce errors.

Diamond vacancies

Quantum Diamond Technologies
Can operate at room temperature.

Image credits: Graphic by C. Bickle/Science data by Gabriel Popkin

What is a quantum computer like?

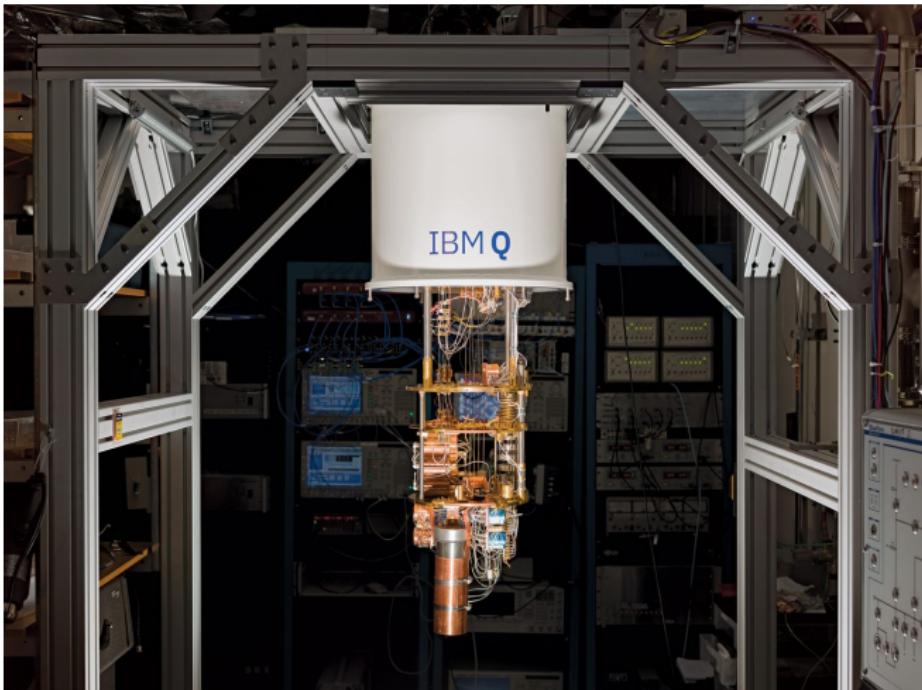


Image credits: IBM

The Sounds of IBM: IBM Q

Programming a quantum computer

- Different frameworks and programming languages:
 - qasm
 - Qiskit (IBM)
 - Cirq (Google)
 - Forest/pyqil (Rigetti)
 - Q# (Microsoft)
 - Ocean (D-Wave)
 - ...
- Most of them for quantum circuit specification

Switch to Composer

Backend: Custom Topology Experiment Units: 3

Simulate

```
1 OPENQASM 2.0;
2     include "qelib1.inc";
3     qreg q[3];
4     creg c0[1];
5     creg c1[1];
6     creg c2[1];
7
8     gate post q { }
9     u3(0.3,0.2,0.1) q[0];
10    h q[1];
11    cx q[0],q[2];
12    barrier q[1];
13    cx q[0],q[1];
14    h q[0];
15    measure q[0] -> c0[0];
16    measure q[1] -> c1[0];
17    if(c0==1) x q[2];
18    if(c1==1) x q[2];
19    post q[2];
20    measure q[2] -> c2[0];
21
22
```

Import QASM Download QASM

Image credits: IBM

What are the elements of a quantum circuit?

- Every computation has three elements: data, operations and results
- In quantum circuits:
 - Data = **qubits**
 - Operations = **quantum gates** (unitary transformations)
 - Results = **measurements**



Image credits: Adobe Stock

Part II

One-qubit systems: one qubit to rule them all

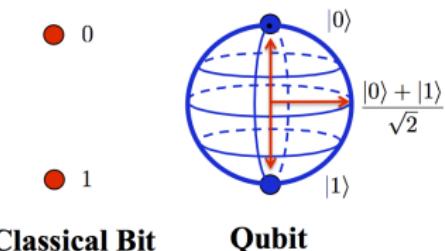
What is a qubit?

- A classical bit can take two different values (0 or 1). It is discrete.
- A qubit can “take” **infinitely** many different values. It is continuous.
- Qubits live in a **Hilbert vector space** with a basis of two elements that we denote $|0\rangle$ y $|1\rangle$.
- A generic qubit is in a **superposition**

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

where α and β are **complex numbers** such that

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



Measuring a qubit

- The way to know the value of a qubit is to perform a measurement. However
 - The result of the measurement is random
 - When we measure, we only obtain one (classical) bit of information
- If we measure the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ we get 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$.
- Moreover, the new state after the measurement will be $|0\rangle$ or $|1\rangle$ depending of the result we have obtained (wavefunction collapse)
- We cannot perform several independent measurements of $|\psi\rangle$ because we cannot copy the state (**no-cloning theorem**)



What are quantum gates?

- Quantum mechanics tells us that the evolution of an isolated state is given by the Schrödinger equation

$$H(t)|\psi(t)\rangle = i\hbar \frac{\partial}{\partial t}|\psi(t)\rangle$$

- In the case of quantum circuits, this implies that the operations that can be carried out are given by unitary matrices. That is, matrices U of complex numbers verifying

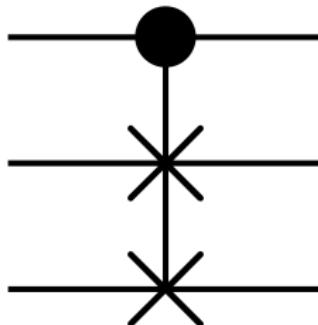
$$UU^\dagger = U^\dagger U = I$$

where U^\dagger is the conjugate transpose of U .

- Each such matrix is a possible quantum gate in a quantum circuit

Reversible computation

- As a consequence, all the operations have an inverse:
reversible computing
- Every gate has the same number of inputs and outputs
- We cannot directly implement some classical gates such as *or*, *and*, *nand*, *xor*...
- But we can simulate any classical computation with small overhead
- Theoretically, we could compute without wasting energy
(Landauer's principle, 1961)



One-qubit gates

- When we have just one qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, we usually represent it as a column vector $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$
- Then, a one-qubit gate can be identified with a matrix $U = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ that satisfies

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \bar{a} & \bar{c} \\ \bar{b} & \bar{d} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

where $\bar{a}, \bar{b}, \bar{c}, \bar{d}$ are the conjugates of complex numbers a, b, c, d .

Action of a one-qubit gate

- A state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is transformed into

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} a\alpha + b\beta \\ c\alpha + d\beta \end{pmatrix}$$

that is, into the state $|\psi\rangle = (a\alpha + b\beta)|0\rangle + (c\alpha + d\beta)|1\rangle$

- Since U is unitary, it holds that

$$|(a\alpha + b\beta)|^2 + |(c\alpha + d\beta)|^2 = 1$$

The X or NOT gate

- The X gate is defined by the (unitary) matrix

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

- Its action (in quantum circuit notation) is

$$|0\rangle \xrightarrow{\boxed{X}} |1\rangle$$

$$|1\rangle \xrightarrow{\boxed{X}} |0\rangle$$

that is, it acts like the classical NOT gate

- On a general qubit its action is

$$\alpha |0\rangle + \beta |1\rangle \xrightarrow{\boxed{X}} \beta |0\rangle + \alpha |1\rangle$$

The Z gate

- The Z gate is defined by the (unitary) matrix

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

- Its action is

$$|0\rangle \xrightarrow{\boxed{Z}} |0\rangle$$

$$|1\rangle \xrightarrow{\boxed{Z}} -|1\rangle$$

The H or Hadamard gate

- The H or Hadamard gate is defined by the (unitary) matrix

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

- Its action is

$$|0\rangle \xrightarrow{H} \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

$$|1\rangle \xrightarrow{H} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

- We usually denote

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

and

$$|-\rangle := \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Other important gates

- Y gate

$$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

- S gate

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix}$$

- T gate

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$$

- The gates X , Y and Z are also called, together with the identity, the Pauli gates. An alternative notation is σ_X , σ_Y , σ_Z .

The Bloch sphere

- A common way of representing the state of a qubit is by means of a point in the surface of the Bloch sphere
- If $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ with $|\alpha|^2 + |\beta|^2 = 1$ we can find angles γ, δ, θ such that

$$\alpha = e^{i\gamma} \cos \frac{\theta}{2}$$

$$\beta = e^{i\delta} \sin \frac{\theta}{2}$$

- Since an overall phase is physically irrelevant, we can rewrite

$$|\psi\rangle = \cos \frac{\theta}{2}|0\rangle + e^{i\varphi} \sin \frac{\theta}{2}|1\rangle$$

with $0 \leq \theta \leq \pi$ and $0 \leq \varphi < 2\pi$.

The Bloch sphere (2)

- From $|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle$ we can obtain spherical coordinates for a point in \mathbb{R}^3

$$(\sin \theta \cos \varphi, \sin \theta \sin \varphi, \cos \theta)$$

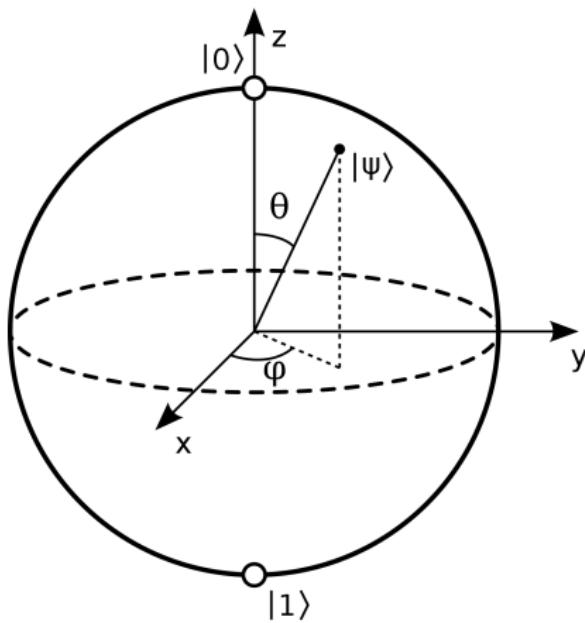


Image credits: wikipedia.org

Rotation gates

- We can define the following rotation gates

$$R_X(\theta) = e^{-i\frac{\theta}{2}X} = \cos \frac{\theta}{2}I - i \sin \frac{\theta}{2}X = \begin{pmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}$$

$$R_Y(\theta) = e^{-i\frac{\theta}{2}Y} = \cos \frac{\theta}{2}I - i \sin \frac{\theta}{2}Y = \begin{pmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}$$

$$R_Z(\theta) = e^{-i\frac{\theta}{2}Z} = \cos \frac{\theta}{2}I - i \sin \frac{\theta}{2}Z = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$$

- Notice that $R_X(\pi) \equiv X$, $R_Y(\pi) \equiv Y$, $R_Z(\pi) \equiv Z$,
 $R_Z(\frac{\pi}{2}) \equiv S$, $R_Z(\frac{\pi}{4}) \equiv T$

Using rotation gates to generate one-qubit gates

- For any one-qubit gate U there exist a unit vector $r = (r_x, r_y, r_z)$ and an angle θ such that

$$U \equiv e^{-i\frac{\theta}{2}r \cdot \sigma} = \cos \frac{\theta}{2} I - i \sin \frac{\theta}{2} (r_x X + r_y Y + r_z Z)$$

- For instance, choosing $\theta = \pi$ and $r = (\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}})$ we can see that

$$H \equiv e^{-i\frac{\theta}{2}r \cdot \sigma} = -i \frac{1}{\sqrt{2}} (X + Z)$$

- Additionally, it can also be proved that there exist angles α , β and γ such that

$$U \equiv R_Z(\alpha)R_Y(\beta)R_Z(\gamma)$$

Inner product, Dirac's notation and Bloch sphere

- The inner product of two states $|\psi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ and $|\psi_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$ is given by

$$\langle\psi_1|\psi_2\rangle = (\overline{\alpha_1} \ \overline{\beta_1}) \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix} = \overline{\alpha_1}\alpha_2 + \overline{\beta_1}\beta_2$$

- Notice that $\langle 0|0\rangle = \langle 1|1\rangle = 1$ and $\langle 0|1\rangle = \langle 1|0\rangle = 0$
- This allows us to compute

$$\begin{aligned}\langle\psi_1|\psi_2\rangle &= (\overline{\alpha_1}\langle 0| + \overline{\beta_1}\langle 1|)(\alpha_2|0\rangle + \beta_2|1\rangle) \\ &= \overline{\alpha_1}\alpha_2\langle 0|0\rangle + \overline{\alpha_1}\beta_2\langle 0|1\rangle + \overline{\beta_1}\alpha_2\langle 1|0\rangle + \overline{\beta_1}\beta_2\langle 1|1\rangle \\ &= \overline{\alpha_1}\alpha_2 + \overline{\beta_1}\beta_2\end{aligned}$$

- Orthogonal states are antipodal on the Bloch sphere

Hello, quantum world!

- Our very first quantum circuit!



- After applying the H gate the qubit state is

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

- When we measure, we obtain 0 or 1, each with 50% probability: we have a circuit that generates perfectly uniform random bits!

Part III

The BB84 protocol: Alice and Bob's
hotline

One-time pad: a Catch-22 situation

- Alice wants to send Bob a message m without Eve being able to learn anything about its content
- This can be achieved if Alice and Bob share in advance a string k of random bits:
 - Alice computes $x = m \oplus k$ and sends x to Bob
 - Eve cannot learn anything from x
 $(Pr(M = m | X = x) = Pr(M = m))$
 - But Bob can recover m by computing $x \oplus k$
- The main problem is that k has to be as long as m and cannot be reused so... how to agree on k ?

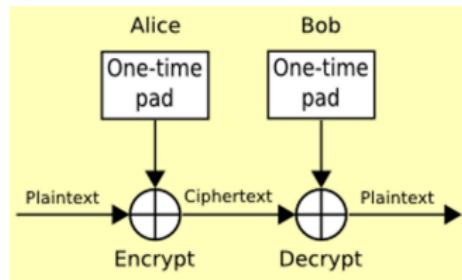
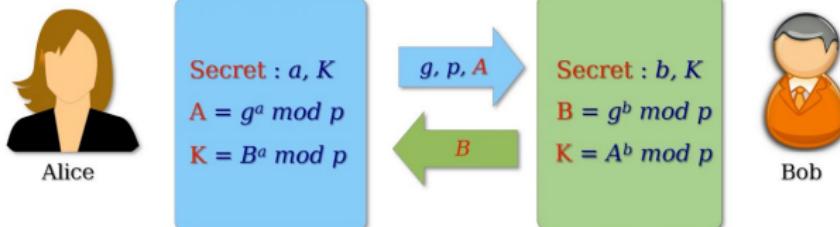


Image credits: nullprogram.com

The problem of key distribution

- Alice and Bob may share several keys for later use when they are together
- But... what if they cannot meet each other?
- There exist key distribution methods like the Diffie-Hellman protocol but...
 - They are not unconditionally secure (they usually rely on hardness assumptions)
 - In fact, DH can be broken with quantum computers!

Diffie - Hellman Key Exchange Protocol



BB84: Alice's part

- In 1984, Charles Bennett and Gilles Brassard proposed the first protocol for quantum key distribution (QKD)
- Alice generates a (private) string of random bits
- She could even do this with a quantum computer (H gate + measure)
- Then, for each bit she randomly chooses if she encodes it in the $\{|0\rangle, |1\rangle\}$ basis or in the $\{|+\rangle, |-\rangle\}$ basis (remember that $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$)
- She can easily do this by using H and X gates (recall that $H|0\rangle = |+\rangle, H|1\rangle = |-\rangle, X|0\rangle = |1\rangle, X|1\rangle = |0\rangle$)
- Alice sends the resulting qubits to Bob (through a quantum but not necessarily secure channel)

BB84: Bob's part

- Each time Bob receives a qubit, he randomly decides whether he will measure it in the $\{|0\rangle, |1\rangle\}$ basis or in the $\{|+\rangle, |-\rangle\}$ basis
- He does this by applying (or not) the H gate before measuring
- He writes down the results and the basis he used:
 - If he used $\{|0\rangle, |1\rangle\}$ he writes down 0 if he gets $|0\rangle$ and 1 if he gets $|1\rangle$
 - If he used $\{|+\rangle, |-\rangle\}$ he writes down 0 if he gets $|+\rangle$ and 1 if he gets $|-\rangle$

BB84: Alice and Bob on the phone

- After this process, Alice and Bob talk on a classical channel (authenticated but not necessarily secure)
- Bob announces the bases he has used for the measurements and Alice announces the bases she used to code the bits
- Bob does NOT announce the results of his measurements
- For those bits in which Bob measured with the same basis that Alice used for coding, he has got the bit that Alice intended to send
- The rest are discarded (they will keep about half of the bits)

BB84: The protocol in an image

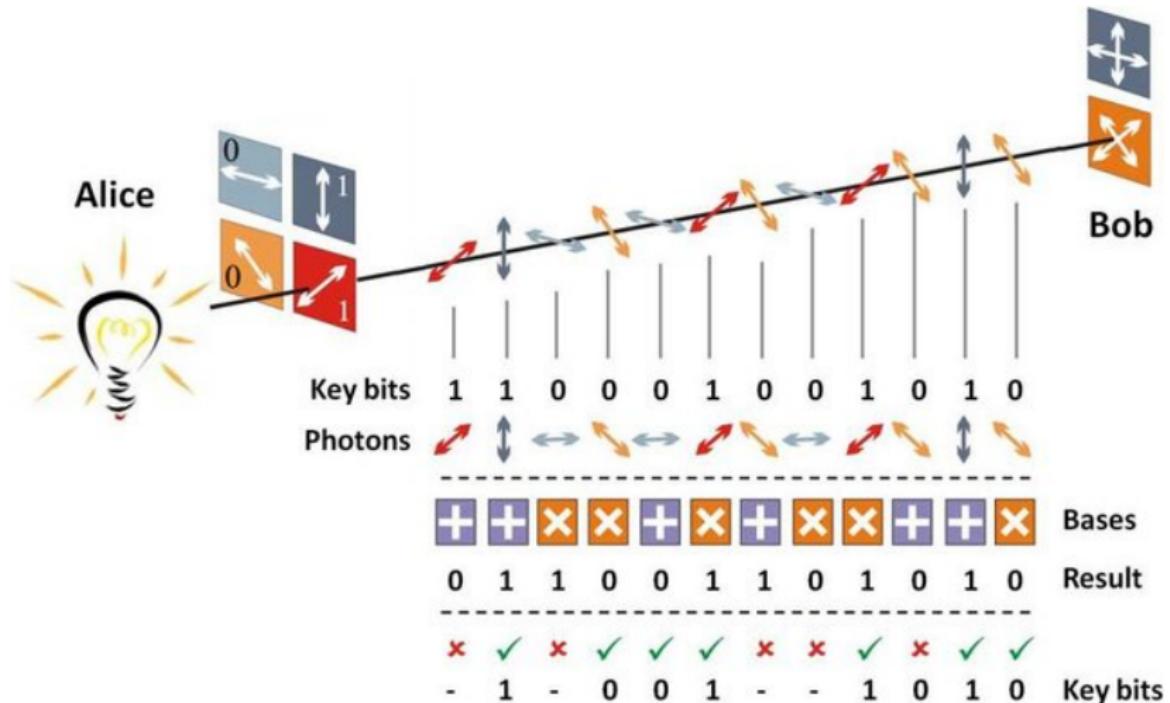


Image credits: A. Carrasco-Casado, V. Fernández, N. Denisenko

Eve tries to intercept and resend...

- Imagine Eve has access to the qubits that Alice sends to Bob
- Eve could try to measure and resend the qubit to Bob
- It is impossible for Eve to distinguish the four possibilities $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$ because she does not know the basis that Alice has chosen
- If Eve chooses a basis at random, she will make an error half of the time and Alice and Bob may detect it (by sharing some of the bits of the key to check that they are equal)
- Eve cannot copy the qubits and wait to check the basis that Alice and Bob have used (no cloning theorem)
- Other more complex attacks are possible, but can be shown to fail

Information reconciliation and privacy amplification

- Because of imperfections in the channel and devices or because of eavesdropping, some of the bits that Alice and Bob have may be different
- They can conduct a process of information reconciliation (for instance, with the cascade protocol)
- After this phase (or even before), some information may have leaked to Eve
- Alice and Bob can perform privacy amplification (for instance, with randomness extractors)

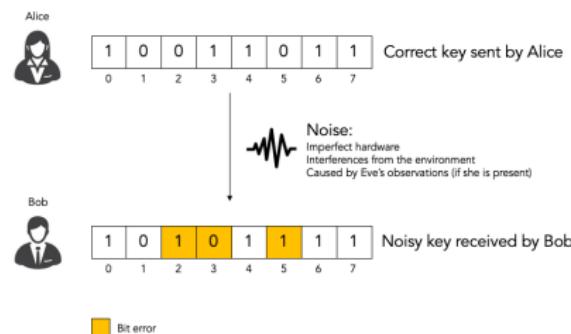


Image credits: hikingandcoding.wordpress.com

QKD at CERN



Image credits: <https://arxiv.org/pdf/1203.4940.pdf>

Kak's three-stage protocol

- Proposed by Kak in 2006
- It needs an authenticated quantum channel
- Suppose Alice wants to send $|x\rangle \in \{|0\rangle, |1\rangle\}$ to Bob:
 - Alice chooses θ_A at random and sends $R_Y(\theta_A)|x\rangle$ to Bob
 - Bob choose θ_B at random and sends $R_Y(\theta_B)R_Y(\theta_A)|x\rangle$ back to Alice
 - Alice applies $R_Y(-\theta_A)$ and sends
$$R_Y(-\theta_A)R_Y(\theta_B)R_Y(\theta_A)|x\rangle = R_Y(\theta_B)|x\rangle$$
to Bob
- Bob can now recover $|x\rangle$ by applying $R_Y(-\theta_B)$

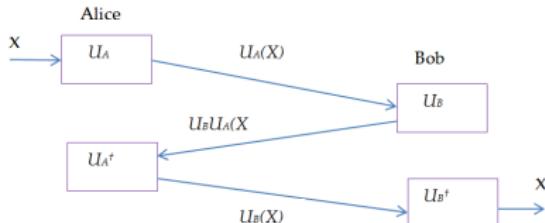


Image credits: wikipedia.org

The quantum one-time pad

- The analogous of the one-time pad with quantum operations would be to choose $a \in \{0, 1\}$ at random and encode $|x\rangle \in \{|0\rangle, |1\rangle\}$ as

$$X^a |x\rangle = |x \oplus a\rangle$$

- This cannot be extended to general qubits $|\psi\rangle$ because $X|+\rangle = |+\rangle$ and $X|-\rangle \equiv |-\rangle$
- We need to choose two bits a and b at random and encode $|\psi\rangle$ as

$$Z^b X^a |\psi\rangle$$

- Bob can now recover $|\psi\rangle$ by applying $X^a Z^b$
- It can be proved that this is unconditionally secure
- The QOTP is the basis of some blind quantum computing protocols

Other protocols that use independent qubits

- The use of independent qubits does not fully exploit the possibilities of quantum information, but there are some additional interesting applications
- For instance:
 - Other QKD protocols: B92, SARG04, Six-state protocol...
 - The concept of quantum money (Wiesner)
 - The Elitzur-Vaidman bomb tester
 - Quantum position verification
 - One-qubit classifier



Image credits: The American Association for the Advancement of Science

Part IV

Two-qubit systems: more than the sum of their parts

Working with two qubits

- Each of the qubits can be in state $|0\rangle$ or in state $|1\rangle$
- So for two qubits we have four possibilities:

$$|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle$$

that we also denote

$$|0\rangle |0\rangle, |0\rangle |1\rangle, |1\rangle |0\rangle, |1\rangle |1\rangle$$

or

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle$$

- Of course, we can have superpositions so a generic state is

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

where α_{xy} are complex numbers such that

$$\sum_{x,y=0}^1 |\alpha_{xy}|^2 = 1$$

Measuring a two-qubit system

- Suppose we have a state

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

- If we measure both qubits, we will obtain:
 - 00 with probability $|\alpha_{00}|^2$ and the new state will be $|00\rangle$
 - 01 with probability $|\alpha_{01}|^2$ and the new state will be $|01\rangle$
 - 10 with probability $|\alpha_{10}|^2$ and the new state will be $|10\rangle$
 - 11 with probability $|\alpha_{11}|^2$ and the new state will be $|11\rangle$
- It is an analogous situation to what we had with one qubit, but now with four possibilities

Measuring just one qubit in a two-qubit system

- If we have a state

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

we can also measure just one qubit

- If we measure the first qubit (for the second one is analogous):
 - We will get 0 with probability $|\alpha_{00}|^2 + |\alpha_{01}|^2$
 - In that case, the new state of $|\psi\rangle$ will be

$$\frac{\alpha_{00} |00\rangle + \alpha_{01} |01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}$$

- We will get 1 with probability $|\alpha_{10}|^2 + |\alpha_{11}|^2$
- In that case, the new state of $|\psi\rangle$ will be

$$\frac{\alpha_{10} |10\rangle + \alpha_{11} |11\rangle}{\sqrt{|\alpha_{10}|^2 + |\alpha_{11}|^2}}$$

Two-qubit states and vector representation

- A general two-qubit quantum state is

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

- We can represent with the column vector

$$\begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix}$$

- We can compute inner products by noticing that

$$\langle 00|00\rangle = \langle 01|01\rangle = \langle 10|10\rangle = \langle 11|11\rangle = 1$$

$$\langle 00|01\rangle = \langle 00|10\rangle = \langle 00|11\rangle = \cdots = \langle 11|00\rangle = 0$$

- A two-qubit quantum gate is a unitary matrix U of size 4×4

Tensor product of one-qubit gates

- The simplest way of obtaining a two-qubit gate is by having a pair of one-qubit gates A and B acting on each of the qubits
- In this case, the matrix for the two-qubit gate is the tensor product $A \otimes B$
- It holds that

$$(A \otimes B)(|\psi_1\rangle \otimes |\psi_2\rangle) = (A|\psi_1\rangle) \otimes (B|\psi_2\rangle)$$

- Of course, either A or B may be the identity
- This does NOT exhaust all possible two-qubit gates

$$\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \otimes \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} = \begin{bmatrix} a_{1,1} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} & a_{1,2} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} \\ a_{2,1} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} & a_{2,2} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_{1,1}b_{1,1} & a_{1,1}b_{1,2} & a_{1,2}b_{1,1} & a_{1,2}b_{1,2} \\ a_{1,1}b_{2,1} & a_{1,1}b_{2,2} & a_{1,2}b_{2,1} & a_{1,2}b_{2,2} \\ a_{2,1}b_{1,1} & a_{2,1}b_{1,2} & a_{2,2}b_{1,1} & a_{2,2}b_{1,2} \\ a_{2,1}b_{2,1} & a_{2,1}b_{2,2} & a_{2,2}b_{2,1} & a_{2,2}b_{2,2} \end{bmatrix}$$

Image credits: wikipedia.org

The *CNOT* gate

- The *CNOT* (or controlled-*NOT* or *cX*) gate is given by the (unitary) matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

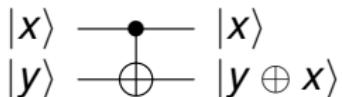
- If the first qubit is $|0\rangle$, nothing changes. If it is $|1\rangle$, we flip the second bit (and the first stays the same)
- That is:

$$|00\rangle \rightarrow |00\rangle \quad |01\rangle \rightarrow |01\rangle$$

$$|10\rangle \rightarrow |11\rangle \quad |11\rangle \rightarrow |10\rangle$$

Action of the *CNOT* gate

- Its action on $x, y \in \{0, 1\}$ is, then:



- This is an extremely important gate for it allows to:
 - Create entanglement (more on this soon)
 - Copy *classical* information, because:

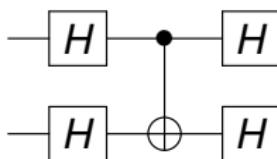
$$|00\rangle \rightarrow |00\rangle$$

$$|10\rangle \rightarrow |11\rangle$$

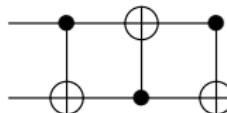
- Construct other controlled gates

Equivalences with CNOT gates

- Sometimes, CNOT gates are not implemented between all pairs of qubits in a quantum computer
- We can use H gates to change the control and target of a CNOT gate



- We can swap states using three CNOT gates



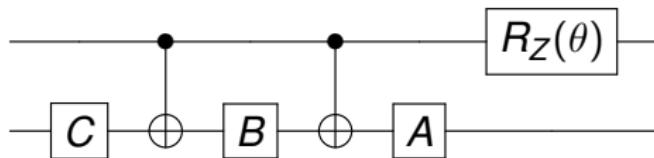
Constructing controlled gates by using the *CNOT* gate

- Any one-qubit gate U can be decomposed in the form

$$e^{i\theta} AXBXC$$

with $ABC = I$

- Then, the circuit



implements a U gate on the lower qubit controlled by the upper qubit

The no-cloning theorem

- There is **no** quantum gate that makes copies of an arbitrary (unknown) qubit
- The proof is easy: suppose we have a gate U such that $U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$
- Then $U|00\rangle = |00\rangle$ and $U|10\rangle = |11\rangle$ and by linearity

$$U\left(\frac{1}{\sqrt{2}}(|00\rangle+|10\rangle)\right) = \frac{1}{\sqrt{2}}(U|00\rangle+U|10\rangle) = \frac{1}{\sqrt{2}}(|00\rangle+|11\rangle)$$

- But

$$\frac{|00\rangle+|10\rangle}{\sqrt{2}} = \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)|0\rangle$$

so we should have

$$U\left(\frac{|00\rangle+|10\rangle}{\sqrt{2}}\right) = \frac{(|0\rangle+|1\rangle)}{\sqrt{2}}\frac{(|0\rangle+|1\rangle)}{\sqrt{2}} \neq \frac{1}{\sqrt{2}}(|00\rangle+|11\rangle)$$

Quantum entanglement: the spooky action at a distance

- We say that a state $|\psi\rangle$ is a product state if it can be written in the form

$$|\psi\rangle = |\psi_1\rangle |\psi_2\rangle$$

where $|\psi_1\rangle$ and $|\psi_2\rangle$ are two states (of at least one qubit)

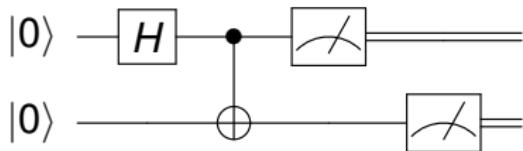
- An **entangled** state is a state that is not a product state
- Example of entangled states (Bell states):

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} \quad \frac{|00\rangle - |11\rangle}{\sqrt{2}}$$

$$\frac{|01\rangle + |10\rangle}{\sqrt{2}} \quad \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

Hello, entangled world!

- We can construct (and measure) Bell states with simple circuits



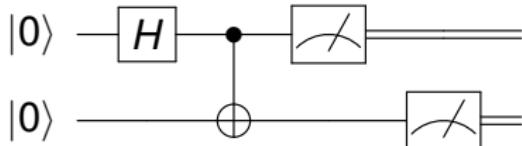
- Initially, the state of the system is $|00\rangle$
- After we apply the H gate, the state is

$$\frac{|00\rangle + |10\rangle}{\sqrt{2}}$$

- When we apply the $CNOT$ gate, the state changes to

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Hello, entangled world!



- Before we measure the first qubit, we have the state $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$
- We will get 0 or 1, each with probability $\frac{1}{2}$
- Suppose we obtain 0. Then, the new state will be $|00\rangle$
- Then, when we measure the second qubit we will obtain 0 with probability 1!
- Also, if we obtain 1 in the first qubit, in the second we will also obtain 1!

Part V

The CHSH game: Nature isn't
classical, dammit

The CHSH game

- Based in an inequality proposed in 1969 by Clauser, Horne, Shimony and Holt based on previous work by John Bell
- Alice and Bob receive bits x and y from a referee
- They have to respond with bits a and b
- They win if

$$a \oplus b = x \cdot y$$

- They can decide on a joint strategy beforehand, but they cannot communicate during the game

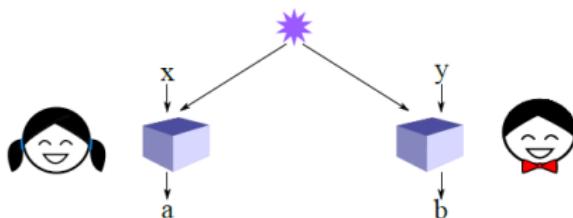


Image credits: quantumcomputing.stackexchange.com

Classical strategies for the CHSH game

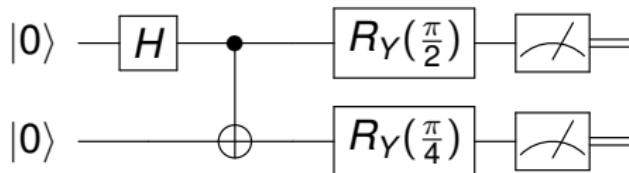
- Alice and Bob can win 75% of the time if they always answer ‘0’
- No other deterministic strategy can do better
- And probabilistic strategies are convex combinations of classical strategies so they cannot improve the 75% success rate

	$a = 0$	$a = 1$	$a = x$	$a = \neg x$
$b = 0$	3/4	1/4	3/4	1/4
$b = 1$	1/4	3/4	1/4	3/4
$b = y$	3/4	1/4	1/4	3/4
$b = \neg y$	1/4	3/4	3/4	1/4

Image credits: Ryan O'Donnell

Quantum strategy for the CHSH game

- Alice and Bob share a Bell pair $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$ before the start of the game
- If Alice receives 0, she measures her qubit and outputs the result
- If she receives 1, she applies $R_Y(\frac{\pi}{2})$ to her qubit and then she measures it
- If Bob receives 0, he applies $R_Y(\frac{\pi}{4})$. Else, he applies $R_Y(-\frac{\pi}{4})$.
- Then, he measures his qubit
- The probability of winning is now $\cos^2(\frac{\pi}{8}) \approx 0.85 > 0.75$



Some comments on the CHSH game

- It can be proved that $\cos^2(\frac{\pi}{8})$ is the highest possible success rate for a quantum strategy (Tsirelson's bound)
- The CHSH game can be used to rule out local realism
- Several experiments have been conducted, including:
 - Aspect et al. (1981-82)
 - Hensen et al. (2005) - Eliminate the locality and detection loopholes
- All of them agree with the predictions of quantum theory

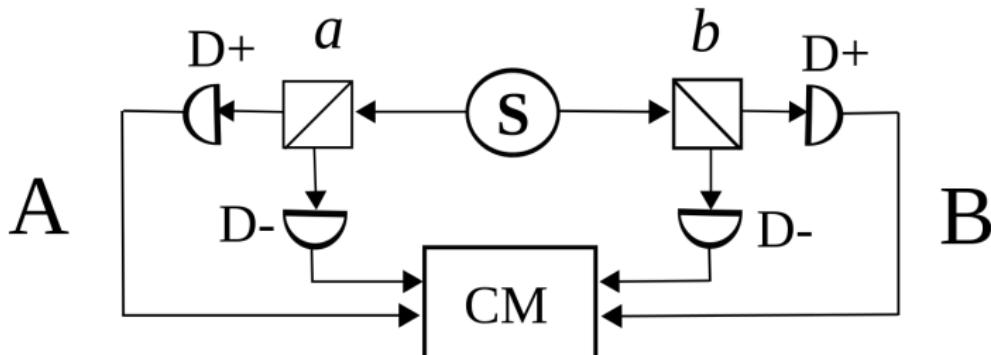


Image credits: George Stamatou based on png file of C.Thompson

The GHZ game

- Introduced by Greenberger, Horne and Zeilinger
- A referee selects rst from $\{000, 011, 101, 110\}$ and sends r to Alice, s to Bob and t to Charlie
- They produce a, b and c and win if

$$a \oplus b \oplus c = r \vee s \vee t$$

- Classically, they can only win with 75% probability
- Quantumly, they can win every single time
 - They share the state

$$\frac{1}{2}(|000\rangle - |011\rangle - |101\rangle - |110\rangle)$$

- They apply H to their qubit if they receive 1
- They measure and return the answer
- This is sometimes called “quantum pseudo-telepathy” (Brassard, Cleve, Tapp)
- Both the CHSH and the GHZ game can be used for randomness certification (and expansion)

Part VI

Quantum teleportation and
superdense coding: entangled up in
blue

Quantum teleportation: Quantum me up, Scotty!

- Can Alice sent a qubit $|\psi\rangle$ to Bob if there is no quantum channel available?
- We are interested in the most general case, even if Alice does not know which state she has
- The problem can be solved if Alice and Bob share an entangled state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

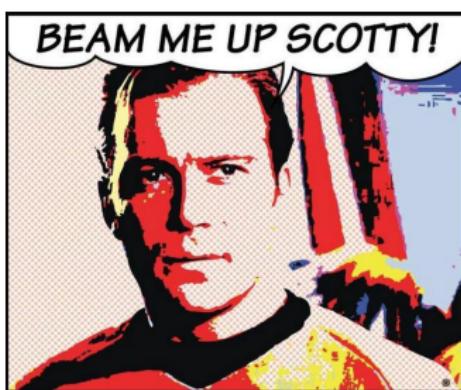


Image credits: www.geeksaresexy.net

Quantum teleportation: Alice's part

- Alice and Bob share an entangled state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$
 - This can be done in advance
 - Or they can rely on a source that distributes entangled pairs
- Alice applies a CNOT gate to the qubit she wants to teleport $|\psi\rangle = a|0\rangle + b|1\rangle$ and to her part of the Bell pair. We will have

$$\frac{1}{\sqrt{2}}(a(|000\rangle + |011\rangle) + b(|110\rangle + |101\rangle))$$

- Alice further applies the H gate to the qubit she wants teleported. Then, we have

$$\begin{aligned} \frac{1}{2}(&|00\rangle(a|0\rangle + b|1\rangle) + |01\rangle(b|0\rangle + a|1\rangle) \\ &+ |10\rangle(a|0\rangle - b|1\rangle) + |11\rangle(-b|0\rangle + a|1\rangle)) \end{aligned}$$

- Alice measures her two qubits and sends the result (two classical bits) to Bob (through a classical channel)

Quantum teleportation: Bob's part

- Bob uses the second bit received from Alice to decide if he applies X to his qubit
- And he uses the first bit to decide if he applies Z

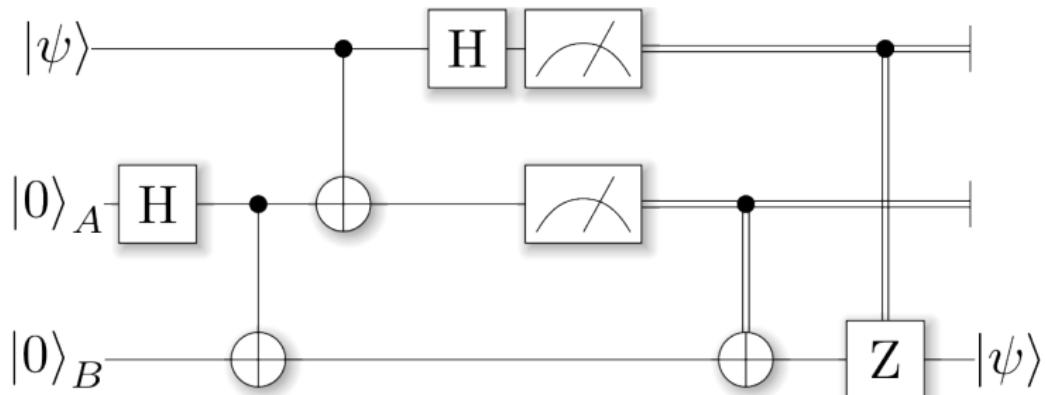


Image credits: ProjectQ

Quantum teleportation: some comments

- It is not matter that is teleported but information
- When Alice measure her qubit, she loses it (if not, we would be contradicting the no-cloning theorem)
- To teleport a qubit, we need two classical bits and one entangled pair:

$$2\text{bits} + 1\text{ebit} \geq 1\text{qubit}$$

- Teleportation is not instantaneous, we need classical communication (no-communication theorem)
- Quantum teleportation has been shown experimentally (current record is 1,400 km)
- Demonstration of quantum teleportation in Quirk

Entanglement swapping

- Quantum teleportation can also be used with entangled qubits
- Alice shares a Bell pair with Bob and another one with Charlie
- In the figure, the top and bottom qubits belong to Alice. The second from the top belongs to Bob and the other to Charlie
- Alice teleports her top qubit to Charlie
- Now Bob's and Charlie's qubits are entangled (although maybe they were never in direct contact)

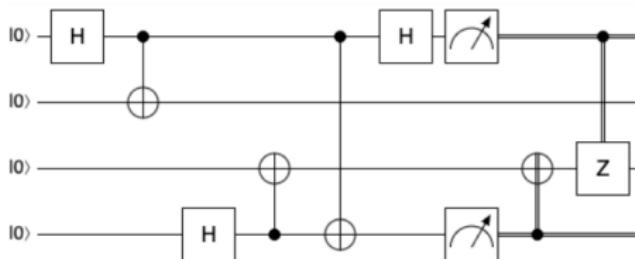
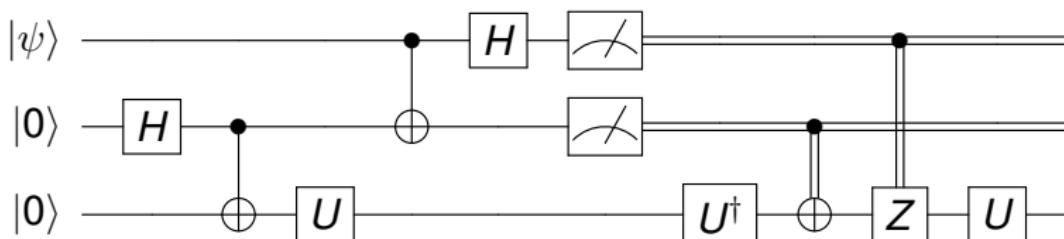


Image credits: Created with Quirk. Click [here](#) to access the circuit

Gate teleportation

- We can generalize the idea of quantum teleportation to teleport the action of gates
- With the circuit of the figure, we can apply gate U to an arbitrary state $|\psi\rangle$
- This is useful if preparing $\frac{1}{\sqrt{2}}(|0\rangle U|0\rangle + |1\rangle U|1\rangle)$ and applying UXU^\dagger , UZU^\dagger , $UZXU^\dagger$ are easy compared to applying U to a general qubit
- Such a situation can happen when $U = T$ in the context of fault-tolerant quantum computing



Superdense coding: two for the price of one (more or less)

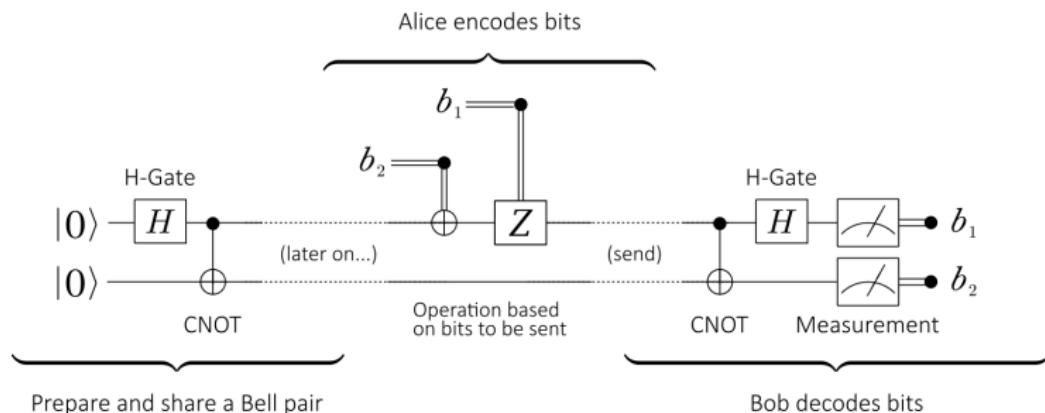
- As we have seen, in the presence of a Bell pair, we can send a qubit with just two classical bits
- But... how many classical bits can we communicate with one qubit?
- Holevo's bound: the accessible information of one qubit is just one bit
- However, if Alice and Bob share in advance a Bell pair... we can send two bits of information with just one qubit!

$$1\text{qubit} + 1\text{ebit} \geq 2\text{bits}$$

- This protocol is, in some sense, the inverse of quantum teleportation

Superdense coding: Alice's part

- Alice and Bob share a Bell pair in advance $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$
- Alice wants to send to Bob two classical bits b_1 and b_2
- If $b_2 = 1$, she applies X to her qubit
- If $b_1 = 1$, she applies Z to her qubit
- Then, she sends her qubit to Bob



Superdense coding: Bob's part

- Bob receives Alice's qubit
- He applies a *CNOT* gate controlled by Alice's qubit
- He applies *H* to Alice's qubit
- He measures and recovers b_1 and b_2

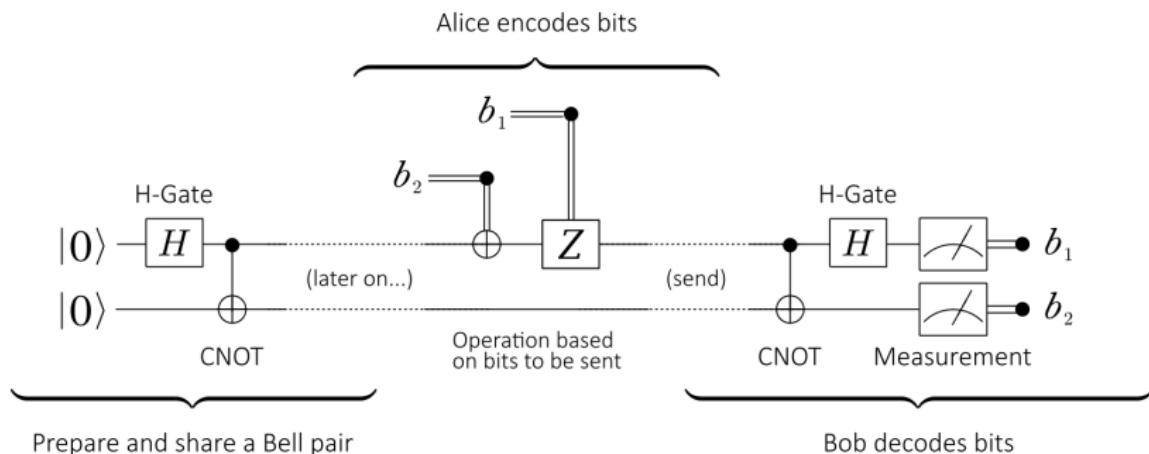


Image credits: www.quantum-bits.org

Superdense coding: an example

- Suppose Alice wants to send 11
- We start with $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$
- After Alice's operations, we will have $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$
- When Bob applies *CNOT* he obtains

$$\frac{1}{\sqrt{2}}(|01\rangle - |11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|1\rangle$$

- And with the *H* gate he gets $|11\rangle$ that now he can measure

Part VII

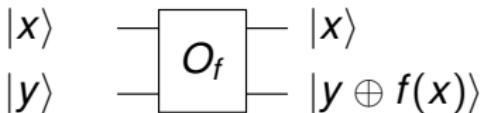
Deutsch's algorithm: the
grandfather of all quantum
algorithms

Deutsch's algorithm: statement of the problem

- In 1985, David Deutsch proposed a very simple algorithm that, nevertheless, hints at the capabilities of quantum computing
- The problem it solves is only of theoretical relevance and was later generalized in a joint work with Jozsa
- We are given a circuit (an **oracle**) that implements a one-bit boolean function and we are asked to determine whether the function is constant (returns the same value for all inputs) or balanced (returns 1 on one input and 0 on the other)
- Alternatively, we can think of the oracle as indexing a bit string of length two and we are asked to compute the XOR of the bits of the string
- In the classical case, we would need to consult the oracle twice, to compute both values of the function
- In the quantum case, we can make just one oracle call... but in superposition

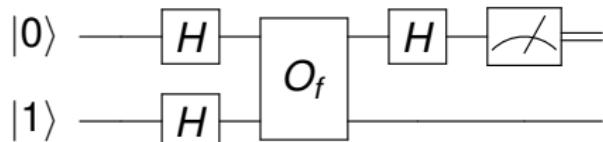
Deutsch's algorithm: the oracle

- An oracle is treated as a black box, a circuit whose interior we cannot know
- This circuit computes, in a reversible way, a certain function f (in our case, of just one input)
- For the computation to be reversible, it uses as many inputs as outputs and “writes the result” with an XOR



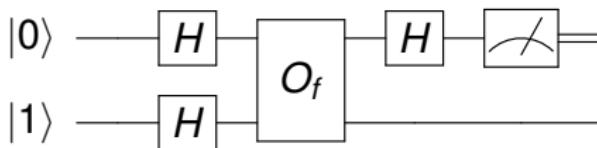
Deutsch's algorithm: the circuit

- The quantum circuit that we need to use to solve the problem is very simple



- If the function is constant, we will measure 0
- If the function is balanced, we will measure 1

Deutsch's algorithm: the magic



- The initial state is $|0\rangle|1\rangle$
- After the H the gates we have

$$\frac{(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)}{2}$$

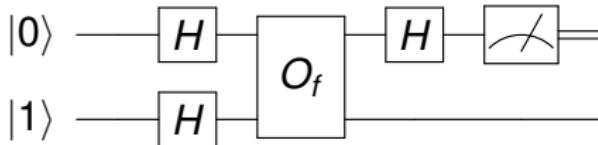
which is the same as

$$\frac{|0\rangle(|0\rangle - |1\rangle)}{2} + \frac{|1\rangle(|0\rangle - |1\rangle)}{2}$$

- When we apply the oracle, by linearity we obtain

$$\frac{|0\rangle(|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle)}{2} + \frac{|1\rangle(|0 \oplus f(1)\rangle - |1 \oplus f(1)\rangle)}{2}$$

Deutsch's algorithm: the magic (2)



- If $f(0) = 0$, we have

$$|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle = |0\rangle - |1\rangle$$

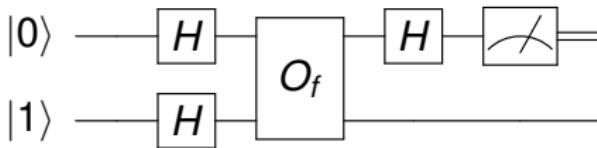
- However, if $f(0) = 1$ we get

$$|0 + f(0)\rangle - |1 \oplus f(0)\rangle = |0 \oplus 1\rangle - |1 \oplus 1\rangle = |1\rangle - |0\rangle = -(|0\rangle - |1\rangle)$$

- For $f(1)$ the situation is the same so the global state is

$$\frac{(-1)^{f(0)} |0\rangle (|0\rangle - |1\rangle)}{2} + \frac{(-1)^{f(1)} |1\rangle (|0\rangle - |1\rangle)}{2}$$

Deutsch's algorithm: the magic (3)



- We can also write that state as

$$\frac{|0\rangle(|0\rangle - |1\rangle)}{2} + \frac{(-1)^{f(0)+f(1)}|1\rangle(|0\rangle - |1\rangle)}{2}$$

- So if $f(0) = f(1)$, we will have

$$\frac{|0\rangle(|0\rangle - |1\rangle)}{2} + \frac{|1\rangle(|0\rangle - |1\rangle)}{2} = \frac{(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)}{2}$$

and when we apply the last H and measure we obtain 0.

- But if $f(0) \neq f(1)$, the state is

$$\frac{|0\rangle(|0\rangle - |1\rangle)}{2} - \frac{|1\rangle(|0\rangle - |1\rangle)}{2} = \frac{(|0\rangle - |1\rangle)(|0\rangle - |1\rangle)}{2}$$

and, then, we obtain 1.

Deutsch's algorithm: some comments

- When we apply the oracle we have a phase kickback: we only act on one qubit, but it affects the whole state
- Deutch's algorithm exploits an interference phenomenon similar to that found in some physical experiments (double-slit experiment, Mach-Zehnder interferometer)

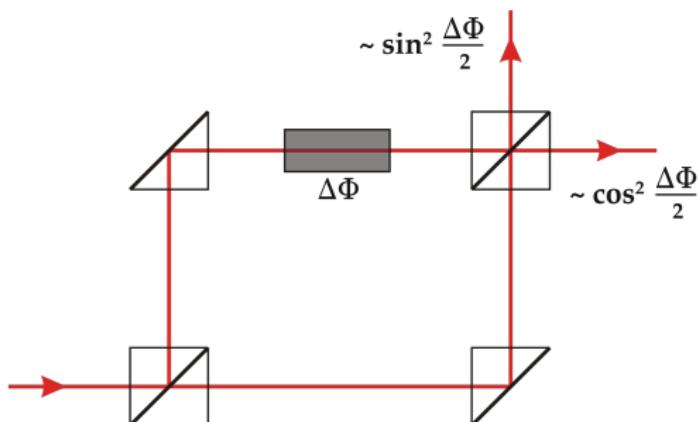


Image credits: Wikipedia

Part VIII

Multiqubit systems: growing up!

n -qubit systems

- When he have n qubits, each of them can be in state $|0\rangle$ and $|1\rangle$
- Thus, for the n -qubit state we have 2^n possibilities:

$$|00\dots0\rangle, |00\dots1\rangle, \dots, |11\dots1\rangle$$

or simply

$$|0\rangle, |1\rangle, \dots, |2^n - 1\rangle$$

- A generic state of the system will be

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \dots + \alpha_{2^n - 1} |2^n - 1\rangle$$

where α_i are complex numbers such that

$$\sum_{i=0}^{2^n - 1} |\alpha_i|^2 = 1$$

Measuring a n -qubit state

- Suppose we have the n -qubit state

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \dots + \alpha_{2^n-1} |2^n - 1\rangle$$

- If we measure all its qubits, we obtain:
 - 0 with probability $|\alpha_0|^2$ and the new state will be $|0\dots00\rangle$
 - 1 with probability $|\alpha_1|^2$ and the new state will be $|0\dots01\rangle$
 - ...
 - $2^n - 1$ with probability $|\alpha_{2^n-1}|^2$ and the new state will be $|1\dots11\rangle$
- It is analogous to what we had with one and two qubits, but now with 2^n possibilities

Measuring one qubit in a n -qubit state

- We have

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \dots + \alpha_{2^n - 1} |2^n - 1\rangle$$

- If we measure the j -th qubit
 - We will get 0 with probability

$$\sum_{i \in I_0} |\alpha_i|^2$$

- where I_0 is the set of numbers whose j -th bit is 0
- In that case, the new state $|\psi\rangle$ will be

$$\frac{\sum_{i \in I_0} \alpha_i |i\rangle}{\sqrt{\sum_{i \in I_0} |\alpha_i|^2}}$$

- The case in which we obtain 1 is analogous

n -qubit quantum gates

- A n -qubit state is

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \dots + \alpha_{2^n-1} |2^n - 1\rangle$$

- We can represent it by the column vector

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2^n-1} \end{pmatrix}$$

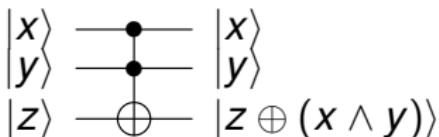
- To compute inner products with Dirac notation we only need to note that

$$\langle i|j\rangle = \delta_{ij}$$

- Thus, a n -qubit quantum gate is a unitary matrix U of size $2^n \times 2^n$

The Toffoli gate

- The Toffoli gate (or *CCNOT*) is a 3-qubit gate. Thus, it can be represented as a 8×8 matrix
- Its action on elements $x, y, z \in \{0, 1\}$ is:



- The Toffoli gate is **universal for classical logic**, and thus **any classical circuit can be simulated with a quantum circuit**
- However, the Toffoli gate, on its own, **is not universal for quantum computing** (and it is not even necessary, because it can be simulated with one and two-qubit gates)

Universal gates in quantum computing

- The number of quantum gates (even for a single qubit) is uncountably infinite. Thus, no finite set of gates is universal in the classical sense
- However, we can obtain finite sets of gates that allow us to **approximate** any other gate as much as we want

Theorem

The one-qubit gates together with the CNOT gate are universal for quantum computing

Theorem

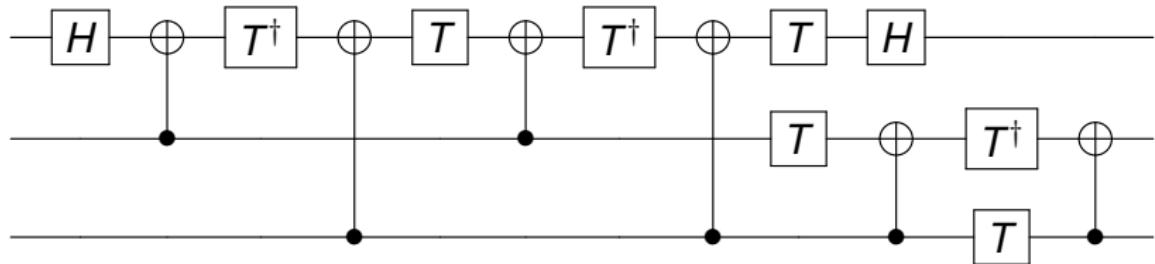
The gates X , H , T and CNOT are universal for quantum computing

Gate equivalences

$$\begin{array}{c} \text{---} \boxed{Z} \text{---} \\ = \\ \text{---} \boxed{H} \text{---} \boxed{X} \text{---} \boxed{H} \text{---} \end{array}$$
$$\begin{array}{c} \text{---} \boxed{S} \text{---} \\ = \\ \text{---} \boxed{T} \text{---} \boxed{T} \text{---} \end{array}$$
$$\begin{array}{c} \text{---} \boxed{Y} \text{---} \\ = \\ \text{---} \boxed{Z} \text{---} \boxed{X} \text{---} \boxed{S} \text{---} \boxed{X} \text{---} \boxed{S} \text{---} \boxed{X} \text{---} \end{array}$$
$$\begin{array}{c} \text{---} \boxed{T^\dagger} \text{---} \\ = \\ \text{---} \boxed{S} \text{---} \boxed{S} \text{---} \boxed{S} \text{---} \boxed{T} \text{---} \end{array}$$
$$\begin{array}{c} \text{---} \boxed{S^\dagger} \text{---} \\ = \\ \text{---} \boxed{S} \text{---} \boxed{S} \text{---} \boxed{S} \text{---} \end{array}$$

However, Z , S , Y , S^\dagger and T^\dagger are usually included among the available gates in most quantum computers (such as the ones in the IBM Q Experience).

Equivalence of the Toffoli gate



Part IX

Everything you always wanted to
know about quantum parallelism but
were afraid to ask

Urban legends about quantum parallelism

- But... don't quantum computers try all 2^n possibilities in parallel?
- The answer is... yes *and* no (this is *quantum* computing after all!)

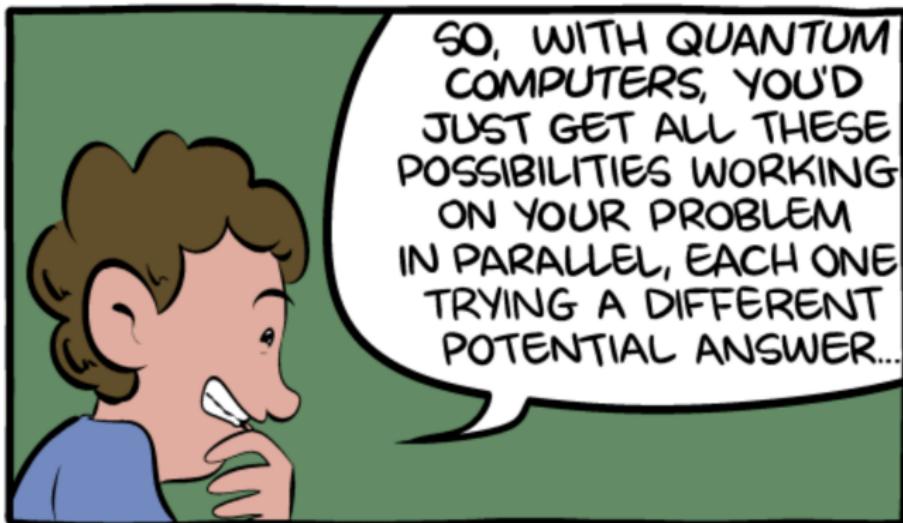
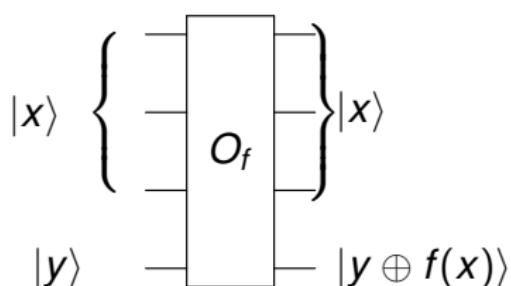


Image credits: [The Talk](#), by Scott Aaronson and Zach Weinersmith

Evaluating a function: querying the oracle

- As we know, in quantum computing every gate is reversible
- To compute a function f we keep the inputs unchanged and xor the result to the output qubits
- This type of circuit is called an oracle for f (we have already used an oracle for a one-bit function in Deutsch's algorithm)



Evaluating a function in parallel: the superposition hocus-pocus

- Suppose that we have an oracle O_f for a function $f(x)$ with a one-bit input
- We know that, using the H gate, we can put a qubit in superposition
- If we start with the state $|0\rangle|0\rangle$ and we apply H on the first qubit, we will have

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

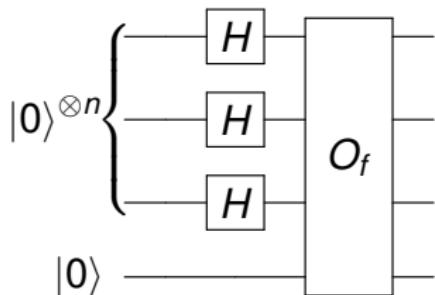
- If we now apply O_f , by linearity we have

$$\frac{1}{\sqrt{2}}|0\rangle|f(0)\rangle + \frac{1}{\sqrt{2}}|1\rangle|f(1)\rangle$$

- We have evaluated the function on two different inputs with just one call!

Evaluating a function in parallel: the tensor-product abracadabra

- We can do something similar with a function $f(x_1, x_2, \dots, x_n)$ on n -variables by using the following circuit



- When we apply the H gates we obtain

$$\frac{(|0\rangle + |1\rangle)(|0\rangle + |1\rangle) \cdots (|0\rangle + |1\rangle)|0\rangle}{\sqrt{2^n}}$$

Evaluating a function in parallel: the tensor-product abracadabra (2)

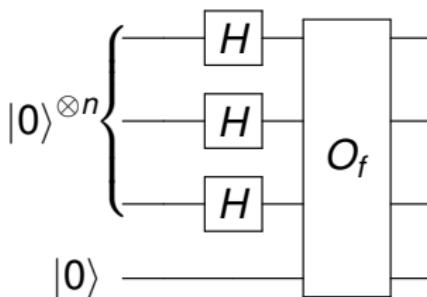
- If we expand the product we get

$$\frac{(|0\dots0\rangle + |0\dots1\rangle + \dots + |1\dots1\rangle)|0\rangle}{\sqrt{2^n}} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|0\rangle$$

- And, when we apply the oracle, we will get the state

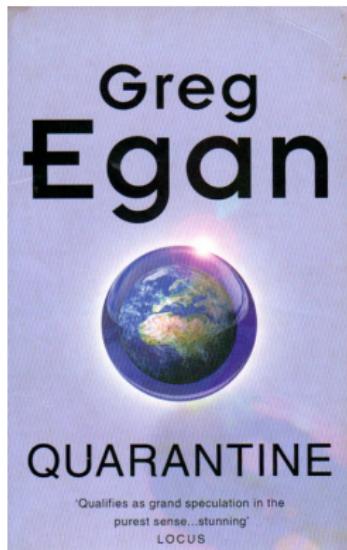
$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|f(x)\rangle$$

- An exponential number of function evaluations with just one call!



Quantum parallelism vs. non-deterministic machines

- With a non-deterministic machine, we could choose at will some value f
- This would allow us to solve NP -complete problems
- A similar idea is used in the plot of *Quarantine*, a science-fiction novel by Greg Egan

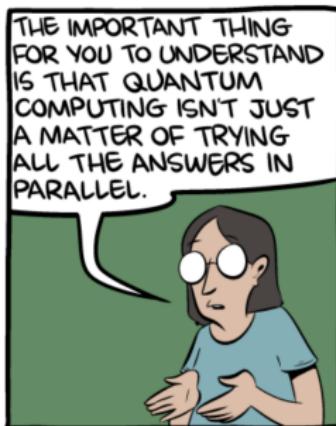


All that glitters ain't gold

- And now... how do we retrieve the values $f(x)$?
- To obtain a result, we need to perform a measurement
- But then we will get a state of the form

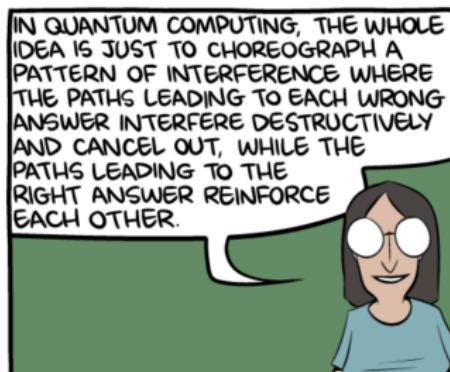
$$|c\rangle |f(c)\rangle$$

- That is, we only obtain the result of the function for a randomly chosen input (this may be even worse than classically evaluating the function)



Interferences come to the rescue

- How can we use the 2^n evaluations to extract useful information?
- One possibility is... to produce interferences!
- The amplitudes of some states can be negative
- If we manage to “annihilate” the amplitudes of states we are not interested in, the probability of obtaining the answer that we need will grow
- This is, in general, no easy task, but we know how to achieve it in some interesting cases



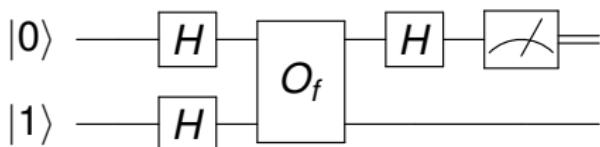
IN QUANTUM COMPUTING, THE WHOLE IDEA IS JUST TO CHOREOGRAPH A PATTERN OF INTERFERENCE WHERE THE PATHS LEADING TO EACH WRONG ANSWER INTERFERE DESTRUCTIVELY AND CANCEL OUT, WHILE THE PATHS LEADING TO THE RIGHT ANSWER REINFORCE EACH OTHER.

Part X

The Deutsch-Jozsa algorithm: a
very fast way of solving a problem
that nobody asked to solve

Reminder: Deutsch's algorithm

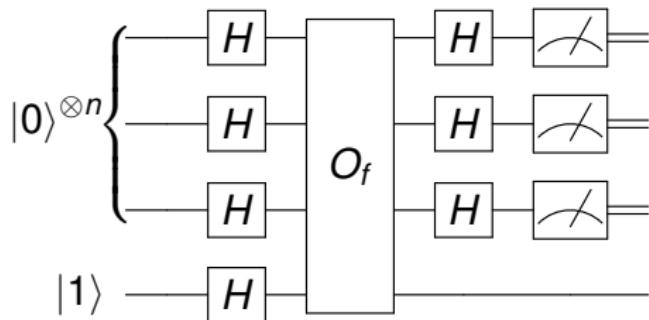
- We have an oracle O_f for a boolean function $f(x)$
- f can be constant (returns the same value for all inputs) or balanced (returns 1 on one input and 0 on the other)
- Distinguishing one situation from the other requires, in the classical case, evaluating the function on the two possible inputs
- With a quantum computer, we can solve the problem with just one call to O_f
- The key is to use quantum parallelism together with interference



Upping the ante: the Deutsch-Jozsa algorithm

- The Deutsch-Jozsa algorithm solves a type of problem called **promise problem**
 - We are given a boolean function $f(x_1, \dots, x_n)$
 - We are **promised** that f is either constant (always 0 or 1) or balanced (0 for half of the inputs and 1 for the rest)
 - We have to decide which of the two cases we are in by calling the function as few times as possible
- With a classical deterministic algorithm we need (in the worst case) $2^{n-1} + 1$ calls to f
- With the Deutsch-Jozsa quantum algorithm it is enough to evaluate f **just once**

Circuit for the Deutsch-Jozsa algorithm



Steps in the Deutsch-Jozsa algorithm

- ① We create the state $|0 \dots 0\rangle |1\rangle$
- ② We use Hadamard gates to create the superposition

$$\sum_{x \in \{0,1\}^n} \frac{1}{\sqrt{2^{n+1}}} |x\rangle (|0\rangle - |1\rangle)$$

- ③ We apply the oracle, getting

$$\sum_{x \in \{0,1\}^n} \frac{1}{\sqrt{2^{n+1}}} |x\rangle (|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle) =$$

$$\sum_{x \in \{0,1\}^n} \frac{(-1)^{f(x)}}{\sqrt{2^{n+1}}} |x\rangle (|0\rangle - |1\rangle)$$

Steps in the Deutsch-Jozsa algorithm (2)

- ④ We apply again Hadamard gates to the n first qubits and we obtain

$$\sum_{y \in \{0,1\}^n} \sum_{x \in \{0,1\}^n} \frac{(-1)^{f(x)+x \cdot y}}{2^n \sqrt{2}} |y\rangle (|0\rangle - |1\rangle)$$

- ⑤ Finally, we measure the n first qubits.
⑥ If the function is constant, we will obtain $|0\rangle$. Otherwise (if the function is balanced), we will obtain $|1\rangle$.

Correctness of the algorithm

- The probability of measuring $|0\rangle$ is exactly

$$\left(\sum_{x \in \{0,1\}^n} \frac{(-1)^{f(x)+x \cdot 0}}{2^n} \right)^2 = \left(\sum_{x \in \{0,1\}^n} \frac{(-1)^{f(x)}}{2^n} \right)^2$$

- If f is constant, the sum is 1
- If f is balanced, the sum is 0

Some comments on the Deutsch-Jozsa algorithm

- The problem we have solved is academical, with no practical interest
- But... it shows how quantum computing can obtain global information about a function with just one evaluation
- The key is to use:
 - Quantum parallelism (because of superposition)
 - Interference (constructive and destructive)
- Similar ideas are used in other algorithms, like the Bernstein-Vazirani and Simon methods

Part XI

Grover's algorithm: finding the
needle in the haystack

Statement of the problem

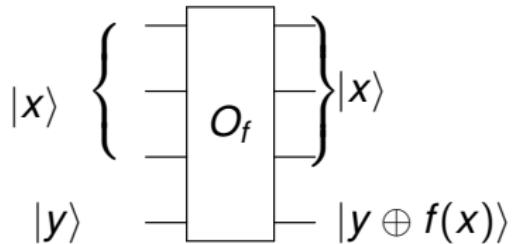
- Grover's algorithm is used to solve search problems
- Imagine we have an unsorted list of N elements
- One of them verifies a certain condition and we want to find it
- Any classical algorithm requires $O(N)$ queries to the list in the worst case
- Grover's algorithm can find the element with $O(\sqrt{N})$ queries



Image credits: Downloaded from www.usnewsglobaleducation.com

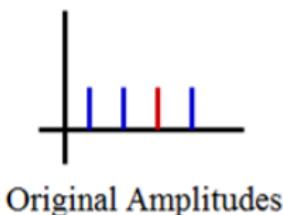
The oracle

- As in Deutsch-Jozsa's algorithm, we will use an oracle
- This oracle computes the function $f : \{0, 1\}^n \Rightarrow \{0, 1\}$ (with $N = 2^n$)
- The element we want to find is the one that verifies $f(x) = 1$

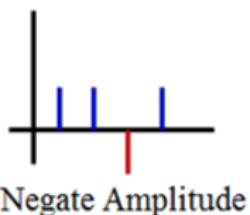


The idea behind the algorithm

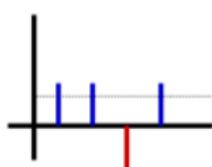
- Grover's algorithm is based on the idea of **inversion about the mean**



Original Amplitudes



Negate Amplitude



Average of all Amplitudes

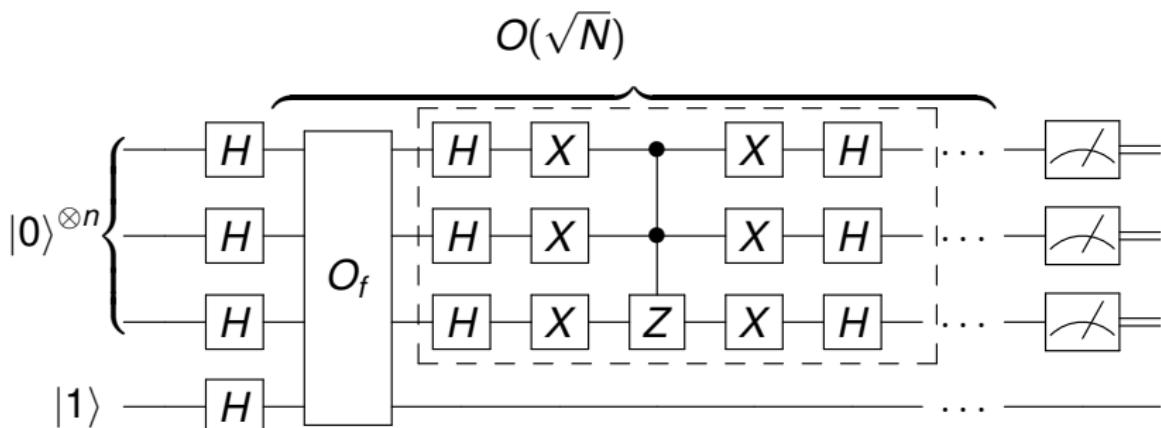


Flip all Amplitudes around Avg

Image credits: quantumcomputing.stackexchange.com

Grover's algorithm

- Grover's algorithm performs $O(\sqrt{N})$ iterations, each one consisting in an oracle query and a call to Grover's diffusion operator
- The oracle “marks” those states that verify the condition
- The diffusion operator “amplifies” the amplitudes of the marked states



Grover's algorithm as a rotation

- Let us denote by $|x_1\rangle$ the marked element
- Then, the initial state of the upper n qubits is

$$\sqrt{\frac{N-1}{N}}|x_0\rangle + \sqrt{\frac{1}{N}}|x_1\rangle$$

where

$$|x_0\rangle = \sum_{x \in \{0,1\}^n, x \neq x_1} \sqrt{\frac{1}{N-1}} |x\rangle$$

- We can choose $\theta \in (0, \frac{\pi}{2})$ such that

$$\cos \theta = \sqrt{\frac{N-1}{N}} \quad \sin \theta = \sqrt{\frac{1}{N}}$$

Grover's algorithm as a rotation (2)

- Define D to be Grover's diffusion operator and $G = DO_f$
- It can be shown that G acts on the 2-dimensional space spawned by $|x_0\rangle$ and $|x_1\rangle$ as a rotation of angle 2θ
- That is

$$G|x_0\rangle = \cos 2\theta |x_0\rangle + \sin 2\theta |x_1\rangle$$

$$G|x_1\rangle = -\sin 2\theta |x_0\rangle + \cos 2\theta |x_1\rangle$$

$$|x_0\rangle = \sum_{x \in \{0,1\}^n, x \neq x_1} \sqrt{\frac{1}{N-1}} |x\rangle$$

- Since the initial state is $\cos \theta |x_0\rangle + \sin \theta |x_1\rangle$, after m iterations we will have

$$\cos(2m+1)\theta |x_0\rangle + \sin(2m+1)\theta |x_1\rangle$$

Grover's algorithm as a rotation (3)

- In order to obtain $|x_1\rangle$ with high probability when we measure we need

$$(2m+1)\theta \approx \frac{\pi}{2}$$

and this gives

$$m \approx \frac{\pi}{4\theta} - \frac{1}{2}$$

- Since

$$\sin \theta = \sqrt{\frac{1}{N}}$$

we will have

$$\theta \approx \sqrt{\frac{1}{N}}$$

and then we can choose

$$m = \left\lfloor \frac{\pi}{4} \sqrt{N} \right\rfloor$$

The case with multiple marked elements

- If the number of marked elements is $k > 1$, a similar argument can be made by defining

$$|x_0\rangle = \sum_{f(x)=0} \sqrt{\frac{1}{N-k}} |x\rangle$$

$$|x_1\rangle = \sum_{f(x)=1} \sqrt{\frac{1}{k}} |x\rangle$$

- In this case

$$\sin \theta = \sqrt{\frac{k}{N}}$$

and if $k \ll N$ we can choose

$$m = \left\lfloor \frac{\pi}{4} \sqrt{\frac{N}{k}} \right\rfloor$$

The case with unknown number of marked elements

- If we do not know how many elements are marked, we can still user Grover's algorithm
- We can use Grover's circuit combined with the Quantum Fourier Transform to estimate k
- Or we can choose m at random. For instance:
 - Uniformly from the set $\{0, \dots, \lceil \sqrt{N} + 1 \rceil\}$
 - With an incremental scheme, starting with an upper bound for m of $b = 1$ and increasing it exponentially up to \sqrt{N}
- In all the cases, it can be shown that a marked element will be found with high probability with $O(\sqrt{N})$ queries to the oracle

Some comments on Grover's algorithm

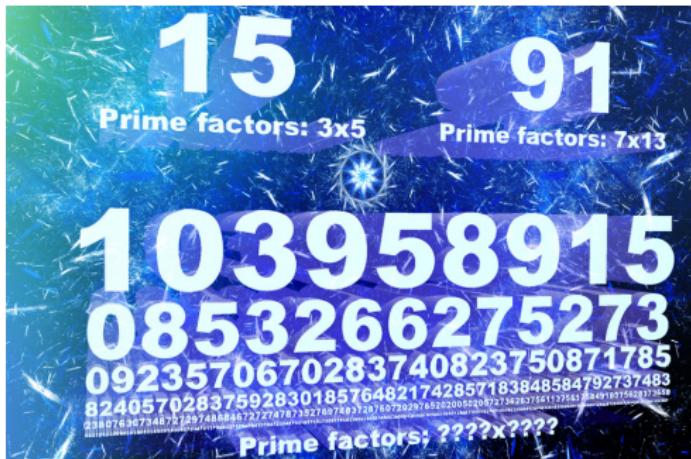
- When we measure, we will obtain x such that $f(x) = 1$ with probability depending on:
 - The number m of iterations
 - The fraction of values x that satisfy the condition
- If we perform too many iterations, we can overshoot and not find a marked element
- On the other hand, if $k = \frac{N}{4}$ then one iteration will find a marked element with certainty
- Grover's algorithm can be used to find minima of functions (Dürr-Hoyer's algorithm)
- It can be shown that no other quantum algorithm can obtain more than a quadratic speed-up over classical algorithms in the same setting
- A generalization of Grover's algorithm called Amplitude Amplification can be used with states prepared by an arbitrary unitary A

Part XII

Shor's algorithm: breaking the
Internet

Shor's algorithm and factoring

- Shor's algorithm is, probably, the most famous quantum algorithm
- It finds a factor of a n -bit integer in time $O(n^2(\log n)(\log \log n))$
- The best classical algorithm that we know of for the same task needs time $O(e^{cn^{\frac{1}{3}}(\log n)^{\frac{2}{3}}})$
- Dramatic consequences for current cryptography (RSA)



Steps of Shor's algorithm

- ① Given N , check that N is not a prime or power of a prime. If it is, stop.
- ② Choose $1 < a < N$ at random
- ③ If $b = \gcd(a, N) > 1$, output b and stop
- ④ Find the order of a mod N , that is, $r > 0$ such that $a^r \equiv 1 \pmod{N}$
- ⑤ If r is odd, go to 2
- ⑥ Compute

$$x = a^{\frac{r}{2}} + 1 \pmod{N}$$

$$y = a^{\frac{r}{2}} - 1 \pmod{N}$$

- ⑦ If $x = 0$, go to 2. If $y = 0$, take $r = \frac{r}{2}$ and go to 5.
- ⑧ Compute $p = \gcd(x, N)$ and $q = \gcd(y, N)$. At least one of them will be a non-trivial factor of N

Correctness of Shor's algorithm

- We know that

$$a^r \equiv 1 \pmod{N}$$

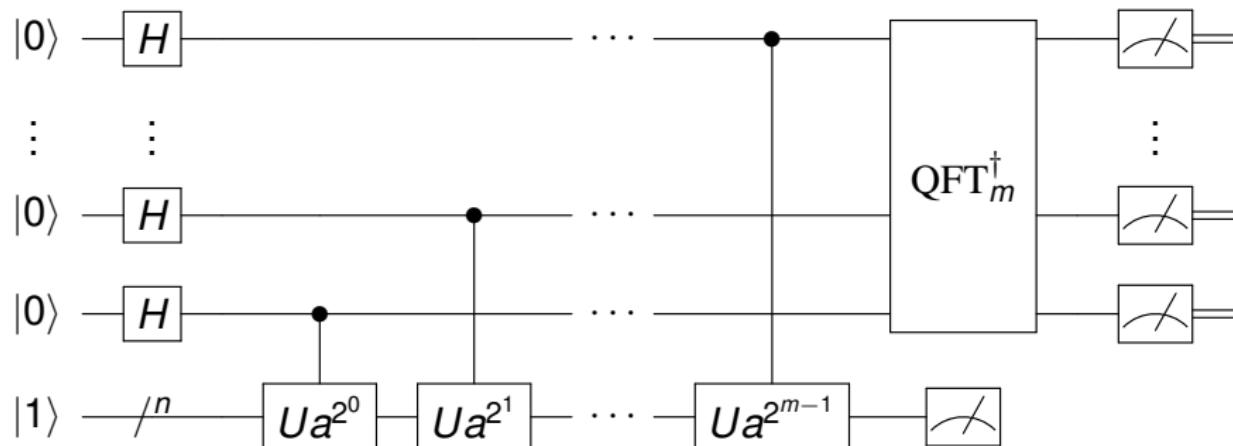
- Thus

$$x \cdot y \equiv (a^{\frac{r}{2}} + 1)(a^{\frac{r}{2}} - 1) \equiv (a^r - 1) \equiv 0 \pmod{N}$$

- This means that $x \cdot y$ is a multiple of N
- Since neither x nor y are multiples of N , either p or q divides N
- It can be proved that step 8 will be reached with high probability

Implementation of Shor's algorithm

- Every step but number 4 are carried out on a classical computer (efficient algorithms exist)
- For step 4, there exists a quantum circuit with a number of gates that is polynomial on n (the number of bits of N)



Preparing a periodic sequence

- The first part of the circuit computes

$$\frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle |a^x \bmod N\rangle$$

- When we measure the bottom qubits, we obtain

$$\frac{1}{\sqrt{|C|}} \sum_{x \in C} |x\rangle |c\rangle$$

where c is some value in $\{0, \dots, N-1\}$ and $C = \{x : a^x \bmod N = c\}$.

Preparing a periodic sequence (2)

- For example, if $a = 2$, $N = 5$, $m = 4$, we would have

$$\frac{1}{4} (|0\rangle|1\rangle + |1\rangle|2\rangle + |2\rangle|4\rangle + |3\rangle|3\rangle + |4\rangle|1\rangle + \dots + |15\rangle|3\rangle)$$

and when we measure we could obtain, for instance

$$\frac{1}{2} (|1\rangle|2\rangle + |5\rangle|2\rangle + |9\rangle|2\rangle + |13\rangle|2\rangle)$$

- Notice that the values of the first register are exactly 4 units apart and that $2^4 = 1 \pmod{5}$.
- In general, we will obtain values that are r units apart, where $a^r = 1 \pmod{N}$.

Measuring the period

- To retrieve the period r we use the (inverse) of the Quantum Fourier Transform (QFT)
- Two properties of the QFT are central here:
 - Shift-invariance (up to an unobservable phase)
 - QFT transforms sequences with period r into sequences with period $\frac{M}{r}$ (where $M = 2^m$)
- After the use of the inverse QFT, we can measure a value of the form $\frac{Mc}{r}$ with high probability and, from it, obtain r

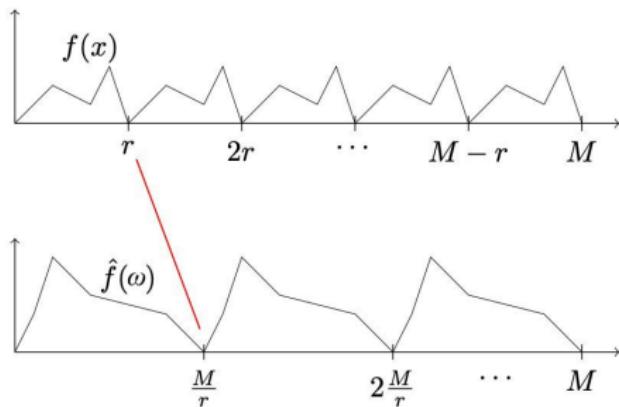


Image credits: Umesh Vazirani

Quantum Fourier Transform: definition and circuit

- The QFT of order m is the unitary transformation defined by

$$QFT |j\rangle = \frac{1}{\sqrt{2^m}} \sum_{k=0}^{2^m-1} e^{\frac{2\pi i j k}{2^m}} |k\rangle$$

- The circuit in the figure implements the QFT
- The R_k gates in the circuit are what we call $R_Z(\frac{2\pi}{2^k})$
- The number of gates is quadratic in m , an exponential speed-up over the classical case (FFT)
- For Shor, m can be chosen to be about $2n$

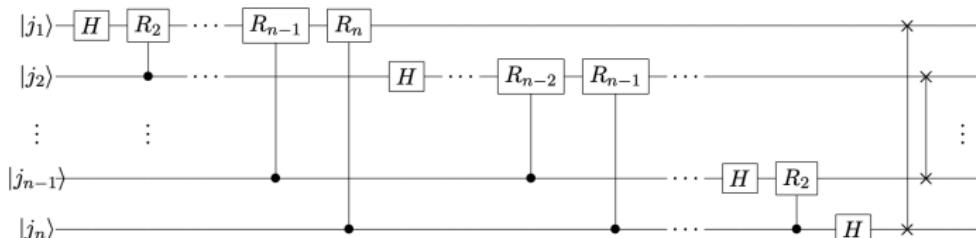


Image credits: Jurgen Van Gael

Using the QFT for phase estimation

- Suppose we are given a unitary operation U and one of its eigenvectors $|\psi\rangle$
- We know that there exists $\theta \in [0, 1)$ such that $U|\psi\rangle = e^{2\pi i \theta}|\psi\rangle$
- We can estimate θ with the circuit shown below
- With the first part, we will obtain $\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i \theta k} |k\rangle$
- By using the inverse QFT we can measure $j \approx 2^n \theta$

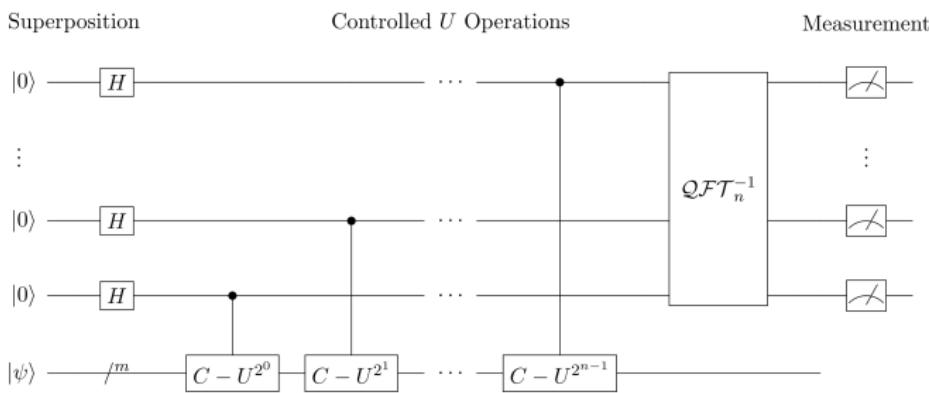


Image credits: Wikipedia

Shor's algorithm as a particular case of quantum phase estimation

- Clearly, the circuit used in Shor's algorithm is a case of quantum phase estimation
- It can be shown that the (unitary) operation of modular multiplication by a has eigenvalues

$$e^{2\pi i \frac{k}{r}} \quad k = 0, \dots, r-1$$

where r is the period of a

- It is not easy to prepare one of the eigenvectors $|\psi_k\rangle$ of the unitary operation
- But we use the fact that

$$|1\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |\psi_k\rangle$$

- We will then measure a value close to $\frac{2^m k}{r}$ for some k

Using quantum phase estimation to count the number of marked elements

- We can use Grover's algorithm together with the QFT to count the number of elements marked by a boolean function
- The eigenvalues of Grover's operator are $e^{\pm 2i\theta}$ where $\sin \theta = \sqrt{\frac{k}{N}}$
- Then, with quantum phase estimation we can recover k , the number of marked elements

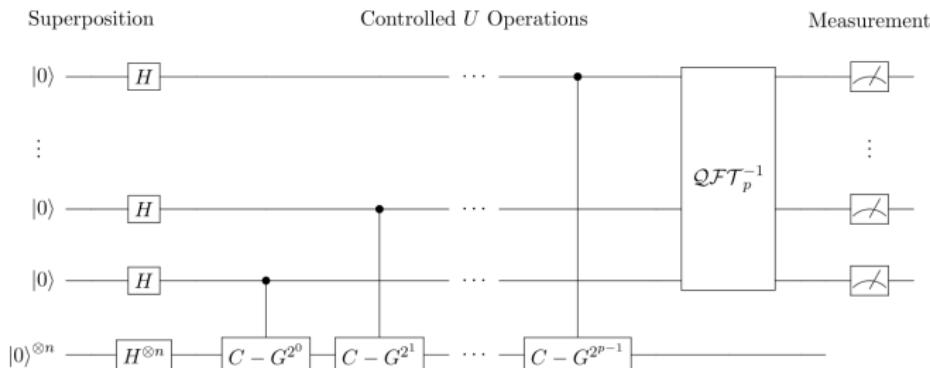


Image credits: Wikipedia

HHL: Applying quantum phase estimations to solve linear systems of equations

- A quantum algorithm proposed in 2009 by Harrow, Hassidim and Lloyd can be used to solve linear systems of equations
- The main steps of the algorithm are
 - Computation of the eigenvalues (quantum phase estimation)
 - Inversion of the eigenvalues
 - Uncomputation of the eigenvalues (inverse of quantum phase estimation)

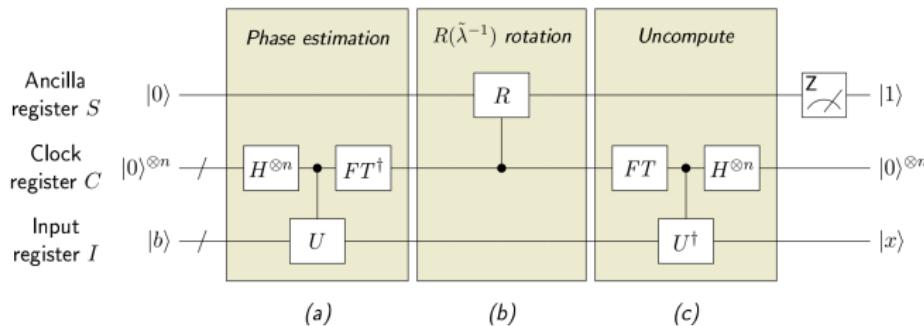


Image credits: Niel de Beaudrap