Quantum Computing
---
an introduction for computing scientists

LPC, 17 April 2020, Clermont-Ferrand

Bogdan Vulpescu

Laboratoire de Physique de Clermont

Service Informatique

Summary of the talk

- Classical bits and classical computing
- Quantum mechanics and quantum bits (qubits)
- Manipulating qubit states, unitary errors
- Quantum gates and circuits
- Quantum teleportation
- Quantum (Discrete) Fourier Transform
- Quantum cryptography
- IBM Q Experience, programming languages for QC

- Classical bits and classical computing
- Quantum mechanics and quantum bits (qubits)
- Manipulating qubit states, unitary errors
- Quantum gates and circuits
- Quantum teleportation
- Quantum (Discrete) Fourier Transform
- Quantum cryptography
- IBM Q Experience, programming languages for QC

Semiconductors

Coming out from a childhood of heavy electro-mechanical devices, the “classical” computing technology succeeded in building commuting devices based on semiconductors: their conductivity can be controlled by doping and driven with electric fields. This lead to the discovery of the “transistor effect” in 1947.

https://commons.wikimedia.org/w/index.php?curid=7353911

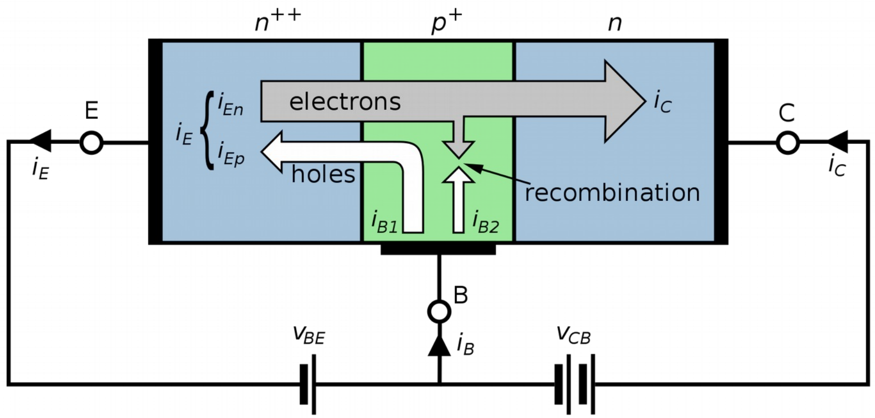
a silicon crystal

The Ebers-Moll equations of the bipolar junction transistor :

$$i_{\mathrm{C}} = I_{\mathrm{S}} \left[\left(e^{\frac{V_{\mathrm{BE}}}{V_{\mathrm{T}}}} - e^{\frac{V_{\mathrm{BC}}}{V_{\mathrm{T}}}} \right) - \frac{1}{\beta_{\mathrm{R}}} \left(e^{\frac{V_{\mathrm{BC}}}{V_{\mathrm{T}}}} - 1 \right) \right]$$

$$i_{\mathrm{B}} = I_{\mathrm{S}} \left[\frac{1}{\beta_{\mathrm{F}}} \left(e^{\frac{V_{\mathrm{BE}}}{V_{\mathrm{T}}}} - 1 \right) + \frac{1}{\beta_{\mathrm{R}}} \left(e^{\frac{V_{\mathrm{BC}}}{V_{\mathrm{T}}}} - 1 \right) \right]$$

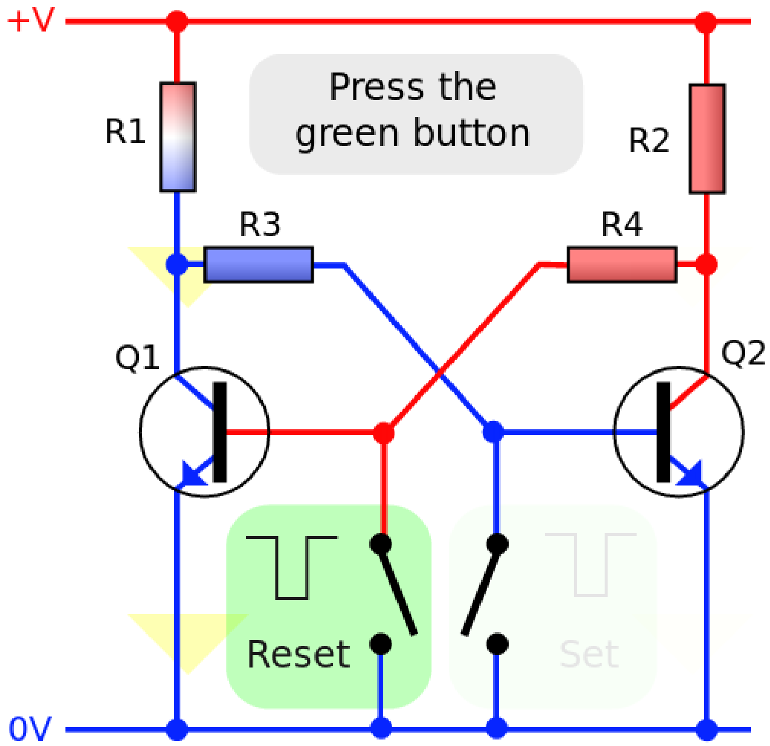
$$i_{\mathrm{E}} = I_{\mathrm{S}} \left[\left(e^{\frac{V_{\mathrm{BE}}}{V_{\mathrm{T}}}} - e^{\frac{V_{\mathrm{BC}}}{V_{\mathrm{T}}}} \right) + \frac{1}{\beta_{\mathrm{F}}} \left(e^{\frac{V_{\mathrm{BE}}}{V_{\mathrm{T}}}} - 1 \right) \right]$$

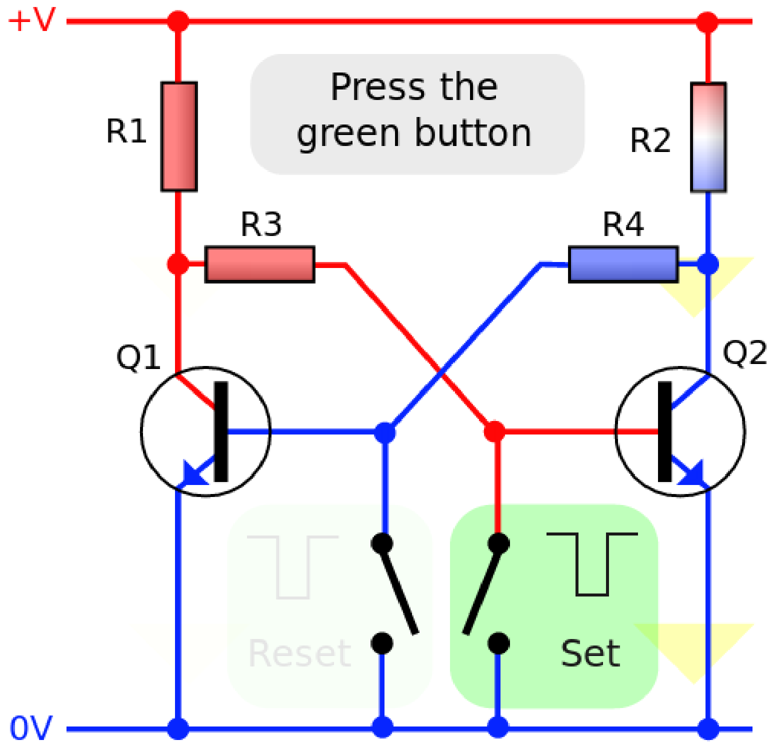


https://en.wikipedia.org/wiki/Bipolar_junction_transistor

The flip-flop circuit (a bi-stable circuit)

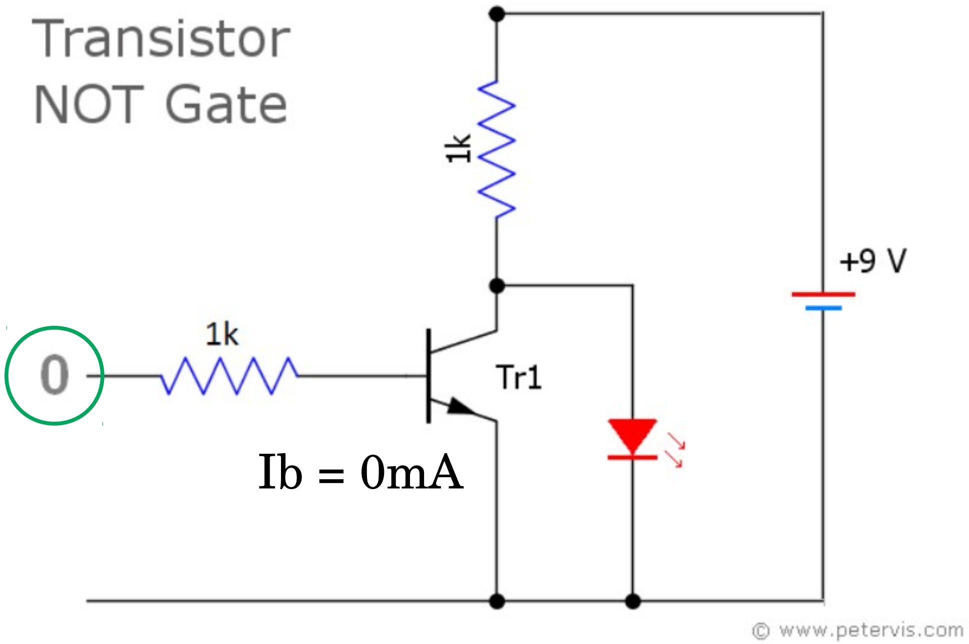
With such a device we can store a single bit of data (0 or 1) :

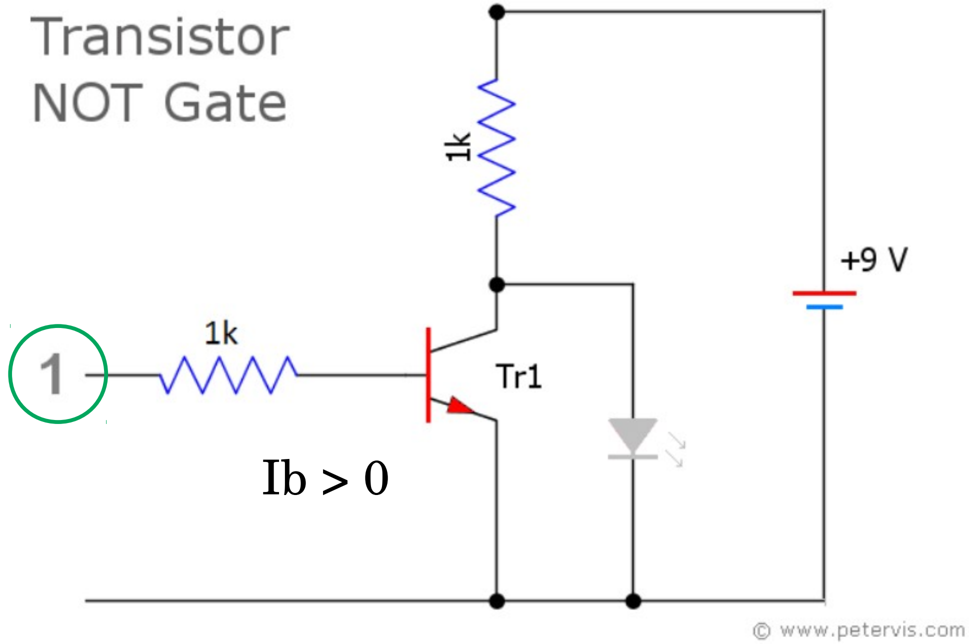




https://en.wikipedia.org/wiki/Flip-flop_(electronics)

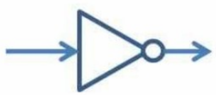
Inverter (0 ⇒1, 1 ⇒ 0) with transistor-transistor logic (TTL)

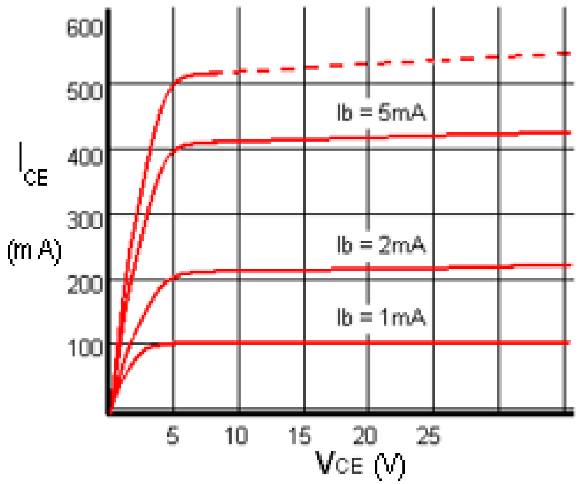




NOT

x	F
0	1
1	0





Elementary logic gates: one-bit logic gates

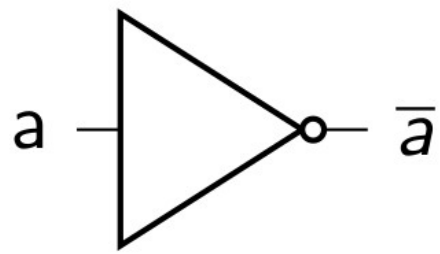
$$f:\{0,1\}\rightarrow\{0,1\}$$

IDENTITY

$$a = a$$

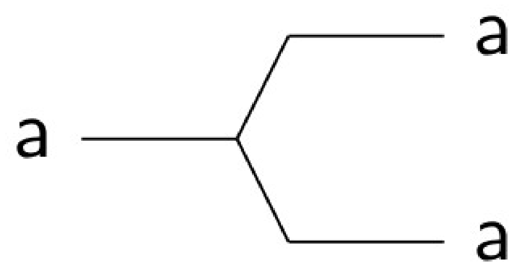
NOT

$$\bar{a} = 1-a$$



a	$\bar{a}$
0	1
1	0

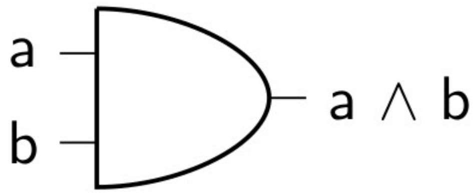
FANOUT (COPY)



Elementary logic gates: two-bit logic gates

$$f:\{0,1\}^2 \rightarrow \{0,1\}$$

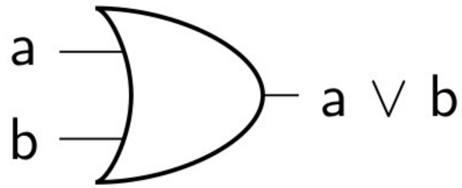
AND



a	b	a ∧ b
0	0	0
0	1	0
1	0	0
1	1	1

$$a \wedge b \;=\; ab$$

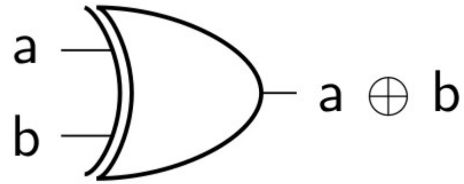
OR



a	b	a ∨ b
0	0	0
0	1	1
1	0	1
1	1	1

$$a \vee b \;=\; a+b-ab$$

XOR



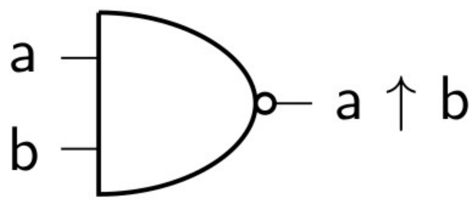
a	b	a ⊕ b
0	0	0
0	1	1
1	0	1
1	1	0

$$a \oplus b \;=\; a+b \quad (mod\, 2)$$

Elementary logic gates: two-bit logic gates (cont.)

$$f:\{0,1\}^2 \rightarrow \{0,1\}$$

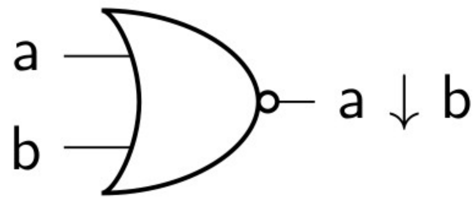
NAND



a	b	a ↑ b
0	0	1
0	1	1
1	0	1
1	1	0

$$a \uparrow b \;=\; \overline{a \wedge b} \;=\; \overline{ab} \;=\; 1-ab$$

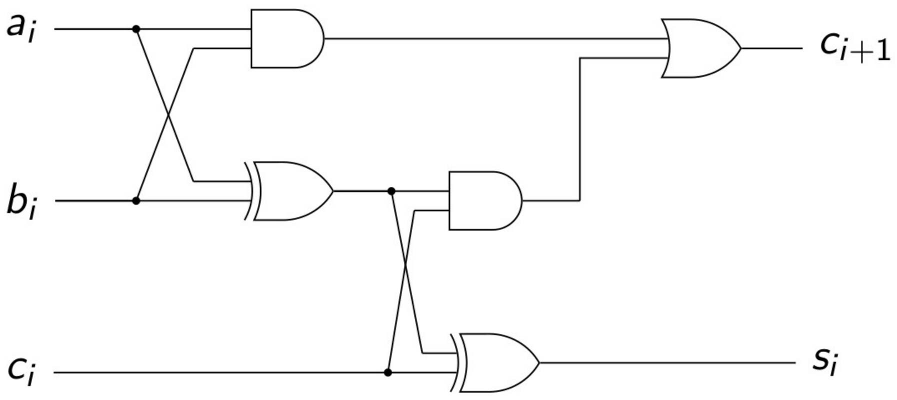
NOR



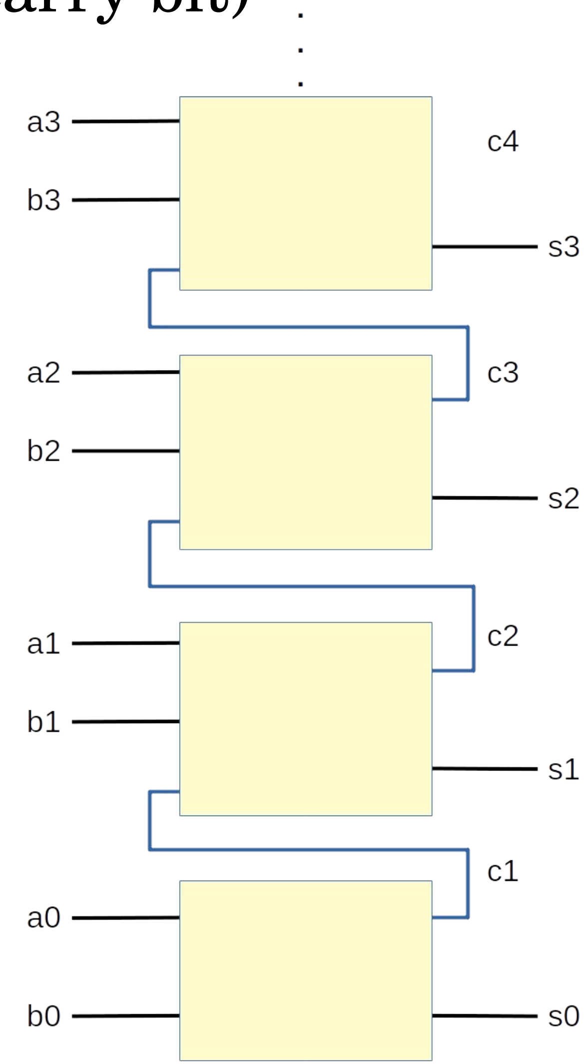
a	b	a ↓ b
0	0	1
0	1	0
1	0	0
1	1	0

$$a \downarrow b \;=\; \overline{a \vee b} \;=\; \overline{a+b-ab} \;=\; 1-a-b+ab$$

A circuit for computing the sum (with carry bit)



Full adder



Given the binary representations

$$a = (a_{n-1}, \ldots, a_1, a_0) \quad , \quad b = (b_{n-1}, \ldots, b_1, b_0),$$

the i-th bit of the sum is

$$s_i = a_i + b_i + c_i \pmod 2)$$

where c_i is the carry over from the sum $a_{i-1} + b_{i-1} + c_{i-1}$. The carry over is set to one if two or more of the input bits a_i, b_i and c_i are 1 and 0 otherwise. This circuit can be built with the following elementary gates: 2 AND, 1 OR, 2 XOR and 4 FANOUT.

Universal (classical) gates

Any function $f:\{0,1\}^m \rightarrow \{0,1\}^n$ can be constructed from the elementary gates :

AND, OR, NOT, FANOUT

We say that AND, OR, NOT and FANOUT constitute **a universal set of gates** for the classical computation.

A smaller universal set is NAND and FANOUT :

OR can be obtained from NOT and AND: $a \vee b \;=\; \overline{\bar{a} \wedge \bar{b}}$ (De Morgan's identities)

and NOT can be obtained from NAND and FANOUT :

$$a \uparrow a \;=\; \overline{a \wedge a} \;=\; 1 - a^2 \;=\; 1 - a \;=\; \bar{a}$$

here we have FANOUT followed by NAND

Classical reversible computing

It is possible to embed any irreversible function into a reversible function :

irreversible function : $f:\{0,1\}^m \rightarrow \{0,1\}^n$ $\quad m>n$

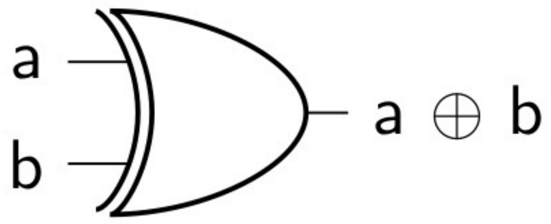
reversible function : $\tilde{f}:\{0,1\}^{m+n} \rightarrow \{0,1\}^{m+n}$

defined such that : $\tilde{f}(x,y) = (x,[y+f(x)](mod\, 2^n))$

where **x** represents **m** bits, while **y** and **f(x)** represent **n** bits. Since the embedding function is **bijective**, it will be **reversible**! So at the logic level it is possible, with the price of introducing more dimensions in the calculations (**ancillary** bits **y**).

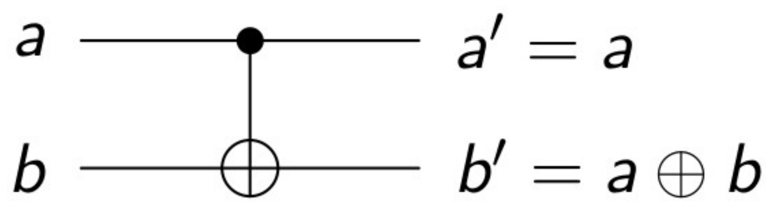
A simple reversible classical gate: the controlled-NOT (CNOT)

The exclusive-OR function (XOR):



a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

The CNOT gate:

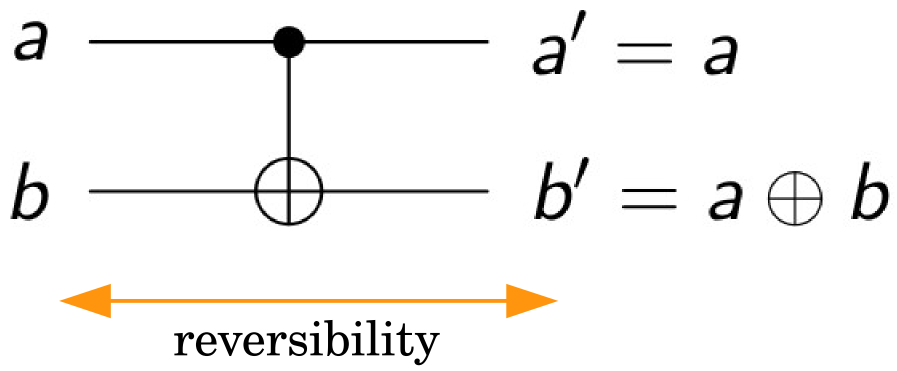


a	b	a'	b'
0	0	**0**	0
0	1	**0**	1
1	0	**1**	1
1	1	**1**	0

The circuit representation of the classical CNOT gate

the control bit:

the target bit:



a) two CNOT gates, applied one after the other :

$$(a,b) \rightarrow (a, a \oplus b) \rightarrow (a, a \oplus (a \oplus b)) = (a,b)$$

CNOT is self-inverse : $(CNOT)^2 = I \quad , \quad CNOT^{-1} = CNOT$

b) if the target bit is set to 0 (b=0), CNOT reproduces the FANOUT gate :

$$(a,0) \rightarrow (a,a)$$

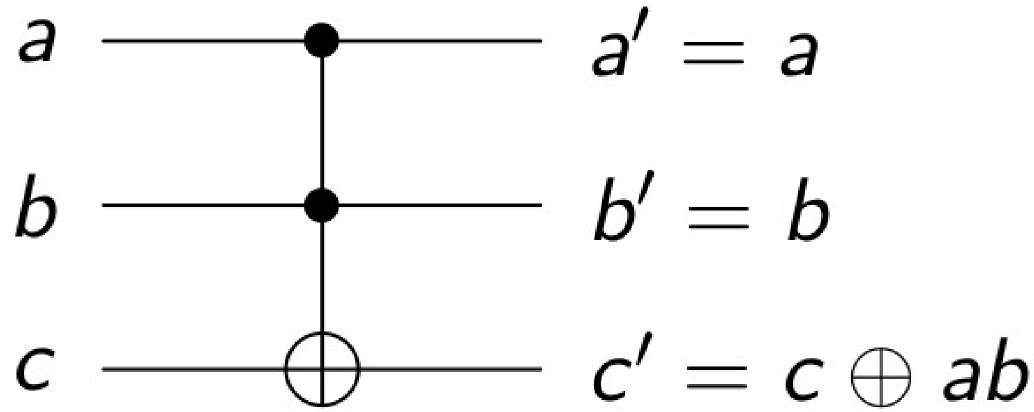
Two-bit reversible gates are not enough for universal computation !
(we can not construct the NAND gate ...)

Three-bit reversible gates: the Toffoli gate (controlled-controlled-NOT, or C^2NOT)

control bit [0]:

control bit [1]:

target bit [2]:



The Toffoli gate is a universal gate!

NOT: $a=b=1$, $c'=\bar{c}$

AND: $c=0$, $c'=a \wedge b$

OR: $a \rightarrow \bar{a}$, $b \rightarrow \bar{b}$, $c=1$, $c'=a \vee b$

- Classical bits and classical computing
- Quantum mechanics and quantum bits (qubits)
- Manipulating qubit states, unitary errors
- Quantum gates and circuits
- Quantum teleportation
- Quantum (Discrete) Fourier Transform
- Quantum cryptography
- IBM Q Experience, programming languages for QC

Quantum bits (qubits)

A **qubit** is a quantum object: a microscopic system whose state and evolution are governed by the laws of **quantum mechanics**. In order to keep a good resemblance with the classical bit, this system will be chosen to have only two possible quantum states, corresponding to some measurable physical property.

The two states are **orthogonal** and any arbitrary state of the system can be described as a linear combination (superposition) of those two states :

$$|\psi\rangle = \alpha\ |0\rangle + \beta\ |1\rangle \qquad\qquad |\alpha|^2 + |\beta^2| = 1 \qquad \alpha, \beta \in \mathbb{C}$$

(the decomposition of a general vector in a 2-dim Hilbert space using the computational basis)

The 1st postulate of quantum mechanics

The state vector (or wave function) completely describes the state of the physical system.

The evolution in time of the state vector is governed by the Schrödinger equation:
(H is the Hamiltonian, a self-adjoint operator)

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

the 6th postulate

The coefficients α and β multiplying the vectors of the computational basis are functions of time:

$$|\psi(t)\rangle = \alpha(t) |0\rangle + \beta(t) |1\rangle$$

$$\hbar \approx 6.626 \times 10^{-34} \, Joule \cdot sec$$

$$i = \sqrt{-1}$$

Vector algebra with qubits

Since we describe our space with two coordinates, we can write the two basis vectors like this :

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \quad \text{and} \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$$

("ket" vectors)

and their superposition in the state vector : $|\psi\rangle = \alpha\begin{pmatrix}1\\0\end{pmatrix} + \beta\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}\alpha\\\beta\end{pmatrix}$

The vectors of the computational basis are normalized orthogonal vectors :

product of a "bra" vector and a "ket" vector

$$\langle 0|0\rangle = (1 \;\; 0)\begin{pmatrix}1\\0\end{pmatrix} = 1 \quad , \quad \langle 0|1\rangle = (1 \;\; 0)\begin{pmatrix}0\\1\end{pmatrix} = 0$$

$$\langle 1|0\rangle = (0 \;\; 1)\begin{pmatrix}1\\0\end{pmatrix} = 0 \quad , \quad \langle 1|1\rangle = (0 \;\; 1)\begin{pmatrix}0\\1\end{pmatrix} = 1$$

The 2nd and 3rd postulates of quantum mechanics

<u>We associate with any observable a **self-adjoint operator** on the Hilbert space of the states. The only possible outcome of a measurement is one of the eigen-values of the corresponding operator (3rd postulate).</u>

A single-qubit operator can be represented as a 2x2 matrix : $\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
(described within a given orthonormal vector base)

$$\sigma_z \, |0\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = +1 \, |0\rangle$$

$$\sigma_z \, |1\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix} = -1 \, |1\rangle$$

$|0\rangle$ and $|1\rangle$ are eigen-vectors of the operator σ_z with eigen-values “+1” and “-1”

The probability of a given measurement outcome (the 4th postulate)

If we expand the state vector over the orthonormal basis formed by the eigen-vectors of the operator corresponding to the observable:

$$|\psi(t)\rangle = \alpha(t)\,|0\rangle + \beta(t)\,|1\rangle$$

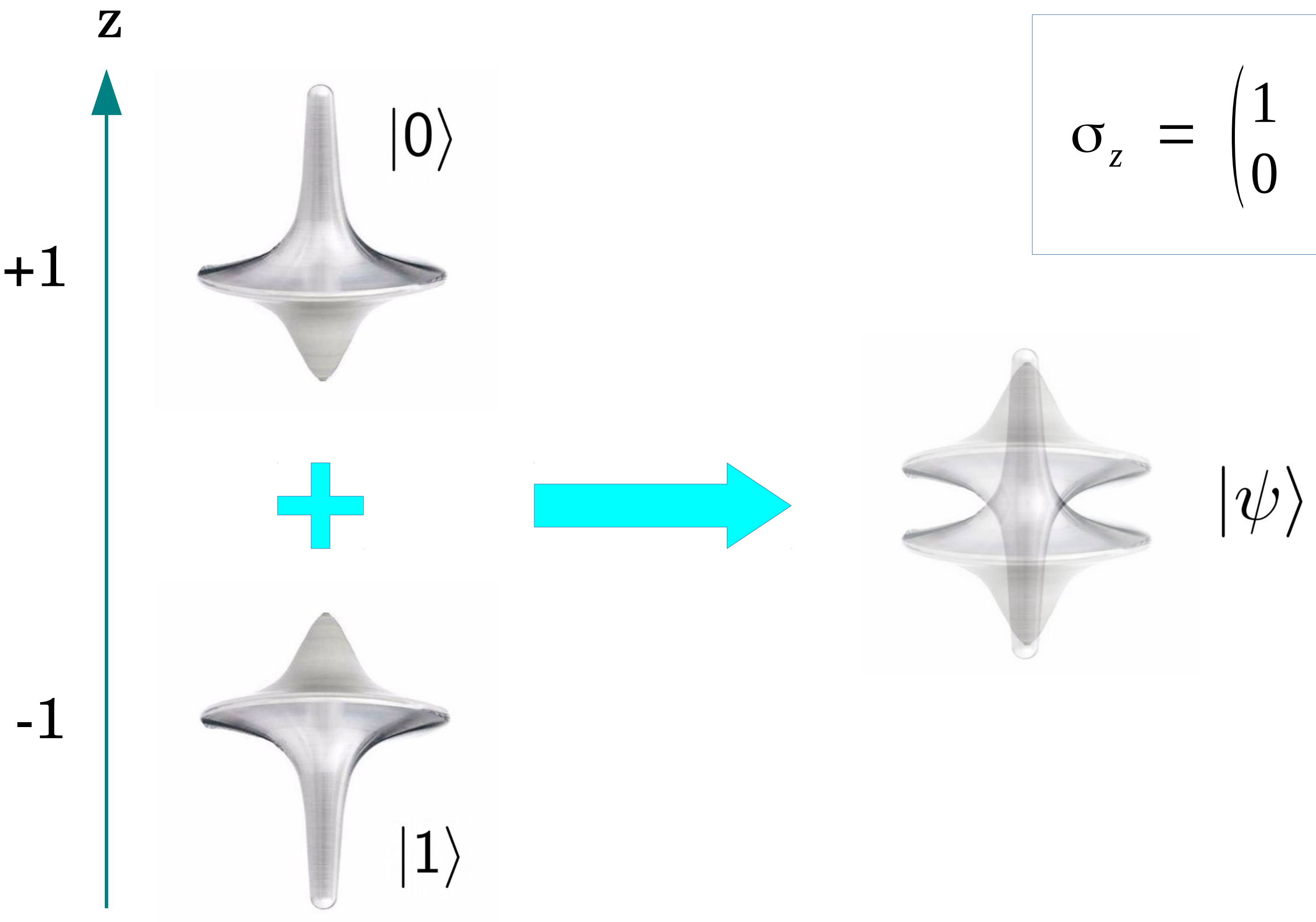
then <u>the probability that a measurement at time t</u> results in outcome "+1" or "-1" is given, respectively, by :

$$p_{+1}(t) = |\,\langle 0|\psi(t)\rangle\,|^2 = |\alpha(t)|^2$$

$$p_{-1}(t) = |\,\langle 1|\psi(t)\rangle\,|^2 = |\beta(t)|^2$$

Note: <u>global phase</u> factors $|\psi'\rangle = e^{i\theta}\,|\psi\rangle$ do not affect physical predictions!

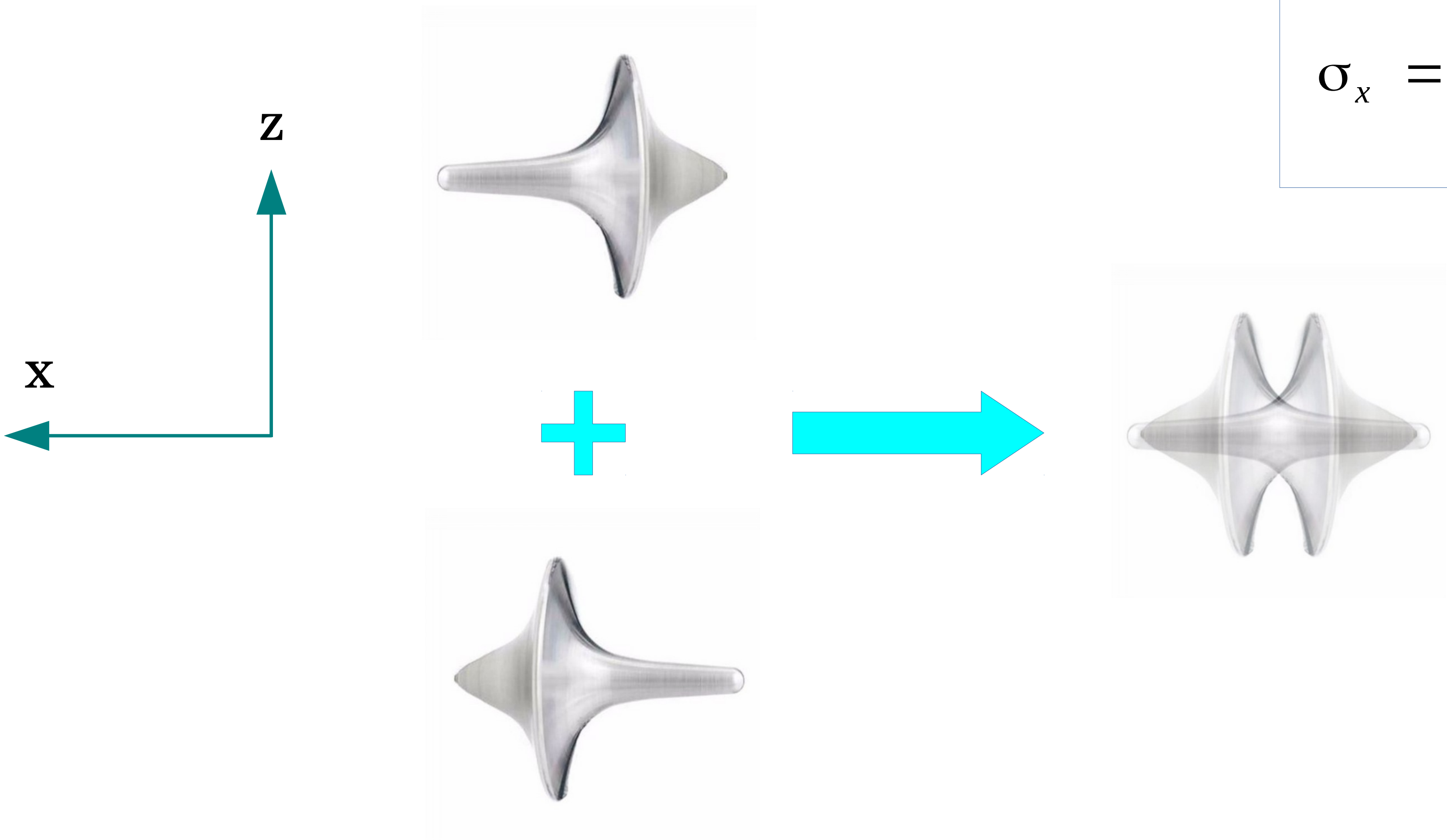
The quantified spin and the choice of the direction of the measurement



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

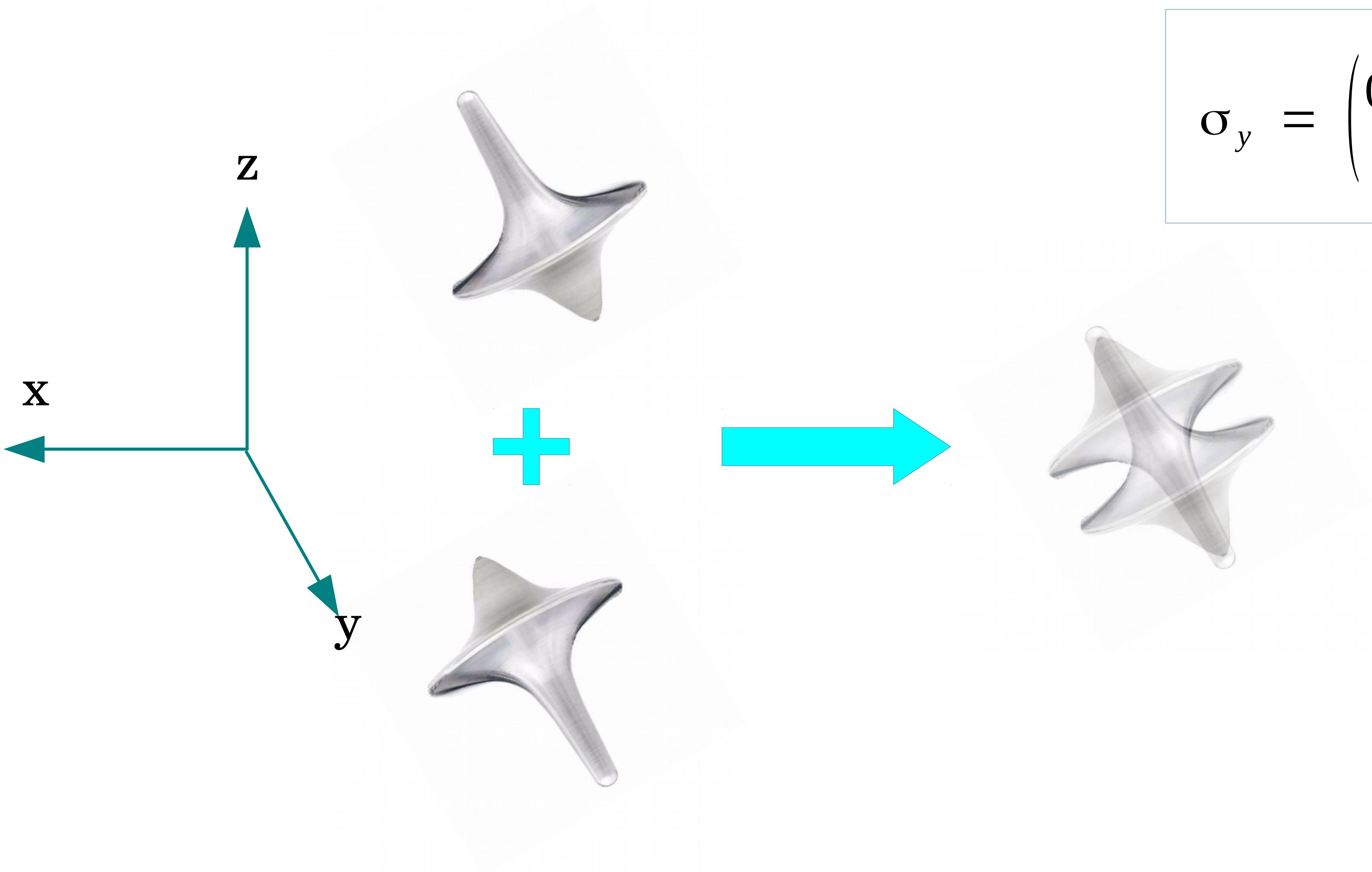
The quantified spin and the choice of the direction of the measurement

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

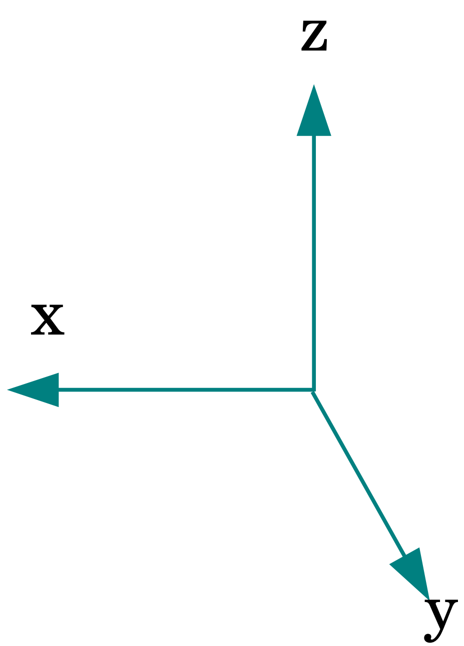


The quantified spin and the choice of the direction of the measurement

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



The quantified spin and the choice of the direction of the measurement



σ_x , σ_y , σ_z = Pauli matrices (operators), also σ_1 , σ_2 , σ_3

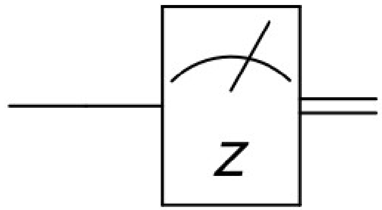
The eigen-vectors of the Pauli operators corresponding to eigen-values "+1" and "-1"

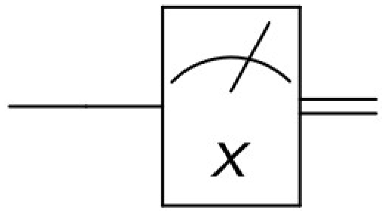
$$\sigma_x : \qquad |+\rangle_x = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad , \quad |-\rangle_x = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

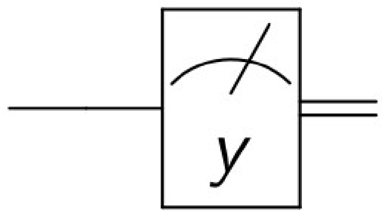
$$\sigma_y : \qquad |+\rangle_y = \frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle) \quad , \quad |-\rangle_y = \frac{1}{\sqrt{2}}(|0\rangle - i\,|1\rangle)$$

$$\sigma_z : \qquad |+\rangle_z = |0\rangle \quad , \quad |-\rangle_z = |1\rangle$$

The circuit symbol for a measurement

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

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

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

Note: double line means that this is a classical information (a bit).

The 5th postulate of quantum mechanics

If a system is described by the state vector $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ and we measure σ_z obtaining the outcome (spin projection) +1 or -1, then, <u>immediately after the measurement, the state of the system is given by the eigen-vector corresponding to that eigen-value</u> : $|0\rangle$ or $|1\rangle$, respectively.

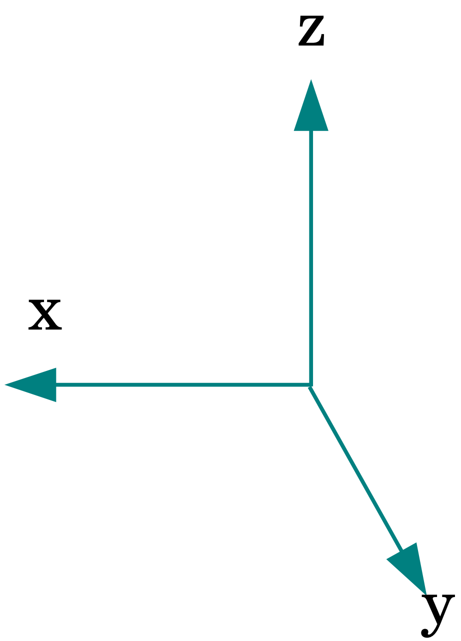
The expected value of an observable will be (4^{th} postulate):

$$\langle\sigma_z\rangle = \sum_n s_n p_n = \sum_n s_n \langle\psi|P_n|\psi\rangle = \langle\psi|(\sum_n s_n P_n)|\psi\rangle = \langle\psi|\sigma_z|\psi\rangle$$

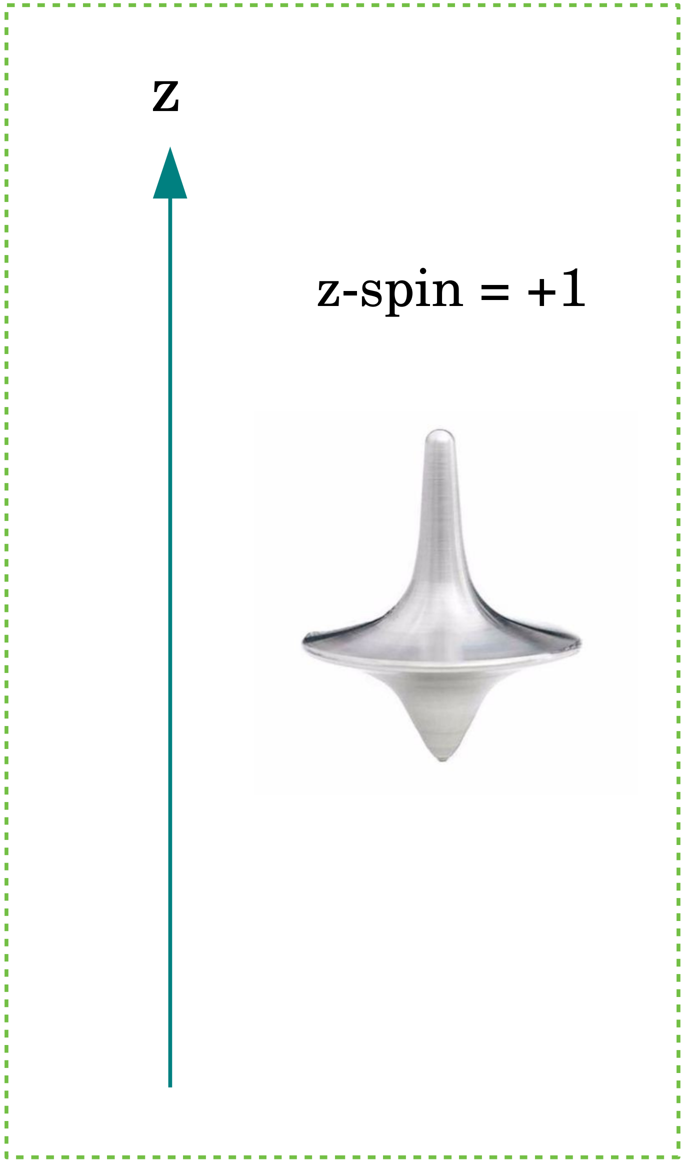
from the outcome probabilities: $p_n = \langle\psi|P_n|\psi\rangle$

with the projector operators: $P_1 = |0\rangle\langle 0|$, $P_2 = |1\rangle\langle 1|$

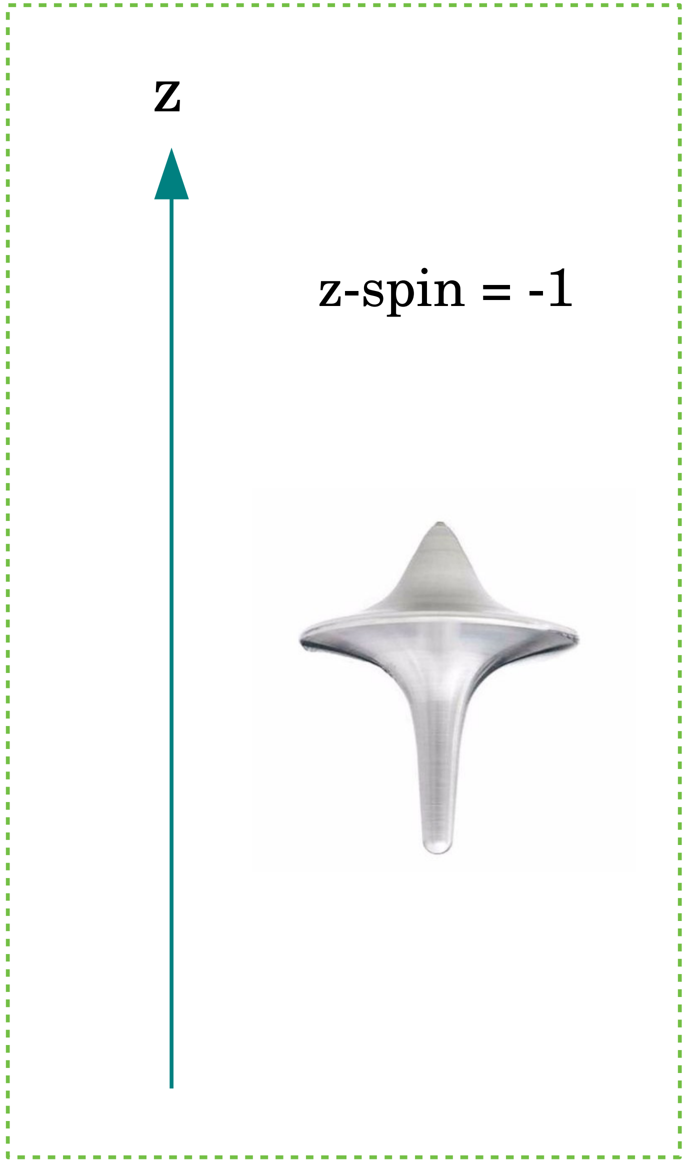
Before the measurement of the z spin component



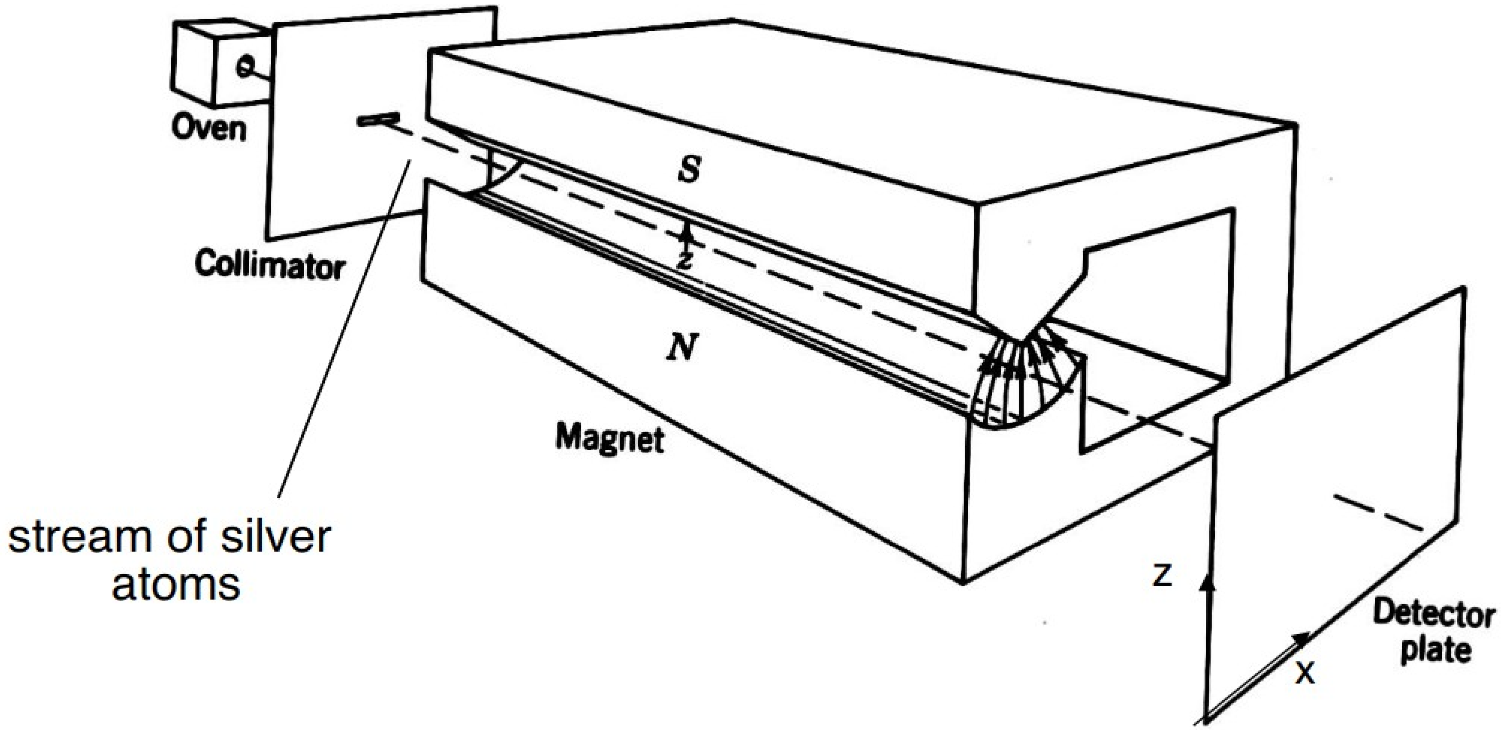
After the measurement



or

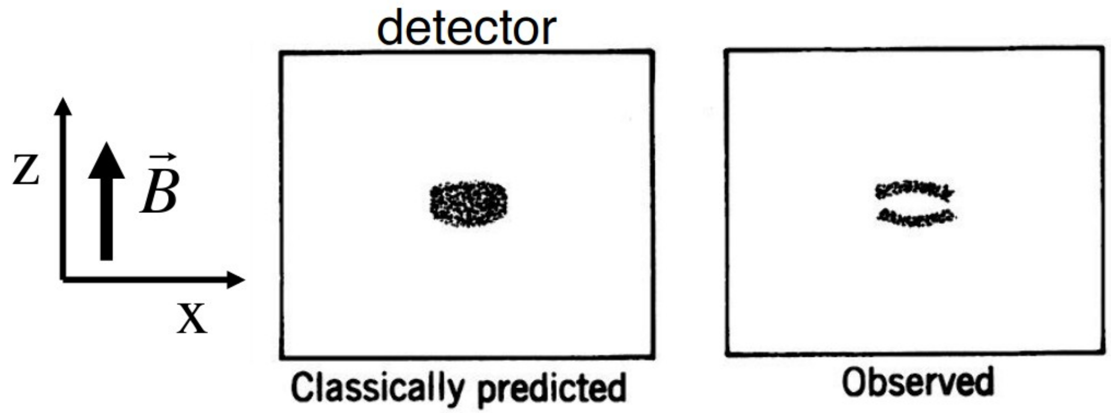


The Stern-Gerlach experiment



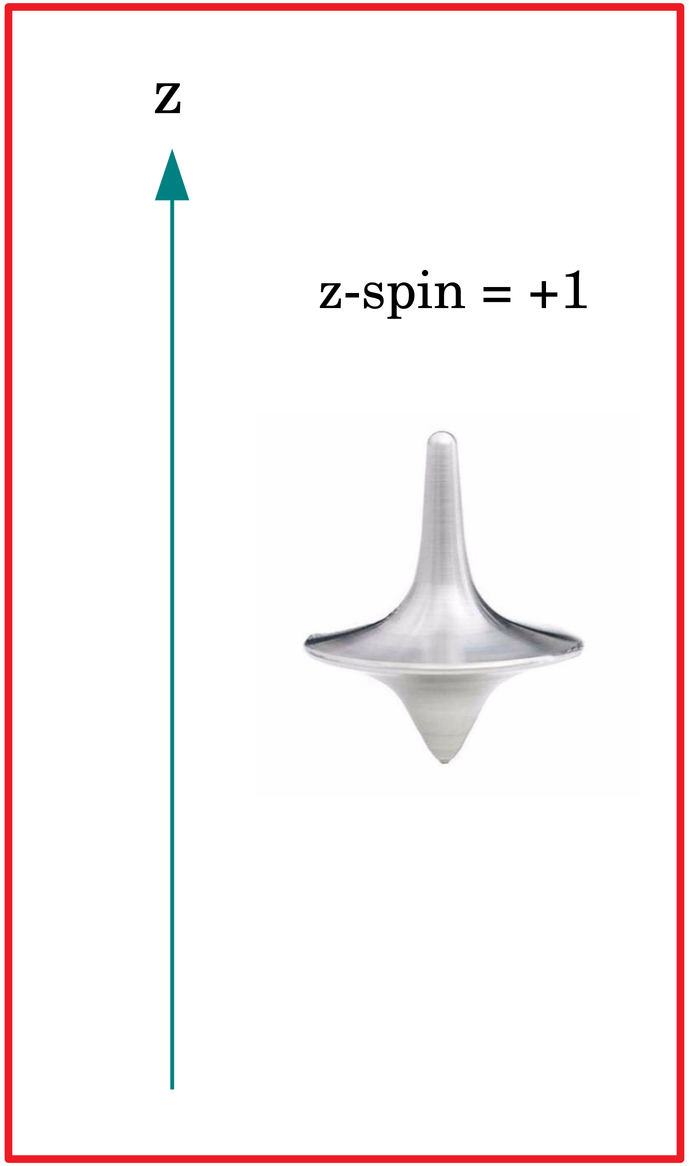
Force proportional to the gradient of the magnetic field

$$F_z = \mu \frac{\partial B_z}{\partial z}$$

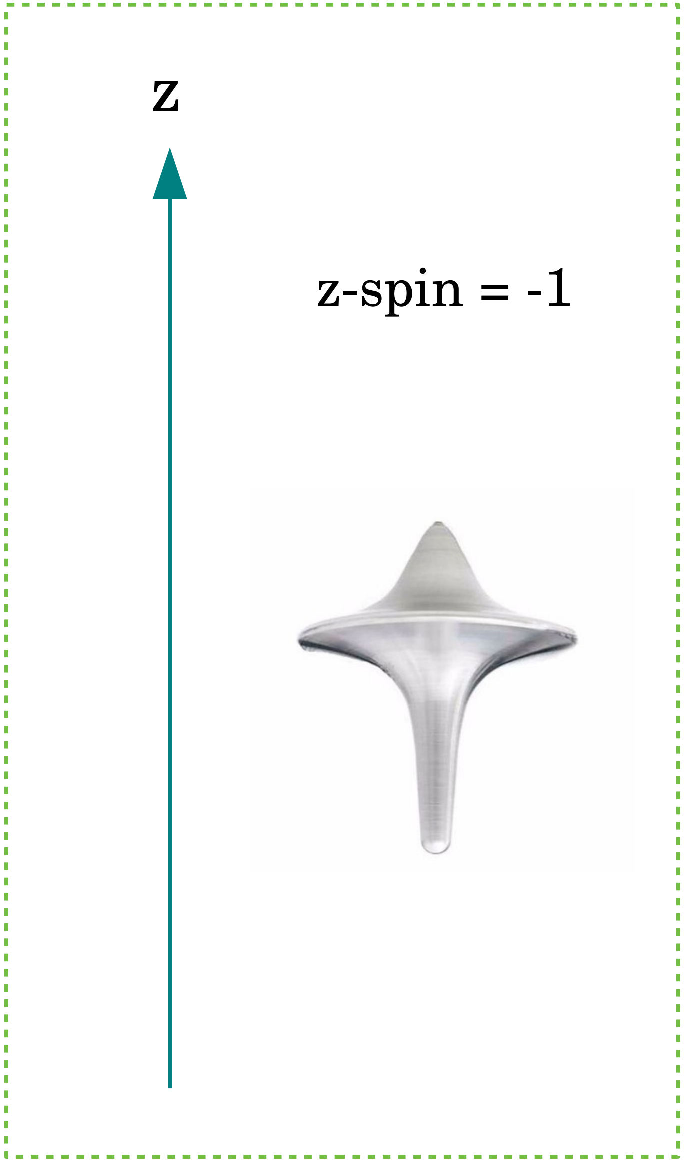


http://web.stanford.edu/class/rad226a/Lectures/Lecture5-2017-Quantum-III.pdf

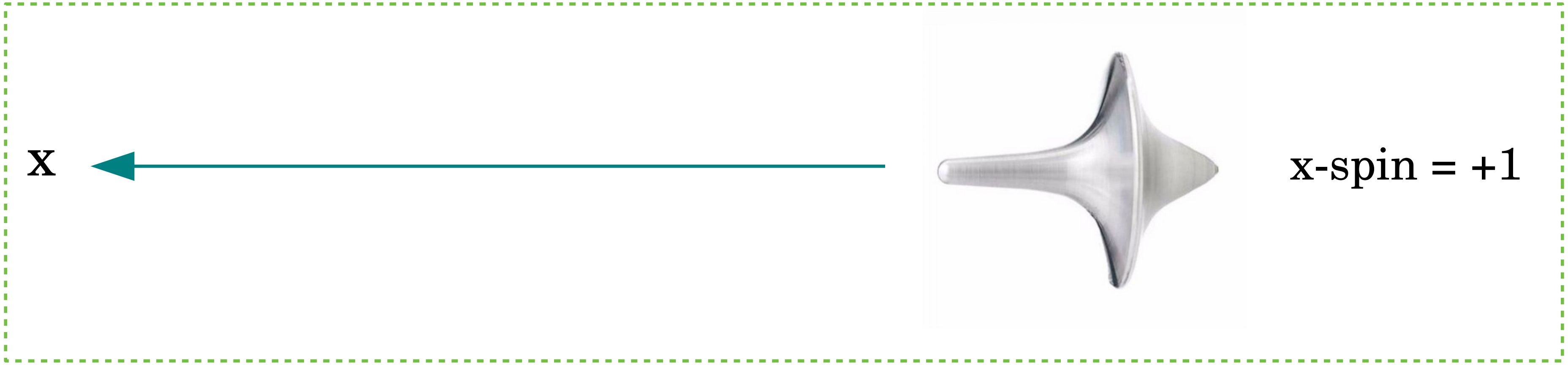
Selection of one z-state



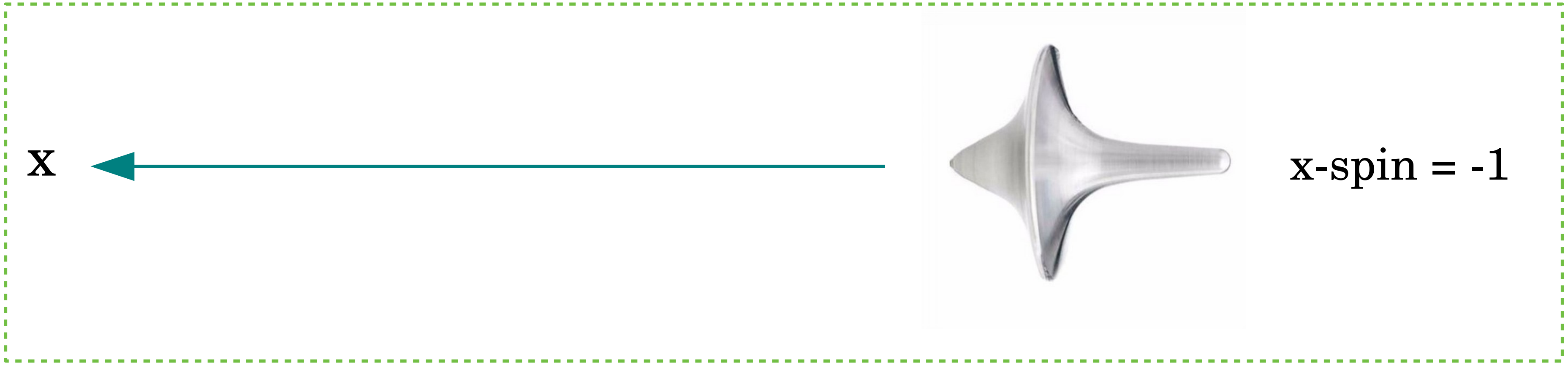
or



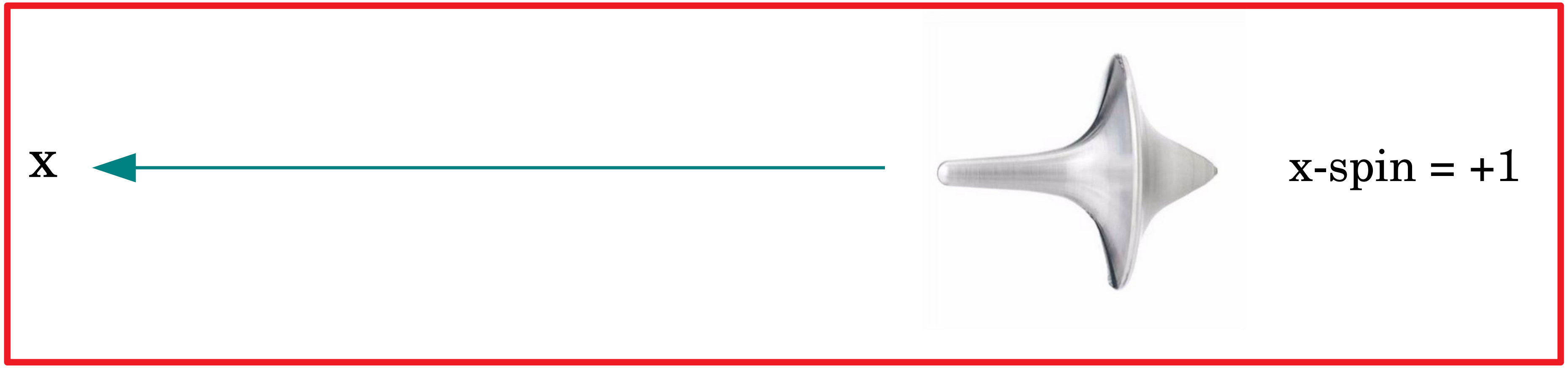
After the second measurement



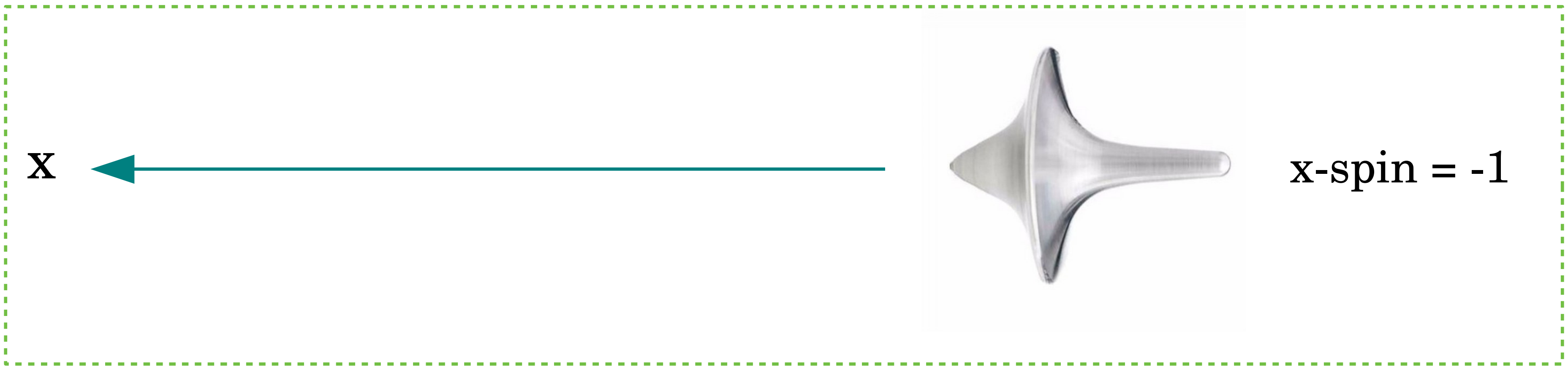
or



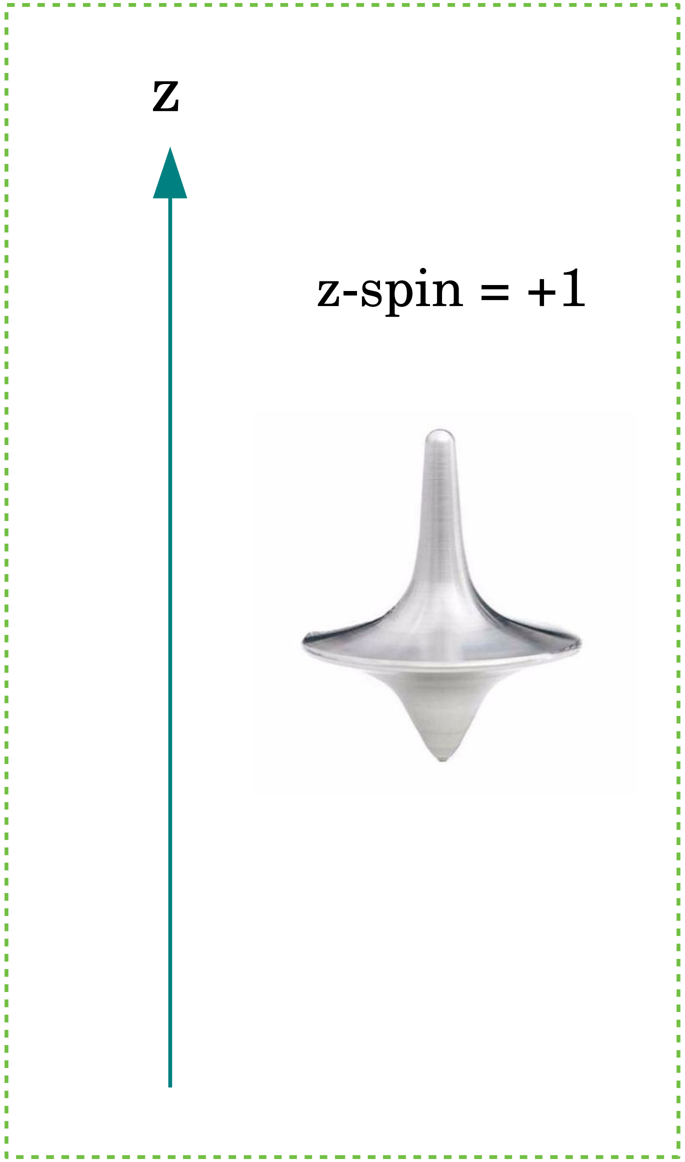
Selection of one x-state



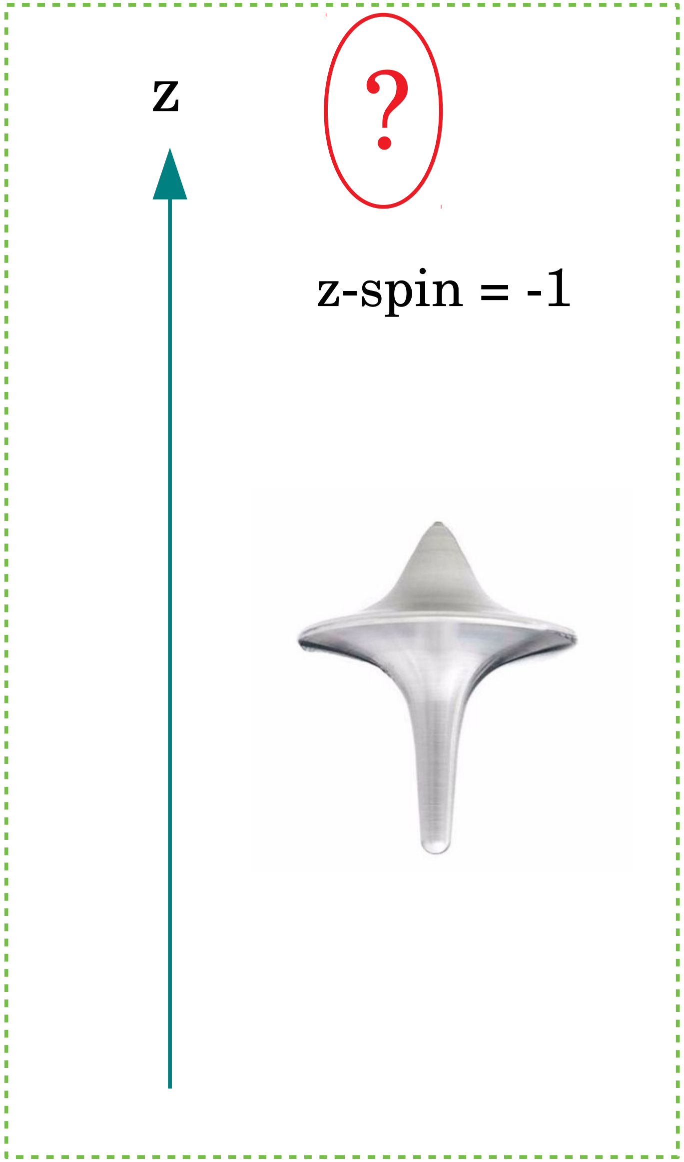
or



After the last measurement



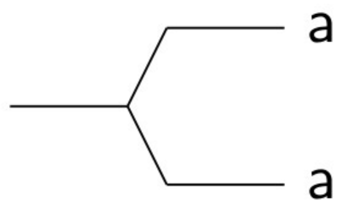
or



- Classical bits and classical computing
- Quantum mechanics and quantum bits (qubits)
- **Manipulating qubit states, unitary errors**
- Quantum gates and circuits
- Quantum teleportation
- Quantum (Discrete) Fourier Transform
- Quantum cryptography
- IBM Q Experience, programming languages for QC

The no-cloning theorem

Contrary to the classical case, it is not possible to clone (FANOUT, COPY) a **generic quantum state**.

The equivalent of this : a 

does not exist in the quantum case!

It is impossible to build a machine that operates unitary transformations and is able to clone the generic state of a qubit.

This has important consequences and leads to interesting applications, like the possibility to do quantum cryptography.

The possibility of cloning would also invalidate the **uncertainty relation of Heisenberg**, because it would be possible to simultaneously measure with infinite precision two physical properties of the system on two identical copies of the same quantum state.

Flipping a qubit using a constant magnetic field

The Schrödinger equation : $i\hbar\frac{d}{dt}\left|\psi(t)\right\rangle = H\left|\psi(t)\right\rangle$

The time-evolution operator :

$$\left|\psi(t)\right\rangle = U(t,t_0)\left|\psi(t_0)\right\rangle \quad , \quad U(t,t_0) = \exp\left[-\frac{i}{\hbar}H(t-t_0)\right]$$

in this particular case, U is a <u>unitary operator</u> : $UU^\dagger = U^\dagger U = I$

The Hamiltonian of a spin interacting with a magnetic field is :

$$H = -\mu\mathcal{H}\cdot\sigma \quad , \quad \mathcal{H} = (\mathcal{H}_x, \mathcal{H}_y, \mathcal{H}_z) \quad , \quad \sigma = (\sigma_x, \sigma_y, \sigma_z)$$

Flipping a qubit with a constant magnetic field (cont.)

Using the notations :

$$\begin{cases} n = \dfrac{1}{\sqrt{\mathcal{H}_x^2 + \mathcal{H}_y^2 + \mathcal{H}_z^2}} (\mathcal{H}_x, \mathcal{H}_y, \mathcal{H}_z) \quad , \quad n = (n_x, n_y, n_z) \\[2em] \alpha(t) = \dfrac{\mu t}{\hbar} \sqrt{\mathcal{H}_x^2 + \mathcal{H}_y^2 + \mathcal{H}_z^2} \end{cases}$$

We obtain for the time-evolution operator :

$$U(t) = \begin{bmatrix} \cos\alpha + i\ n_z \sin\alpha & (n_y + i\ n_x)\sin\alpha \\ (-n_y + i\ n_x)\sin\alpha & \cos\alpha - i\ n_z \sin\alpha \end{bmatrix}$$

Flipping a qubit with a constant magnetic field (cont.)

For instance, with a magnetic field : $\mathcal{H} = (\mathcal{H}_x, 0, 0) \quad , \quad n = (1, 0, 0)$

we can flip the state $|0\rangle$ into the state $|1\rangle$ if :

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} = U \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\alpha(t_{01}) & i \;\sin\alpha(t_{01}) \\ i \;\sin\alpha(t_{01}) & \cos\alpha(t_{01}) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

which is fulfilled when :

$$\cos\alpha(t_{01}) = 0 \quad , \quad t_{01} = \frac{\pi\hbar}{2\mu|\mathcal{H}_x|}$$

Unitary errors

Any quantum computation is given by a sequence of quantum gates applied to some initial state :

$$|\psi_n\rangle = \prod_{i=1}^{n} U_i \, |\psi_0\rangle$$

If the errors are unitary (there is no coupling to the environment, although any realistic implementation of a unitary operation will still involve some error, since unitary operators form a continuous set), instead of the operators U_i we apply slightly different operators V_i :

$$|\psi_i\rangle = U_i \, |\psi_{i-1}\rangle \longrightarrow V_i \, |\psi_{i-1}\rangle = |\psi_i\rangle + |E_i\rangle$$

exact transformation

and with error vector E_i

Unitary errors (cont.)

$|\widetilde{\psi_n}\rangle = \prod_{i=1}^{n} V_i \, |\psi_0\rangle$ we start from the product of "real" unitary operators

$V_1 \, |\psi_0\rangle = |\psi_1\rangle + |E_1\rangle$ and calculate recurrently the result of each member of the product

$$V_2(|\psi_1\rangle + |E_1\rangle) = V_2 \, |\psi_1\rangle + V_2 \, |E_1\rangle = |\psi_2\rangle + |E_2\rangle + V_2 \, |E_1\rangle$$

$$V_3(|\psi_2\rangle + |E_2\rangle + V_2 \, |E_1\rangle) = \ldots = |\psi_3\rangle + |E_3\rangle + V_3 \, |E_2\rangle + V_3 V_2 \, |E_1\rangle$$

$\ldots$

$$|\widetilde{\psi_n}\rangle = |\psi_n\rangle + |E_n\rangle + V_n V_{n-1} \, |E_{n-2}\rangle + \cdots + V_n V_{n-1} \ldots V_2 \, |E_1\rangle$$

Unitary errors (cont.)

Each product of unitary operators V_i does not change the amplitude of the error vectors E_i, which we can consider to be upper bounded by some value ε :

$$|\widetilde{\psi_n}\rangle = |\psi_n\rangle + |E_n\rangle + V_n V_{n-1}|E_{n-2}\rangle + \cdots + V_n V_{n-1} \ldots V_2 |E_1\rangle$$

this means that we can upper limit the error on the final state like this

$$\left\| |\widetilde{\psi_n}\rangle - |\psi_n\rangle \right\| < n\epsilon$$

In the "classical" case, from the rule of the errors propagation, we have a weaker increase of the overall error with the number of operations :

$$\sigma^2 = \sum_{i=1}^{n} \sigma_i^2 \quad \rightarrow \quad \sigma < \sqrt{n}\ \epsilon$$

- Classical bits and classical computing
- Quantum mechanics and quantum bits (qubits)
- Manipulating qubit states, unitary errors
- Quantum gates and circuits
- Quantum teleportation
- Quantum (Discrete) Fourier Transform
- Quantum cryptography
- IBM Q Experience, programming languages for QC

Single-qubit gates σ_x , σ_y , σ_z

(Pauli operators)

$$\begin{aligned}
\sigma_x |0\rangle &= |1\rangle \\
\sigma_x |1\rangle &= |0\rangle \\
\sigma_y |0\rangle &= i\,|1\rangle \\
\sigma_y |1\rangle &= -i\,|0\rangle \\
\sigma_z |0\rangle &= |0\rangle \\
\sigma_z |1\rangle &= -\,|1\rangle
\end{aligned}$$

The Hadamard gate

$$H\,|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \equiv |+\rangle_x$$

$$H\,|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \equiv |-\rangle_x$$

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

$$|x\rangle \;\text{——}\boxed{H}\text{——}\; (-1)^x\,|x\rangle + |1-x\rangle \qquad , \qquad |x\rangle = \{|0\rangle\,, |1\rangle\}$$

Transforms the
computational basis : $|0\rangle\,, |1\rangle \to |+\rangle_x\,, |-\rangle_x$

The exponential power of the states superposition

$$|0\rangle \;-\!\!\boxed{H}\!\!-\; \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$|0\rangle \;-\!\!\boxed{H}\!\!-\; \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$|0\rangle \;-\!\!\boxed{H}\!\!-\; \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

3 qubits in a network :
the application of the 3 Hadamard gates is synchronized and in the total product state we have a superposition of the values from 0 to 7.

$$= \frac{1}{2^{3/2}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle)$$

$$= \frac{1}{2^{3/2}}(|0\rangle + |1\rangle + |2\rangle + |3\rangle + |4\rangle + |5\rangle + |6\rangle + |7\rangle)$$

The generic state of a qubit in spherical coordinates

$$|\psi\rangle = \cos\frac{\theta}{2}\,|0\rangle + e^{i\phi}\sin\frac{\theta}{2}\,|1\rangle = \begin{bmatrix}\cos\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2}\end{bmatrix}$$

We can write this because :

- the two coefficients α and β are complex
- we have the total probability normalization condition
- a state vector is defined only up to a global phase of no physical significance (we can take one of the coefficients to be real)

$$p_{+1,z} = |\,\langle 0,\psi\rangle\,|^2 = \cos^2\frac{\theta}{2} \quad , \quad p_{-1,z} = |\,\langle 1,\psi\rangle\,|^2 = \sin^2\frac{\theta}{2}$$

The phase-shift gate

$$R_z(\delta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\delta} \end{bmatrix}$$

$$R_z(\delta)\,|\psi\rangle = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\delta} \end{bmatrix} \begin{bmatrix} \cos\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta}{2} \\ e^{i(\phi+\delta)}\sin\frac{\theta}{2} \end{bmatrix}$$

$$|x\rangle \;\text{———}\boxed{R_z(\delta)}\text{———}\; e^{ix\delta}\,|x\rangle \qquad , \qquad |x\rangle = \{|0\rangle\,, |1\rangle\}$$

Universality of Hadamard and phase-shift gates

Any unitary operation on a single qubit can be constructed using only Hadamard and phase-shift gates. In particular, the generic state can be reached starting from $|0\rangle$ in the following way:

$$e^{i\frac{\theta}{2}}\,|\psi\rangle = e^{i\frac{\theta}{2}}(\cos\tfrac{\theta}{2}\,|0\rangle + e^{i\phi}\sin\tfrac{\theta}{2}\,|1\rangle) = R_z(\tfrac{\pi}{2}+\phi)\;H\;R_z(2\theta)\;H\;\;|0\rangle$$

Two-qubit states and gates

The total vector space of two qubits is the result of a tensor product, the computational base of the resulting space is given by the 4 possible combinations of the computational basis vectors of each of the two qubits:

$$|\psi\rangle = \alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle$$

$$|ij\rangle \equiv |i\rangle\,|j\rangle \equiv |i\rangle \otimes |j\rangle \qquad i = \{0,1\}\ ,\ j = \{0,1\}$$

with the probability normalization constraint:

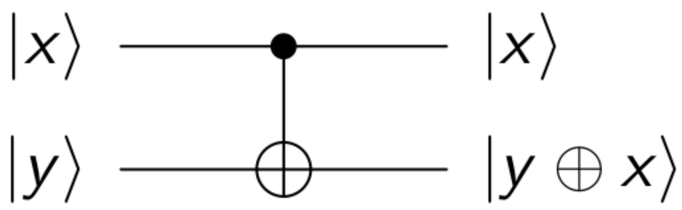
$$\alpha^2+\beta^2+\gamma^2+\delta^2=1$$

The quantum CNOT gate

It acts on the computational basis of the system of two qubits like this:

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

The circuit diagram:

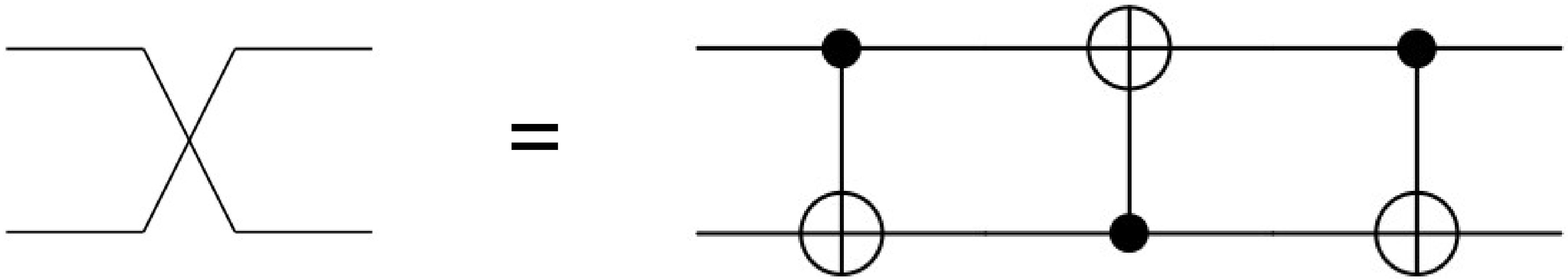


The 4×4 unitary matrix:

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The state of target qubit (y) flips only if the control qubit (x) is in the $|1\rangle$ state.

Obtaining a SWAP gate from CNOT gates



The CNOT gate generates <u>entanglement</u> of two qubits

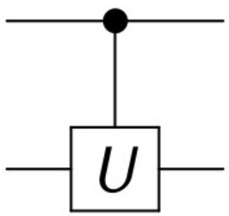
$$\boxed{CNOT(\alpha|0\rangle + \beta|1\rangle) \otimes |0\rangle = \alpha|0\rangle \otimes |0\rangle + \beta|1\rangle \otimes |1\rangle}$$

(the final state is non-separable, it can not be expressed as a single product of two single-qubit states)

Universal quantum gates

Any unitary operation in the Hilbert space of n qubits, $U^{(n)}$ can be decomposed into one-qubit gates and (two-qubit) CNOT gates.

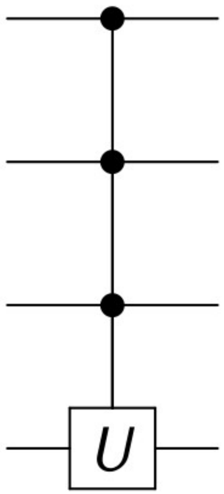
- we need few more special gates, like the controlled-U gate, where the U operator is applied to the target qubit only if the control qubit is in the $|1\rangle$ state



- the controlled-U gate can be generalized to the C^k-U gate, with k control qubits

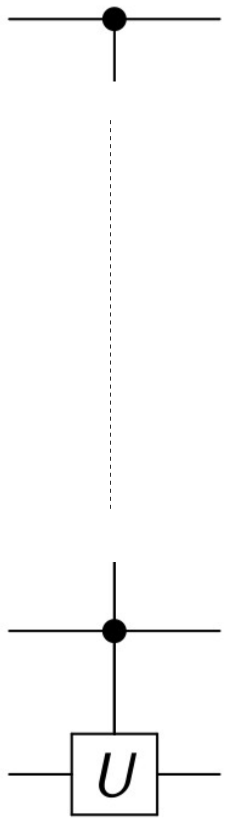
- the three-qubit C^2-NOT gate is the Toffoli gate

C^3-U



. . . . .

C^k-U

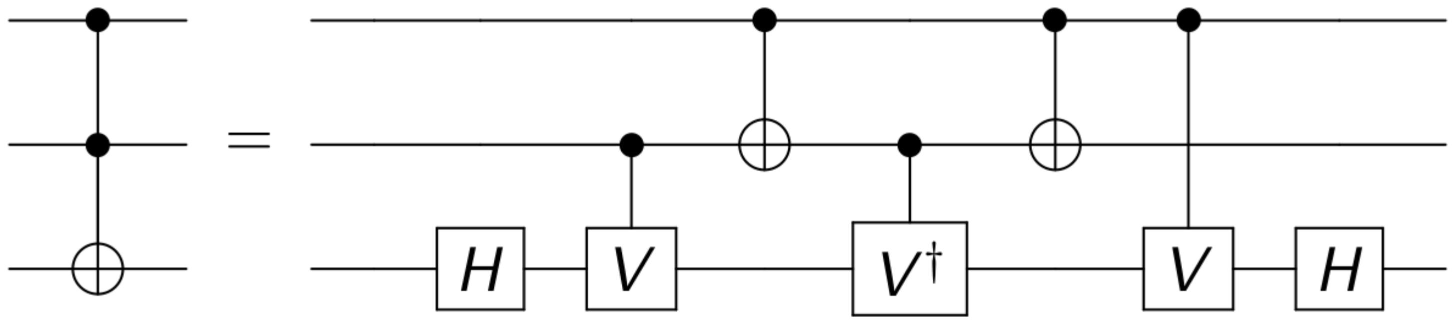


Universal quantum gates (cont.)

The Toffoli gate can be implemented using the Hadamard gate and a special unitary operator V :

$$V = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

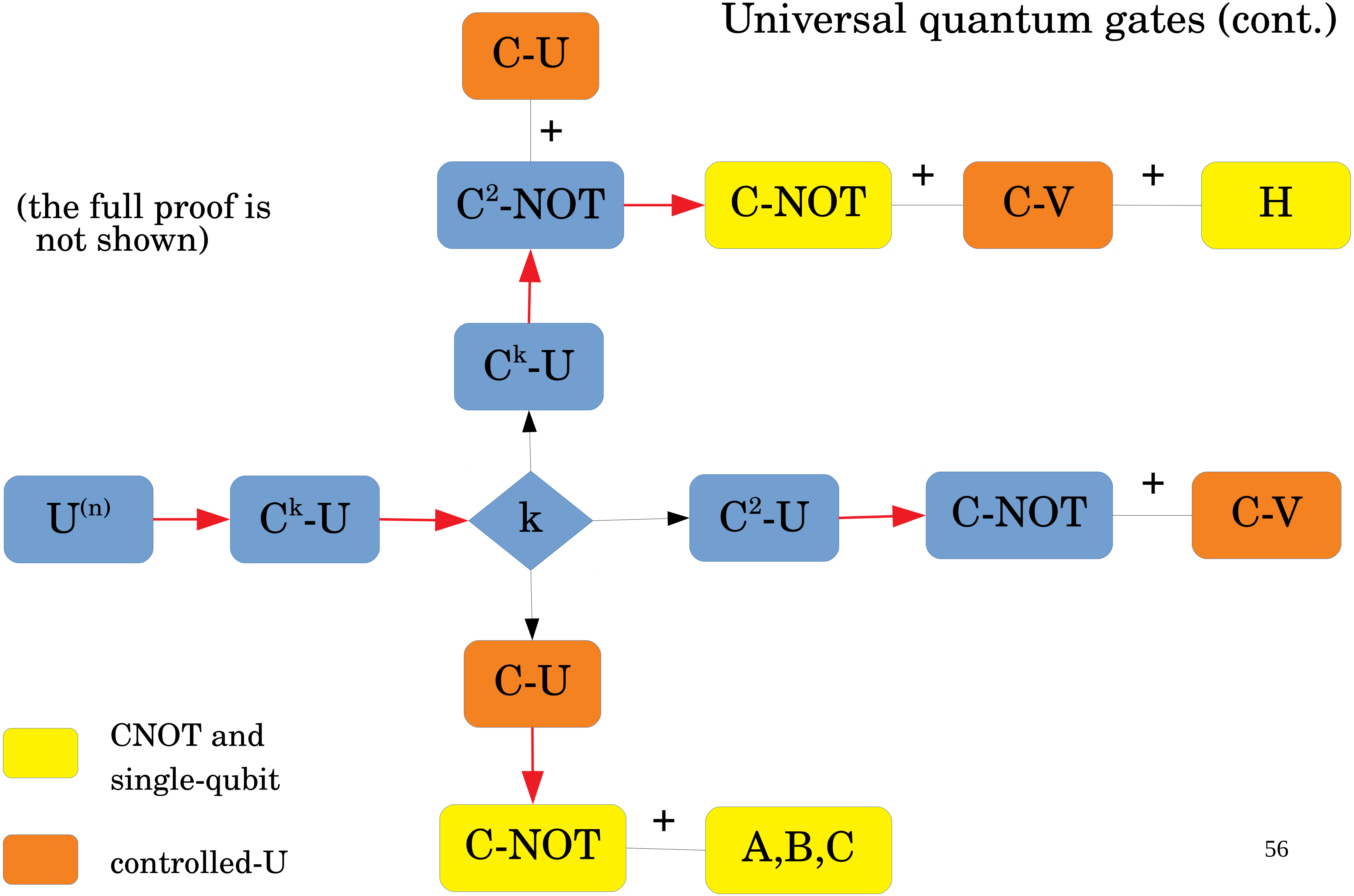
Toffoli



V is a single-qubit operator, so we know how to decompose it in Hadamard and phase-shift gates.

Universal quantum gates (cont.)

(the full proof is not shown)



CNOT and single-qubit

controlled-U

- Classical bits and classical computing
- Quantum mechanics and quantum bits (qubits)
- Manipulating qubit states, unitary errors
- Quantum gates and circuits
- Quantum teleportation
- Quantum (Discrete) Fourier Transform
- Quantum cryptography
- IBM Q Experience, programming languages for QC

Quantum information : teleportation

Suppose Alice owns a qubit in some unknown generic state :

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

and wishes to send to Bob this qubit state (not the physical realization of the qubit), using only a classical communication channel (send only classical bits).

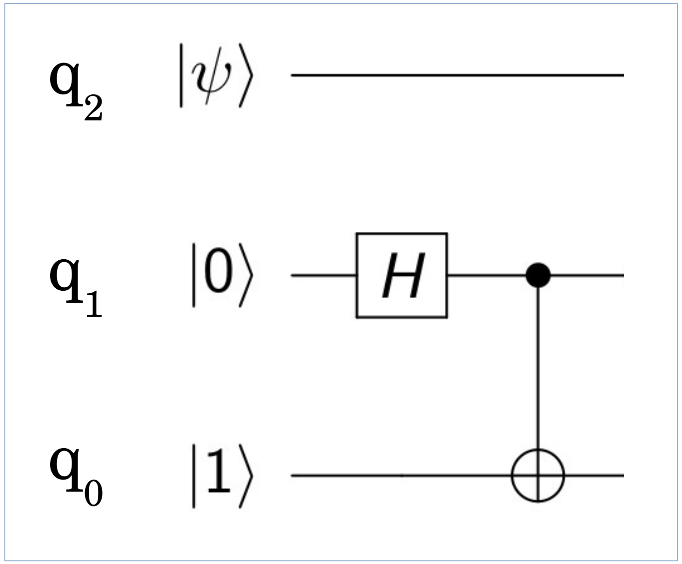
- Alice can not simply measure the state of her qubit, because this will immediately destroy that state, with the price of obtaining only one bit of information, while describing the generic state requires an infinite amount of classical information

- we also know that Alice can not clone that state ; if she could do that, she could do as many clones and measurements needed to describe the full state (even if, in practice, this would not be really possible)

Quantum information : teleportation (cont.)

Quantum teleportation is possible, providing that Alice and Bob share at the beginning **a pair of entangled qubits**.

For instance, starting from the computational basis, we can create an entangled state of two qubits in this way :



the state to be teleported

the Bell pair $\quad CNOT(H \otimes I)\,|01\rangle = |\psi^{+}\rangle$

$$|\psi^{+}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

Quantum information : teleportation (cont.)

The three-qubit state obtained by putting in the same register the two qubits and the qubit to be teleported is given by the tensor product :

$$|\psi\rangle \otimes |\psi^+\rangle = \frac{\alpha}{\sqrt{2}}(|001\rangle + |010\rangle) + \frac{\beta}{\sqrt{2}}(|101\rangle + |110\rangle)$$

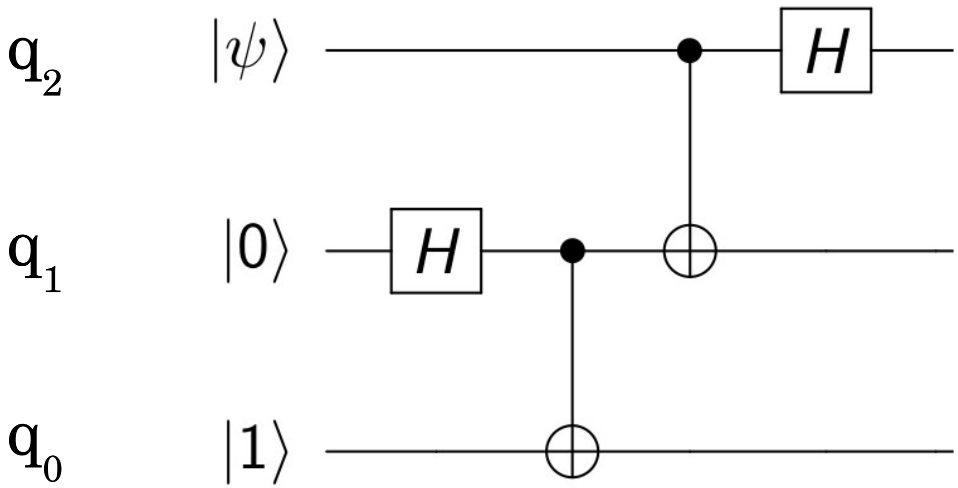
Alice will let her qubit interact with her half of the Bell pair, which means that she will perform a measurement not in the computational basis but in the Bell basis (see appendix) :

$$|\psi\rangle \otimes |\psi^+\rangle = \quad \tfrac{1}{2}|\psi^+\rangle(\alpha|0\rangle + \beta|1\rangle) + \tfrac{1}{2}|\psi^-\rangle(\alpha|0\rangle - \beta|1\rangle)$$

$$+\tfrac{1}{2}|\phi^+\rangle(\alpha|1\rangle + \beta|0\rangle) + \tfrac{1}{2}|\phi^-\rangle(\alpha|1\rangle - \beta|0\rangle)$$

Quantum information : teleportation (cont.)

and after the application of the two last gates $(H \otimes I)CNOT$ we obtain :

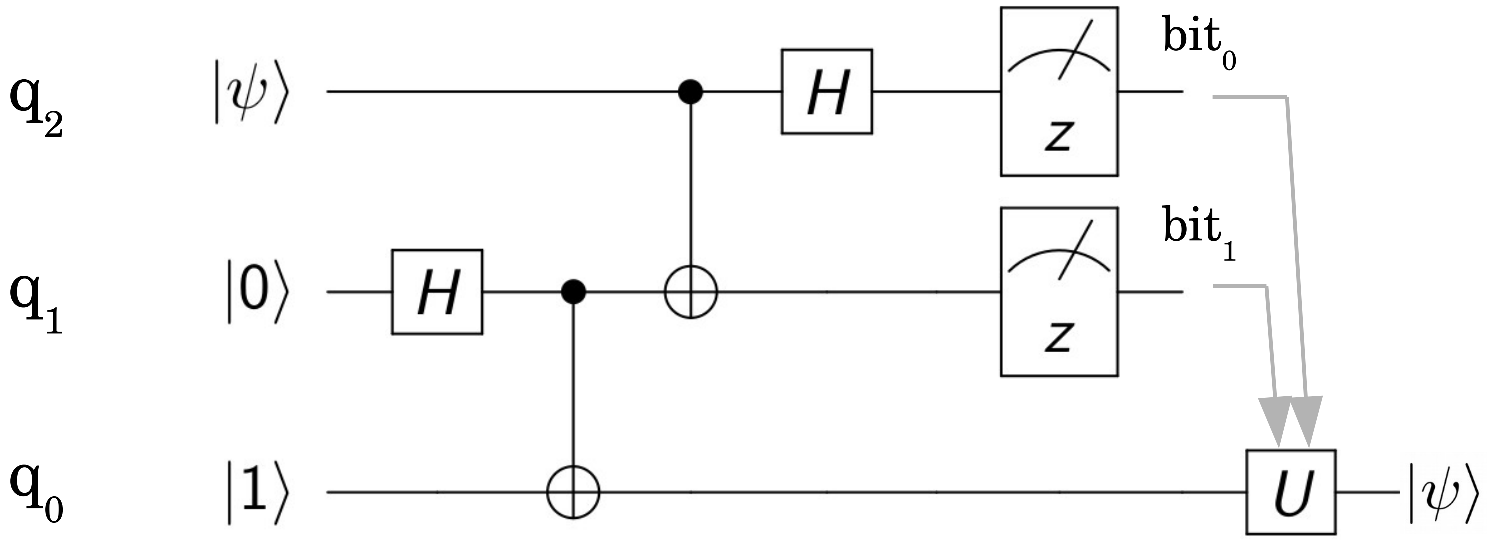


$$|\psi\rangle \otimes |\psi^+\rangle = \frac{1}{2}|01\rangle(\alpha|0\rangle + \beta|1\rangle) + \frac{1}{2}|11\rangle(\alpha|0\rangle - \beta|1\rangle)$$

$$+\frac{1}{2}|00\rangle(\alpha|1\rangle + \beta|0\rangle) + \frac{1}{2}|10\rangle(\alpha|1\rangle - \beta|0\rangle)$$

Quantum information : teleportation (cont.)

Finally, Alice makes a measurement on his two qubits in the computational basis "z" and sends the result to Bob, in the form of two classical bits over a classical transmission channel :



Quantum information : teleportation (cont.)

Now it is Bob's turn to act : he chooses a unitary operator U and applies it to his qubit, doing this according to the pair of bits sent by Alice and having a look in a table like this one :

Alice measures	Bob gets the bits	and applies to his qubit
$\|01\rangle$	0,1	I
$\|11\rangle$	1,1	σ_z
$\|00\rangle$	0,0	σ_x
$\|10\rangle$	1,0	$i\sigma_y$

As a consequence of this last operation, he will obtain exactly the initial generic state which Alice wanted to transmit (he does not need to check, he must have full confidence in the theory...).

- Classical bits and classical computing
- Quantum mechanics and quantum bits (qubits)
- Manipulating qubit states, unitary errors
- Quantum gates and circuits
- Quantum teleportation
- Quantum (Discrete) Fourier Transform
- Quantum cryptography
- IBM Q Experience, programming languages for QC

The Fourier Transform (FT), continuous and discrete

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

direct : time domain to frequence domain

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega$$

inverse : frequence domain to time domain

$$X_k = \sum_{n=0}^{N-1} x_n \mathrm{e}^{-i2\pi kn/N} \quad k = 0, \dots, N-1$$

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \mathrm{e}^{i2\pi kn/N} \quad n = 0, \dots, N-1$$

(discrete, DFT)

The discrete quantum Fourier transform (QFT)

How to do the Fourier transform of (a vector of) N complex values :

$$f(0), f(1), \dots, f(N-1) \quad \rightarrow \quad \tilde{f}(0), \tilde{f}(1), \dots, \tilde{f}(N-1)$$

Build a generic state with n = $\log_2$N qubits, in the computational basis :

$$|\psi\rangle = \sum_{j=0}^{2^n-1} f(j)\,|j\rangle$$

a vector of the computational basis is the tensor product :

$$|j\rangle = |j_0\rangle \otimes |j_1\rangle \otimes \dots \otimes |j_{n-1}\rangle \quad , \quad j_m = \{0,1\} \quad , \quad m = 0, \dots, n-1$$

The discrete quantum Fourier transform (cont.)

Define a unitary operator F, fully described by its action on the "n" vectors of the computational basis :

$$F(|j\rangle) = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i \frac{jk}{2^n}} |k\rangle$$

With this definition, an arbitrary state is transformed into :

$$|\widetilde{\psi}\rangle = F(|\psi\rangle) = \sum_{k=0}^{2^n-1} \tilde{f}(k)\,|k\rangle$$

where the coefficients are exactly the discrete transform we were looking for :

$$\tilde{f}(k) = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i \frac{jk}{N}} f(j) \qquad , \qquad N = 2^n$$

The discrete quantum Fourier transform (cont.)

We introduce the following notations for the binary representations of the indices of the n-qubit vectors from the computational basis :

$$j = j_{n-1}j_{n-2}\ldots j_0 = j_{n-1}2^{n-1} + j_{n-2}2^{n-2} + \ldots + j_0 2^0$$

$$0.j_l j_{l+1}\ldots j_m = j_l 2^{-1} + j_{l+1}2^{-2} + \ldots + j_m 2^{-(m-l+1)}$$

and we notice that we can re-write the terms of the sum by taking into account that :

$$\sum_{k=0}^{2^n-1} |k\rangle = \sum_{k_{n-1}=0}^{1} \cdots \sum_{k_0=0}^{1} |k_{n-1}\ldots k_0\rangle \qquad \text{and} \qquad \frac{k}{2^n} = \sum_{l=1}^{n} \frac{k_{n-l}}{2^l}$$

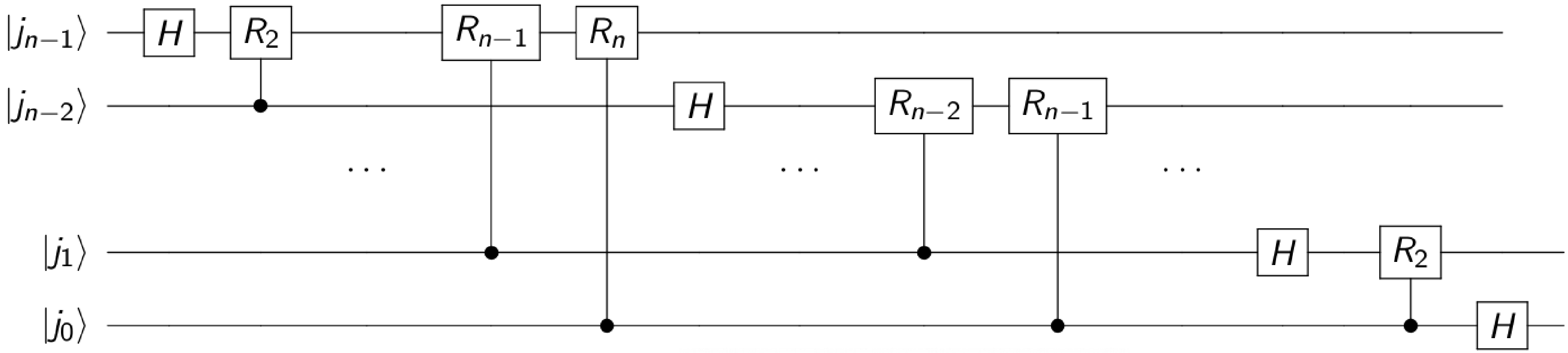
The discrete quantum Fourier transform (cont.)

Finally we obtain the expression for the result of the action of the F operator on a vector of the n-qubit computational basis in this form :

$$F(|j\rangle) = \frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i \; 0.j_0}\,|1\rangle)(|0\rangle + e^{2\pi i \; 0.j_1 j_0}\,|1\rangle)\cdots$$
$$\cdots(|0\rangle + e^{2\pi i \; 0.j_{n-1}j_{n-2}\cdots j_0}\,|1\rangle)$$

We notice that this state is not entangled, since it can be factorized in “n” single-qubit states.
Starting from this expression, it is possible to create the circuit which performs the transformation describing the operator F.

The discrete quantum Fourier transform (cont.)



with the unitary operator : $R_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp(\frac{2\pi i}{2^k}) \end{bmatrix}$

It is using n Hadamard gates and n(n-1)/2 single qubit gates, so the computation requires $O(n^2)$ elementary quantum gates.

The classical Fast Fourier Transform on a vector of $N = 2^n$ complex values, needs O(N logN) elementary operations.

The “brute-force” Discrete Fourier Transform needs $O(N^2)$ operations.

- Classical bits and classical computing
- Quantum mechanics and quantum bits (qubits)
- Manipulating qubit states, unitary errors
- Quantum gates and circuits
- Quantum teleportation
- Quantum (Discrete) Fourier Transform
- Quantum cryptography
- IBM Q Experience, programming languages for QC

The unbreakable cypher

Gilbert Vernam (AT&T Bell Labs engineer, 1917):

- the text is written as a binary sequence of 0's and 1's

0 0 1 0 1 0 0 1 1

- the secret key is a completely random binary sequence <u>of the same length</u> as the text

1 0 0 1 1 1 0 1 0

1 0 1 1 0 1 0 0 1

- the cypher text is obtained by adding the secret key bitwise modulo 2 to the plain text

$$c_i = p_i \oplus k_i \quad (i = 1, 2, \ldots, N)$$

Note: a key must not be reused for another message!

and to go back to the text:

$$p_i = q_i \oplus k_i \quad (i = 1, 2, \ldots, N)$$

The BB84 quantum protocol (Bennett and Brassard, 1984)

BB84 is using four quantum states of a single qubit and is coding the classical bits into states of a qubit, by using two alphabets :

$$|0\rangle \quad , \quad |1\rangle \quad , \quad |+\rangle \equiv |0\rangle_x = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad , \quad |-\rangle \equiv |1\rangle_x = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

which are the eigen-states of the Pauli matrices σ_z and σ_x respectively (the z-alphabet and the x-alphabet), a pair on non-commuting observables.

The coding rules :

$$0 = \begin{cases} |0\rangle & \text{, z-alphabet} \\ |+\rangle & \text{, x-alphabet} \end{cases} \qquad 1 = \begin{cases} |1\rangle & \text{, z-alphabet} \\ |-\rangle & \text{, x-alphabet} \end{cases}$$

The first part of the BB84 protocol

Alice's data bits	1	0	0	0	1	1	0	1	0	1

1. Alice generates a random sequence of 0's and 1's

The first part of the BB84 protocol

Alice's data bits	1	0	0	0	1	1	0	1	0	1
Alice's alphabet	x	z	x	z	x	x	x	z	z	x

2. Alice encodes each data bit in a qubit, by choosing randomly between the z- and the x-alphabet

The first part of the BB84 protocol

Alice's data bits	1	0	0	0	1	1	0	1	0	1
Alice's alphabet	x	z	x	z	x	x	x	z	z	x
Transmitted qubits	$\|-\rangle$	$\|0\rangle$	$\|+\rangle$	$\|0\rangle$	$\|-\rangle$	$\|-\rangle$	$\|+\rangle$	$\|1\rangle$	$\|0\rangle$	$\|-\rangle$

3. The resulting string of qubits is sent by Alice and received by Bob (by teleportation)

The first part of the BB84 protocol

Alice's data bits	1	0	0	0	1	1	0	1	0	1
Alice's alphabet	x	z	x	z	x	x	x	z	z	x
Transmitted qubits	$\|-\rangle$	$\|0\rangle$	$\|+\rangle$	$\|0\rangle$	$\|-\rangle$	$\|-\rangle$	$\|+\rangle$	$\|1\rangle$	$\|0\rangle$	$\|-\rangle$
Bob's alphabet	x	z	x	x	z	x	z	x	z	z

4. For each qubit, Bob decides at random which alphabet (axis) to use for the measurement, z or x.

The first part of the BB84 protocol

Alice's data bits	1	0	0	0	1	1	0	1	0	1
Alice's alphabet	x	z	x	z	x	x	x	z	z	x
Transmitted qubits	$\|-\rangle$	$\|0\rangle$	$\|+\rangle$	$\|0\rangle$	$\|-\rangle$	$\|-\rangle$	$\|+\rangle$	$\|1\rangle$	$\|0\rangle$	$\|-\rangle$
Bob's alphabet	x	z	x	x	z	x	z	x	z	z
Bob's measurement	1	0	0	0	0	1	0	0	0	1

If Bob chooses the same alphabet as Alice, he gets the same bit value (if there are no eavesdroppers or noise) ; this happens on average for half of his choices. When Bob chooses a different axis, the resulting bit will agree with the one of Alice only half of the time, on average.

The first part of the BB84 protocol

Alice's data bits	1	0	0	0	1	1	0	1	0	1
Alice's alphabet	x	z	x	z	x	x	x	z	z	x
Transmitted qubits	$\|-\rangle$	$\|0\rangle$	$\|+\rangle$	$\|0\rangle$	$\|-\rangle$	$\|-\rangle$	$\|+\rangle$	$\|1\rangle$	$\|0\rangle$	$\|-\rangle$
Bob's alphabet	x	z	x	x	z	x	z	x	z	z
Bob's measurement	1	0	0	0	0	1	0	0	0	1
Bob's data bits	1	0	0	0	0	1	0	0	0	1

These are Bob's results following his choice of alphabets.

The first part of the BB84 protocol

5. Bob communicates to Alice over a classical public channel his choices of the alphabet (but not the results of his measurements!)

6. Alice communicates to Bob over a classical public channel which alphabet she used for the transmitted qubits.

7. Alice and Bob delete all bits corresponding to the cases in which they used different alphabets. The remaining bits form the “raw key” (or rather a part of it).

The key is smaller in size than it was initially intended, so, probably they have to repeat the procedure several times, and there are other steps performed in order to minimize the effects of **eavesdropping** and especially the transmission **noise**.

The first part of the BB84 protocol

Alice's data bits	1	0	0	0	1	1	0	1	0	1
Alice's alphabet	x	z	x	z	x	x	x	z	z	x
Transmitted qubits	$\|-\rangle$	$\|0\rangle$	$\|+\rangle$	$\|0\rangle$	$\|-\rangle$	$\|-\rangle$	$\|+\rangle$	$\|1\rangle$	$\|0\rangle$	$\|-\rangle$
Bob's alphabet	x	z	x	x	z	x	z	x	z	z
Bob's measurement	1	0	0	0	0	1	0	0	0	1
Bob's data bits	1	0	0	0	0	1	0	0	0	1
Raw key	1	0	0			1			0	

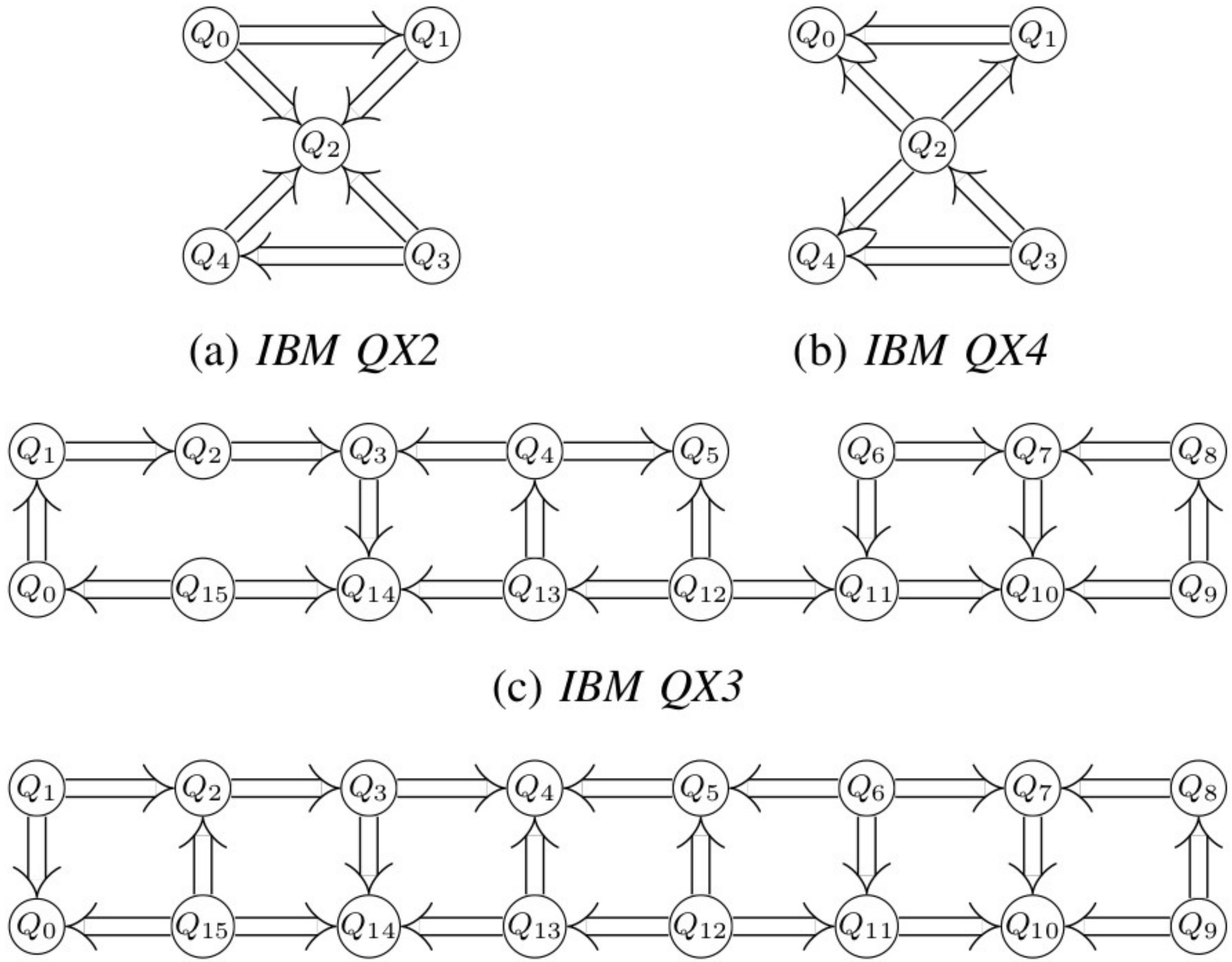
The raw key is now: 10010 (because in this process 5 bits out of 10 were lost)

- Classical bits and classical computing
- Quantum mechanics and quantum bits (qubits)
- Manipulating qubit states, unitary errors
- Quantum gates and circuits
- Quantum teleportation
- Quantum (Discrete) Fourier Transformation
- Quantum cryptography
- IBM Q Experience, programming languages for QC

The IBM Q project (launched in March 2017, [1], [2])

Processor	Qubits
IBM QX2	5
IBM QX3	16
IBM QX4	5
IBM QX5	16

The coupling-maps



(a) *IBM QX2* (b) *IBM QX4*

(c) *IBM QX3*

(d) *IBM QX5*

[1] "An Efficient Methodology for Mapping Quantum Circuits to the IBM QX Architectures"
Alwin Zulehner, Alexandru Paler, Robert Wille https://arxiv.org/abs/1712.04722
[2] http://iic.jku.at/eda/research/ibm_qx_mapping/

Approximating continuous gates with discrete gates

One of the gates necessary for describing any unitary operation on a set of qubits is the phase-shift gate, which is a **continuous** gate.
Its practical implementation will raise technical problems, for the reasons discussed before.
However, it is possible to approximate such a transformation with an arbitrary accuracy ε using a discrete set of quantum gates. It is possible to show that using Hadamard and T gates (T is a $\pi/4$ phase-shift around the z-axis) we can approximate any single-qubit rotation in

$$O\left(\log^{c}\left(1/\epsilon\right)\right) \quad , \quad c \sim 2$$

steps, where ε is the desired accuracy (Nielsen and Chuang, 2000).

Such T gates are implemented in the IBM QX processors and together with the Hadamard and the CNOT gates form a universal set.

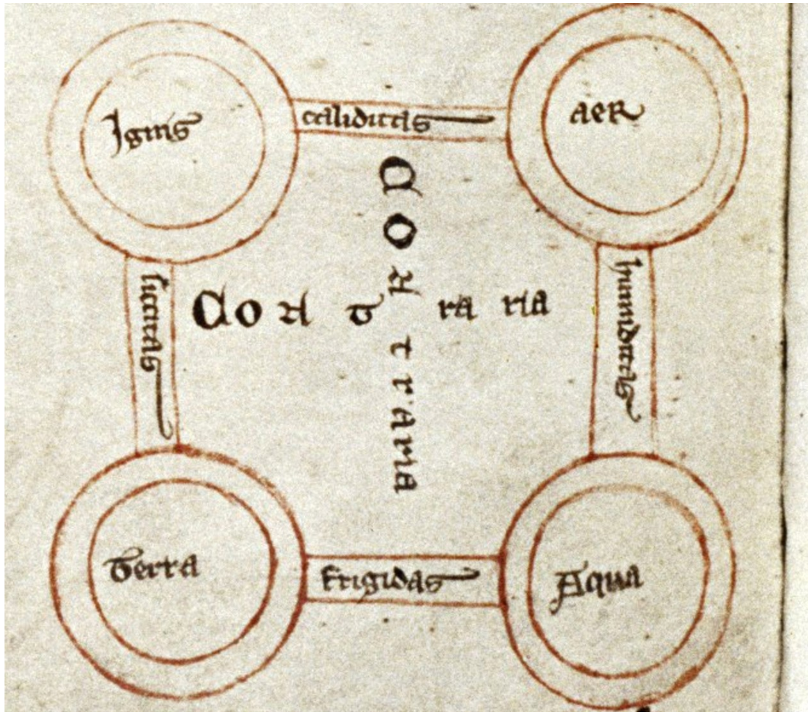
The programming language

For the programming of its QX devices, IBM provides a Software Development Kit (SDK) written in Python, named Qiskit (https://qiskit.org):

```
$ pip install qiskit
```

Qiskit has several components ("elements"):

- **Terra** = is the foundation on which the Qiskit framework lies
- **Aer** = provides a simulation framework for quantum circuits (contains a C++ simulator backend)
- **Ignis** = characterization of errors, improving gates, and computing in the presence of noise
- **Aqua** = applications and algorithms

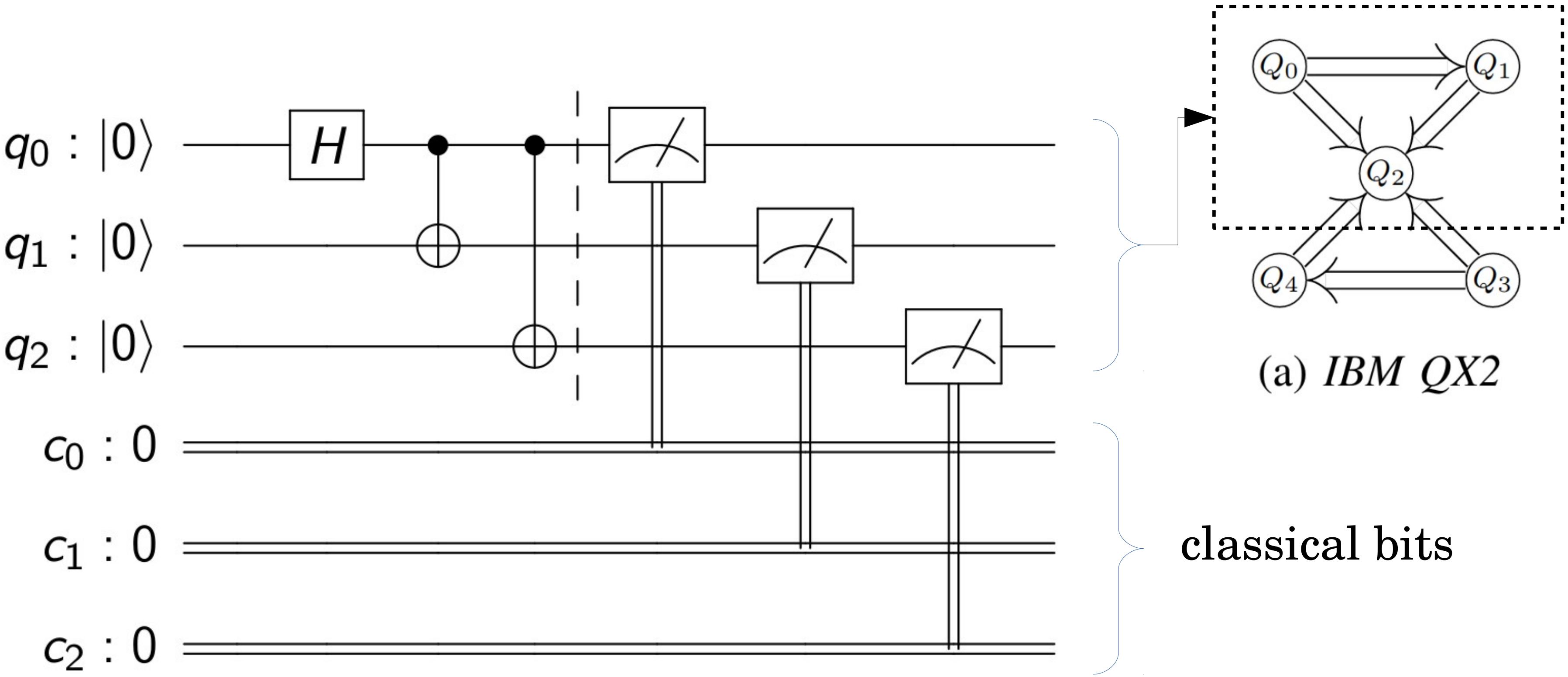


Bodleian Library, MS. Digby 107
William de Conchis, Dragmaticon
France, 13th century, end

Running an example on IBM QX2

Create the 3-qubit entangled state GHZ (Greenberger-Horne-Zeilinger):

$$|GHZ\rangle = \frac{|0\rangle^{\otimes 3} + |1\rangle^{\otimes 3}}{\sqrt{2}} = \frac{|000\rangle + |111\rangle}{\sqrt{2}}$$

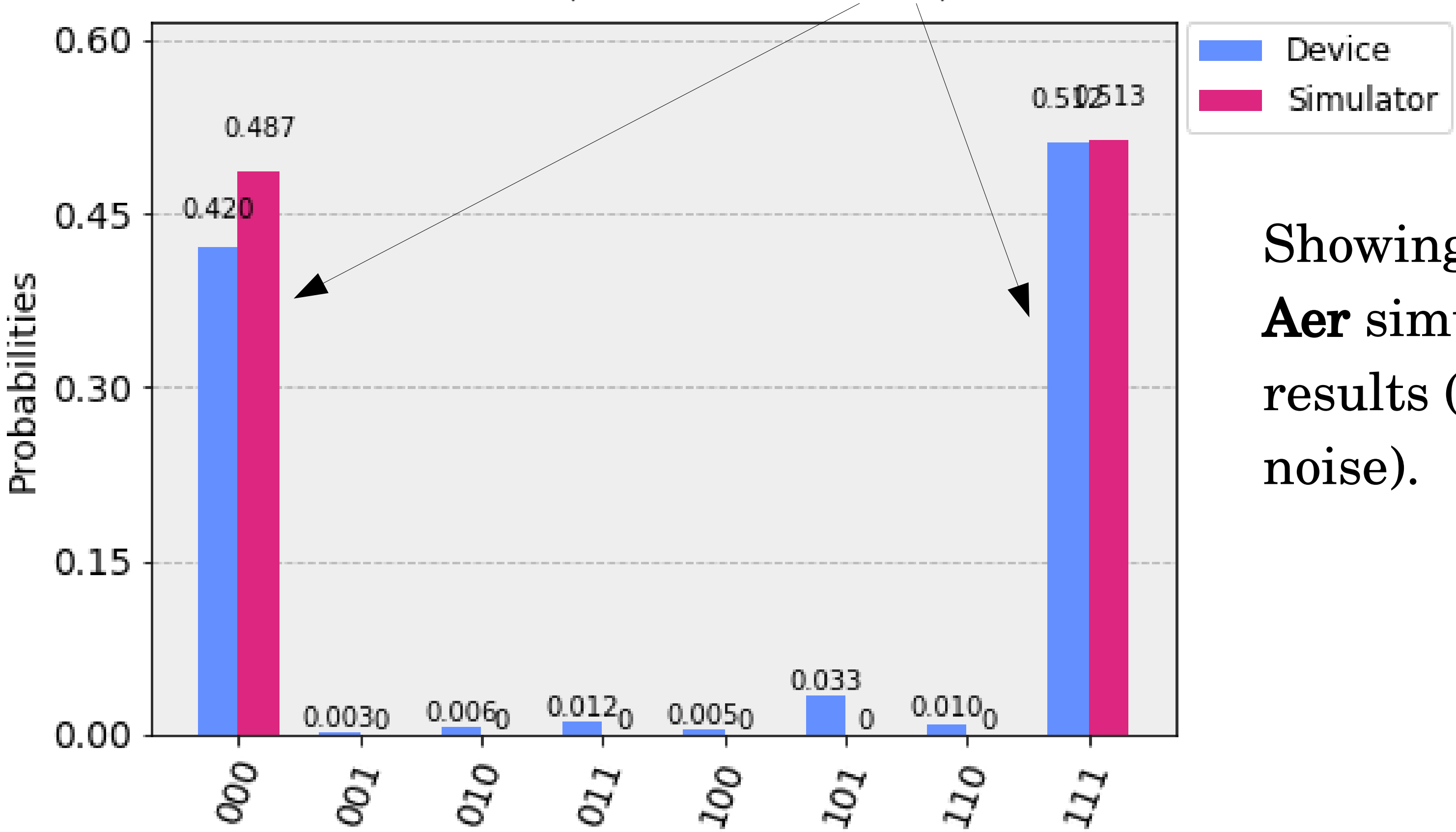


(a) *IBM QX2*

Running an example on IBM QX2 (cont.)

The final state should contain only (000) and (111), in reality we see other states, (001, 010, 011, …) with smaller probability:

$$|GHZ\rangle = \frac{|0\rangle^{\otimes 3} + |1\rangle^{\otimes 3}}{\sqrt{2}} = \frac{|000\rangle + |111\rangle}{\sqrt{2}}$$



Showing also **Aer** simulation results (without noise).

The GHZ state

The Qiskit code

This was a basic example from:

https://github.com/Qiskit/qiskit-iqx-tutorials.git

Jupyter notebook: **1_getting_started_with_qiskit.ipynb**

```
from qiskit import *
```

```
circ = QuantumCircuit(3)
```

; create a circuit with 3 qubits
(qubits are initialized in $|0\rangle$ state)

```
circ.h(0)
```

; apply Hadamard gate to Q0
(put Q0 in superposition)

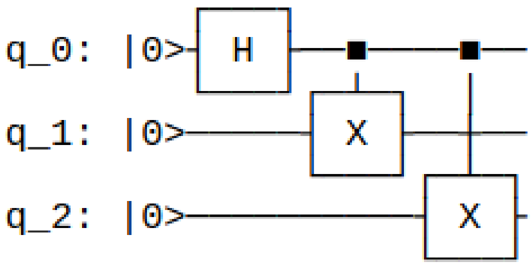
```
circ.cx(0, 1)
```

; apply CNOT Q0→Q1
(put Q0 and Q1 in a Bell state)

```
circ.cx(0, 2)
```

; apply CNOT Q0→Q2
(put Q0, Q1 and Q2 in a GHZ state)

```
circ.draw()
```



; draw the circuit (with `matplotlib`)

The Qiskit code (cont.)

Doing a simulation with Qiskit Aer:

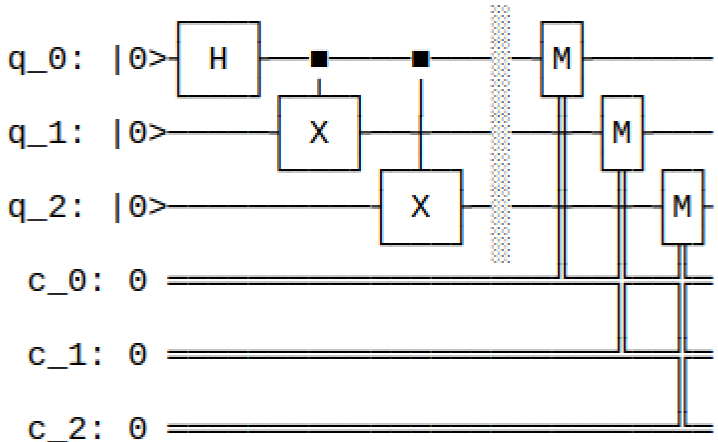
```
from qiskit import Aer
backend = Aer.get_backend('statevector_simulator')
job = execute(circ, backend)
result = job.result()
outputstate = result.get_statevector(circ, decimals=3)
print(outputstate)
  [0.707+0.j
   0.   +0.j
   0.   +0.j
   0.   +0.j
   0.   +0.j
   0.   +0.j
   0.   +0.j
   0.707+0.j]
```

complex coefficients of the 8 basis vectors (without errors)

The Qiskit code (cont.)

Doing a simulation with Qiskit Aer and **OpenQASM** as a back-end:

```
meas = QuantumCircuit(3, 3)                ; add 3 classical bits to the 3 qubits
meas.barrier(range(3))                     ; set a size 3 barrier over the 3 qubits
meas.measure(range(3),range(3))            ; map the 3 qubits to the 3 bits
qc = circ+meas                             ; add the previous circuit with the gates
qc.draw()
```



OpenQASM = an intermediate representation for quantum instructions, a kind of “hardware description language”
(IBM, "Open Quantum Assembly Language". ArXiv:1707.03429)
Example:

```
H q[0]                measure q[0] → c[0]
CX q[0],q[1]          etc.
CX q[0],q[2]
barrier q
```

The Qiskit code (cont.)

Doing a simulation with Qiskit Aer and OpenQASM as a back-end:

```
backend_sim = Aer.get_backend('qasm_simulator')
job_sim = execute(qc, backend_sim, shots=1024)
result_sim = job_sim.result()
counts = result_sim.get_counts(qc)
print(counts)
  {'000': 515, '111': 509}
```

(without errors)

The Qiskit code (cont.)

Running on a IBM QX device:

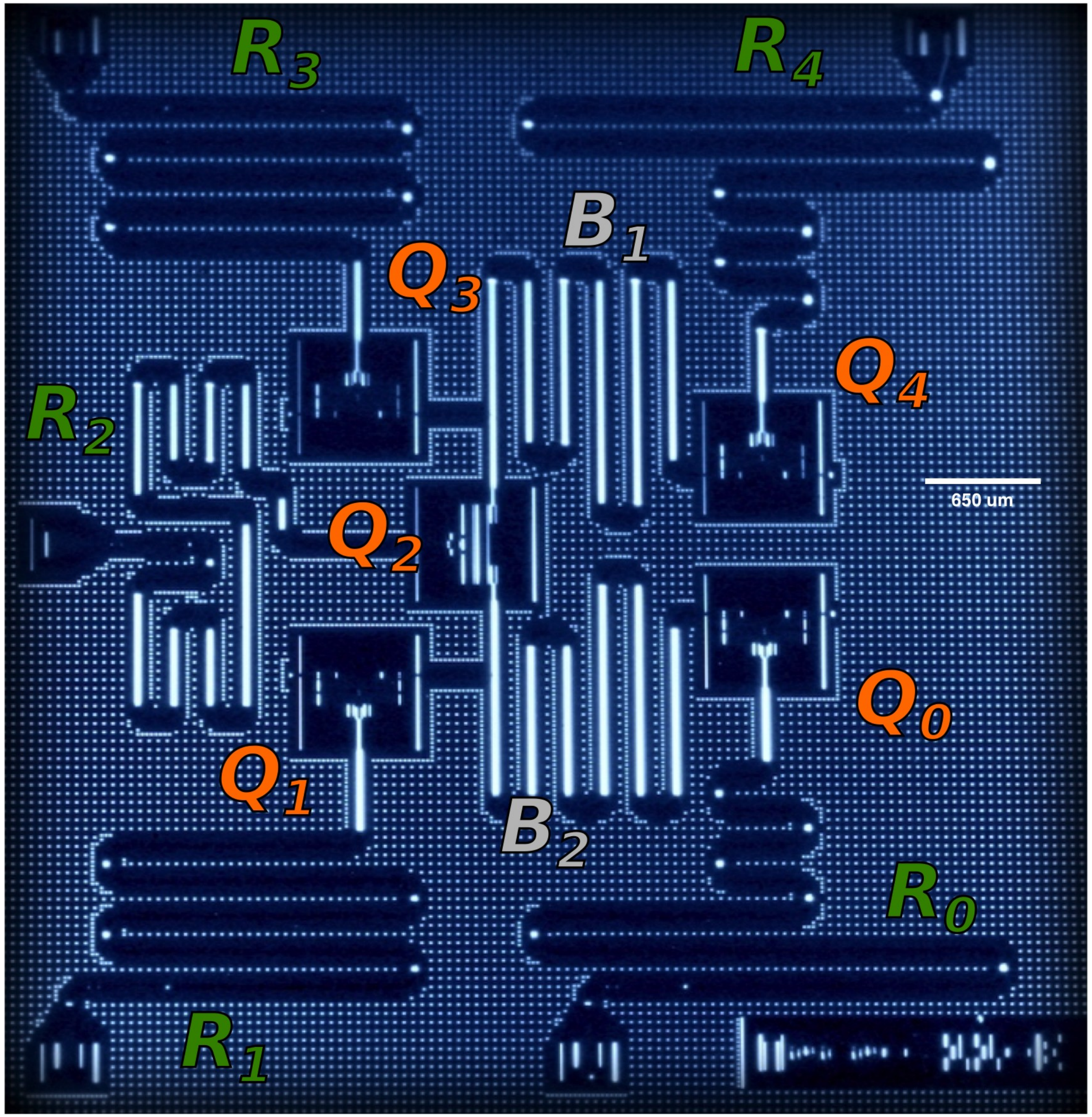
```
from qiskit import IBMQ
IBMQ.load_account()                                   ; user account
provider = IBMQ.get_provider(group='open')            ; device providers
backend = provider.get_backend('ibmqx2')              ; select device

from qiskit.tools.monitor import job_monitor
job_exp = execute(qc, backend=backend)
job_monitor(job_exp)                                  ; launch on cloud
result_exp = job_exp.result()                         ; wait for result
counts_exp = result_exp.get_counts(qc)                ; count statistics
{'001': 3, '100': 5, '111': 524, '101': 34,
'011': 12, '010': 6, '110': 10, '000': 430}
```

$$|\psi\rangle = |\mathsf{T}\rangle + |\mathsf{h}\rangle + |\mathsf{a}\rangle + |\mathsf{n}\rangle + |\mathsf{k}\rangle + |\mathsf{y}\rangle + |\mathsf{o}\rangle + |\mathsf{u}\rangle$$

Extra slides

IBM QX2 device information

https://github.com/Qiskit/ibmq-device-information/tree/master/backends/yorktown/V1



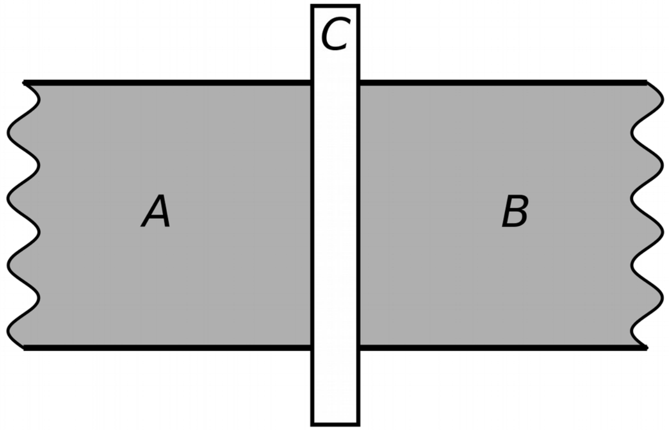
The Josephson junction

In a structure formed by a thin layer of non-superconducting material (or even an insulator) placed between two layers of superconducting material, pairs of superconducting electrons could “tunnel” through the non-superconducting barrier from one superconductor to the other (Brian Josephson, 1962, Nobel Prize 1973).

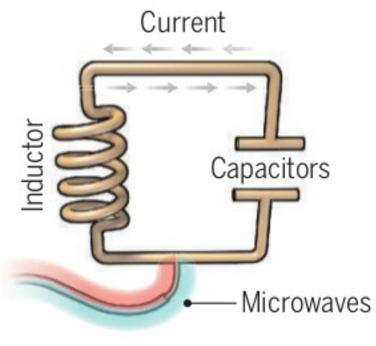
Superconductivity: below a critical temperature (depending on the material) the overall interaction between two electrons becomes slightly attractive.

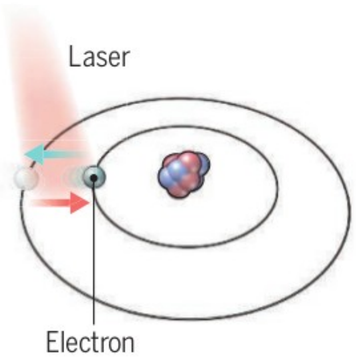
Josephson structure in electronic circuits: SQUID = Superconducting Quantum Interference Device.

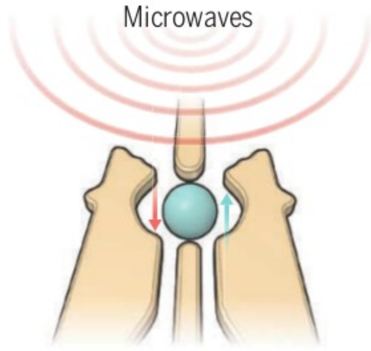
A, B = superconductors
C = insulator

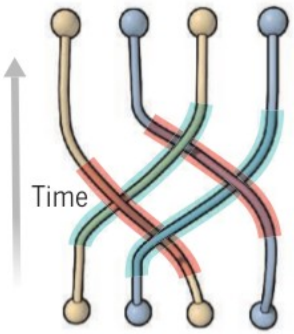


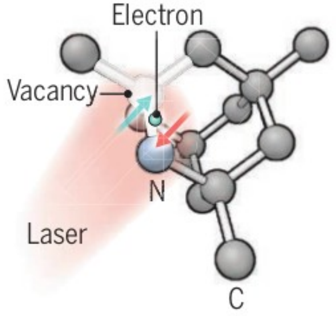
https://commons.wikimedia.org/w/index.php?curid=13302040











	Superconducting loops	Trapped ions	Silicon quantum dots	Topological qubits	Diamond vacancies
	A resistance-free current oscillates back and forth around a circuit loop. An injected microwave signal excites the current into superposition states.	Electrically charged atoms, or ions, have quantum energies that depend on the location of electrons. Tuned lasers cool and trap the ions, and put them in superposition states.	These "artificial atoms" are made by adding an electron to a small piece of pure silicon. Microwaves control the electron's quantum state.	Quasiparticles can be seen in the behavior of electrons channeled through semiconductor structures. Their braided paths can encode quantum information.	A nitrogen atom and a vacancy add an electron to a diamond lattice. Its quantum spin state, along with those of nearby carbon nuclei, can be controlled with light.
Longevity (seconds)	0.00005	>1000	0.03	N/A	10
Logic success rate	99.4%	99.9%	~99%	N/A	99.2%
Number entangled	9	14	2	N/A	6
Company support	Google, IBM, Quantum Circuits	ionQ	Intel	Microsoft, Bell Labs	Quantum Diamond Technologies
Pros	Fast working. Build on existing semiconductor industry.	Very stable. Highest achieved gate fidelities.	Stable. Build on existing semiconductor industry.	Greatly reduce errors.	Can operate at room temperature.
Cons	Collapse easily and must be kept cold.	Slow operation. Many lasers are needed.	Only a few entangled. Must be kept cold.	Existence not yet confirmed.	Difficult to entangle.

Note: Longevity is the record coherence time for a single qubit superposition state, logic success rate is the highest reported gate fidelity for logic operations on two qubits, and number entangled is the maximum number of qubits entangled and capable of performing two-qubit operations.

Gabriel Popkin, Quest for Qubits, in Science Magazine, December 2016

Universal gates for classical computation

AND, OR, NOT and FANOUT constitute a universal set of gates for classical computation.

Proof.
The m-bit function is equivalent to m one-bit (or Boolean) functions

$$f_i : \{0,1\}^n \to \{0,1\}, \qquad (i = 1, 2, ..., m)$$

where $f = (f_1, f_2, ..., f_m)$. For any values of the input argument $a = (a_{n-1}, a_{n-2}, ..., a_1, a_0)$, one way to compute the boolean function $f_i(a)$ is to consider the *minterms* $f_i^{(l)}(a)$, defined as

$$f_i^{(l)} = \begin{cases} 1, & \text{if } a = a^{(l)} \\ 0, & \text{otherwise} \end{cases}$$

Universal gates for classical computation (cont.)

for instance, if the particular value of $a^{(l)} = 110100\ldots001$, then $f_i^{(l)}$ can be defined as follows

$$f_i^{(l)} = a_{n-1} \wedge a_{n-2} \wedge \bar{a}_{n-3} \wedge a_{n-4} \wedge \bar{a}_{n-5} \wedge \bar{a}_{n-6} \wedge \ldots \wedge \bar{a}_2 \wedge \bar{a}_1 \wedge a_0$$

the one-bit function f_i can be calculated for all possible a values as follows

$$f_i(a) = f_i^{(1)} \vee f_i^{(2)} \vee \ldots \vee f_i^{(k)}$$

as the logical OR of all k minterms, with $0 \leq k \leq 2^n - 1$ (2^n is the number of all possible values of the input a). The FANOUT gate is required to feed the input a to the k minterms.

Universal gates for classical computation (example)

Consider the Boolean function $f(a)$, where $a = (a_2, a_1, a_0)$ defined as follows

	a	a_2	a_1	a_0	$f(a)$
	$a^{(1)} = 1$	0	0	1	1
	$a^{(2)} = 3$	0	1	1	1
	$a^{(3)} = 6$	1	1	0	1
t	$a^{(4)} = 0$	0	0	0	0
	$a^{(5)} = 2$	0	1	0	0
	$a^{(6)} = 4$	1	0	0	0
	$a^{(7)} = 5$	1	0	1	0
	$a^{(8)} = 7$	1	1	1	0

$$f^{(1)} = \bar{a}_2 \wedge \bar{a}_1 \wedge a_0$$

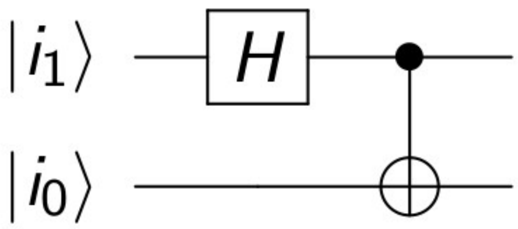
$$f^{(2)} = \bar{a}_2 \wedge a_1 \wedge a_0$$

$$f^{(3)} = a_2 \wedge a_1 \wedge \bar{a}_0$$

$$f(a) = f^{(1)}(a) \vee f^{(2)}(a) \vee f^{(3)}(a)$$

The Bell (EPR) basis

This circuit:



transforms the computational basis states into the Bell states:

$$\left\{\begin{aligned}
|00\rangle &\rightarrow |\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\
|10\rangle &\rightarrow |\phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\
|01\rangle &\rightarrow |\psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\
|11\rangle &\rightarrow |\psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)
\end{aligned}\right.$$

(EPR = Einstein-Podolski-Rosen, a paradox about the quantum nature of the reality)