ECOLE CENTRALE CASABLANCA

ANNÉE UNIVERSITAIRE 2020-2021

Introduction to quantum computing

Kenneth MAUSSANG

Université de Montpellier

Version du 10 février 2021

Contents

Contents

Chapter I

Quantum Mechanics postulates

1. Quantum states

1.1. Hilbert space

A quantum system is described by a quantum state, formally a vector of a Hilbert space $\mathscr{H}$. This vector contains all the information necessary to describe the quantum state of the quantum system consider, and might be used to obtain the quantum wavefunction (either in X or P representation.

Dirac notation : the Dirac notation is nowadays the most commonly used formalism used to describe quantum states of a system. For instance, let's consider a quantum system with d possible states. Let's consider a basis $\mathscr{B}$ of this Hilbert space associated, of dimension $d \times d$, noted $\mathscr{H}$. For any state of this quantum system, it is possible to find $a_i \in \mathbb{C}^d$ so that the corresponding vector state is decomposed over the basis $\mathscr{B}$ with a_i components. The Dirac notation consists in denoting a given vector, corresponding to a quantum state, by the so-called *ket* notation as follow

$$\textbf{ket} : |a\rangle = \begin{pmatrix} a_1 \\ \vdots \\ a_d \end{pmatrix} \in \mathscr{H},$$

A so-called *bra* is the complex conjugated and transposed of the corresponding vector (*i.e.* a ket), such as follow

$$\textbf{bra} : \langle b| = |b\rangle^\dagger = \begin{pmatrix} b_1 \\ \vdots \\ b_d \end{pmatrix}^\dagger = \begin{pmatrix} b_1^* & \dots & b_d^* \end{pmatrix}.$$

A so-called *braket* is a scalar product of two vectors

$$\textbf{braket} : \langle b|a\rangle = a_1 b_1^* + \cdots + a_d b_d^* = \langle a|b\rangle^* \in \mathbb{C}.$$

In the particular case of the scalar product of a vector with itself, one obtains the square of the norm of the vector

$$\langle a|a\rangle = \||a\rangle\|^2 \in \mathbb{R}^+ \text{ is the norm of the state } |a\rangle.$$

If a ket describes a physical state, the conservation of probability imposes

$$\boxed{\langle \psi|\psi\rangle = 1}.$$

As a consequence, a quantum state is formally described by a vector with the constraint on its norm to be unitary.

$$\boxed{\textbf{All quantum states are normalized}}.$$

Let $\{|\alpha_1\rangle, \cdots, |\alpha_d\rangle\}$ a basis of $\mathscr{H}$ and eigenvectors of an observable $\hat{A}$.

$$\forall |\psi\rangle \in \mathscr{H}, \exists (a_1, \cdots, a_d) \text{ such that}$$

$$|\psi\rangle = a_1|\alpha_1\rangle + \cdots + a_d|\alpha_d\rangle = \sum_{i=1}^{d} a_i|\alpha_i\rangle.$$

Since $|\alpha_i\rangle$ are eigenvectors of the observable $\hat{A}$, let's denote λ_i the corresponding eigenvalues, such that

$$\hat{A}|\alpha_i\rangle = \lambda_i|\alpha_i\rangle.$$

Each component a_i on a given basis vector $|\alpha_i\rangle$ has a physical meaning through its square modulus. Indeed, let's call it $P_i = |\langle\alpha_i|\psi\rangle|^2 = |a_i|^2$. P_i is the probability that a measurement of $\hat{A}$ of the quantum state $|\psi\rangle$ gives α_i as result.

Example of observables:

— position $\hat{\vec{R}} = (\hat{X},\hat{Y},\hat{Z})$;
— impulsion $\hat{\vec{P}} = (\hat{P}_X,\hat{P}_Y,\hat{P}_Z)$;
— hamiltonian $\hat{H}$ ($\Leftrightarrow$ energy);
— orbital momentum $\hat{\vec{L}} = (\hat{L}_X,\hat{L}_Y,\hat{L}_Z)$;
— spin momentum $\hat{\vec{S}} = (\hat{S}_X,\hat{S}_Y,\hat{S}_Z)$;

If braket is interpreted as a scalar product as a direct consequence of their definition, a *ket-bra* might also be considered, corresponding to a $d \times d$ square matric in the case of a finite size Hilbert space.

ket-bra:

$$|a\rangle\langle b| = \begin{pmatrix} a_1 b_1^* & a_1 b_2^* & \cdots & a_1 b_d^* \\ a_2 b_1^* & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ a_d b_1^* & \cdots & \cdots & a_d b_d^* \end{pmatrix}.$$

A $d \times d$ matrix is an operator on $\mathscr{H}$.

Examples:

— Projector on a state $|\psi\rangle$

$$\hat{P}_{|\psi\rangle} = |\psi\rangle\langle\psi|.$$

— Projector on a subspace defined by a basis $\{|\beta_1\rangle,\cdots,|\beta_p\rangle\}$

$$\hat{P} = \sum_{i=1}^{p} |\beta_i\rangle\langle\beta_i|.$$

1.2. Multiparticule quantum states

We use tensor product to describe multiple particle states. Indeed, for $|a\rangle \in \mathscr{H}_1$ (dimension $d \times d$) a state of a particle 1 and $|b\rangle \in \mathscr{H}_2$ (dimension $p \times p$ a state of a particle 2, the global quantum state $|\psi\rangle$ of the two particle is formally describe by a vector in the Hilbert space $\mathscr{H}$ given by the tensor product of $\mathscr{H}_1$ and $\mathscr{H}_2$

$$\mathscr{H} = \mathscr{H}_1 \otimes \mathscr{H}_2.$$

Then, the global vector $|\psi\rangle$ is given by

$$|\psi\rangle = |a\rangle \otimes |b\rangle = \begin{pmatrix} a_1 \\ \vdots \\ a_d \end{pmatrix} \otimes \begin{pmatrix} b_1 \\ \vdots \\ b_p \end{pmatrix} = \begin{pmatrix} a_1 b_1 \\ a_1 b_2 \\ \vdots \\ a_1 b_p \\ a_2 b_1 \\ \vdots \\ a_d b_p \end{pmatrix}.$$

Note that the dimension of $\mathscr{H}_1$ is not necessary the same than $\mathscr{H}_2$. It is possible to construct a basis $\mathscr{B}$ of this Hilbert space $\mathscr{H}$ so that any state might be decomposed of this basis. For example, if $|\alpha_i\rangle$ is a basis of $\mathscr{H}_1$ and $|\beta_j\rangle$ is a basis of $\mathscr{H}_2$, then $|\mu_{ij}\rangle = |\alpha_i\rangle \otimes |\beta_i\rangle$ is a basis of $\mathscr{H}$.

Example: Let consider a system A is in state $|\psi_A\rangle$ and a system B is in state $|\psi_B\rangle$. The total system $\{A \bigcup B\}$ is

$$|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle,$$

sometimes noted

$$|\psi\rangle = |\psi_A \psi_B\rangle = |\psi_A\rangle|\psi_B\rangle.$$

A state $|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$ is a state of the total system but a state of $\{A \bigcup B\}$ might not necessarily be written as a tensor product of two states of A and B. In the latter case, particules are so-called **entangled**.

Example:
— $|\psi_{A,1}\rangle$ and $|\psi_{A,2}\rangle$ two states of A;
— $|\psi_{B,1}\rangle$ and $|\psi_{B,2}\rangle$ two states of B;
— then,

$$|\psi\rangle = \frac{1}{\sqrt{2}} \left(|\psi_{A,1}\rangle|\psi_{B,1}\rangle + |\psi_{A,2}\rangle|\psi_{B,2}\rangle \right),$$

is entangled.
But the following state is **not entangled**

$$\begin{aligned} |\psi\rangle &= \frac{1}{2} \left(|\psi_{A,1}\rangle + |\psi_{A,2}\rangle \right) \otimes \left(|\psi_{B,1}\rangle + |\psi_{B,2}\rangle \right), \\ &= \frac{1}{2} \left(|\psi_{A,1}\rangle|\psi_{B,1}\rangle + |\psi_{A,1}\rangle|\psi_{B,2}\rangle + |\psi_{A,2}\rangle|\psi_{B,1}\rangle + |\psi_{A,2}\rangle|\psi_{B,2}\rangle \right). \end{aligned}$$

1.3. Measurements

A measurement is performed by a detector sensitive to an observable. We choose orthogonal bases states to describe and measure quantum states. Let's consider an observable $\hat{A}$ used for measurements, of eigenvectors $\{|\alpha_1\rangle, \cdots, |\alpha_d\rangle\}$. The probability to measure the eigenvalue λ_i of the eigenvector $|\alpha_1\rangle$ for a state $|\psi\rangle$ is

$$P_i = |\langle \alpha_i | \psi \rangle|^2.$$

If the measurement of $\hat{A}$ provides the value λ_i, the state $|\psi\rangle$ collapses after measurement on state $|\alpha_i\rangle$. It's a so-called **projective measurement.**

2. Classical bits VS quantum bits

2.1. Classical information

Classical information is encoded in bits. Instead of using a decimal system, computers are using binary system for calculation due to its simplicity.
— Multiplying by 2 is performed by adding a 0.

$$11 \times 2 = 22 \text{ in decimal system;}$$

$$11_{10} = 1011_2 \text{ and } 22_{10} = 10110_2;$$

$$\Rightarrow 1011 \times 10 = 10110 \text{ in binary system.}$$

— Dividing by 2 is performed by removing the last number corresponding to the rest of the division

$$11 \operatorname{div} 2 = 5 \text{ rest } 1 \text{ in decimal system;}$$

$$1011 \operatorname{div} 10 = 101 \text{ rest } 1 \text{ in binary system.}$$

In classical information, a bit is a unit of a binary number : 0110101... In the hardware, it might correspond to the state of a transistor, a voltage, or a flux of photons in an optic fiber. It might take only two values: either 1 or 0. Usually, information is encoded on 8 bits, so-called an octet. It's related to base 3 ($2^3 = 8$).

$$1 \text{ octet} = 8 \text{ bits} = 2^8 = 256 \text{ numbers encoded.}$$

Hexadecimal is often used to have a more compact description of binary numbers. It's a base $16 = 2^4$, noted

$$0, 1, \cdots, 9, A, B, C, D, E \text{ and } F.$$

Each batch of 4 bits is a binary representation of a number in base 16.

$$\begin{array}{ll} A_{16} = 10_{10} = 1010_2 & D_{16} = 13_{10} = 1101_2 \\ B_{16} = 11_{10} = 1011_2 & E_{16} = 14_{10} = 1110_2 \\ C_{16} = 12_{10} = 1100_2 & F_{16} = 15_{10} = 1111_2 \end{array}$$

Example:

$$10101101110_2 = \underbrace{101}_{5} \vdots \underbrace{0110}_{6} \vdots \underbrace{1110}_{E} = 56E_{16} = 1390_{10}.$$

A bit might take only two values : 0 or 1. An ensemble of bits permits to encode an integer number. Information is stored as a succession of bits: 011001010... Values 1 and 0 might be seen as logical values : TRUE or FALSE. A classical numerical calculation is performed by the mean of **logic gates**: NOT, OR, XOR, NAND,...

2.2. Digitization

Digitization corresponds to the process that convert a decimal number into a digital number (binary). Let $(n, N) \in \{0,1\}^N$ such that

$$n = \sum_{i=0}^{N} a_i \times 2^i,$$

where $\{a_i\}$ is then the digital number corresponding to n

$$n \leftrightarrow a_N a_{N-1} a_{N-2} \cdots a_2 a_1 a_0.$$

With N bits, one might encode 2^N integer numbers.

Example: coding on 4 bits.

$$\begin{aligned} 1 &= 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \quad \leftrightarrow \quad 0001, \\ 2 &= 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \quad \leftrightarrow \quad 0010, \\ 3 &= 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \quad \leftrightarrow \quad 0011, \\ 4 &= 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \quad \leftrightarrow \quad 0100. \end{aligned}$$

Only positive integers might be converted to binaries. For a real number (a "floating" number), one converts it as an integer and the power of 10 corresponding. For example

$$1.321 = \underbrace{1321}_{\text{integer}} \cdot 10^{-3} \nwarrow_{\text{signe + integer}} .$$

The larger is the precision on the number, the larger the number of coding bits required will be. The choice of the number of bits coding is a compromise between the precision required, the memory available, the fastness of acquisition or the fastness of calculation. Consequently, precise calculations are "slow" and requires memory.

Examples

Coding in 8 bits (1 octet)

$$n_{max} = 2^8 - 1 = 255 \quad \rightarrow \quad \text{coding } n \in [\![0, 255]\!].$$

Coding in 16 bits

$$n_{max} = 2^{16} - 1 = 65535 \text{ values} \quad \rightarrow \quad \text{coding } n \in [\![0, 65535]\!].$$

Color scales
Gray scales are usually encoded on 8 bits, which corresponds to $2^8 = 256$ values of gray (from white to black). It is usually enough for human's eye.

Color coding: **RGB coding**. Colors are encoded from their decomposition on three primary colors which are red (R), green (G) and blue (B). Human's eye is made of different sensitive cells called *cone cells* and *rod cells*. Cone cells themselves consist of three cell types, each "tuned" to a distinct wavelength response maximum centered at either 430, 535, or 590 nanometers. Consequently, any color seen by a human eye might be obtained by the superposition of red, green and blue lights (so-called *additive color synthesis*). And thus, a color image is actually encoded from the amount of red, green and blue on each pixel. Usually, each color is encoded over 8 bits. Then, with 8 bits for red R, 8 bits for green G, 8 bits for blue B, RGB coding permits to encode up over $8 \times 3 = 24$ bits for color coding, resulting in more than 16 millions of color available (usually enough for most of standard applications).

$$n_{max} = 2^{24} - 1 = 16777216 \approx 16 \text{ millions of color available.}$$

2.3. Classical logical gates

Logical circuits might be represented in a diagram where logical gates inputs and outputs are connected by wires. Symbolic representation of logical gates for such circuits diagrams are represented Fig. I.1.

2.4. Classical computing circuit

Logic gate and more generally electronical circuit are made from basic elements like capacitors or transistors. Such elements are governed by classical physics, without any quantum effects, and consequently no state superposition. The calculation power of a computer is related to the number of transistor in the processor. From observation an empirical law has been established so that the number of transistors in a dense integrated circuit (such as processors) doubles about every 18 months: that's Moore's law. It is a direct consequence of the downsizing of the key element of a processor, the transistor, thanks to technological improvement in their fabrication process. For example, Intel Core i7 8th generation (2017) is base on 14nm transistors only! IBM has announced in 2017 being able to produce chips with 5nm transistors! Going further down to lower size, one will reach the limit where quantum effects are no longer negligible (atomic size devices).

IBM roadmap 2014:

— quantum effects are no longer negligeable;
— new materials ? (post-silicon era);
— toward quantum computing ?
$\Longrightarrow$ **it requires to change radically the vision of computing, either with a complete new technology (new materials such as carbone nanotubes) or rethink computing to benefit from quantum effects instead of trying to deal with them in a classical architecture.**

Instead of dealing with quantum effects, exploiting them for calculation: **it a totally new vision of computing!**

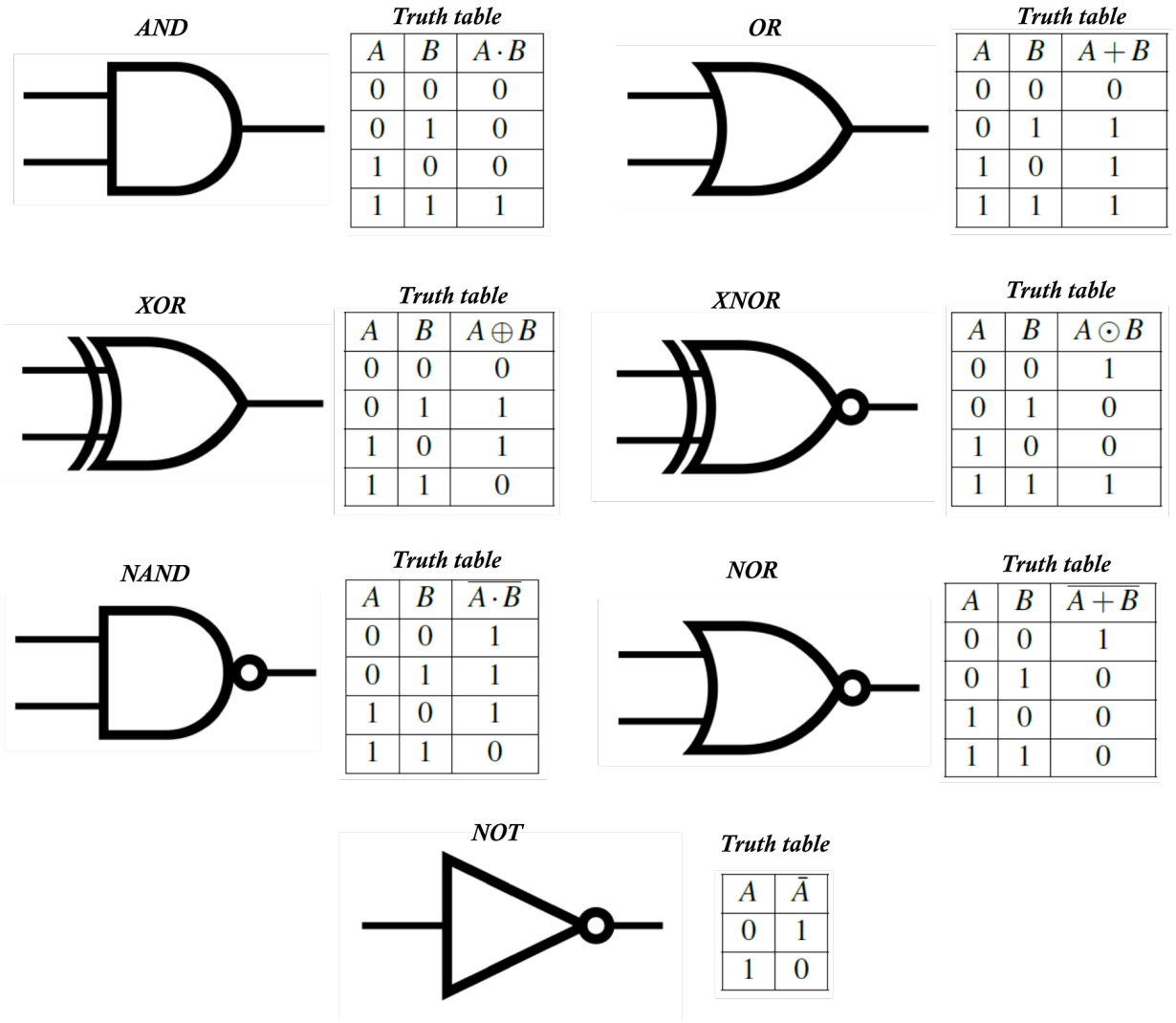


Fig. I.1. Symbolic representation of classical logical gates and corresponding truth tables: AND, OR, XOR, XNOR, NAND, NOR and NOT gates.

2.5. Notion of qubit

The simplest quantum system is a two-state system (so-called two-level system).

E.g.: spin 1/2 in a $\vec{B}$ field, polarization of a photon,...

Let consider the case of a spin 1/2 in a $\vec{B}$ field. The Hilbert space associated if of dimension 2×2, of the following basis

$$\mathscr{B} = \{|\uparrow\rangle, |\downarrow\rangle\}.$$

In the context of quantum information, those two states are labelled as $|0\rangle$ and $|1\rangle$ (e.g. $|\uparrow\rangle = |1\rangle$ and $|\downarrow\rangle = |0\rangle$) and are the quantum equivalent of the classical bits 0 and 1. What is the main difference between a classical bit and a quantum qubit ? The huge difference with a qubit is the possibility to be a superposition of $|0\rangle$ and $|1\rangle$, which is so-called *quantum parallelism*. It's the key point that will permit to a quantum computer to speed up calculations compared to a classical computer, with the ability, in a simple view, to perform several calculation at the same time thank to quantum superposition of states of each qubits. Moreover, it increases the amount of information that might be encoded in n qubits compared to n bits.

Quantum superpositions allow to perform calculation on many states **at the same time**. Thank to quantum superpostion, **quantum algorithms** could improve classical ones with exponential speed-up. **But**, as a draw-

Classical bit	**Qubit**
0 or 1	$\|0\rangle$ or $\|1\rangle$ **OR** $\alpha\|0\rangle + \beta\|1\rangle$

back, once we measure the superposition of states, it collapses to one of its states. Therefore, we can only get one "answer" and not all answers to all states in the superposition. This imposes strong constraints on the quantum algorithm to keep the benefit of the quantum superposition, with "tricks" to deal with the projective measurement: that the mean difficulty of quantum algorithms.

Then, it is not that easy to design quantum algorithms, but one can use interferences effects.

2.6. Dynamics of a qubit

The time evolution of a state $|\psi\rangle$ of a closed quantum system is described by the Schrödinger equation

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

where $\hat{H}(t)$ is the hamiltonian. A closed quantum system does not interact with any other systems. When $\hat{H}(t) = \hat{H}$ is not time dependent, the general solution is

$$|\psi(t)\rangle = \exp\left(-i\frac{\hat{H}t}{\hbar}\right)|\psi(0)\rangle.$$

Moreover, $\hat{H}$ is hermitian

$$\hat{H}^\dagger = \hat{H} \;\Rightarrow\; \left(\hat{H}|\psi\rangle\right)^\dagger = \langle\psi|\hat{H}.$$

Any states might be decomposed on a basis made out of eigenvectors of the Hamiltonian $\hat{H}$. Consequently, the observable $\hat{H}$ might be decomposed as a superposition of projectors on eigen subspaces ponderated by their eigenvalue (spectral decomposition of $\hat{H}$) as follow

$$\hat{H} = \sum_i E_i|\psi_i\rangle\langle\psi_i|,$$

with eigenvalues E_i and eigenvectors $|\psi_i\rangle$. The smallest value of $E_i = E_0$ is the ground state energy with the corresponding eigenstate $|\psi_0\rangle$.

Example: let considers a two-level system (e.g. electronic spin in a B field), as depicted Fig. I.2. The corresponding hamiltonian is the following

$$\hat{H} = -\frac{\hbar\omega}{2}\left(|0\rangle\langle 0| - |1\rangle\langle 1|\right).$$

What is the temporal evolution of a quantum state governed by such an hamiltonian?
— If $|\psi(0)\rangle = |0\rangle$,then $|\psi(t)\rangle = e^{i\frac{\omega t}{2}}|0\rangle$;
— if $|\psi(0)\rangle = |1\rangle$, then $|\psi(t)\rangle = e^{-i\frac{\omega t}{2}}|1\rangle$;
— If $|\psi(0)\rangle = \alpha|0\rangle + \beta|1\rangle$ with $|\alpha|^2 + |\beta|^2 = 1$ (normalized state), then

$$\begin{aligned} |\psi(t)\rangle &= \alpha e^{i\frac{\omega t}{2}}|0\rangle + \beta e^{-i\frac{\omega t}{2}}|1\rangle, \\ &= e^{i\frac{\omega t}{2}}\left(\alpha|0\rangle + \beta e^{-i\omega t}|1\rangle\right). \end{aligned}$$

The global phase of a quantum state is not relevant for the final probability of a measurement (equivalent to a choice of energies' origin). So it is physically equivalent to the following state (quantum states are defined within a global phase)

$$\boxed{|\psi(t)\rangle = \alpha|0\rangle + \beta e^{-i\omega t}|1\rangle}.$$

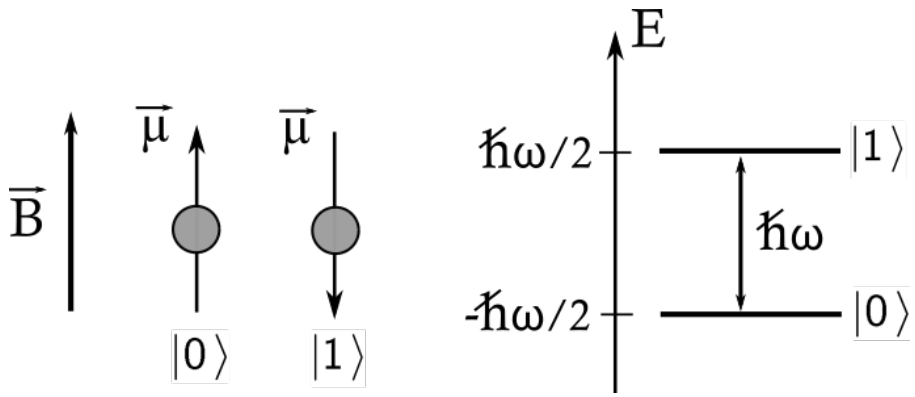


Fig. I.2. Electronic spin in a B field as a two-level system.

2.7. Manipulation of a qubit

A two level state might be manipulated by applying external operation on it. Such an operation should preserve the total probability, so that the final state is still a physical state (*i.e.* normalized). Let's call $\hat{U}$ the operator applied for such an operation, such that

$$|\psi'\rangle = \hat{U}|\psi\rangle.$$

Since the probability is conserved, one should have

$$\langle\psi'|\psi'\rangle = \langle\psi|\psi\rangle \;\Rightarrow\; \langle\psi|\hat{U}^\dagger\hat{U}|\psi\rangle = \langle\psi|\psi\rangle.$$

The operator $\hat{U}$ preserves the norm of a vector: it is a unitary opertor properties.

$$\hat{U} \text{ is unitary } \Leftrightarrow\; \hat{U}^\dagger\hat{U} = \hat{U}\hat{U}^\dagger = \mathbb{I}.$$

Such unitary operator has eigenvalues with specific properties. Let note α such an eigenvalue of $\hat{U}$

$$\hat{U}|\alpha\rangle = \alpha|\alpha\rangle.$$

Since $\hat{U}$ is unitary, $\hat{U}^\dagger\hat{U} = \mathbb{I}$ so that

$$\Rightarrow \langle\alpha|\hat{U}^\dagger\hat{U}|\alpha\rangle = |\alpha|^2 = 1 = \langle\alpha|\mathbb{I}|\alpha\rangle.$$

Then,

$$\alpha = e^{j\theta_\alpha},\ \theta_\alpha \in \mathbb{R}.$$

Eigenvalues of $\hat{U}$ are of complex values with a unitary modulus, *i.e.* $|\alpha| = 1$.

Let $\hat{A}$ being an operator on a quantum system. The exponential operator of $\hat{A}$ is defined as follow

$$\exp\left(i\hat{A}x\right) = \sum_{n=0}^{+\infty} \frac{\left(i\hat{A}x\right)^n}{n!},\ \text{with } x \in \mathbb{R}.$$

Let's consider the particular case where $\hat{A}^2 = \mathbb{I}$, then

$$\begin{aligned}\exp\left(i\hat{A}x\right) &= \sum_{p=0}^{+\infty} \frac{\left(i\hat{A}x\right)^{2p}}{2p!} + \sum_{p=0}^{+\infty} \frac{\left(i\hat{A}x\right)^{2p+1}}{(2p+1)!},\\ &= \sum_{p=0}^{+\infty} \frac{(ix)^{2p}}{2p!} + \sum_{p=0}^{+\infty} \frac{(ix)^{2p+1}}{(2p+1)!}\hat{A}.\end{aligned}$$

Consequently, $\forall\hat{A}$ such that $\hat{A}^2 = \mathbb{I}$,

$$\boxed{\exp\left(i\hat{A}x\right) = \cos x + i\sin x\hat{A}}.$$

Chapter II

Manipulation of a single qubit

1. Bloch sphere representation of a two-level system

Let's consider a two-level quantum system. The corresponding state space is an hilbert space of dimension 2×2. With a basis made out of two vectors denoted $|0\rangle$ and $|1\rangle$, it is possible to decompose any quantum state on this basis. Then, the general expression of a quantum state $|\psi\rangle$ of a two level quantum system is the following

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

with $(\alpha,\beta) \in \mathbb{C}^2$ and $|\alpha|^2 + |\beta|^2 = 1$. The coefficients (α,β) might be expressed as follow

$$\alpha = \cos\frac{\theta}{2} \text{ and } \beta = e^{i\varphi}\sin\frac{\theta}{2},$$

such that

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

for $\theta \in [0, 2\pi[$ and $\varphi \in [0, 2\pi[$. The parameters θ and φ might be seen as the angle of the direction of the equivalent spin of the two level system. The geometrical representation of this direction on a unitary sphere is the **Bloch sphere** representation of the quantum state (introduced initially by Felix Bloch, Nobel Prize in 1952 for NMR). Then, any pure quantum state of a qubit, which is a two level system, can be visualized as a point on this unit sphere. The basis vectors $|0\rangle$ and $|1\rangle$ are given by two points on the Bloch sphere, for $\theta = 0$ and $\theta = \pi$, *i.e.* diametrically opposed on the vertical axis (see Fig. II.1).

$$\theta = 0 \rightarrow |\psi\rangle = |0\rangle,$$

$$\theta = \pi \rightarrow |\psi\rangle = |1\rangle.$$

Any state on the equator of the Bloch sphere corresponds to a quantum superposition of $|0\rangle$ and $|1\rangle$ with equal probability of measuring each state. The angle φ corresponds then to the relative phase between $|0\rangle$ and $|1\rangle$ in such a state

$$\theta = \frac{\pi}{2} \rightarrow |\psi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\varphi}|1\rangle\right).$$

Since any state is a point on a sphere, any change of state correspond to a displacement on a sphere. Therefore, as an important consequence **any unitary operator $\hat{U}$ might be seen as a rotation on the Bloch sphere.** Moreover, any unitary operator $\hat{U}$ on a two-level system is a 2×2 matrix, and might be expressed as a function of four basis operators. A commonly used basis consists in **Pauli's matrices.**

Pauli's matrices:

$$\mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}; \sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}; \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}; \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Alternative notations:

$$\hat{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}; \hat{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}; \hat{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

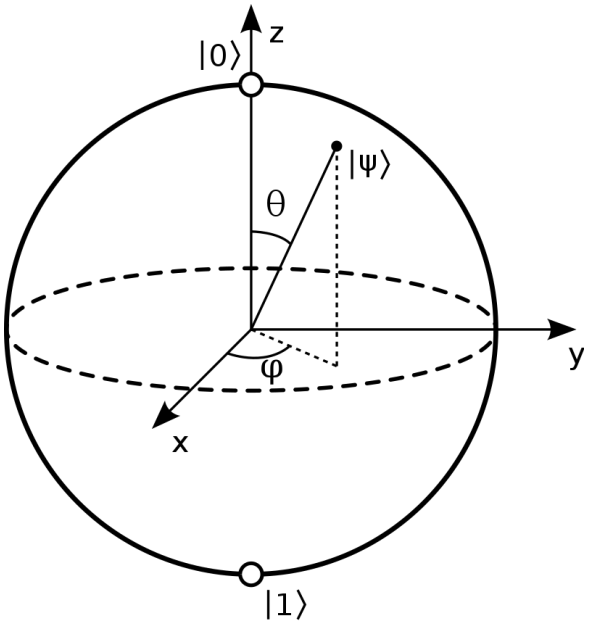


Fig. II.1. Bloch's sphere representation of a qubit.

Pauli's matrices are the generators of rotations for a two-level system. Let's interpret this results in term of rotation of the Bloch sphere. For instance, let's consider a rotation of an angle ψ around the X axis, which is formally given by the following matrix in the $\{|0\rangle, |1\rangle\}$ basis

$$R_X(\psi) = \begin{bmatrix} \cos\frac{\psi}{2} & -i\sin\frac{\psi}{2} \\ -i\sin\frac{\psi}{2} & \cos\frac{\psi}{2} \end{bmatrix}.$$

Such a matrix might be expressed with Pauli's matrices as follow

$$R_X(\psi) = \cos\left(\frac{\psi}{2}\right)\mathbb{I} - i\sin\left(\frac{\psi}{2}\right)\hat{X}.$$

Pauli's matrices properties

$$\hat{X}^2 = \hat{Y}^2 = \hat{Z}^2 = \mathbb{I},$$

$$[\sigma_i, \sigma_j] = 2i\varepsilon_{ijk}\sigma_k \text{ and } \{\sigma_i, \sigma_j\} = 2\delta_{ij}\mathbb{I},$$

where ε_{ijk} is the Levi-Civita symbol and δ_{ij} is the Kronecker symbol.

Consequently

$$\sigma_i\sigma_j = i\varepsilon_{ijk}\sigma_k + \delta_{ij}\mathbb{I}.$$

Since $\hat{X}^2 = \mathbb{I}$,

$$\boxed{\hat{R}_X(\psi) = \cos\left(\frac{\psi}{2}\right)\mathbb{I} - i\sin\left(\frac{\psi}{2}\right)\hat{X} = \exp\left(-i\frac{\psi}{2}\hat{X}\right)}.$$

Similarly, one gets

$$\boxed{\hat{R}_Y(\psi) = \cos\left(\frac{\psi}{2}\right)\mathbb{I} - i\sin\left(\frac{\psi}{2}\right)\hat{Y} = \exp\left(-i\frac{\psi}{2}\hat{Y}\right)},$$

$$\boxed{\hat{R}_Z(\psi) = \cos\left(\frac{\psi}{2}\right)\mathbb{I} - i\sin\left(\frac{\psi}{2}\right)\hat{Z} = \exp\left(-i\frac{\psi}{2}\hat{Z}\right)}.$$

One may defines the rotation around an arbitrary direction $\vec{n}$

$$\vec{n} = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix}.$$

Let introduce the following operator

$$\hat{\vec{\sigma}} = \begin{pmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{pmatrix}.$$

Then, any rotation around the direction $\vec{n}$ might be expressed as the exponential matrix of a superposition of Pauli's matrices

$$\begin{aligned} \hat{R}_{\vec{n}}(\psi) &= \exp\left(-i\frac{\psi}{2}\vec{n}\cdot\hat{\vec{\sigma}}\right) \\ &= \cos\left(\frac{\psi}{2}\right)\mathbb{I} - i\sin\left(\frac{\psi}{2}\right)\left(n_x\hat{X} + n_y\hat{Y} + n_z\hat{Z}\right) \end{aligned}$$

How to perform such a rotation on a "real" system ? In the next section, let's consider the case of a spin 1/2 system in a magnetic field (NMR).

2. Case of NMR: single qubit manipulation

2.1. Nuclear spin in a static $\vec{B}$ field

Let's consider a static magnetic field $\vec{B}_0 = B_0\vec{u}_z$, so that z is the quantification axis. Let's consider an atom with a nuclear spin $\hat{\vec{I}}$, which is a spin 1/2. It corresponds to a magnetic momentum $\hat{\vec{M}} = \gamma\hat{\vec{I}}$, where γ is the gyromagnetic factor (g-factor). Magnetic interaction between the $\vec{B}_0$ field and the magnetic momentum results in the following hamiltonian

$$\hat{H}_0 = -\hat{\vec{M}}\cdot\vec{B}_0 = \gamma\hat{\vec{I}}\cdot\vec{B}_0 = -\gamma\frac{\hbar}{2}B_0\hat{Z}.$$

Introducing the **Larmor frequency** $\boxed{\omega_0 = \gamma B_0}$, one gets the following Hamiltonian expressed with $\hat{Z}$ Pauli's matrix

$$\boxed{\hat{H}_0 = -\frac{1}{2}\hbar\omega_0\hat{Z}}.$$

$\hat{H}_0$ is time-independent, so the time-evolution of any state is obtained with the evolution operator as follow

$$|\psi(t)\rangle = \exp\left(-i\frac{\hat{H}_0 t}{\hbar}\right)|\psi(0)\rangle,$$

then

$$\boxed{|\psi(t)\rangle = \exp\left(i\frac{\omega_0 t}{2}\hat{Z}\right)|\psi(0)\rangle}.$$

One recognizes a rotation operator around the z axis, so that **the evolution of the nuclear spin is a rotation along z axis at larmor frequency ω_0.**

2.2. Static $\vec{B}_0$ field and RF $\vec{B}_{rf}$ field

Now let's consider the same spin in a static $\vec{B}_0$ field superimposed to a radio-frequency (RF) rotating field $\vec{B}_{rf}$, perpendicular to the static magnetic field $\vec{B}_0$. Then, the total magnetic field is expressed as follow

$$\vec{B}_{rf} = B_1\left(\cos\left(\omega_{rf}t + \phi\right)\vec{u}_x - \sin\left(\omega_{rf}t + \phi\right)\vec{u}_y\right).$$

II. Manipulation of a single qubit

If one notes $\omega_1 = \gamma B_1$, the hamiltonian of the spin becomes

$$\hat{H} = -\frac{1}{2}\hbar\omega_0\hat{Z} - \frac{1}{2}\hbar\omega_1\left(\cos(\omega_{rf}t+\phi)\hat{X} - \sin(\omega_{rf}t+\phi)\hat{Y}\right).$$

To describe the dynamics of such a system, it is common to change the frame of study, more specifically the frame in which the spin is static in presence of $\vec{B}_0$ only, *i.e.* the frame rotating at Larmor frequency around z. Let's then consider the rotating frame at $-\omega_{rf}$ around z, so the state becomes

$$\boxed{|\psi\rangle^{rot} = \exp\left(-\frac{i}{2}\omega_{rf}t\hat{Z}\right)|\psi\rangle}.$$

What is the expression of the hamiltonian in the rotating frame ?

Let's consider a unitary transformation $\hat{U}(t)$ on a state $|\psi\rangle$

$$|\psi'\rangle = \hat{U}(t)|\psi\rangle \;\Leftrightarrow\; |\psi\rangle = \hat{U}^\dagger(t)|\psi'\rangle.$$

Then, since this state is governed by Schrödinger's equation, one gets

$$i\hbar\frac{\partial}{\partial t}|\psi\rangle = i\hbar\left(\frac{\partial\hat{U}^\dagger(t)}{\partial t}\right)|\psi'\rangle + i\hbar\hat{U}^\dagger(t)\frac{\partial|\psi'\rangle}{\partial t},$$

$$\hat{H}|\psi\rangle = \hat{H}\hat{U}^\dagger(t)\hat{U}(t)|\psi\rangle = \hat{H}\hat{U}^\dagger(t)|\psi'\rangle.$$

Therefore

$$i\hbar\left(\frac{\partial\hat{U}^\dagger(t)}{\partial t}\right)|\psi'\rangle + i\hbar\hat{U}^\dagger(t)\frac{\partial|\psi'\rangle}{\partial t} = \hat{H}\hat{U}^\dagger(t)|\psi'\rangle,$$

$$\Leftrightarrow i\hbar\frac{\partial|\psi'\rangle}{\partial t} = \left(-i\hbar\hat{U}(t)\frac{\partial\hat{U}^\dagger(t)}{\partial t} + \hat{U}(t)\hat{H}\hat{U}^\dagger(t)\right)|\psi'\rangle.$$

Then, if states are transformed by a unitary operator $\hat{U}(t)$, the transformed state is governed by the transformed hamiltonian $\hat{H}'$

$$\boxed{\hat{H}' = \hat{U}(t)\hat{H}\hat{U}^\dagger(t) - i\hbar\hat{U}(t)\frac{\partial\hat{U}^\dagger(t)}{\partial t}}.$$

For a rotation along z at ω_{rf},

$$\hat{U}(t) = \exp\left(-i\frac{\omega_{rf}}{2}t\hat{Z}\right) \;\Rightarrow\; \hat{H}^{rot} = \frac{\hbar\omega_{rf}}{2}\hat{Z} + \hat{U}(t)\hat{H}\hat{U}^\dagger(t),$$

thus, one gets

$$\hat{H}^{rot} = -\frac{\hbar}{2}(\omega_0-\omega_{rf})\hat{Z} - \frac{\hbar}{2}\omega_1\left(\cos(\omega_{rf}t+\phi)\hat{U}(t)\hat{X}\hat{U}^\dagger(t)\right) + \frac{\hbar}{2}\omega_1\left(\sin(\omega_{rf}t+\phi)\hat{U}(t)\hat{Y}\hat{U}^\dagger(t)\right)$$

The expression of Hamiltonian $\hat{H}^{rot}$ is rather complex, and requires to obtain the explicit values of $\hat{U}(t)\hat{X}\hat{U}^\dagger(t)$ and $\hat{U}(t)\hat{Y}\hat{U}^\dagger(t)$. In the following, one will focus on those terms. For instance, let's evaluate explicitly $\hat{U}(t)\hat{X}\hat{U}^\dagger(t)$ from the matrix expressions

$$\begin{aligned}
\hat{U}(t)\hat{X}\hat{U}^\dagger(t) &= \begin{pmatrix} e^{-i\omega_{rf}t/2} & 0 \\ 0 & e^{i\omega_{rf}t/2}\end{pmatrix}\begin{pmatrix}0&1\\1&0\end{pmatrix}\begin{pmatrix} e^{i\omega_{rf}t/2} & 0 \\ 0 & e^{-i\omega_{rf}t/2}\end{pmatrix}, \\
&= \begin{pmatrix} e^{-i\omega_{rf}t/2} & 0 \\ 0 & e^{i\omega_{rf}t/2}\end{pmatrix}\begin{pmatrix} 0 & e^{-i\omega_{rf}t/2} \\ e^{i\omega_{rf}t/2} & 0\end{pmatrix}, \\
&= \begin{pmatrix} 0 & e^{-i\omega_{rf}t} \\ e^{i\omega_{rf}t} & 0\end{pmatrix}.
\end{aligned}$$

Similarly, one gets

$$\begin{aligned}\hat{U}(t)\hat{Y}\hat{U}^{\dagger}(t) &= \begin{pmatrix} e^{-i\omega_{rf}t/2} & 0 \\ 0 & e^{i\omega_{rf}t/2} \end{pmatrix}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\begin{pmatrix} e^{i\omega_{rf}t/2} & 0 \\ 0 & e^{-i\omega_{rf}t/2} \end{pmatrix}, \\ &= \begin{pmatrix} e^{-i\omega_{rf}t/2} & 0 \\ 0 & e^{i\omega_{rf}t/2} \end{pmatrix}\begin{pmatrix} 0 & -ie^{-i\omega_{rf}t/2} \\ ie^{i\omega_{rf}t/2} & 0 \end{pmatrix}, \\ &= \begin{pmatrix} 0 & -ie^{-i\omega_{rf}t} \\ ie^{i\omega_{rf}t} & 0 \end{pmatrix}.\end{aligned}$$

Consequently,

$$\hat{U}(t)\hat{X}\hat{U}^{\dagger}(t)\cos(\omega_{rf}t+\phi) = \frac{1}{2}\begin{pmatrix} 0 & e^{i\phi}+e^{-2i\omega_{rf}t}e^{-i\phi} \\ e^{-i\phi}+e^{2i\omega_{rf}t}e^{i\phi} & 0 \end{pmatrix},$$

and

$$\hat{U}(t)\hat{Y}\hat{U}^{\dagger}(t)\sin(\omega_{rf}t+\phi) = \frac{1}{2i}\begin{pmatrix} 0 & -ie^{i\phi}+ie^{-2i\omega_{rf}t}e^{-i\phi} \\ -ie^{-i\phi}+ie^{2i\omega_{rf}t}e^{i\phi} & 0 \end{pmatrix}.$$

The terms $e^{-2i\omega_{rf}t}$ and $e^{2i\omega_{rf}t}$ are fast rotation compared to the state dynamics. Their effect on the state dynamics is negligible. One then neglect those terms, which is the so-called **rotating wave approximation.**

In the rotating wave approximation, one might approximate the following term

$$\begin{aligned}\hat{U}(t)\hat{X}\hat{U}^{\dagger}(t)\cos(\omega_{rf}t+\phi) - \hat{U}(t)\hat{Y}\hat{U}^{\dagger}(t)\sin(\omega_{rf}t+\phi) \approx & \ \tfrac{1}{2}\begin{pmatrix} 0 & e^{i\phi} \\ e^{-i\phi} & 0 \end{pmatrix} - \tfrac{1}{2i}\begin{pmatrix} 0 & -ie^{i\phi} \\ -ie^{-i\phi} & 0 \end{pmatrix}, \\ = & \ \begin{pmatrix} 0 & e^{i\phi} \\ e^{-i\phi} & 0 \end{pmatrix}, \\ = & \ \begin{pmatrix} 0 & \cos\phi + i\sin\phi \\ \cos\phi - i\sin\phi & 0 \end{pmatrix}, \\ = & \ \cos\phi \underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}}_{=\hat{X}} - \sin\phi \underbrace{\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}}_{=\hat{Y}},\end{aligned}$$

Consequently, in the rotating frame, in the rotating wave approximation, the hamiltonian becomes

$$\boxed{\hat{H}^{rot} = -\frac{\hbar}{2}(\omega_0-\omega_{rf})\hat{Z} - \frac{\hbar}{2}\omega_1\left(\cos\phi\hat{X}+\sin\phi\hat{Y}\right)}.$$

Introducing the detuning δ between RF magnetic field frequency and Larmor frequency in the static field $\boxed{\delta = \omega_0 - \omega_{rf}}$, the hamiltonian reduces to

$$\boxed{\hat{H}^{rot} = -\frac{\hbar\delta}{2}\hat{Z} - \frac{\hbar\omega_1}{2}\left(\cos\phi\hat{X}+\sin\phi\hat{Y}\right)}.$$

The Hamilonian in the rotating frame is a superpostion of three Pauli's matrices such the evolution operator is a arbitrary rotation on the Bloch sphere. One may conclude that any rotation of the state on the Bloch sphere is achievable, using the appropriate values of δ, ω_1 and ϕ (detuning, amplitude and phase of the RF magnetic field) : **it is that which is used in NMR sequences,** the prepare spin states and measure them.

2.3. Qubit measurement

Usually, the measurement of a qubit corresponds to the measurement of the observable $\hat{Z}$. If a measurement of $\hat{X}$ or $\hat{Y}$ is required, one simply has to apply the appropriate rotation on the qubit to transform it as a $\hat{Z}$ measurement after rotation.

2.4. Few realization of qubits

— Nuclear spins in molecules;
— ions in an ion trap;
— quantum dots;
— **superconducting circuits** → Google, IBM, D-waves,...
— polarization states of photons;
— NV center in diamonds;
— ...

3. Quantum gates and quantum circuits

3.1. Definitions

In classical computing, algorithms are implemented by the mean of logical gates, acting on bits. In quantum computing, quantum algorithm are implemented by the mean of **quantum gates**, acting on qubits. A quantum gate is a unitary operator acting on several qubits. A single-qubit gate is a quantum gate acting on a single qubit. Since it is unitary, **a single qubit gate is a rotation on the Bloch sphere.** A gate is represented in

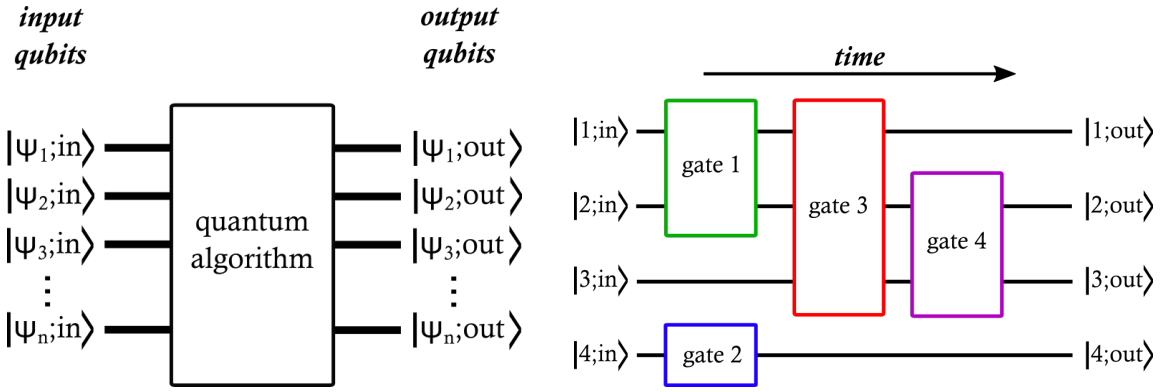


Fig. II.2. Quantum algorithm - I/O approach. Example of quantum circuit.

the **"circuit model"** as a block with an input and an output (see Fig. II.2). Gates are connected by "wires", representing the time evolution of qubits. In the circuit model, a quantum algorithm is represented as a sequence of building blocks that carry out elementary computations (=gates) connected by wires. In quantum computing, a **quantum register** is a system comprising multiple qubits. It's the quantum analog of the classical processor register. Quantum computers perform calculations by manipulating qubits within a quantum register. The size of a quantum register correspond to the number of qubits in it. The corresponding Hilbert space $\mathscr{H}$ in which data of the quantum register are stored is the tensorial product of the Hilbert space of each qubit

$$\mathscr{H} = \mathscr{H}_n \otimes \mathscr{H}_{n-1} \otimes \mathscr{H}_{n-2} \otimes \cdots \mathscr{H}_1.$$

The size of $\mathscr{H}$ is $2^n \times 2^n$ where n is the size of the quantum register. To access to the result of a calculation, a measurement on at least one of the qubits of the quantum register has to be realized, with the block representation of Fig. II.5.a). Note that it's not necessarily all the quantum register that has to be measured.

3.2. Single-qubit gates

A **NOT gate** (bit flip) is a unitary operator that flips $|0\rangle$ and $|1\rangle$, such that

$$\hat{U} = |0\rangle\langle 1| + |1\rangle\langle 0| \equiv \hat{X},$$

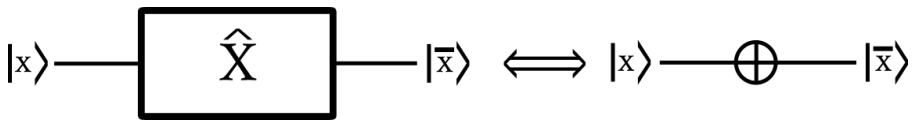


Fig. II.3. NOT gate and its synthetic representation.

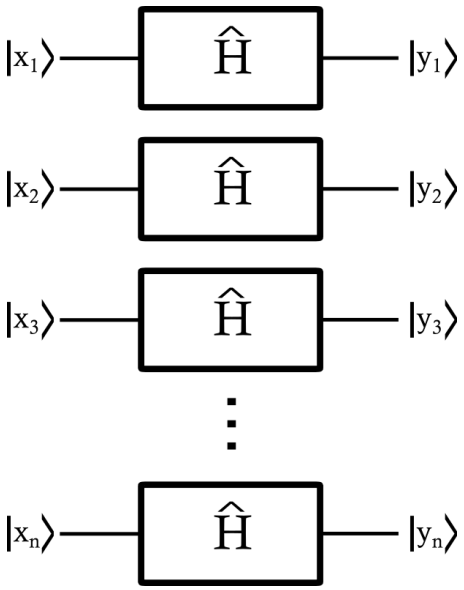


Fig. II.4. Hadamard gate on a quantum register to generate states superposition.

so that a NOT gate is actually an **X gate** (see Fig. II.5.b)). A NOT gate has a synthetic representation in diagrams using the symbol $\bigoplus$, as represented in Fig. II.3. A **Z gate** (phase flip, see Fig. II.5.d)) is a unitary operator such that if the qubit is $|1\rangle$, its phase is shifted by π, and nothing if the qubit is $|0\rangle$.

$$\hat{U} = |0\rangle\langle 0| - |1\rangle\langle 1| \equiv \hat{Z}.$$

The Hadamard gate, noted $\hat{H}$ (see Fig. II.5.e)), is one of the most important gates for quantum circuits.

$$\boxed{\hat{H} \overset{def}{=} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}}.$$

Hadamard gate is mainly used to prepare quantum state superposition. A Hadamard gate might be expressed with a rotation around y axis and a $\hat{Z}$ gate.

$$\hat{H} = \hat{R}_Y\left(\frac{\pi}{2}\right)\hat{Z}.$$

The Hadamard gate allows to transform a pure state ($|0\rangle$ or $|1\rangle$) into a superposition of states such as

$$\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right), \quad \text{or} \quad \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right).$$

The Hadamard gate is a key gate to fully exploit quantum superposition.

Case of a quantum register: a Hadamard gate might be applied on each qubit of a quantum register in ordre to prepare each qubit in a quantum superposition for example. Let's consider a state $|x\rangle$ describing N qubits of a quantum register. This state might be view as the tensor product of N individual states as follow

$$\begin{aligned} |x\rangle &= |x_1\rangle \otimes |x_2\rangle \otimes \cdots |x_N\rangle \\ &= \bigotimes_{i=1}^{N} |x_i\rangle, \end{aligned}$$

After applying a Hadamard on each qubit of $|x\rangle$, one gets the state $|y\rangle\mathscr{H}^{\otimes n}|x\rangle$ that might be decomposed also as a tensor product of individual states as follow

$$\begin{aligned} |y\rangle &= |y_1\rangle \otimes |y_2\rangle \otimes \cdots |y_N\rangle \\ &= \bigotimes_{i=1}^{N} |y_i\rangle. \end{aligned}$$

To understand quantum algorithm, it is important to write this decomposition on the $\{|y_i\rangle\}$ explicitly.

If $N = 1$
For simplicity, let's first consider the case where $N = 1$

$$\begin{aligned} \hat{H}|x\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^x|1\rangle\right), \ x \in \{0,1\} \\ &= \frac{1}{\sqrt{2}}\left((-1)^{0\cdot x}|0\rangle + (-1)^{1\cdot x}|1\rangle\right), \\ &= \frac{1}{\sqrt{2}}\sum_{K\in\{0,1\}} (-1)^{K\cdot x}|K\rangle, \end{aligned}$$

where $K\cdot x$ is a formal scalar product.

If $N \in \mathbb{N}$
The previous result might be easily generalized for N qubits as follow

$$\hat{H}^{\otimes n}|x\rangle = \bigotimes_{i=1}^{N} \hat{H}|x_i\rangle = \bigotimes_{i=1}^{N}\left(\frac{1}{\sqrt{2}}\sum_{K_i\in\{0,1\}} (-1)^{K_i\cdot x_i}|K_i\rangle\right),$$

$$\boxed{\hat{H}^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}}\sum_{K\in\{0,1\}^n} (-1)^{\langle K|x\rangle}|K\rangle}.$$

The later expression will be used afterwards to describe several quantum algorithms.

A **phase gate** $\hat{R}_\theta$ (see Fig. II.5.g)) is a single qubit gate that apply a phase θ only on the $|1\rangle$ state of a qubit (*i.e.* multiply $|1\rangle$ by $e^{i\theta}$)

$$\hat{R}_\theta = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix} = |0\rangle\langle 0| + e^{i\theta}|1\rangle\langle 1|.$$

A **S-gate** $\hat{S}$ is a phase gate in the particular case where $\theta = \pi/2$ (see Fig. II.5.h)).

$$\hat{S} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = |0\rangle\langle 0| + i|1\rangle\langle 1|.$$

A **T-gate** is a phase gate in the particular case where $\theta = \frac{\pi}{4}$ (see Fig. II.5.i)). It is also defined as the square root of $\hat{S}$, $\hat{T} = \sqrt{\hat{S}}$.

$$\hat{T} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix} = |0\rangle\langle 0| + e^{i\frac{\pi}{4}}|1\rangle\langle 1|.$$

General structure of a quantum algorithm: a quantum algorithm is generaly composed of four steps as follow

1. Initialization of qubits in a pure state (e.g. $|0\rangle^{\otimes n} = |0\rangle\cdots|0\rangle$).
2. Set the system in a superposition of states.
3. Unitary manipulation of qubits.
4. Measurement of qubits.

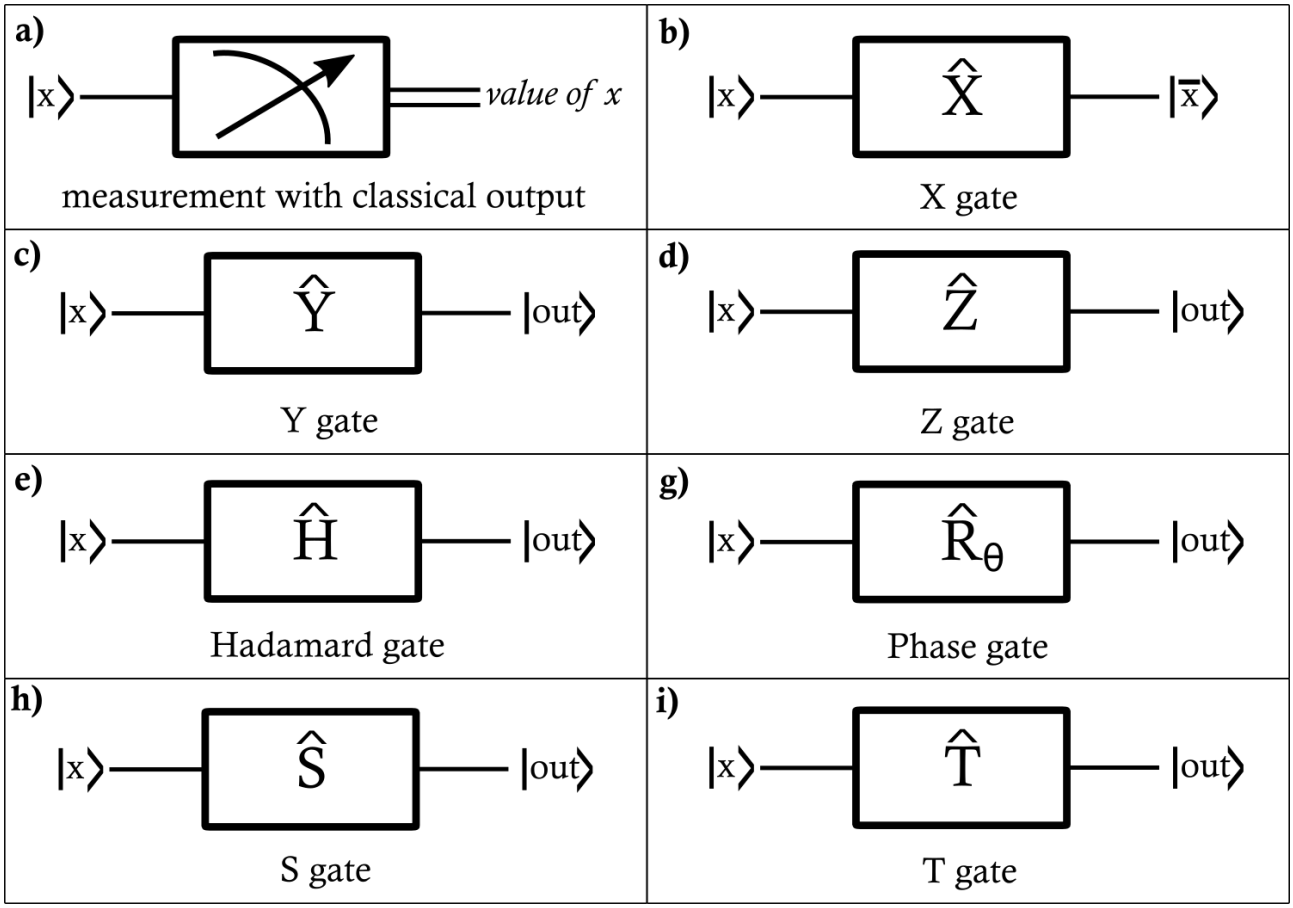


Fig. II.5. a) Representation of a measurement with a classical output in the circuit diagram formalism. b) $\hat{X}$ gate. c) $\hat{Y}$ gate. d) $\hat{Z}$ gate. e) Hadamard gate f) Phase gate of phase θ. h) $\hat{S}$ gate. i) $\hat{T}$ gate.

Chapter III

Multiqubit gates and C-gates

1. Binary quantum gates

1.1. Definition

A binary quantum gate is a unitary operation on two qubits, *i.e.* a unitary map $\mathscr{H}_2 \otimes \mathscr{H}_2 \to \mathscr{H}_2 \otimes \mathscr{H}_2$, where $\mathscr{H}_2$ is a hilbert space of dimension 2×2. A basis of $\mathscr{H}_2 \otimes \mathscr{H}_2$ is $\{|00\rangle, |10\rangle, |01\rangle, |11\rangle\}$.

C-gate: let's A and B be two qubits. Let's M be a unitary quantum gate acting on B. The controlled-M gate (or C-M gate) is the binary gate acting on $A \otimes B$ defined as follow

$$\text{C-}M = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes \mathbb{I}_B + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes M,$$

where $\mathbb{I}_B$ is the identity operator on qubit B. A C-gate (controlled gate) is the operation such that M is applied to B only if the qubit A is in the state $|1\rangle$.

Important example: C-NOT gate. The C-NOT gate is a controlled gate that applies a NOT gate on a qubit only if the other qubit is in the $|1\rangle$ state.

$$\text{C-NOT} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\hat{U}_{\text{C-NOT}}|00\rangle = |00\rangle, \quad \hat{U}_{\text{C-NOT}}|01\rangle = |01\rangle, \quad \hat{U}_{\text{C-NOT}}|10\rangle = |11\rangle, \quad \hat{U}_{\text{C-NOT}}|11\rangle = |10\rangle.$$

$$\hat{U}_{\text{C-NOT}}\left(\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)\right) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

1.2. Circuit representation of a C-gate

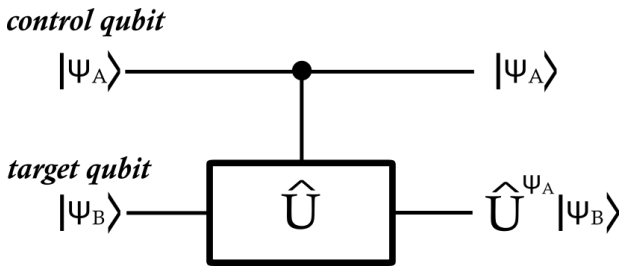


Fig. III.1. C-U gate.

Let's consider a C-U gate where the controlled qubit is A and the applied gate is $\hat{U}$. Such a C-gate is represented as in Fig. III.1. It is formally equivalent to apply the operator $\hat{U}^{\psi_A}$ on $|\psi_B\rangle$ when the qubit A is in state $|\psi_A\rangle$ (either 0 or 1). The qubit A controls the action on qubit B, *i.e.* applying $\hat{U}$ on B depending on the state of A. Then, A is called a *control qubit* while B is called the *target qubit*.

Example: C-NOT gate is a NOT gate where the controlled qubit is A. A C-NOT gate might be seen as a way to implement the XOR classical gate.

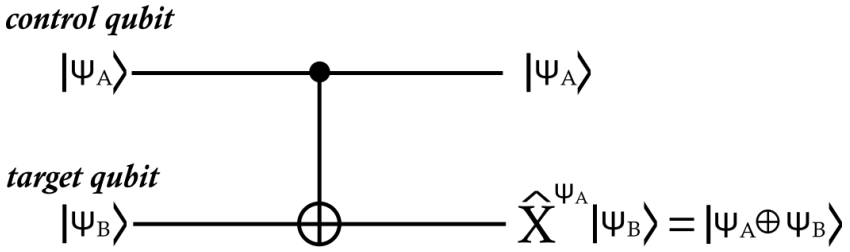


Fig. III.2. C-NOT gate.

1.3. Importance of the C-NOT gate

Theorem: *All quantum circuits can be constructed using only C-NOT gates and single-qubit gates* [5].

That is the reason why the development of a C-NOT gate is so important, in the context of hardware development. Hardware players focus on the achievement of high quality C-NOT gate, so that any other controlled-gate would be achievable afterwards. C-NOT gate is self inverse gates

$$(\text{C-NOT}) \cdot (\text{C-NOT}) = \mathbb{I} \otimes \mathbb{I}.$$

2. Examples of multiqubit gates

2.1. The Toffoli gate

The **Toffoli gate**, originally devised as a universal, reversible classical logic gate by Toffoli, is especially interesting because depending on the input, the gate can perform logical AND, XOR and NOT operations, making it universal for classical computing [25]. Toffoli is often referred to a "controlled-controlled-NOT" gate (C^2-NOT). The circuit diagramm of a Toffoli gate is represented Fig.III.3. Toffoli gate is self inversed

$$\text{Toffoli} \cdot \text{Toffoli} = \mathbb{I} \otimes \mathbb{I} \otimes \mathbb{I}.$$

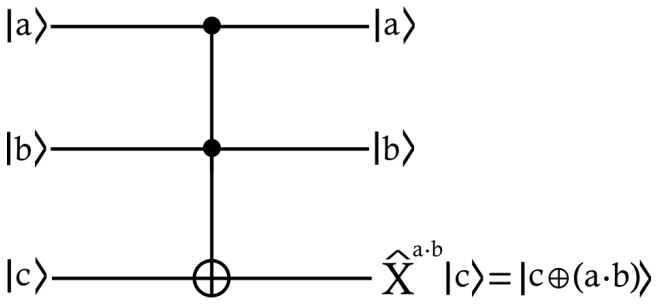


Fig. III.3. Toffoli gate.

The matrix representation of a Tofolli gate is the following

$$\text{Toffoli} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Theorem: *All quantum circuits can be constructed (in some approximated sense) using only Hadamard gates and Toffoli gates* [24].

Example of circuit made of only Toffoli and C-NOT gates

The example of the classical and quantum half-adder is proposed Fig. III.4. A classical half-adder compute the sum and carry for two bits x_1 and x_0.

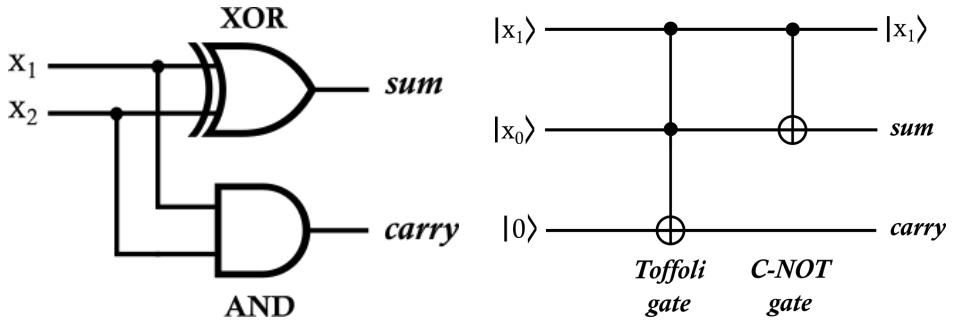


Fig. III.4. Classical and quantum half-adder.

2.2. C-$\hat{Z}$ gate

A C-$\hat{Z}$ gate is a two qubit gate that apply a phase π on the target qubit if the state in the control qubit is $|1\rangle$. Let's note $|x\rangle$ the control qubit and $|y\rangle$ the target qubit. The C-$\hat{Z}$ acts on $|y\rangle$ only if $x = 1$ so that at the output one has $|1\rangle|y\rangle \longrightarrow -|1\rangle|y\rangle$. It is possible to formally compute the output as $(-1)^{x\cdot y}|x\rangle|y\rangle$ for any state $|x\rangle$ since if $x = 0$, $(-1)^{x\cdot y} = 1$, as represented Fig. III.5. One may notice that the gobal output state present a symmetry between x and y. Thus, the output state of a C-$\hat{Z}$ gate is the same, $(-1)^{x\cdot y}|x\rangle|y\rangle$ no matter which qubit was the target one or the control one (see Fig. III.6).

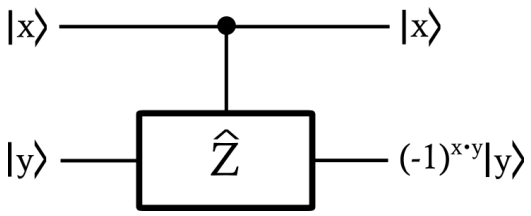


Fig. III.5. C-Z gate.

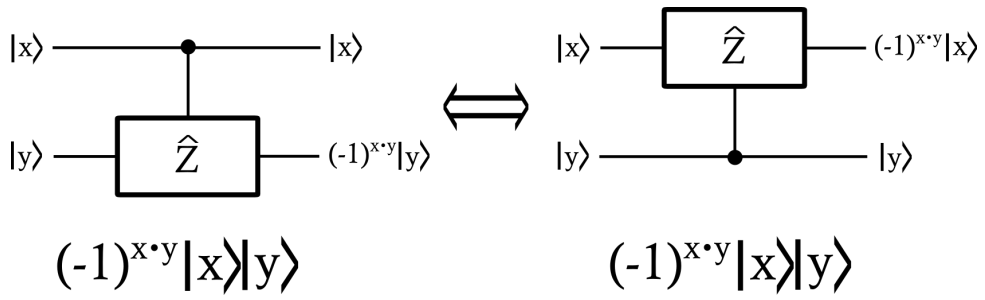


Fig. III.6. C-Z gate.

The matrix representation of a C-$\hat{Z}$ gate is the following

$$\text{C-}Z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

and the formal action on an input state $|x\rangle \otimes |y\rangle$ is the following

$$|x\rangle \otimes |y\rangle \xrightarrow{\text{C}_Z} (-1)^{x \cdot y} |x\rangle \otimes |y\rangle.$$

Since $|x\rangle$ and $|y\rangle$ play symmetric roles, a C-Z gate is usually represented without specification of the control qubit with the simplified notation of Fig. III.7.

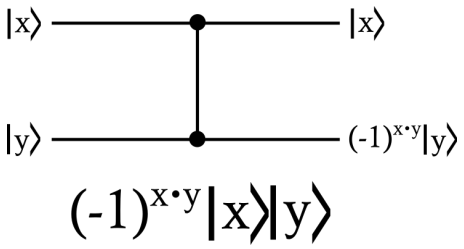


Fig. III.7. C-Z gate compact representation.

2.3. SWAP gate

A SWAP gate is a gate that exchange the state of two qubits. It is very useful in pratice to "displace" a state in a quantum register so that two non-adjacent qubits are bring close together that that is it possible to use them in a multiqubit gate. It another important gate for hardware developers. Indeed, if they may provide a SWAP gate with high performances, it allows them to have more flexibility of the topology of the quantum processor chip, and qubit pairwise connectivity. It is possible to implement a SWAP gate with three C-NOT gate, as demonstrated in Fig. III.8. Consequently, the efficiency of a SWAP gate is in pratice closely related to the efficiency of a C-NOT gate. Therefore, error rates measurement of a C-NOT is on important information during the calibration of hardware, as a pertinent metric of performances of the quantum processor.
A SWAP gate has the following map

$$\text{SWAP} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

A SWAP gate might be implemented with 3 C-NOT gates, as illustrated Fig. III.8.

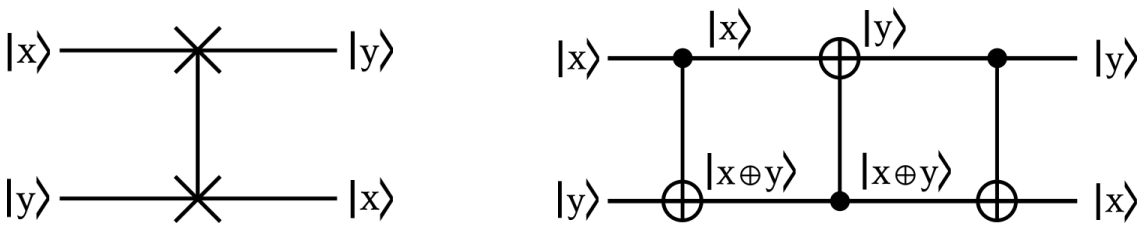


Fig. III.8. SWAP gate and its implementation with 3 C-NOT gates.

$$
\begin{aligned}
|x\rangle \otimes |y\rangle &\xrightarrow{C_{12}} |x\rangle \otimes |y \oplus x\rangle \\
&\xrightarrow{C_{21}} |x \oplus (y \oplus x)\rangle \otimes |y \oplus x\rangle = |y\rangle \otimes |y \oplus x\rangle \\
&\xrightarrow{C_{12}} |y\rangle \otimes |\,(y \oplus x) \oplus y\rangle = |y\rangle \otimes |x\rangle
\end{aligned}
$$

2.4. Logical gates

It is possible to use quantum gates to compute the classical logical function, in particular classical boolean gates such as AND, OR, XOR,... For example, gates AND and OR calculated with a quantum circuits have circuit diagrams represented Fig. III.9. However, one may notice that the output of the calculation is encoded in a third qubit. Such additionnal qubits used in quantum implementation of boolean function is common and a strong constraint on the development of quantum algorithms, and will be discussed afterwards.

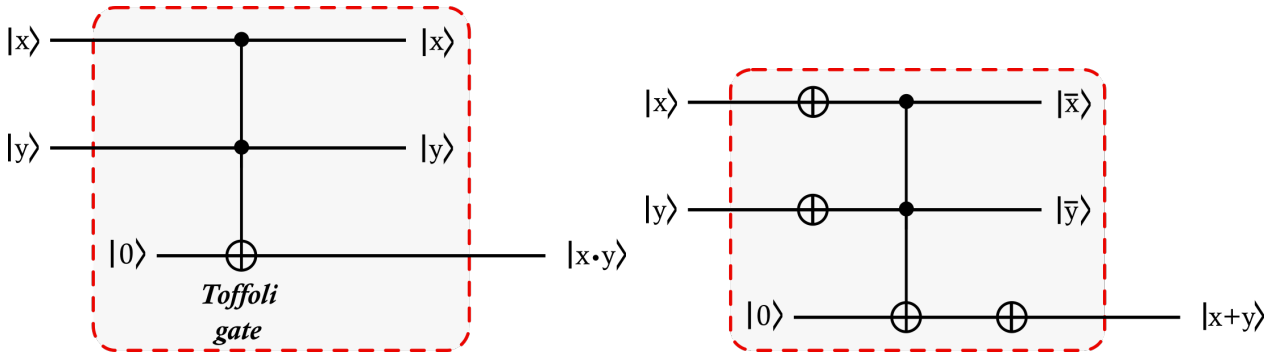


Fig. III.9. AND and OR logical gates based on quantum circuits.

2.5. Boolean circuits

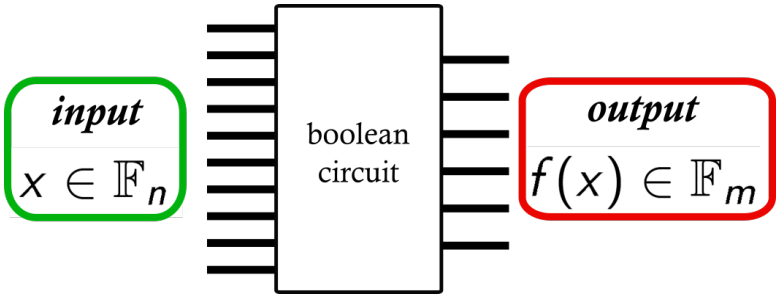


Fig. III.10. Boolean function $f : \mathbb{F}_n \to \mathbb{F}_m$ can't be unitary operation.

III. Multiqubit gates and C-gates

Let's note $\mathbb{F}_n = \{0,1\}^n$. A boolean function $f : \mathbb{F}_n \to \mathbb{F}_m$ can't be a unitary operation. Indeed, if $n \neq m$, the number of inputs do not equal the number of outputs, so the map is not invertible (Fig. III.10). However, it is in fact possible to construct a quantum circuit that performs the same function than any classical boolean circuit. In that aim, let's consider $f : \mathbb{F}_n \to \mathbb{F}_m$ a boolean function with k gates. It is possible to construct a quantum circuit, $\hat{R}$, that performs the same function. This quantum circuit uses $\mathcal{O}(k)$ gates and requires $q = \mathcal{O}(k)$ additional qubits, so-called **ancilla qubits**, only used for the calculation. These ancilla qubits are not part of the quantum register, and are all initially in the pure state $|0\rangle$. Then, such a circuit outputs $n+q-m$ garbage qubits (Fig. III.11).

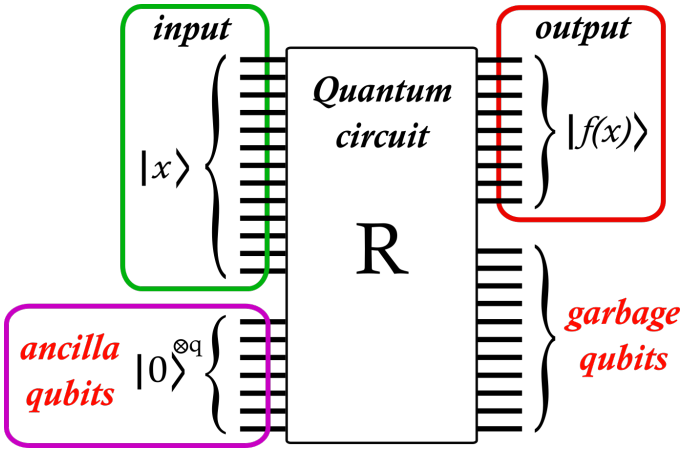


Fig. III.11. Boolean function $f : \mathbb{F}_n \to \mathbb{F}_m$ implemented in a quantum qubit using $q = \mathcal{O}(k)$ ancilla qubits. However, such a circuit outputs $n+q-m$ garbage qubits.

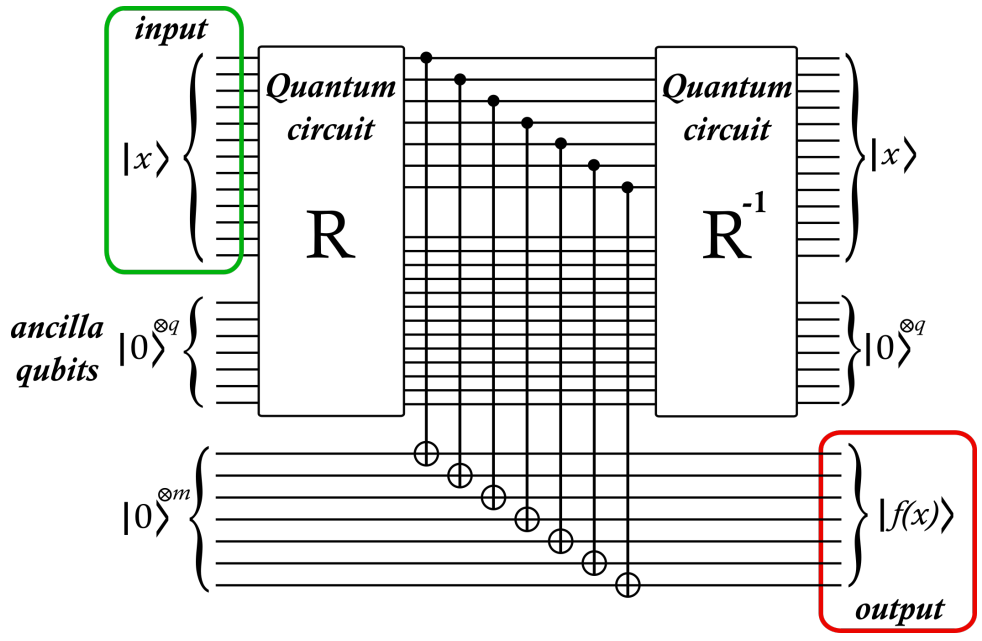


Fig. III.12. Quantum circuit emulating a boolean circuit that performs the function $f : \mathbb{F}_n \to \mathbb{F}_m$ with ancilla qubits recycling.

To obtain the invert $\hat{R}^{-1}$ of $\hat{R}$, one just has to take the mirror image of the circuit $\hat{R}$, fully exploiting the self-inversed properties of gates. This is used to "recycle" the ancilla qubits, so that they are reset to $|0\rangle$. So a quantum circuit emulating a boolean circuit that performs the function $f : \mathbb{F}_n \to \mathbb{F}_m$ has the structure of Fig. III.12.

2.6. Oracle

Oracles are quantum operators which one uses in the context of query complexity. An Oracle is some sort of *black box* associated to a given problem, which provide answers when one submit a query to it, related to the problem considered. We assume to have access to an Oracle, for example a physical device that we cannot look inside, but to which we can pass queries and which returns answers. For a classical computer, the Oracle is given by a function f that from a given input bit string return an output bit string as follow

$$f:\ \mathbb{F}_n \longrightarrow \mathbb{F}_m.$$

$$\text{input string } \underbrace{01\ldots1101\ldots01}_{n \text{ bits}} \longrightarrow \text{output string } \underbrace{00\ldots0111\ldots11}_{m \text{ bits}}.$$

For a quantum computer, the oracle must be **unitary**. Let's then introduce the operator $\hat{O}_f$, a quantum oracle, which can be seen as a unitary operator that performs a map that encode the answer $f(x)$ to the query x. For example, it might perform the following map

$$\hat{O}_f|x\rangle \otimes |y\rangle = |x\rangle \otimes |y \oplus f(x)\rangle,$$

with $|x\rangle \in \mathscr{H}_{2^n}$ and $|y\rangle \in \mathscr{H}_{2^m}$. The answer of the Oracle is encoded in the output state of the target qubit $|y\rangle$ as a XOR boolean operation $y \oplus f(x)$ between y string and $f(x)$ string. It is not possible to directly encoded such a string in the input qubit since the input string length n is not necessary identical to the output string length m. As a consequence, such an Oracle requires the use of ancilla qubits.

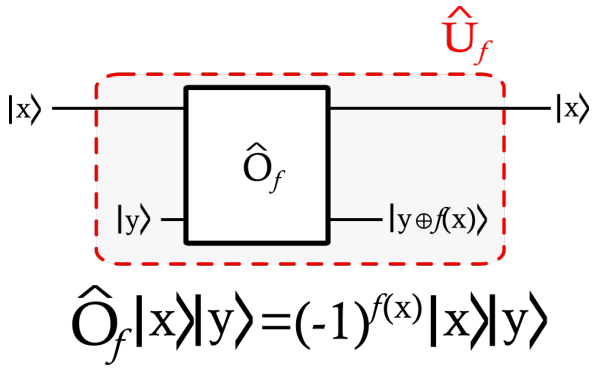


Fig. III.13. Implementation of an oracle. $|y\rangle$ is an ancilla qubit.

Example: for a given function $f:\ \{0,1\}^n \longrightarrow \{0,1\}$, we can construct $\hat{U}_f$ as shown Fig. III.13, with the use of an ancilla qubit $|y\rangle$. For $|y\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, then

$$\boxed{\hat{O}_f|x\rangle \otimes |y\rangle = (-1)^{f(x)}|x\rangle \otimes |y\rangle}.$$

Then, forgetting the ancilla qubit, $\hat{U}_f|x\rangle = (-1)^{f(x)}|x\rangle$. In such a case, the answer $f(x)$ to the query x is encoded in the phase of the state: $\hat{U}_f$ is a so-called **phase oracle**.
Demonstration:

$$\begin{aligned}\hat{O}_f|x\rangle \otimes |y\rangle &= \frac{1}{\sqrt{2}}\left(\hat{O}_f|x\rangle \otimes |0\rangle - \hat{O}_f|x\rangle \otimes |1\rangle\right),\\ &= \frac{1}{\sqrt{2}}\left(|x\rangle \otimes |0 \oplus f(x)\rangle - |x\rangle \otimes |1 \oplus f(x)\rangle\right),\end{aligned}$$

$$\hat{O}_f|x\rangle \otimes |y\rangle = \frac{1}{\sqrt{2}}\begin{cases} |x\rangle \otimes (|0\rangle - |1\rangle) & \text{if} \quad f(x) = 0 \\ |x\rangle \otimes (|1\rangle - |0\rangle) & \text{if} \quad f(x) = 1 \end{cases},$$

$$\hat{O}_f|x\rangle\otimes|y\rangle = \frac{1}{\sqrt{2}}\begin{cases} |x\rangle\otimes|y\rangle & \text{if} \quad f(x)=0 \\ -|x\rangle\otimes|y\rangle & \text{if} \quad f(x)=1 \end{cases},$$

such that

$$\boxed{\hat{O}_f|x\rangle\otimes|y\rangle = (-1)^{f(x)}|x\rangle\otimes|y\rangle}.$$

3. Deutsch-Josa algorithm

3.1. Deutsch algorithm

Deutsch algorithm is a quantum algorithm to solve a very simple problem but for which quantum superposition is illustrated and result in solving the problem with only one query of the Oracle, while a classical algorithm would require two queries. Let's consider the simpliest function possible on bits, with only one bit input and one bit output, such that $f:\ \{0,1\}\longrightarrow\{0,1\}$. There is four possibilities for f

$$\underbrace{\begin{cases} f(0)=0 \\ f(1)=1 \end{cases}}_{\text{identity}}, \underbrace{\begin{cases} f(0)=1 \\ f(1)=0 \end{cases}}_{\text{swap}}, \underbrace{\begin{cases} f(0)=0 \\ f(1)=0 \end{cases}, \begin{cases} f(0)=1 \\ f(1)=1 \end{cases}}_{\text{constant function}}.$$

The function f is said to be **balanced** if $f(0)\neq f(1)$. The function f is said to be **constant** if $f(0)=f(1)$. With a classical computer, one need to evaluate the function f twice to determine whether it is balanced or constant.

Example: f is constant : $f(0)=f(1)=1$.

$$\hat{N}_f = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix},\ \hat{N}_f|0\rangle = |1\rangle,\ \hat{N}_f|1\rangle = |1\rangle.$$

$\hat{N}_f$ is **not** unitary: $\hat{N}_f^\dagger\hat{N}_f \neq \mathbb{I}$. $\hat{N}_f$ does not preserve the norm and consequently the probability. Thus to impliment the function f, it requires to have an Oracle with ancilla qubits. One constructs this Oracle as a

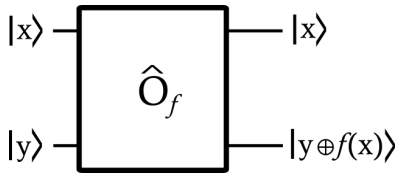


Fig. III.14. Implementation of an oracle for Deutsch and Deutsch-Josa algorithm.

quantum gate that realizes f with a unitary operator $\hat{U}_f$ (see Fig. III.14), where $|x\rangle$ is the qubit on which one wants to evaluate the function f. $|y\rangle$ is an ancilla qubit, allowing the operation to be unitary. Then

$$\hat{U}_f|x\rangle\otimes|y\rangle = |x\rangle\otimes|y\oplus f(x)\rangle,$$

where $\oplus$ is a XOR operation. The oracle $\hat{U}_f$ is reversible (see Fig. III.15), thus

$$\hat{U}_f\dagger\hat{U}_f = \mathbb{I}.$$

Demonstration:

$$\hat{U}_f\hat{U}_f|x\rangle\otimes|y\rangle = \hat{U}_f|x\rangle\otimes|y\oplus f(x)\rangle = |x\rangle\otimes|\,(y\oplus f(x))\oplus f(x)\rangle.$$

Or

$$\begin{aligned}(y\oplus f(x))\oplus f(x) &= y\oplus(f(x)\oplus f(x)) \\ &= y\oplus 0 \\ &= y\end{aligned}$$

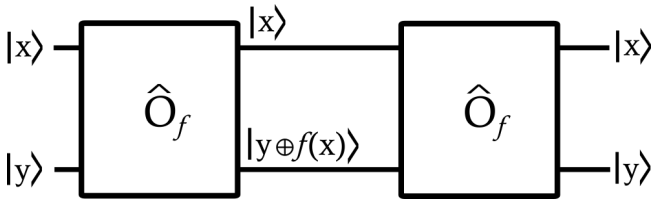


Fig. III.15. Oracle for Deutsch and Deutsch-Josa algorithm is reversible.

Remark: one has used the following identity

$$\forall k \in \{0,1\},\ k \oplus k = 0.$$

$\hat{U}_f$ is unitary and might be used to evaluate f. If the control qubit $|y\rangle = |0\rangle$, then

$$|x\rangle \otimes |y \oplus f(x)\rangle = |x\rangle \otimes |0 \oplus f(x)\rangle,$$

so that

$$|x\rangle \otimes |y \oplus f(x)\rangle = |x\rangle \otimes |f(x)\rangle.$$

So when the ancilla qubit is initiate into the state $|y\rangle = |0\rangle$, this Oracle directly encode the value $f(x)$ as the state of this ancilla qubit.

3.2. Implementation of Deutsch algorithm

Now let's assume that one has an Oracle as described in the previous section associated to a function f. It is possible then to answer the question whether the function f is either constant or balanced with a single query of the Oracle: that is Deutsch algorithm. The circuit diagram of the implementation of Deutsch algorithm is proposed Fig. III.16.

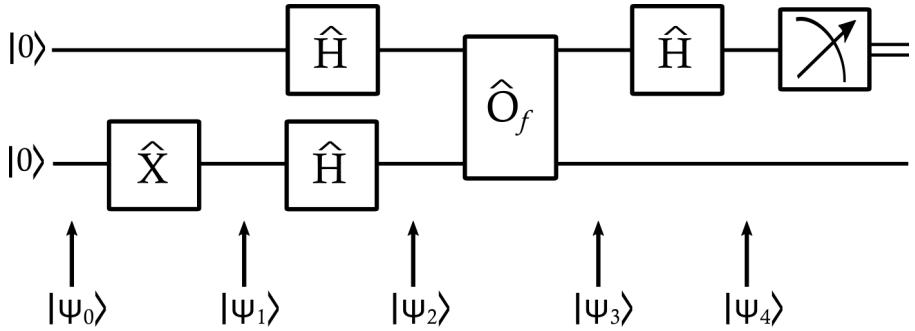


Fig. III.16. Implementation of Deutsch algorithm.

Such an implementation is based on a two qubit register. Let's consider an input state of this register noted $|\psi_0\rangle$ with

$$|\psi_0\rangle = |00\rangle.$$

The global action of the quantum circuit is the following

$$|\psi_4\rangle = \left(\hat{H} \otimes \mathbb{I}\right) \hat{U}_f \left(\hat{H} \otimes \hat{H}\right) \left(\mathbb{I} \otimes \hat{X}\right) |00\rangle.$$

One may describe this action gate after gate to analyze the algorithm. From the initial state $|\psi_0\rangle$, applying a $\hat{X}$-gate on the lower qubit results in a state $|\psi_1\rangle$, with a simple bit flip on the lower qubit

$$|\psi_1\rangle = |01\rangle.$$

Then, a Hadamard gate is applied on each qubit of the register and result in the state $|\psi_2\rangle$

$$|\psi_2\rangle = \left(\hat{H}\otimes\hat{H}\right)|01\rangle = \frac{1}{2}\left(\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\otimes\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\right)\left(\begin{pmatrix}1\\0\end{pmatrix}\otimes\begin{pmatrix}0\\1\end{pmatrix}\right)$$

$$\Leftrightarrow |\psi_2\rangle = \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\otimes\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) = \frac{|00\rangle-|01\rangle+|10\rangle-|11\rangle}{2}.$$

Then, one applies the Oracle operator on the state $|\psi_2\rangle$ with a state $|\psi_3\rangle$ as an output

$$|\psi_3\rangle = \hat{U}_f|\psi_2\rangle = |x\rangle\left(\frac{|0\oplus f(x)\rangle-|1\oplus f(x)\rangle}{\sqrt{2}}\right).$$

For any state $|x\rangle$ in a pure state.

$$\text{If } f(x)=0,\ |\psi_3\rangle = |x\rangle\left(\frac{|0\oplus 0\rangle-|1\oplus 0\rangle}{\sqrt{2}}\right) = |x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right),$$

$$\text{If } f(x)=1,\ |\psi_3\rangle = |x\rangle\left(\frac{|0\oplus 1\rangle-|1\oplus 1\rangle}{\sqrt{2}}\right) = |x\rangle\left(\frac{|1\rangle-|0\rangle}{\sqrt{2}}\right).$$

So finally, any output $|\psi_3\rangle$ might be expressed formally as follow

$$\boxed{|\psi_3\rangle = (-1)^{f(x)}|x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)}.$$

In the algorithm considered, the upper qubit of the quantum register is prepared in a superposition of state by a Hadamard gate prior to the Oracle, as follow

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

then

$$|\psi_3\rangle = \hat{U}_f\frac{|0\rangle}{\sqrt{2}}\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)+\hat{U}_f\frac{|1\rangle}{\sqrt{2}}\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right),$$

$$\Leftrightarrow |\psi_3\rangle = \left(\frac{(-1)^{f(0)}|0\rangle+(-1)^{f(1)}|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right).$$

If f is constant:

$$|\psi_3\rangle = (\pm 1)\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right).$$

If f is balanced:

$$|\psi_3\rangle = (\pm 1)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right).$$

Finally, by applying an Hadamard gate on each qubit of the register after the Oracle, one gets

$$|\psi_3\rangle = \left(\frac{(-1)^{f(0)}|0\rangle+(-1)^{f(1)}|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right).$$

The final state $|\psi_4\rangle$ depends whether f is balanced or constant. Indeed, if f is constant:

$$|\psi_4\rangle = (\pm 1)|0\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right).$$

whereas if f is balanced:

$$|\psi_4\rangle = (\pm 1)|1\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right).$$

The measurement of the first qubit permits to determine unambiguously if the function f is balanced or constant. If the measurement provides 0, the function f is constant. If the measurement provides 1, the function f is balanced. Therefore, The Deutsch problem (determining whether a function $f : \{0,1\} \longrightarrow \{0,1\}$ is balanced or constant) is solved with only one query of the Oracle. A classical computer requires two solicitation of the Oracle, while a quantum algorithm requires only one solicitation of the oracle to get the result. **That's a direct consequence of the quantum parallelism.**

3.3. Deutsch-Josa algorithm

In the case of Deutsch-Josa algorithm, a more general case is considered with a function $f : \{0,1\}^n \longrightarrow \{0,1\}$.

— f is balanced if half of the inputs return 0 and the others return 1.

— f is constant if f only returns 0 or 1.

The Deutsch-Josa algorithm is based on the same principles than the Deutsch algorithm (case $n = 1$), with the implementation of Fig. III.17.

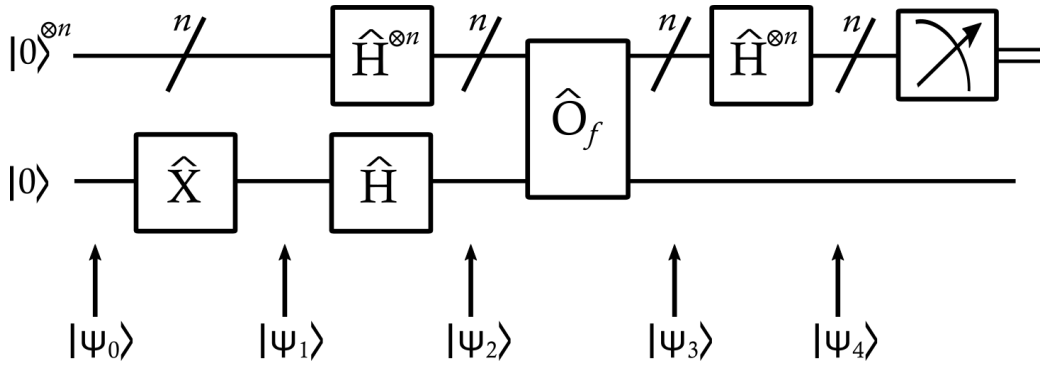


Fig. III.17. Implementation Deutsch-Joza algorithm.

Classical solution: we need to ask the Oracle at least twice, but if we get twice the same value, we need to ask again... corresponding to at most $\frac{N}{2} + 1 = 2^{n-1} + 1$ queries of the Oracle, with n the number of input bits and $N = 2^n$ the number of realizable bit string.

The quantum solution with the Deutsch-Josa algorithm needs only one query !!!

Proof:

Initial state:

$$|\psi_0\rangle = |0\rangle^{\otimes n}|0\rangle = |000\cdots00\rangle|0\rangle.$$

Preparation of the ancilla qubit with a $\hat{X}$ gate:

$$|\psi_1\rangle = |0\rangle^{\otimes n}|1\rangle.$$

Hadamard gate on the quantum register input:

$$|\psi_2\rangle = \left(\hat{H}^{\otimes n}|0\rangle^{\otimes n}\right) \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right).$$

And

$$\hat{H}^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{\langle x|0\rangle}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle,$$

so that $|\psi_2\rangle$ is a superposition of all states as follow

$$\boxed{|\psi_2\rangle = \left(\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle\right)\otimes\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)}.$$

Oracle:

$$|\psi_3\rangle = \hat{U}_f|\psi_2\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle\otimes\left(\frac{|0\oplus f(x)\rangle-|1\oplus f(x)\rangle}{\sqrt{2}}\right),$$

$$\boxed{|\psi_3\rangle = \left(\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle\right)\otimes\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)}.$$

Hadamard gate on the quantum register:

$$|\psi_4\rangle = \left(\hat{H}^{\otimes n}\left(\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle\right)\right)\otimes\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right),$$

$$|\psi_4\rangle = \left(\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}\hat{H}^{\otimes n}|x\rangle\right)\otimes\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right),$$

$$|\psi_4\rangle = \left(\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}\frac{1}{\sqrt{2^n}}\sum_{K\in\{0,1\}^n}(-1)^{\langle K|x\rangle}|K\rangle\right)\otimes\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right),$$

$$|\psi_4\rangle = \left(\sum_{K\in\{0,1\}^n}\left(\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{f(x)+\langle K|x\rangle}\right)|K\rangle\right)\otimes\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right),$$

Let's define

$$C_K = \frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{f(x)+\langle K|x\rangle}, \quad \text{and} \quad |\phi\rangle = \sum_{K\in\{0,1\}^n}C_K|K\rangle.$$

Then

$$|\psi_4\rangle = |\phi\rangle\otimes\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right).$$

The state $|\phi\rangle$ is measured at the end and the probability to measure the string $|000\cdots000\rangle$ is then

$$P(y=00\cdots00) = |\langle 00\cdots00|\phi\rangle|^2.$$

$$\begin{aligned}\Leftrightarrow P(y=00\cdots00) &= \left|\sum_{K\in\{0,1\}^n}C_K\langle 00\cdots00|K\rangle\right|^2 = |C_{00\cdots00}|^2,\\ &= \left|\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}\right|^2\end{aligned}$$

Or

$$\sum_{x\in\{0,1\}^n}(-1)^{f(x)} = \begin{cases}+2^n & \text{if} \quad f(x)=0\\ -2^n & \text{if} \quad f(x)=1\\ 0 & \text{if} \quad f(x) \text{ is balanced}\end{cases},$$

then

$$P(y=00\cdots00) = \begin{cases}1 & \text{if} \quad f(x) \text{ is constant}\\ 0 & \text{if} \quad f(x) \text{ is balanced}\end{cases}$$

So that the probability permits to answer to Deutsch-Josa problem with only one query since

$$P(y = 00\cdots00) = 1 \text{ for a balanced function,}$$

$$P(y = 00\cdots00) = 0 \text{ for a constant function.}$$

In a nutshell, after the measurement of the output state of the quantum register, if one measures the string $00\cdots00$, the function is balanced, otherwise the function is constant. Note that Deutsch-Josa algorithm only works well for balanced function with exactly half of the output 0 and the other half 1. Answer is obtained with a single query to be compared to $2^{n-1}+1$ queries at most with a classical algorithm. If the function is neither balanced nor constant, then

$$P(y = 00\cdots00) \in \,]\,0,1\,[\,,$$

and then Deutsch-Josa algorithm is no longer appropriated for an answer with a single query of the Oracle.

Chapter IV

Implementation of multiqubit gates — Case of NMR

1. Implementation of a C-NOT gate

1.1. Ising interaction

In order to implement a two qubit gate, one should introduce controlable interactions between qubits. Let's consider an ensemble of N qubits labeled with i and j index. Those qubits interacts with an ising-type interaction, *i.e.* a pairwise spin interaction described by the following hamiltonian

$$\hat{H}_{int} = -\sum_{i,j} J_{ij} \hat{Z}_i \hat{Z}_j,$$

where J_{ij} is the intensity of interaction between qubits i and j. It corresponds to a generic two-qubits interaction

$$\hat{H}_{int} = -J \hat{Z}_1 \hat{Z}_2,$$

with $J > 0$ for a ferromagnetic coupling, and $J < 0$ for an anti-ferromagnetic coupling.

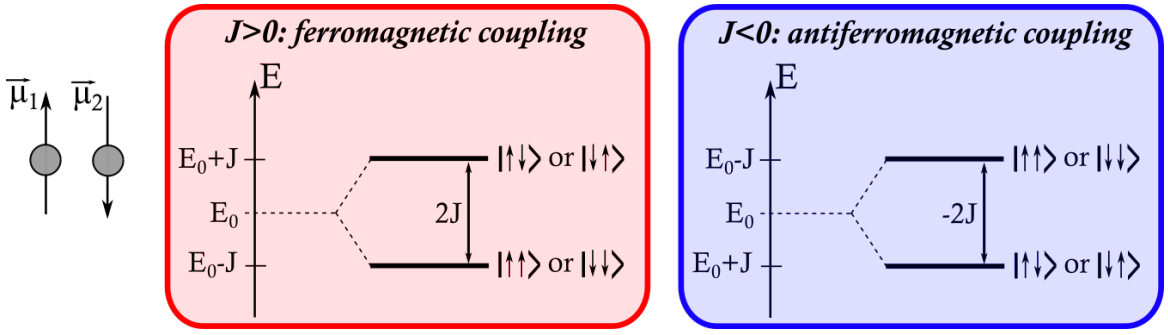


Fig. IV.1. Two spins in interaction: ferromagnetic and anti-ferromagnetic coupling.

1.2. Two-qubit unitary evolution

Applying the hamiltonian

$$\hat{H}_{int} = -J \hat{Z}_1 \hat{Z}_2,$$

for a time T, the unitary evolution of the system of two qubits will be

$$\hat{C}(\gamma) = e^{-i\frac{\gamma}{2}\hat{Z}_1\hat{Z}_2},$$

with

$$\gamma = -\frac{2J}{\hbar}T.$$

It is important to note that $\hat{C}(\gamma)$ **does not realize a C-NOT gate yet**. Indeed, additional single qubit operations on each of the qubits are required. One may demonstrate that a C-NOT gate might be realized by the following

sequence of single qubit gates and Ising type interactions operator $\hat{C}(\gamma)$ as follow

$$e^{-i\frac{3\pi}{4}}\hat{R}_{X2}\left(\frac{3\pi}{2}\right)\hat{C}\left(\frac{3\pi}{2}\right)\hat{R}_{Z2}\left(\frac{\pi}{2}\right)\hat{R}_{X2}\left(\frac{\pi}{2}\right)\hat{R}_{Z2}\left(\frac{\pi}{2}\right)\hat{R}_{Z1}\left(\frac{\pi}{2}\right)\hat{C}\left(\frac{3\pi}{2}\right).$$

Any physical two-qubit interaction that can produce entanglement can be tuned into a universal two-qubit gate (such as the C-NOT gate) when it is augmented by arbitrary number of single qubit operations [8].

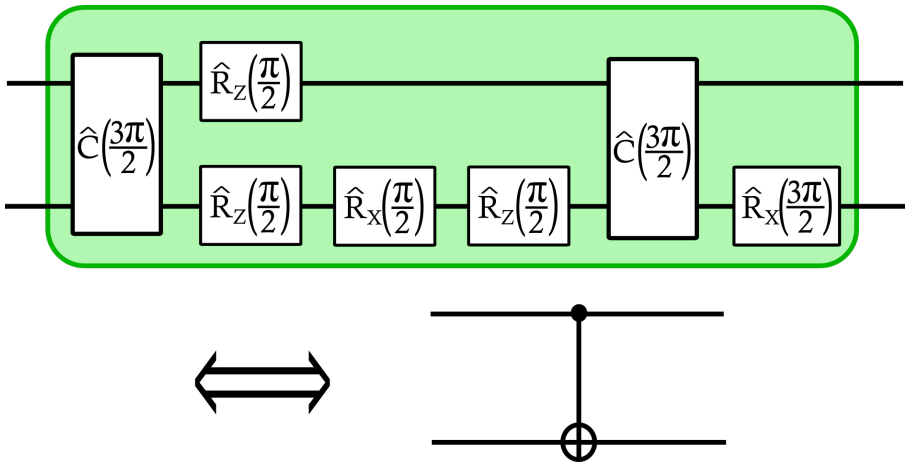


Fig. IV.2. Circuit representation of a C-NOT gate made with a two-qubit interaction $\hat{C}(\gamma)$.

2. Example with NMR quantum computing

2.1. NMR quantum computing

NMR (Nuclear Magnetic Resonance) is an experimental technic based on manipulation and measurement of nuclear spins. It is meanly used for medical imaging and analytical chemistry. It permits to obtain information on the chemical environment of any spin type in a sample (nuclear spin of hydrogen for example), based on resonance shift induced by nearest neighbourhood interactions. NMR technics have been developed in order to prepare, manipulate and measure spin states of a system. Consequently, it's also well adapted for quantum computing implementation and the first quantum calculation in 2001 was implemented in NMR systems [27], where researchers have demonstrate the ability to factorize $15 = 3 \times 5$ in molecules by NMR. Since then, several review paper have been written on NMR technics applied to quantum computing [26, 12].

2.2. Manipulation of qubits and NMR

In the context of quantum computing, how one may exploite NMR technics ?

— **Qubits:** qubits in NMR implementation are nuclear spins 1/2 in a static magnetic field $\vec{B}_0$, which are non-degenerated two level systems, with an energy level splitting proportional to the intensity of the magnetic field. The two eigenstates, spin up $|\uparrow\rangle$ and spin down $|\downarrow\rangle$ are labeled as $|0\rangle$ and $|1\rangle$ in the context of quantum computing.

— **Quantum gates:** quantum gates correspond to manipulation of the spins, which is achievable in NMR by applying radiofrequency (RF) electromagnetic waves pulses and adjusting delay times between pulses applied.

— **Input:** the input state correspond to the Boltzmann distribution of spins at room temperature.

— **Read out:** it is possible to detect spin states with RF coil, from the free-induction decay signal (relaxation of spins after manipulations by RF pulses.

— **Coherence times:** in NMR, the coherence of spin superposition is rather long, easily up to several seconds.

So the realize a quantum computation with NMR, one has to consider a nuclear spin in a static $\vec{B}_0$ field. Let's note $\hat{\vec{I}}$ the nuclear spin operator of this nuclear spin and $\vec{B}_0 = B_0\vec{u}_z$ so that z is the quantification axis of this spin. Then, the hamiltonian of this nuclear spin is

$$\hat{H}_0 = -\hbar\gamma B_0 \hat{I}_Z = -\frac{\hbar\omega_0}{2}\hat{Z},$$

where ω_0 is the Larmor frequency: precession of the qubit around $\vec{u}_Z$ at Larmor frequency. Even without any

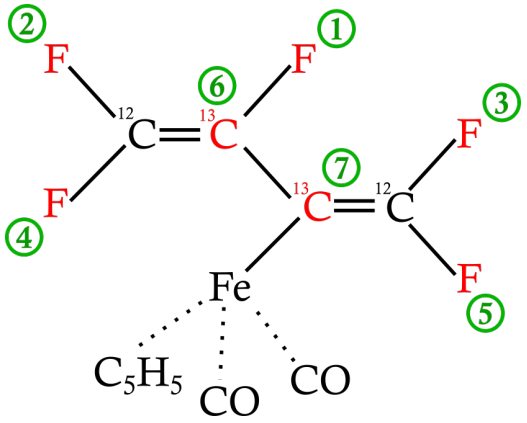
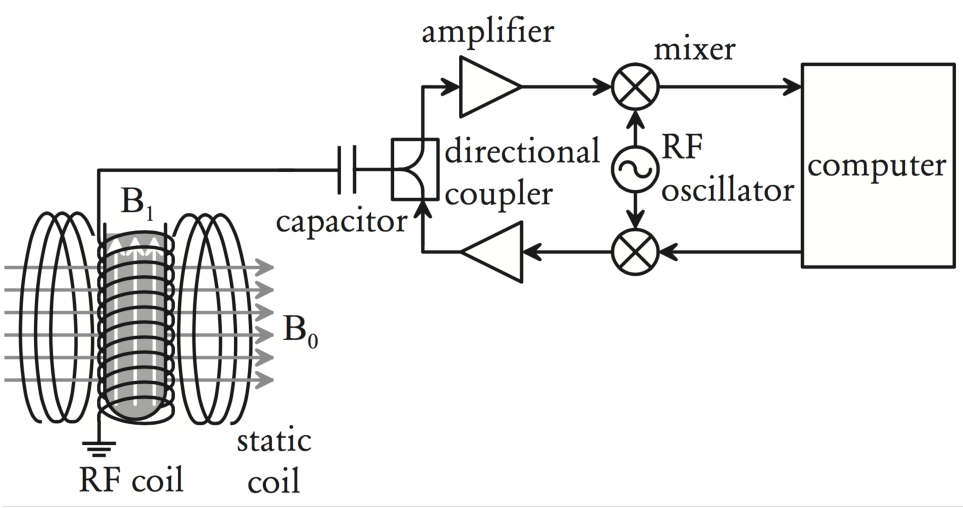


Fig. IV.3. Case of $C_4F_5Fe(CH)_5(CO)_2$ - perfluorobutadienyl iron complex [27]. Nuclei labeled in red are qubits (F and ^{13}C), numbered in green. $B_0 = 11.7$T.



I. L. Chuang et al., Proceedings of the Royal Society A 454, pp. 447-467 (1998).

Fig. IV.4. Experimental setup for NMR qubits implementation.

qubit-qubit coupling, the Larmor frequency depends on the atoms. If without coupling this frequency is only atomic species specific, those frequencies are shifted depending on the coupling with the chemical environment so that the hamiltonian of n spins might be rewritten as follow

$$\hat{H}_0 = -\sum_{j=1}^{n} \hbar(1-\tilde{\sigma}_i)\gamma_i B_0 \hat{I}_Z^i = -\sum_{i=1}^{n} \frac{\hbar}{2}\omega_0^i(1-\tilde{\sigma}_i),$$

with ω_0^i the Larmor frequency of the nuclei i without qubit/qubit coupling and $\tilde{\sigma}_i$ the chemical frequency shift due to coupling with neighborhooding spins. The chemical frequency shift $\tilde{\sigma}_i$ is used in analytical chemistry

Atom	ω_0^i
^{1}H	500 MHz
^{13}C	126 MHz
^{15}N	-51 MHz
^{19}F	470 MHz
^{31}P	202 MHz

Table IV.1 – Larmor frequency at 11.7 T for different atoms with 500 MHz equivalent to 25 mK.

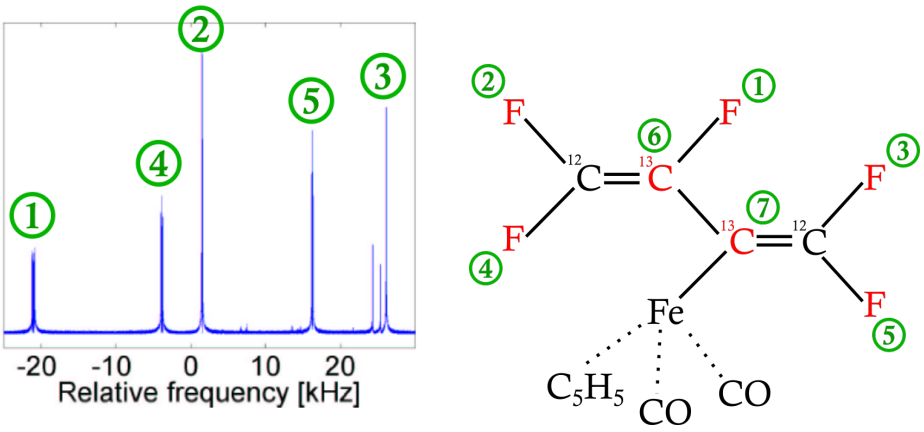


Fig. IV.5. Chemical shifts of the five F qubits of perfluorobutandienyl iron complex.

to determine the environment of an atom, while here it is used to address different qubits with different RF frequencies independently. Thus, in the case of perfluorobutandienyl iron complexe (Fig. IV.5), this molecule permits to have five resonance frequencies corresponding to five different qubits.

2.3. RF field interaction: single qubit rotation

Now one consider that the spins are in a static magnetic field and an oscillating radiofrequency magnetic field. In the rotating frame and the rotating wave approximation, the hamiltonian becomes

$$\hat{H}^{rot} = -\frac{\hbar\delta}{2}\hat{Z} - \frac{\hbar\omega_1}{2}\left(\cos\phi\hat{X} + \sin\phi\hat{Y}\right),$$

with $\delta = \omega_0 - \omega_{RF}$ the detuning between RF and Larmor frequency, ϕ the phase of the RF field and $\omega_1 = \gamma B_1$ where B_1 is the amplitude of the RF field. For n qubits of Larmor frequencies ω_0^i, gyromagnetic factor γ_i and $\omega_1^i = \gamma_i B_1$, frequency shift $\tilde{\sigma}_i$, one notes

$$\delta_i = \omega_0^i(1-\tilde{\sigma}_i) - \omega_{RF},$$

the detuning between RF and qubit i frequency, such that

$$\boxed{\hat{H}^{rot} = -\sum_{i=1}^{n}\frac{\hbar}{2}\delta_i\hat{Z}_i - \frac{\hbar}{2}\sum_{i=1}^{n}\left(\cos\phi\,\omega_1^i\hat{X}_i + \sin\phi\,\omega_1^i\hat{Y}_i\right)}.$$

Let's call T the time during which the RF field is applied to the system of n qubits. If T is long enough, each spin resonance is non overlapping with the others ones. So if ω_{RF} is not close enough to a corrected Larmor frequency $\omega_0^i(1-\tilde{\sigma}_i)$, the effect of the RF field in negligeable. If $\omega_{RF} \approx \omega_0^i(1-\tilde{\sigma}_i)$, only the qubit i will be affected and rotate on the Bloch sphere.

If qubits have different Larmor frequencies, it is possible to manipulate each qubit individually choosing the corresponding frequency.

2.4. Coupled spins

When spins of different atoms are coupled, the coupling interaction energy usually results in level splitting. Consequently, the Larmor frequency of a qubit, corrected by the chemical frequency shift $\tilde{\sigma}_i$, will depend on the state of neighborhooding qubits. In other words, the resonance RF frequency of a qubit will depends on the state of the neighborhooding qubits.

This phenomena is the key physical effect used to implement multiqubit gates.

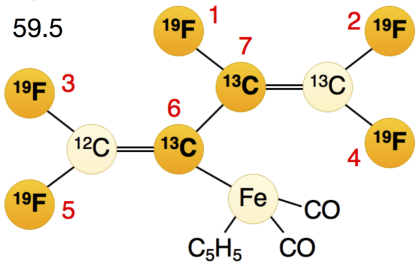
The coupling hamiltonian might be written as follow

$$\hat{H}_J = \hbar \sum_{i<j} 4 J_{ij} \hat{I}_Z^i \hat{I}_Z^j \Leftrightarrow \boxed{\hat{H}_J = \hbar \sum_{i<j} J_{ij} \hat{Z}^i \hat{Z}^j}.$$

For $J > 0$, one has a antiferromagnetic interaction, while for $J < 0$ one has a ferromagnetic interaction. In the case of perfluorobutadienyl iron complex, one has **5 resonances frequencies corresponding to 5 different qubits.**

i	$\omega_i/2\pi$	$T_{1,i}$	$T_{2,i}$	J_{7i}	J_{6i}	J_{5i}	J_{4i}	J_{3i}	J_{2i}
1	–22052.0	5.0	1.3	–221.0	37.7	6.6	–114.3	14.5	25.16
2	489.5	13.7	1.8	18.6	–3.9	2.5	79.9	3.9	
3	25088.3	3.0	2.5	1.0	–13.5	41.6	12.9		
4	–4918.7	10.0	1.7	54.1	–5.7	2.1			
5	15186.6	2.8	1.8	19.4	59.5				
6	–4519.1	45.4	2.0	68.9					
7	4244.3	31.6	2.0						

At B_0 = 11.7 T:
$\omega_{0,F}/2\pi$ = 470 MHz
$\omega_{0,C}/2\pi$ = 125 MHz
$[\omega_i/2\pi]$ = Hz, $[T]$ = s, $[J]$ = Hz



L. M. K. Vandersypen et al., Nature 414, 883 (2001)

Fig. IV.6. Coupling constants of the five F qubits of perfluorobutandienyl iron complex.

Orders of magnitude: in the case of perfluorobutadienyl iron complex at 11.7T
- — Larmor frequency of F-type qubit ~ 470 MHz ;
- — chemical frequency shift $\sim 10 - 20$ kHz;
- — typical RF strength $\omega_1 \sim 2\pi \times 100$ kHz ;
- — typical level splitting due to qubit/qubit coupling $J \sim$ few 100 Hz max.

$$\frac{100 \text{ Hz}}{470 \text{ MHz}} \sim 2 \cdot 10^{-7}, \quad \frac{100 \text{ Hz}}{10 \text{ kHz}} \sim 10^{-2}.$$

The ratio of the qubit level splitting due to interactions to the central Larmor frequency is of the order of $2\dot{1}0^{-7}$, requiring high stability of the RF frequency used. The ratio of the qubit level splitting due to interaction to the chemical frequency shift is of the order of 10^{-2}, so that one may unambiguously distinguish the shift due to qubit/qubit coupling from the chemical shift of different qubits.

2.5. Controlled-NOT gate in NMR

Let's consider two spins A and B as qubits. Let's consider a C-NOT gate that flips A when B is $\downarrow$, which corresponds to the truth table of Tab. IV.2.

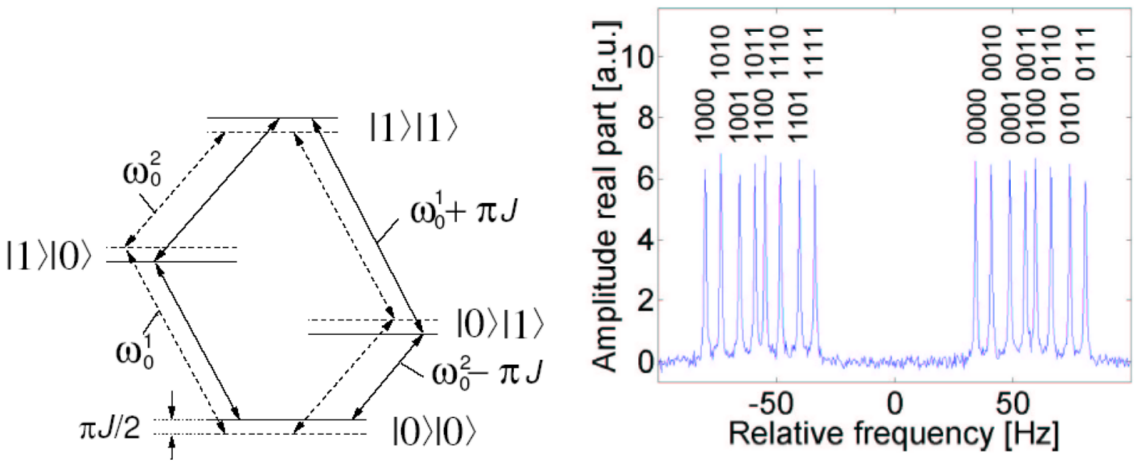


Fig. IV.7. Level scheme of uncoupled and coupled qubits system and NRM spectrum of coupled qubits.

Before	
A	*B*
↑	↑
↑	↓
↓	↑
↓	↓

After	
A	*B*
↑	↑
↓	↓
↓	↑
↑	↓

Table IV.2 – C-NOT gate with two spins A and B as qubits. States of spins before and after the C-NOT gate is applied.

Let's consider then the following sequence : apply $\hat{R}_{Y,A}\left(\frac{\pi}{2}\right)$, wait for a delay

$$\Delta t = \frac{\pi}{4J_{AB}},$$

and finally apply $\hat{R}_{X,A}\left(\frac{\pi}{2}\right)$. For clarity, it is more convenient to describe this sequence on the Bloch sphere of spin A in the rotating frame at the larmor frequency of A, ν_A (no coupling), as represented in Fig. IV.8.

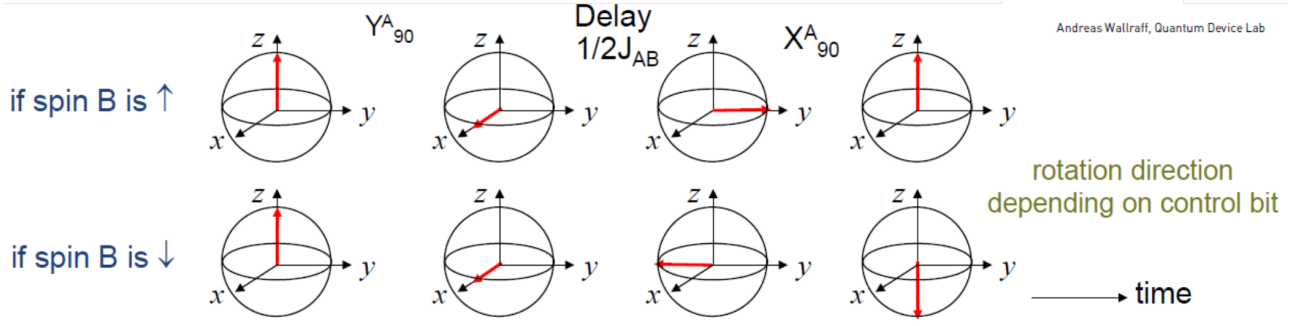


Fig. IV.8. C-NOT in the case of NMR implementation.

Initially, the vector on the Bloch sphere is oriented in the direction $+\vec{u}_z$, in the $|\uparrow\rangle$ state. After $\hat{R}_{Y,A}\left(\frac{\pi}{2}\right)$ rotation, the vector of spin A is oriented in the $+\vec{u}_x$ axis, on the equator of the sphere. If there was no coupling with spin B, the spin would remain steady in the $+\vec{u}_x$ direction, since one is in the rotating frame. Due to interaction with spin B, the Larmor frequency of spin A is shifted so that now A precess in the rotating frame, at a frequency given by the frequency shift that results from spin-spin interaction. **The rotation direction depends on the control qubit states, which is the key element of a C-NOT gate implementation with**

NMR technics. If spin B is in state $|\uparrow\rangle$, spin A precess with an angular frequency of J_{AB} around $+\vec{u}_z$, while if spin B is in state $|\uparrow\rangle$, spin A precess with an angular frequency of J_{AB} around $-\vec{u}_z$ (*i.e.* in the opposite direction). One considers free evolution of spin A with such an interaction for a delay time Δt such that $J_{AB}\Delta t = \pi/4$, so that the spin A will turn by $\pm\pi/4$ around $\vec{u}_z$ depending on the state of B. After this free evolution, the spin A will point in the direction $+\vec{u}_y$ if B is in $|\uparrow\rangle$ state, and in the direction $-\vec{u}_y$ if B is in $|\downarrow\rangle$ state. After a final rotation $\hat{R}_{X,A}\left(\frac{\pi}{2}\right)$ of $\pi/2$ along $\vec{u}_x$, the spin A will point in the direction $+\vec{u}_z$ if B is in $|\uparrow\rangle$ state, and in the direction $-\vec{u}_z$ if B is in $|\downarrow\rangle$ state. Then, the final state of A is changed only if B is in the down state $|\downarrow\rangle$, which is a key element of a conditional gate.

2.6. Read-out in NMR

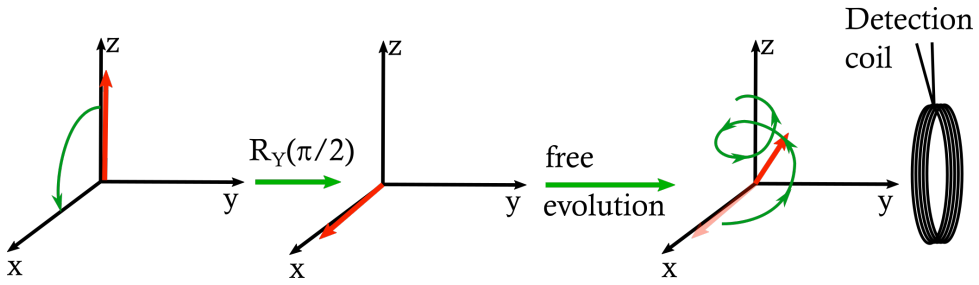


Fig. IV.9. Read-out of the qubit state in the case of NMR implementation : Bloch's sphere sequence.

The measurement of a qubit correspond to the measurement of the observable $\hat{Z}$ of the qubit, which correspond to the observable $\hat{S}_z$ of the spin state. Such a measurement is actually straightforward from technics develops initially in NMR methodology. Let's consider a spin which is initially in the $|\uparrow\rangle$ state, pointing in the $+\vec{u}_z$ direction. Then, apply a rotation $\hat{R}_Y\left(\frac{\pi}{2}\right)$ so the spin is aligned on the equator of the sphere. Afterwards, this spin will precess at Larmor frequency around $+\vec{u}_z$ axis, and at the same time relax towards the Boltmann distribution state (aligned with the $\vec{u}_z$ direction). To such a spin is associated a magnetic moment. Then, this free evolution after the initial rotation correspond to an oscillating dipole, that induce an oscillating magnetic field. Such an oscillating magnetic field is measurable by the mean of a small detection coil, that results in a induction voltage measured. Then, the induction signal measured is an image of the spin relaxation, as depicted in Fig. IV.9.

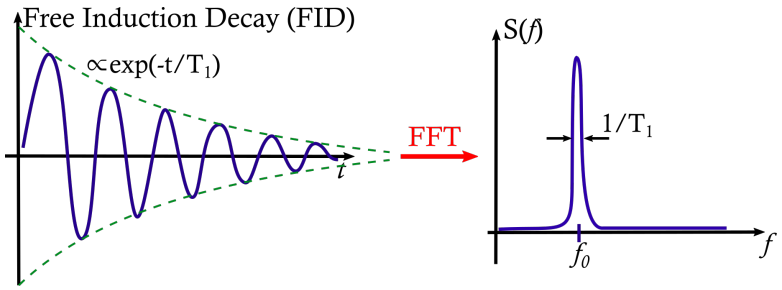


Fig. IV.10. Read-out of the qubit state in the case of NMR implementation : free induction decay signal (FID) and its FFT to recover spectrum.

The induction signal measured, related to free relaxation of spin toward the thermal equilibrium, is called *Free Induction Decay* (FID) signal. This signal contains two informations: the Larmor frequency of the free evoluating spin and it's relaxation time, called T_1 relaxation time, or *population decoherence time*. With typical frequencies in the radiofrequency range, it is easy to measure the time trace of it, and compute the fourier transform (FFT) to obtain the spectral density (both in amplitude and phase). For a single spin, one has a peak

centered at Larmor frequency, and a full width at half maximum (FWHM) corresponding to the inverse of the population coherence time $1/T_1$ (Fig. IV.10).

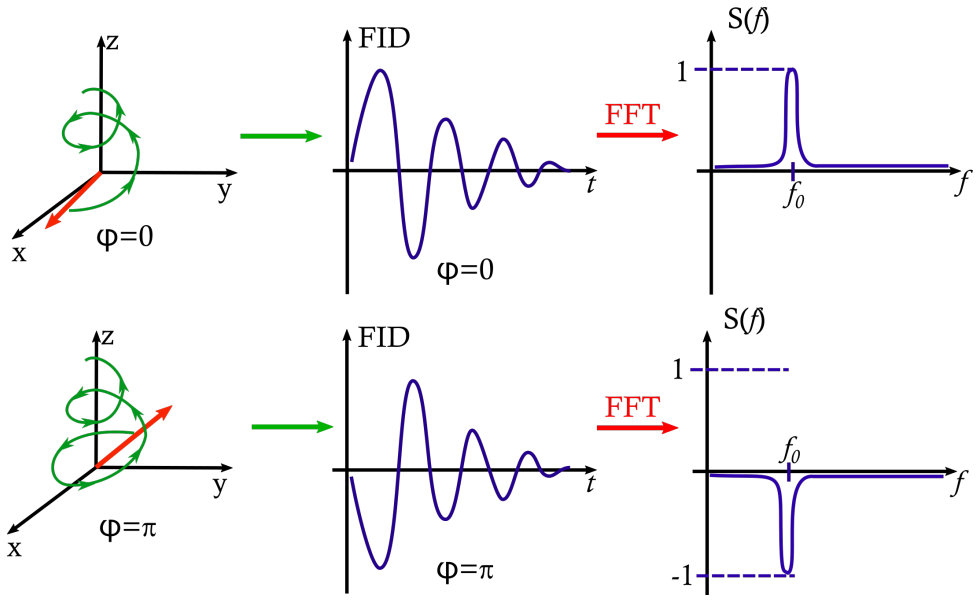


Fig. IV.11. Read-out of the qubit state in the case of NMR implementation : influence of the initial state on the FID signal and its FFT.

But if the initial state if the spin is $|\downarrow\rangle$, the spin while point in the $-\vec{u}_x$ direction after the initial rotation. This results in a π phase shift of the oscillation of the spin during the free evolution phase, as illustrated in Fig. IV.11. Thus, depending on the initial position of the equator of the Bloch sphere, the phase of the free induction decay will be different. Then, a π phase shift in a time domain signal results in a $e^{i\pi} = -1$ term that multiply the spectral density in the Fourier domain, so that the spectrum measured is of opposite sign if the initial state is $|\downarrow\rangle$ instead of $|\uparrow\rangle$.

The initial angle φ on the Bloch sphere is measured by the FID signal in NMR : **NMR is a phase sensitive detection method.** A qubit state is measured with a $\hat{R}_Y\left(\frac{\pi}{2}\right)$ pulse followed by relaxation FID signal measurement, which sign will permit to determine the state of the qubit, as illustrated in Fig. IV.12.

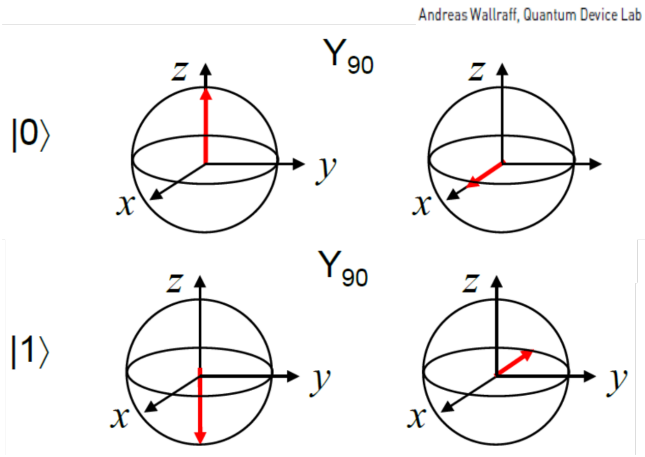


Fig. IV.12. Read-out of the qubit state in the case of NMR implementation.

For a $|0\rangle$ state, $\varphi = 0$ after the $\hat{R}_Y\left(\frac{\pi}{2}\right)$ pulse leading in a positive signal in the FFT of FID. For a $|1\rangle$ state, $\varphi = \pi$ after the $\hat{R}_Y\left(\frac{\pi}{2}\right)$ pulse leading in a negative signal in the FFT of FID.

Two coupled qubits in their four computational basis states have the following schematized Fig. IV.13 : **NMR detection signal (FFT of FID) depends on states.** This dependance is exploit to detect the state of the coupled qubits system, as illustrated in Fig. IV.14

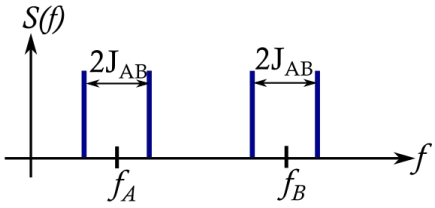


Fig. IV.13. Spectrum of two coupled qubits in their four computational basis states.

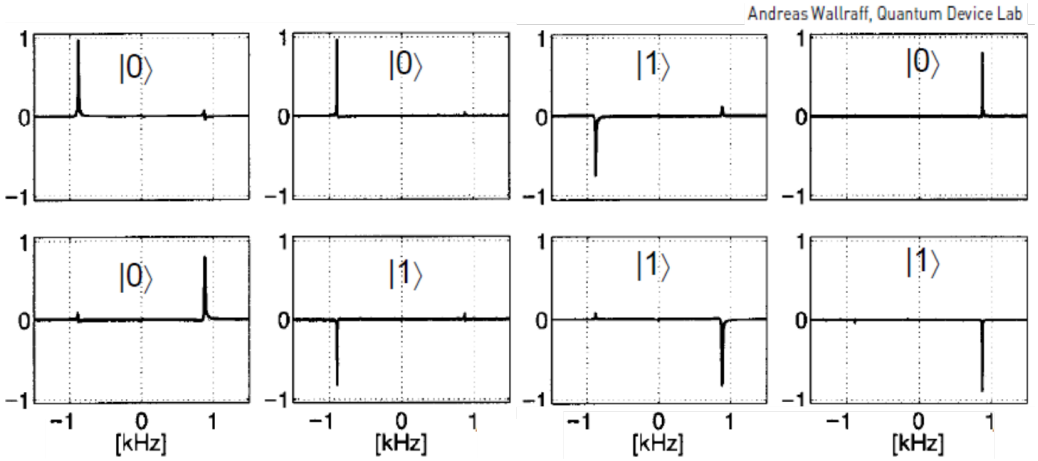


Fig. IV.14. Experimental signal of state read out in NRM implementation (figure from Andreas Walfraff Quantum Device Lab).

It's easier and faster experimentally to detect the sign of the spectral density $S(\nu)$ rather than the frequency shift J_{AB} which is only of few hundreds of Hz.

State detection is based on a sign (phase) measurement of the NMR signal $S(\nu)$.

2.7. Example of Shor's algorithm

In this section is exposed briefly the results obtained with a real sequence for factorization with perfluorobutadienyl iron complex [27], that has been used to demonstrat the ablility of such a system to factorize $15 = 3 \times 5$ in molecules by NMR with Shor's algorithm.

In Fig. IV.15 is represented the sequence used in such algorithm. It is based on a 7 qubit quantum register and this sequence is the circuit diagram of the algorithm in the context of quantum computing. Each red stick corresponds to a $\pi/2$ rotation, and blue sticks correspond to π rotation. This rotations are meanly used to manage coherence of the spin during the sequence. Green sticks correspond to Z rotation and C-gates are represented in black structures. Without going into the detail of the algorithm, this sequence provides an idea of the number of operation required to manipulate spins to obtain the result of a rather simple calculation such as factorizing 15.

In Fig. IV.16 is represented the spectrum of a given spin depending on states of other spin, illustrating the change in resonance frequency and in phase of the spectrum depending on states. This change in phase permits to measure the state of the corresponding spin. Experimental measurement are compared to theoretical predictions.

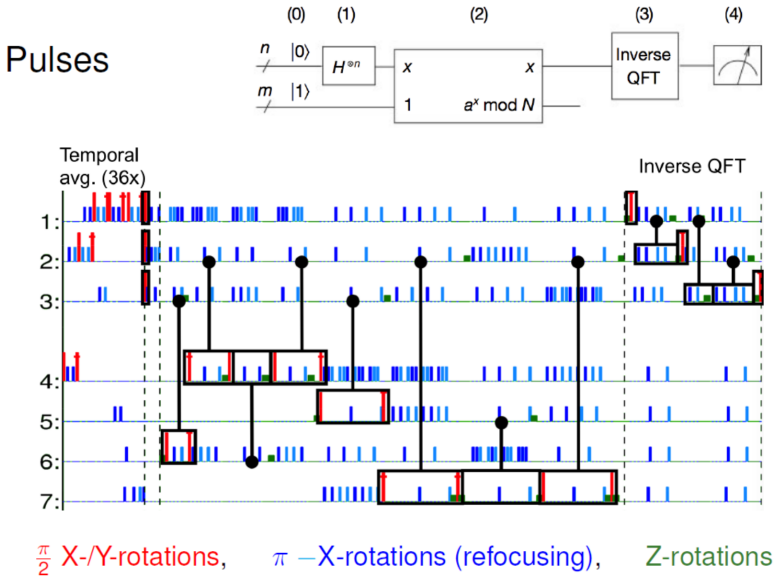


Fig. IV.15. Shor's algorithm: experimental sequence. Extracted from [27].

2.8. Quantum state tomography

If a quantum calculation output a qubit in a non-pure state $|0\rangle$ or $|1\rangle$, one may only perform a measurement of the observable $\hat{Z}$, which is a projective measurement that will result in either 0 or 1 as a result. Therefore, it is not possible to have access to such a state after a signal. If the quantum calculation is repeated several time, it is possible to have access the probability of being in state $|0\rangle$ ($P_{|0\rangle} = |\langle 0|\Psi\rangle|^2$) or in state $|1\rangle$ ($P_{|1\rangle} = |\langle 1|\Psi\rangle|^2$). But it is not enough to access to the full quantum state, especially the relative phase between $|0\rangle$ and $|1\rangle$. It is however possible to reconstruct a quantum state if it is possible to produce it several times. The principle consists in applying different rotations to look at the qubits from different angle, as illustrated Fig. IV.17. It is called **quantum state tomography**.

3. Molecules for quantum computing based on NMR techniques

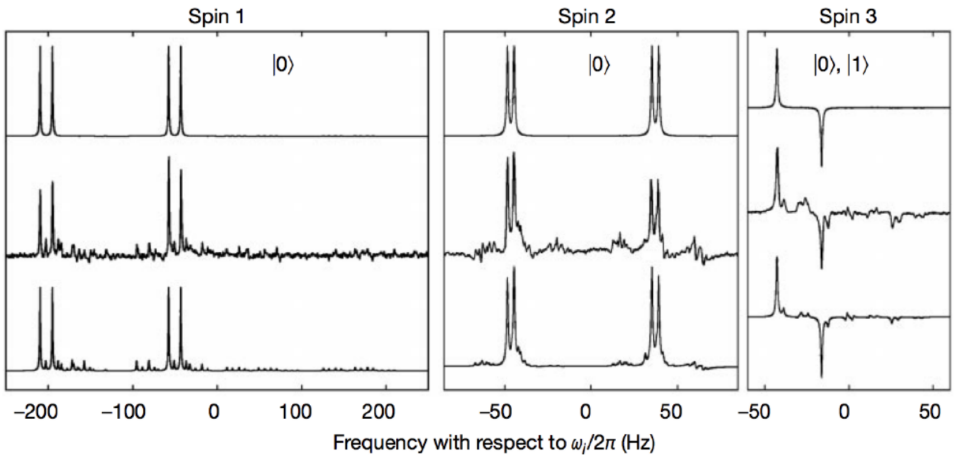
3.1. Properties of molecules

In order to develop a quantum computer with NMR in molecule, one should chose a molecule with the following properties to get good qubits

— spins 1/2 in order to have a two-level system equivalent to a qubit (^{1}H, ^{13}C, ^{19}F, ^{15}N,...) ;
— long T_1's and T_2's coherence time, so coherence between spin is long enough compared to the time required to manipulate spins according to the sequence of the algorithm considered ;
— heteronuclear molecules, and/or large chemical shifts (required to address different spin types independently) ;
— good J-coupling network ; such coupling will impact the interaction time required to achieve C-gates and consequently will affect to clock-speed of the calculation ;
— a molecule *"easy to use"*: stable, available, soluble,...

3.2. Examples of molecules used

Several molecules have been already used to implement different quantum algorithms. For example in molecules represented Fig. IV.18, where red atoms nuclei are used as qubit.



L. M. K. Vandersypen et al., Nature 414, 883 (2001)

Fig. IV.16. Extracted from [27].

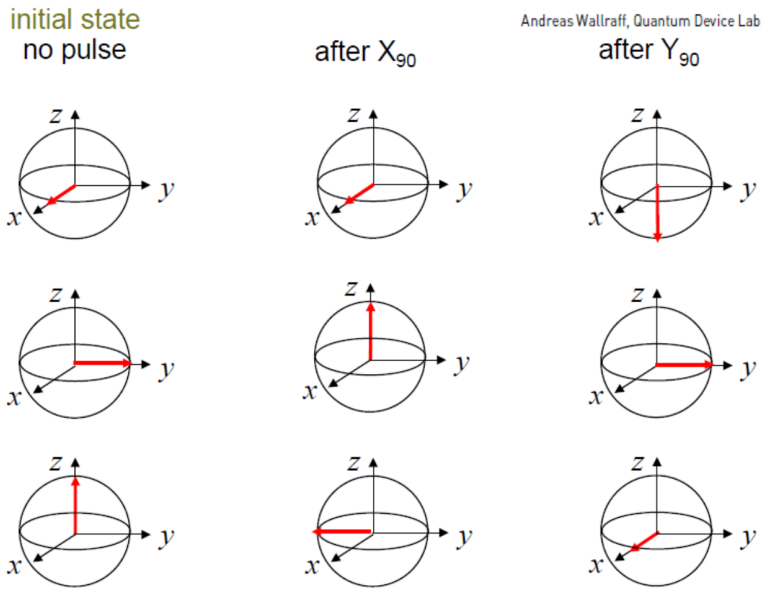


Fig. IV.17. Principle of quantum state tomography.

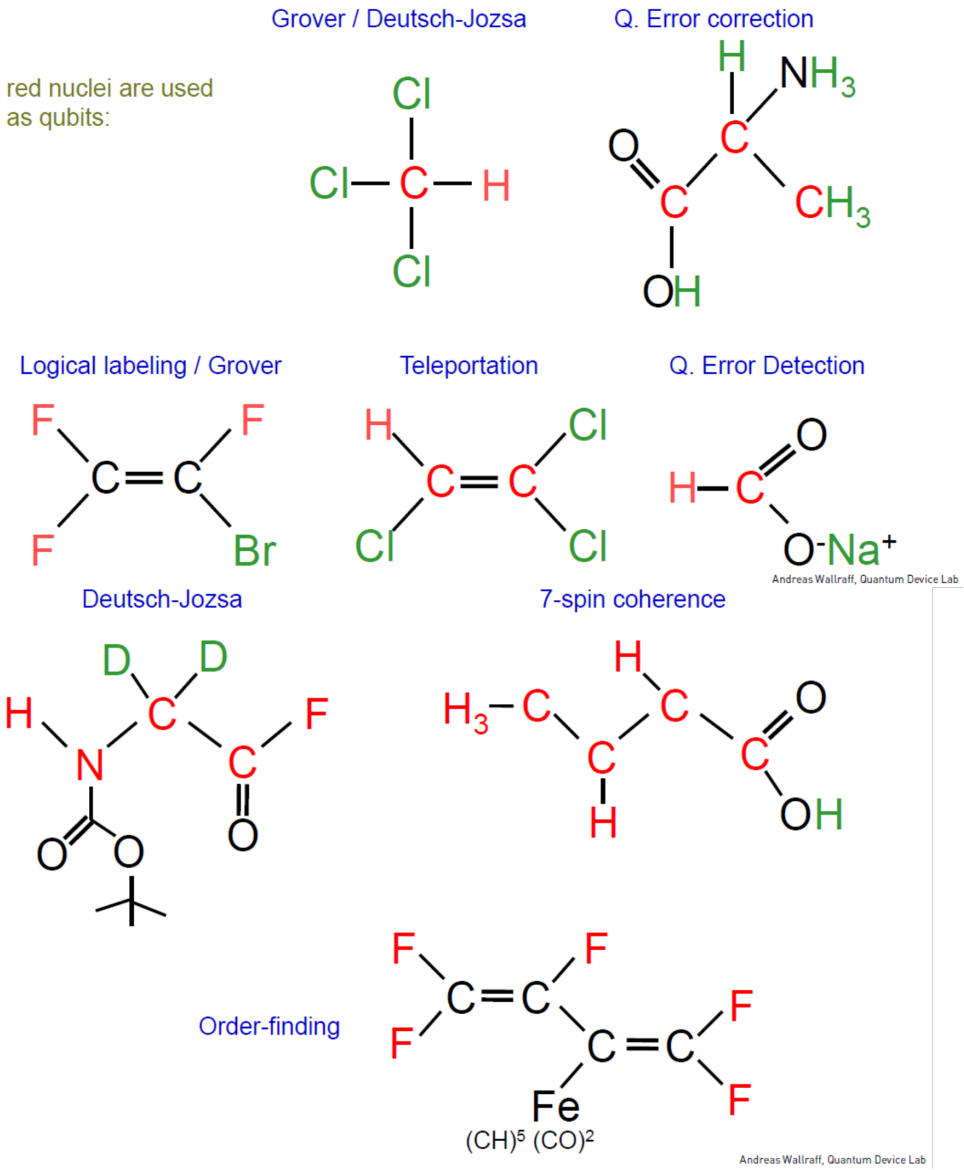


Fig. IV.18. Several molecules used for quantum algorithm implementation by NMR technics (from Andreas Walralf Quantum Device Lab, ETH Zürich, Switzerland).

Chapter V

Quantum algorithms

1. Introduction

There are a lot of quantum algorithms. The *quantum algorithm zoo* [1] cites more than 400 papers [2] on quantum algorithms. Mostly, they solve specific mathematical problems : factoring, matrix inversion,... We will focus on a few important algorithms / sub-routines: Grover's search, phase estimation, factoring, matrix inversion (HHL), hamiltonian simulation.

We might divide quantum algorithms in two categories

Polynomial speed up	Exponential speed up
Grover's search	Integer
Quantum walks	Matrix Inversion
Graph algorithms	Phase Estimation
Minimum finding	Quantum Fourier Transform

2. Bernstein-Vazirani algorithm

Berstein-Vazirani algorithm is a restricted version of Deutsch-Josa algorithm. Instead of distinguishing between two different classes of functions, it tries to learn a string encoded in a function. One is given an oracle implementing a function f

$$f:\ \{0,1\}^n \longrightarrow \{0,1\}.$$

It is given that $f(x)$ is a dot product between x and a secret string $s \in \{0,1\}^n$ modulo 2

$$f(x) = x \cdot s = x_1 \cdot s_1 + x_2 \cdot s_2 + \cdots + x_n \cdot s_n.$$

Bernstein-Vazirani algorithm aims at finding s.

Classicaly, it requires to evaluate n times the function $f(x)$, with $x = 2^k$, $k \in \{0,1,\cdots,n-1\}$.

$$f(1000\cdots00) = s_1,$$

$$f(0100\cdots00) = s_2,$$

$$\vdots$$

$$f(0000\cdots01) = s_n,$$

Thanks to Bernstein-Vazirani algorithm, only one query is needed with a quantum computer.

1. `https://quantumalgorithmzoo.org/`
2. 404 papers on October 29th, 2019

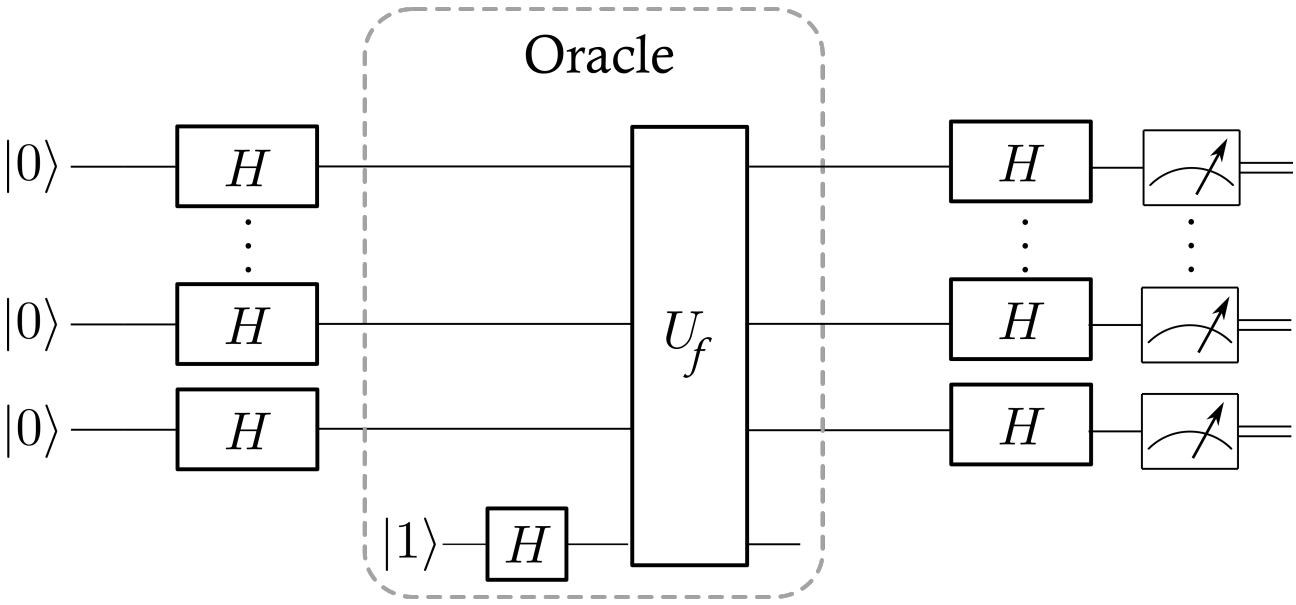


Fig. V.1. Circuit diagram of Bernstein-Vazirani algorithm.

Algorithm

1. Initialize a quantum register of n qubit in the state $|0\rangle^{\otimes n}$.
2. Apply a Hadamard gate to each qubit of the quantum register, providing the state

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n} |x\rangle.$$

3. Apply the Oracle to the previous superposed state to obtain the following state

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n} (-1)^{f(x)} |x\rangle.$$

4. Apply Hadamard gate on each qubit of the quantum register. If $s_i = 1$, it converts the state $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ to $|1\rangle$. If $s_i = 0$, it converts the state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ to $|0\rangle$.
5. To obtain s, the classical measurement on the $\{|0\rangle, |1\rangle\}$ basis provides the result.

The circuit diagram of Bernstein-Vazirani algorithm is proposed in Fig. V.1.

3. Grover's algorithm

3.1. Grover's problem - unstructured search

A simple example of a problem that fits into the query complexity model is **unstructured search on a set of N elements, in which only on element is marked**. In this problem, we are given a function

$$f: \{x_i, i \in [\![0, N-1]\!]\} \longrightarrow \{0,1\},$$

with the promise that it exists only one $p \in [\![0, N-1]\!]$ such that

$$f(x_p) = 1, \text{ and for } q \neq p,\ f(x_q) = 0.$$

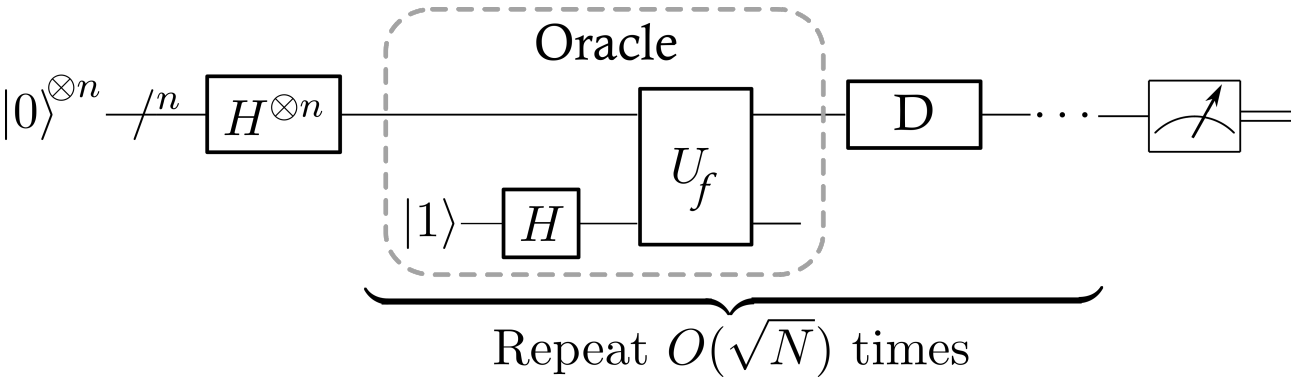


Fig. V.2. Circuit diagram of Grover's algorithm.

Then, x_p **is the "marked" element.** Our task is to output x_p, f being given by an Oracle. It is intuitively clear that the unstructured search problem requires about N queries the be solved classically. Let $\mathscr{A}$ be a classical algorithm which solves the unstructured search problem on a set of N elements with a failure probability $\lesssim 1/2$. Then, $\mathscr{A}$ makes $\mathscr{O}(N)$ queries in the worst case.

Grover (1997): *there is a quantum algorithm which solves the unstructured search problem using* $\mathscr{O}(\sqrt{N})$ *queries.*

For simplicity, we assume that $N = 2^n$, $n \in \mathbb{N}$ (this is not an essential restriction). Thus, we associate any element of $\{x_i\}$ with an n-bits string.

3.2. Grover's algorithm

We are given access to the following function

$$f:\ \{0,1\}^n \longrightarrow \{0,1\},$$

with the property that $f(x_p) = 1$ for a unique element x_p. We use a quantum circuit on n qubits with an initial state

$$|\psi_0\rangle) = |0\rangle^{\otimes n}.$$

Let $\hat{H}$ denote the Hadamard gate, and let $\hat{U}_0$ denote the n-qubit operation which inverts the phase of only $|0\rangle^{\otimes n}$

$$\begin{cases} \hat{U}_0|0\rangle^{\otimes n} = -|0\rangle^{\otimes n} \\ \hat{U}_0|x\rangle = |x\rangle \text{ for } |x\rangle \neq |0\rangle^{\otimes n} \end{cases} \tag{V.1}$$

Grover's algorithm:

1. Apply $\hat{H}^{\otimes n}$;
2. Repeat the following operation T times, for some T to be determined later
   a) Apply $\hat{U}_f$;
   b) Apply $\hat{D} = -\hat{H}^{\otimes n}\hat{U}_0\hat{H}^{\otimes n}$.
3. Measure all the qubits and output the results.

The overall operation performed, applied on the initial state $|0\rangle^{\otimes n}$, is unitary

$$\hat{D}^T\hat{H}^{\otimes n} = \left(-\hat{H}^{\otimes n}\hat{U}_0\hat{H}^{\otimes n}\right)^T\hat{H}^{\otimes n}.$$

3.3. Analysis of Grover's algorithm

To describe Grover's algorithm, we introduce unitary operators

$$\hat{I}_{|\psi\rangle} = \mathbb{I} - 2|\psi\rangle\langle\psi| \quad \text{and} \quad \hat{R}_{|\psi\rangle} = -\hat{I}_{|\psi\rangle} = 2|\psi\rangle\langle\psi| - \mathbb{I},$$

where $\mathbb{I}$ is the identity operator, and $|\psi\rangle$ is an arbitrary state. $\hat{I}_{|\psi\rangle}$ can be seen as an inversion around $|\psi\rangle$ operation, while $\hat{R}_{|\psi\rangle}$ can be seen as a reflection around $|\psi\rangle$ operation. An arbitrary state $|\phi\rangle$ can be expressed as

$$|\phi\rangle = \alpha|\psi\rangle + \beta|\psi^{\perp}\rangle,$$

with $(\alpha,\beta) \in \mathbb{C}$ and $|\psi^{\perp}\rangle$ belongs to the subspace perpendicular to $|\psi\rangle$

$$\langle\psi|\psi^{\perp}\rangle = 0.$$

Then

$$\hat{I}_{|\psi\rangle}|\phi\rangle = -\alpha|\psi\rangle + \beta|\psi^{\perp}\rangle.$$

$\hat{I}_{|\psi\rangle}$ has flipped the phase of the component corresponding to $|\psi\rangle$.

$\hat{R}_{|\psi\rangle}$ has the opposite effect

$$\hat{R}_{|\psi\rangle}|\phi\rangle = \alpha|\psi\rangle - \beta|\psi^{\perp}\rangle.$$

$\hat{U}_f$ is an Oracle such that

$$\hat{U}_f|x\rangle = (-1)^{f(x)}|x\rangle,$$

where one forgets the ancilla qubit required for unitary evolution. In the unstructured search problem with a marked element x_p

$$\boxed{\hat{U}_f = \hat{I}_{|x_p\rangle}}.$$

Furthermore,

$$\hat{H}^{\otimes n}\hat{U}_0\hat{H}^{\otimes n} = \hat{H}^{\otimes n}\left(\mathbb{I} - 2|0\rangle^{\otimes n}\langle 0|^{\otimes n}\right)\hat{H}^{\otimes n} = \mathbb{I} - 2\hat{H}^{\otimes n}|0\rangle^{\otimes n}\langle 0|^{\otimes n}\hat{H}^{\otimes n}.$$

Introducing the $|+\rangle$ state defined as follow

$$|+\rangle = \hat{H}^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle,$$

one obtains immediately $\boxed{\hat{D} = -\hat{I}_{|+\rangle}}$. After T iterations, the final state $|\psi_f\rangle$ measured is the following

$$\begin{aligned} |\psi_f\rangle &= \left(\hat{D}\hat{U}_f\right)^T \hat{H}^{\otimes n}|0\rangle^{\otimes n} = \left(\hat{D}\hat{U}_f\right)^T|+\rangle, \\ &= \left(-\hat{I}_{|+\rangle}\hat{I}_{|x_p\rangle}\right)^T|+\rangle = \left(-\hat{R}_{|+\rangle}\hat{R}_{|x_p\rangle}\right)^T|+\rangle, \end{aligned}$$

$$\boxed{|\psi_f\rangle = \left(-\hat{R}_{|+\rangle}\hat{R}_{|x_p\rangle}\right)^T|+\rangle}.$$

Properties:

1. For any states $|\psi\rangle$, $|\phi\rangle$, and any state $|\xi\rangle$ in the plan defined by $|\psi\rangle$ and $|\phi\rangle$, the states $\hat{R}_{|\psi\rangle}|\xi\rangle$ and $\hat{R}_{|\phi\rangle}|\xi\rangle$ remain in this plan.
2. For two orthogonal states, $\hat{R}_{|\phi^{\perp}\rangle} = -\hat{R}_{|\phi\rangle}$.
   Demonstration:

$$\begin{aligned} -\hat{R}_{|\phi\rangle}\left(\alpha|\psi\rangle + \beta|\psi^{\perp}\rangle\right) &= -\alpha|\psi\rangle + \beta|\psi^{\perp}\rangle \\ &= \hat{R}_{|\phi^{\perp}\rangle}\left(\alpha|\psi\rangle + \beta|\psi^{\perp}\rangle\right) \end{aligned}$$

3. If $|\xi\rangle$ is in the plan defined by two orthogonal states $|\phi\rangle$ and $|\phi^\perp\rangle$

$$\hat{R}_{|\phi\rangle}|\phi\rangle = \langle\psi|\xi\rangle|\phi\rangle - \langle\psi^\perp|\xi\rangle|\phi^\perp\rangle.$$

Demonstration is straightforward.

Consequently

$$\boxed{|\psi_f\rangle = \left(\hat{R}_{|+^\perp\rangle}\hat{R}_{|x_p\rangle}\right)^T |+\rangle}.$$

Grover's algorithm is based on successive rotations around $|x_p\rangle$ and $|+^\perp\rangle$, starting from the initial states superposition $|+\rangle$.

3.4. Geometrical interpretation of Grover's algorithm

After each iteration, $|\xi\rangle$ moves closer to $|x_p\rangle$ (see Fig V.3). In fact, the composition of two reflections around $|x_p\rangle$ and $|+^\perp\rangle$ is a rotation. Let not θ the angle between $|\xi\rangle$ and $|x_p\rangle$, and γ the angle between $|x_p\rangle$ and $|+^\perp\rangle$. After $\hat{R}_{|x_p\rangle}$, $|\xi\rangle$ rotates by an angle 2θ anticlockwise. After $\hat{R}_{|+^\perp\rangle}$, $|\xi\rangle$ rotates by an angle $2(\theta-\gamma)$ clockwise. Thus, after $\hat{R}_{|+^\perp\rangle}\hat{R}_{|x_p\rangle}$, $|\xi\rangle$ rotates by an angle of

$$\Delta\theta = 2\theta - 2(\theta-\gamma) = 2\gamma.$$

$$\boxed{\Delta\theta = 2\gamma}.$$

After each iteration, the state has rotates within the plane defined by $|x_p\rangle$ and $|+^\perp\rangle$ by an angle of 2γ.

3.5. Number of iterations

The iteration has to be stopped when $|\xi\rangle$ is close as much as possible to $|x_p\rangle$. We start with $|\xi\rangle = |+\rangle$, so the initial angle between $|\xi\rangle$ and $|x_p\rangle$ is $\frac{\pi}{2}-\gamma$. We can calculate γ as follow

$$\begin{cases} \cos\gamma = \langle x_p|+^\perp\rangle, \\ \sin\gamma = \langle x_p|+\rangle = \frac{1}{\sqrt{N}}, \end{cases} \tag{V.2}$$

because

$$|+\rangle = \frac{1}{\sqrt{N}}\sum_{x\in\{0,1\}^n}|x\rangle.$$

For large N,

$$\sin\gamma \approx \gamma \approx \frac{1}{\sqrt{N}}.$$

So the number of iterations M required to move from an angle $\frac{\pi}{2}-\gamma$ down to approximately 0 is

$$M \approx \frac{\frac{\pi}{2}-\gamma}{2\gamma} = \frac{\pi}{4\gamma} - \frac{1}{2} \approx \frac{\pi}{4}\sqrt{N} - \frac{1}{2}.$$

So in the limit where $N \ggg 1$,

$$\boxed{M \approx \frac{\pi}{4}\sqrt{N}}.$$

The number of iteration with a quantum algorithm scales as $\sqrt{N}$ while with a classical algorithm, it scales as N. After T iterations, the angle between $|\xi\rangle$ and $|x_p\rangle$ is

$$\gamma_T = \frac{\pi}{2} - (2T+1)\arcsin\left(\frac{1}{\sqrt{N}}\right),$$

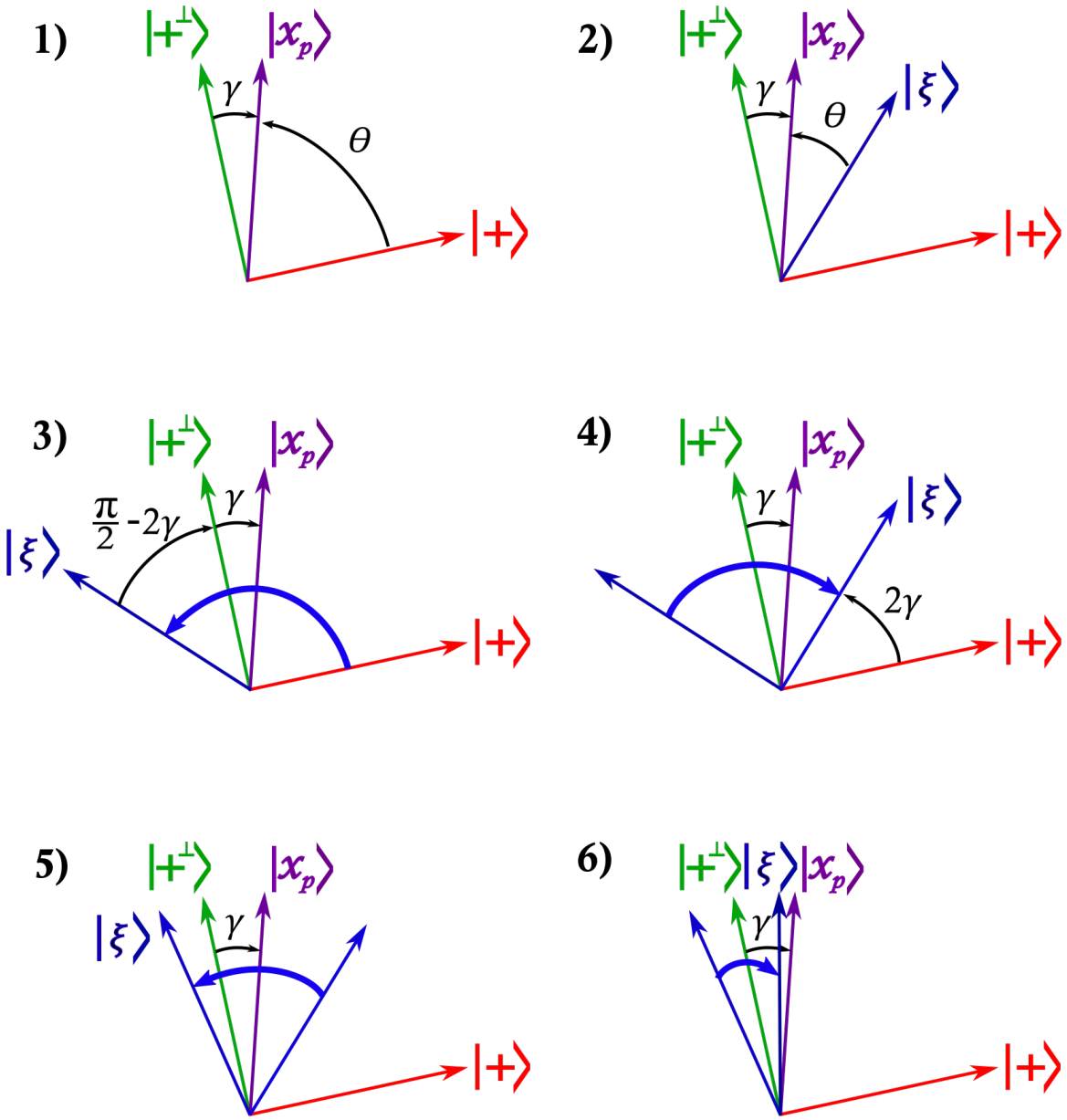


Fig. V.3. Geometrical interpretation of Grover's algorithm.

so the probability of obtaining the outcome $|x_p\rangle$ when we measure it is precisely

$$|\langle\xi\,|x_p\rangle|^2 = \cos^2\gamma_T = \sin^2\left((2T+1)\arcsin\left(\frac{1}{\sqrt{N}}\right)\right).$$

Maximising this by taking T as the nearest integrer to

$$\frac{\pi}{4\arcsin\left(\frac{1}{\sqrt{N}}\right)} - \frac{1}{2} = \frac{\pi}{4}\sqrt{N} - \frac{1}{2} - \mathscr{O}\left(\frac{1}{N}\right).$$

We have access to x_p with a probability $1-\mathscr{O}\left(\frac{1}{N}\right)$ using $\mathscr{O}\left(\sqrt{N}\right)$ queries (for small x, $\arcsin x \approx a + \mathscr{O}\left(x^3\right)$).

Remark:
the optimum number of iteration is independent of x_p.

A particular nice case, where we can determine an exact solution for T, is for $N=4$. Indeed,

$$\arcsin\frac{1}{2} = \frac{\pi}{6},$$

so if we plug in $T=1$, the probability of getting x_p at the outcome is $\sin^2\frac{\pi}{2} = 1$. So we get the right answer only after 1 query for 4 possibilities of x_p !

4. Grover's algorithm in the case of multiple marked elements

4.1. Number of marked elements known

Grover's algorithm can also be used when there are $M>1$ marked elements. In this setting, the operator $\hat{U}_f$ inverts the phase of inputs elements $x\in S$, for S an unknown subset of $\{0,1\}^n$, and $\mathrm{Card}(S)=M$. $\hat{U}_f$ is still related to a reflection operator, but now an inversion around a M-dimensional subspace

$$\hat{U}_f + \mathbb{I} - 2\hat{\Pi}_S,$$

where

$$\hat{\Pi}_S = \sum_{x\in S} |S\rangle\langle S|,$$

is the projector on the subspace generated by S. Let's define the state $|S\rangle$ as follow

$$\boxed{|S\rangle = \frac{1}{\sqrt{M}}\sum_{x\in S}|x\rangle}.$$

Then,

$$\begin{aligned}\hat{I}_{|S\rangle} &= (\mathbb{I} - 2|S\rangle\langle S|)\,|+\rangle \\ &= |+\rangle - 2\left(\frac{1}{M}\sum_{x,y\in S}|x\rangle\langle y|\right)\left(\frac{1}{\sqrt{N}}\sum_{x\in\{0,1\}^n}|x\rangle\right).\end{aligned}$$

So

$$\hat{I}_{|S\rangle} = |+\rangle - \frac{2}{\sqrt{N}}\sum_{x\in S}|x\rangle = \left(\mathbb{I} - 2\hat{\Pi}_S\right)|+\rangle = \hat{U}_f|+\rangle.$$

Similarly

$$\hat{I}_{|S\rangle}|S\rangle = -|S\rangle = \left(\mathbb{I} - 2\hat{\Pi}_S\right)|S\rangle = \hat{U}_f|S\rangle.$$

Then, $\hat{U}_f$ operation behaves like a reflection around $|S\rangle$ operator for any states in the subspace spanned by $|+\rangle$ and $|S\rangle$. The whole of the previous analysis goes through, except that now the angle γ moved at each step satisfies

$$\boxed{\sin\gamma = \langle S|+\rangle = \sqrt{\frac{M}{N}}}.$$

Thus, after T iterations, we have

$$|\langle\xi|S\rangle|^2 = \cos^2\gamma_T = \sin^2\left((2T+1)\arcsin\sqrt{\frac{M}{N}}\right).$$

To obtain an overlap with $|S\rangle$ close to 1, it requires T iterations with

$$\boxed{T \approx \frac{\pi}{4}\sqrt{\frac{N}{M}}}.$$

At the end of the algorithm, one get an element of the subset S at the measurement (a uniformly random distribution of elements of S) with a probability $|\langle\xi|S\rangle|^2$. For $M = \frac{N}{4}$, we again measure an element of S with certainty using only one query.

4.2. Number of marked elements unknown

Now the number of marked elements is not known (noted M'). In that case, one first runs the algorithm assuming there is only 1 marked element. If it fails, try again assuming there are 2 marked elements. Then 4, 8, etc... The total number of queries used is roughly

$$\sum_{k=0}^{\log_2 N} \frac{\pi}{4}\sqrt{\frac{N}{2^k}} = \frac{\pi}{4}\sqrt{N}\sum_{k=0}^{\log_2 N} 2^{-k/2} = \mathcal{O}\left(\sqrt{N}\right).$$

If the number of marked elements is M', at least one of the iterations must choose a guess M for M' such that

$$\frac{M'}{2} \leq M \leq 2M'.$$

This corresponds to a value of T which is within a factor of about $\sqrt{2}$ of the optimal value $T' \approx \frac{\pi}{4}\sqrt{\frac{N}{M'}}$. Since

$$(2T'+1)\arcsin\sqrt{\frac{M'}{N}} = \frac{\pi}{2} + \mathcal{O}\left(\sqrt{\frac{M'}{N}}\right),$$

then

$$\begin{aligned}\sin^2\left((2T+1)\arcsin\sqrt{\frac{M'}{N}}\right) &= \sin^2\left(\tfrac{2T+1}{2T'+1}(2T'+1)\arcsin\sqrt{\tfrac{M'}{N}}\right)\\ &= \sin^2\left(\tfrac{2T+1}{2T'+1}\left(\tfrac{\pi}{2} + \mathcal{O}\left(\sqrt{\tfrac{M'}{N}}\right)\right)\right),\end{aligned}$$

which is lower-bounded by a strictly positive constant of M is small with respect to N. Repeating the whole algorithm $\mathcal{O}(1)$ times, and checking each time whether the returned element is marked, allos to achieve an arbitrary high success probability.

This algorithm might still have a high probability of failing in the case where $M = \mathcal{O}(N)$. To find a marked element in this case, we can just sample $\mathcal{O}(1)$ random values of $f(x)$ classically ; we will find a marked element with high probability.

4.3. Amplitude amplification

The idea of Grover's algorithm might be generalized to an algorithm for finding heuristic solutions to any problems. This algorithm is known as **amplitude amplification**. Imagine we have $N = 2^n$ possible solutions, of which a subset S are "good", and we would like to find a good solution. As well as having access to a "checking" algorithm f as before, where $f(x) = 1$ if and only if x is marked, we now have access to a "guessing" algorithm $\hat{\mathscr{A}}$, which has the job of producing potential solution to the problem. It performs the following map

$$\hat{\mathscr{A}}|0\rangle^{\otimes n} = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle,$$

with $\alpha_x \in \mathbb{C}$. After applying $\hat{\mathscr{A}}$, the probability that we would obtain a good solution after measurement is

$$p = \sum_{x\in S} |\alpha_x|^2.$$

We may consider $\hat{\mathscr{A}}$ as an heuristic try for output of a good solution. We can use f afterwards to check whether a claimed solution is actually good. If we repeated the algorithm $\hat{\mathscr{A}}$ until we got a good solution, the expected number of trials we would need is $\mathscr{O}\left(\frac{1}{p}\right)$.

Amplitude amplification algorithm:
We are given access to $\hat{\mathscr{A}}$ and $\hat{U}_f$.

1. Apply $\hat{\mathscr{A}}$ to initial state $|0\rangle^{\otimes n}$.
2. Repeat the following operations T times, for some T to be determined
   a) Apply $\hat{U}_f$.
   b) Apply $-\hat{\mathscr{A}}\hat{U}_0\hat{\mathscr{A}}^{-1}$.
3. Measure all the qubits and output the result.

Let introduce

$$|\psi\rangle = \hat{\mathscr{A}}|0\rangle^{\otimes n},$$

and

$$|G\rangle = \frac{\hat{\Pi}_S|\psi\rangle}{\|\hat{\Pi}_S|\psi\rangle\|}, \quad \text{with} \quad \hat{\Pi}_S = \sum_{x\in S} |x\rangle\langle x|.$$

The previous analysis is still valid, replacing $|+\rangle$ with $|\psi\rangle$ and $|S\rangle$ with $|G\rangle$. The first operation applied is equivalent to $\hat{I}_{|G\rangle}$ and the second is equivalent to $-\hat{I}_{|\psi\rangle}$. We start with the state $|\psi\rangle$ and rotate it toward $|G\rangle$. The angle γ moved at each step is such that

$$\boxed{\sin\gamma = \langle\psi|G\rangle = \|\hat{\Pi}_S|\psi\rangle\| = \sqrt{p}},$$

so the number of iterations required to move from $|\psi\rangle$ to $|G\rangle$ is $\mathscr{O}\left(\frac{1}{\sqrt{p}}\right)$, which is a quadratic improvement.

5. Phase estimation

5.1. Quantum Fourier Transformation

We now introduce an important unitary transformation which is used in a number of different contexts in quantum information theory

the quantum Fourier transform (QFT) over $\mathbb{Z}_N$,

where $\mathbb{Z}_N$ is the ensemble of integer modulo N. QFT might be seen as a generalization of the Hadamard gate, which has the following map

$$\hat{H}^{\otimes n} = \frac{1}{\sqrt{2^n}} \left(|0\rangle\langle 0| + |1\rangle\langle 0| + |0\rangle\langle 1| - |1\rangle\langle 1| \right)^{\otimes n}.$$

The QFT map is the following

$$\boxed{\hat{Q}_N |x\rangle = \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega_N^{x \cdot y} |y\rangle},$$

where $\omega_N = e^{\frac{2i\pi}{N}}$, but $x \cdot y$ is the product of x and y as integer of $\mathbb{Z}_N$.

Exemple:

$$Q_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad Q_3 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 1 \\ 1 & e^{\frac{2i\pi}{3}} & e^{\frac{-2i\pi}{3}} \\ 1 & e^{\frac{-2i\pi}{3}} & e^{\frac{2i\pi}{3}} \end{pmatrix},$$

$$Q_4 = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}.$$

Note that the QFT is unitary.

Demonstration: Let consider the inner product of rows x and z

$$\frac{1}{N} \sum_{y \in \mathbb{Z}_N} \left(\omega_N^{x \cdot y} \right)^* \omega_N^{z \cdot y} = \frac{1}{N} \sum_{y \in \mathbb{Z}_N} \omega_N^{(z-x) \cdot y}.$$

Or

$$\sum_{k=0}^{r-1} x^k = \begin{cases} \frac{1-x^r}{1-x} & \text{if} \quad x \neq 1, \\ r & \text{if} \quad x = 1. \end{cases} \tag{V.3}$$

Given that $\omega_N^N = 1$, the inner product is then 0 if $z \neq x$, and 1 otherwise ($z = x$). More generally, for any integer j,

$$\frac{1}{N} \sum_{y \in \mathbb{Z}_N} \omega_N^{j \cdot y} = \begin{cases} 0 & \text{if} \quad j \neq 0\,[N], \\ 1 & \text{if} \quad j = 0\,[N]. \end{cases} \tag{V.4}$$

Then the QFT is unitary.

The QFT is a similar transformation than the Discrete Fourier Transformation (DFT) used for classical computation and signal processing, up to the non standard normalization of $1/\sqrt{N}$.

5.2. Periodicity determination with QFT

Let consider a function

$$f : \mathbb{Z}_N \longrightarrow \mathbb{Z}_M,$$

for $(N, M) \in \mathbb{N}^2$ such that

1. f is periodic: there is a r such that

$$\forall x \in \mathbb{Z}_N, \; f(x+r) = f(x),$$

2. f is one-to-one on each period

$$\forall (x, y) \in \mathbb{Z}_N \text{ such that } |x-y| < r, \; f(x) \neq f(y).$$

The goal is to determine r.

The periodicity determination algorithm is the following. We start in the state $|0\rangle^{\otimes N}|0\rangle^{\otimes M}$.

1. Apply $\hat{Q}_N$ to the first register.
2. Apply $\hat{O}_f$ to the two registers (the Oracle).
3. Measure the second register.
4. Apply $\hat{Q}_N$ to the first register.
5. Measure the first register ; let the answer be k.
6. Simplify the fraction $\frac{k}{N}$ as far as possible and return the denominator.

$$|0\rangle^{\otimes N}|0\rangle^{\otimes M} \xrightarrow{1)} \frac{1}{\sqrt{N}} \sum_{x\in\mathbb{Z}_N} |x\rangle|0\rangle^{\otimes M} \xrightarrow{2)} \frac{1}{\sqrt{N}} \sum_{x\in\mathbb{Z}_N} |x\rangle|f(x)\rangle.$$

When the second register is measured, we receive an answer, say z. Since f is periodic and one-to-one,

$$\exists x_0 \text{ such that } f(x_0) = z.$$

Consequently

$$\forall x \in \mathbb{Z}_N \text{ such that } f(x_0) = z, \exists j \in \mathbb{Z} \text{ such that } x = x_0 + jr.$$

The state collapses then to something of the following form

$$\sqrt{\frac{r}{N}} \sum_{j=0}^{\frac{N}{r}-1} |x_0 + jr\rangle,$$

which means there is N/r states in a period. After we apply the QFT, we get the state

$$\frac{\sqrt{r}}{N} \sum_{j=0}^{\frac{N}{r}-1} \left(\sum_{y\in\mathbb{Z}_N} \omega_N^{y\cdot(x_0+jr)} |y\rangle \right) = \frac{\sqrt{r}}{N} \sum_{y\in\mathbb{Z}_N} \omega_N^{y\cdot x_0} \left(\sum_{j=0}^{\frac{N}{r}-1} \omega_N^{j\cdot r\cdot y} \right) |y\rangle$$

Observe that, as r divides N,

$$\omega_N^r = e^{\frac{2i\pi r}{N}} = \omega_{\frac{N}{r}}.$$

This state is then equivalent to

$$\frac{\sqrt{r}}{N} \sum_{y\in\mathbb{Z}_N} \omega_N^{y\cdot x_0} \left(\sum_{j=0}^{\frac{N}{r}-1} \omega_{\frac{N}{r}}^{j\cdot y} \right) |y\rangle$$

$$\sum_{j=0}^{\frac{N}{r}-1} \omega_{\frac{N}{r}}^{j\cdot y} = 0 \quad \text{unless } y \equiv 0 \left[\frac{N}{r}\right], \text{ in other words, if } y = l\frac{N}{r}, l \in \mathbb{Z}.$$

This state might be rewrite as follow

$$\frac{1}{\sqrt{r}} \sum_{l=0}^{r-1} \omega_N^{l\cdot x_0 \cdot \frac{N}{r}} |l\frac{N}{r}\rangle.$$

When me perform the final measurement, we receive an outcome

$$k = \frac{l_0 N}{r},$$

for some l_0 picked uniformly at random from $\{0, \cdots, r-1\}$ so

$$\frac{k}{N} = \frac{l_0}{r}.$$

If l_0 is coprime to r, we could cancel the fraction $\frac{k}{N}$ and output the denominator. For a integer, b picked-up uniformly at random from 0 to an the probability that b is coprime to a is

$$\mathscr{O}\left(\frac{1}{\log(\log a)}\right).$$

The, if we repeat the procedure $\mathscr{O}(\log(\log r)) = \mathscr{O}(\log(\log N))$ times, we are likely to find the period r. We have a probabilistic procedure which succeeds with probability p ; the probability that it fails every time over R repetitions is exactly

$$(1-p)^R \leq e^{-pR},$$

so it suffices to take $R = \mathscr{O}\left(\frac{1}{p}\right)$ to achieve $\sim 99\%$ success probability. Each time the algorithm returns a claimed period, we can check whether it is really a period of the function using two additional queries of the Oracle. Each use of the quantum algorithm therefore makes 3 queries of $\hat{O}_f$ so it makes $\mathscr{O}(\log(\log N))$ queries in total.

5.3. Phase estimation

Phase estimation is an important quantum computing primitive routine. Often used as an ingredient of more complex algorithms:

— integer factorisation ;
— matrix inversion ;
— quantum counting ;
— quantum walks.

Let consider an operator $\hat{A}$, of eigenvectors $\vec{x}$ and eigenvalues λ

$$\hat{A}\vec{x} = \lambda\vec{x}.$$

In the case of a unitary matrix

$$\hat{U}|x\rangle = e^{2i\pi\theta}|x\rangle,$$

with $|x\rangle$ an eigenvector and θ the phase of the eigenvalue $e^{2i\pi\theta}$.

Phase estimation algorithm: given $\hat{U}$ and $|x\rangle$, estimate θ.

The circuit diagram implementation of the phase estimation algorithm is described Fig. **??**

6. Shor's algorithm

6.1. Factoring

Shor's algorithm is the most famous application of quantum computers [15]. It consists in solving the following factorization problem

$$N = a \times b \longrightarrow \text{find } a \text{ and } b \text{ given } N,$$

where a and b are prime numbers.

Exemple: the recommended key size for RSA is 2048 bits.
— Best known classical algorithm $\sim$ 1 billion years.
— Shor's algorithm $\sim$ 100 seconds !!!! (QC at 1 GHz).

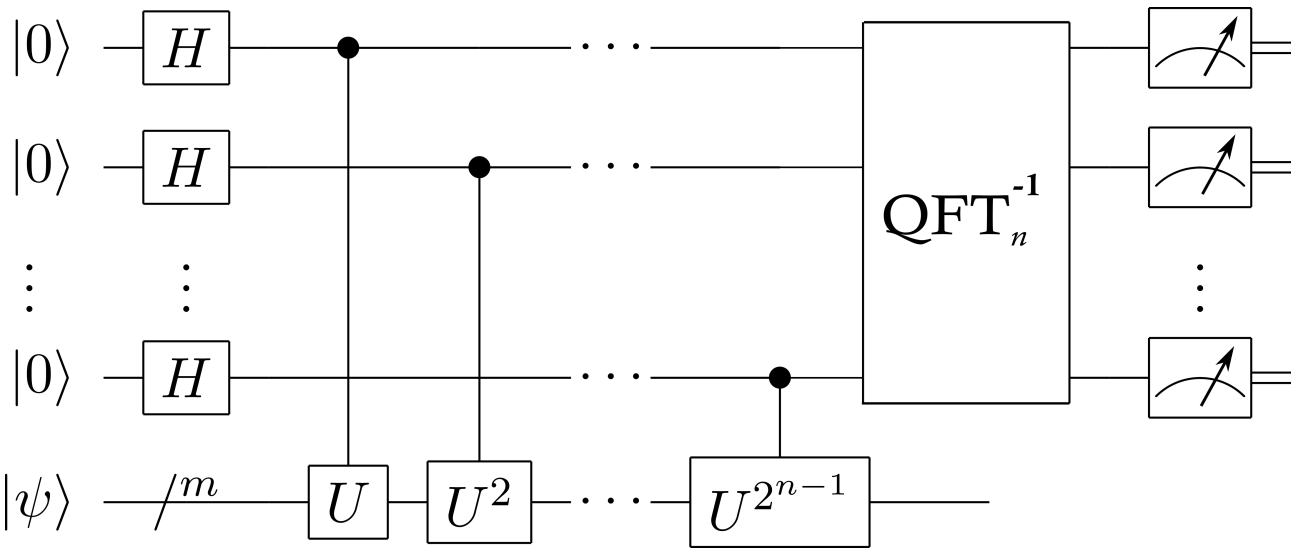


Fig. V.4. Circuit diagram of Quantum Phase Estimation algorithm.

6.2. Shor's algorithm

Shor's algorithm consists in reducing the problem of factoring to the problem of period finding. It uses a quantum algorithm for fast period finding.

Shor's algorithm

1. If N is even, return $f = 2$.
2. If $N = p^k$ for p prime, return p.
3. Randomly choose $1 < q < N-1$.
   If $f = gcd(q,N) > 1$, return f
4. Determine the order k of q modulo N. (Phase estimation).
   If k is odd, repear from step 3.
5. Write $k = 2l$ and determine q^l mod N with $1 < r < N$.
   a) If $1 < f = gcd(r-1,N) < N$, return f.
   b) If $1 < f = gcd(r+1,N) < N$, return f.
   c) Else repeat step 3.

All steps, excepted step 4, can be performed efficiently by a classical computer.

Given an n-bit integer
— classical number field sieve: $\mathcal{O}\left(2^{n^{1/3}}\right)$.
— Shor's algorithm: $\mathcal{O}\left(n^3\right)$.

7. Hamiltonian simulation

7.1. Context

In quantum mechanics, physical systems are described by Hamiltonians. The evolution is given by Schrödinger's equation

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

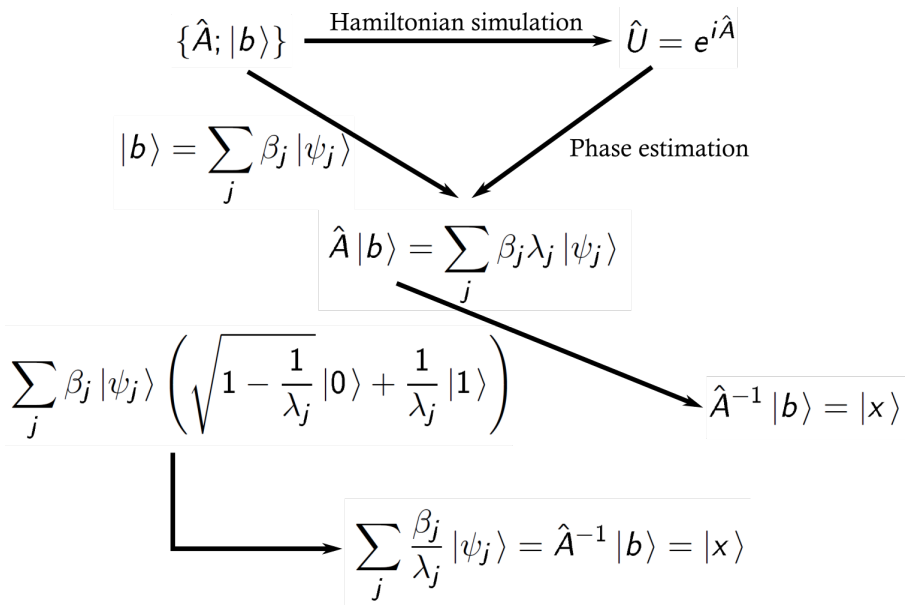


Fig. V.5. HHL algorithm outline.

For a stationnary hamiltonian, $|\psi(t)\rangle = e^{-i\frac{\hat{H}t}{\hbar}}|\psi(0)\rangle$

Hamiltonian simulation:
Given a Hamiltonian $\hat{H}$, construct a quantum circuit that approximates $e^{-i\frac{\hat{H}t}{\hbar}}$.

There are a number of quantum algorithm that can do this efficiently for certain type of Hamiltonian. To simulate a classical system like a plane, a classical computer is appropriated. But to simulate a quantum system like a molecule, a quantum computer is better suited.

7.2. HHL

Named after Harrow, Hassidim and Lloyd, who invented it in 2008. It attacks one of the most fundamental tasks in science: solving systems of linear equations

$$\boxed{A\vec{x}=\vec{b},\ \text{ solve for } \vec{x}}.$$

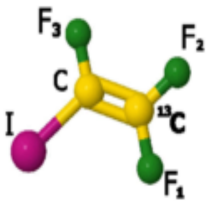
Classically: it takes polynomial time in the size of the matrix, whereas HHL "solves" this problem in logarithmic time. **Quantum algorithm HHL:** inputs $|b\rangle$ and $\hat{A}$, outputs quantum state $|x\rangle$. The general outline of HHL algorithm is represented Fig. V.5

7.3. Applications

— Solving systems of differential equations (finite element method, FEM).
— Data fitting.
— Various tasks in machine learning (clustering, support-vector machines, principal component analysis).

Run time: for a system of n equations.
— classical: $\mathcal{O}\left(n^3\right)$;
— quantum: $\mathcal{O}\left(\kappa s\frac{\log n}{\varepsilon}\right)$, with κ the condition number, s the sparsity and ε the accuracy.

	^{13}C	F_1	F_2	F_3
^{13}C	15479.7 Hz			
F_1	-297.7 Hz	-33122.4 Hz		
F_2	-275.7 Hz	64.6 Hz	-42677.7 Hz	
F_3	39.1 Hz	51.5 Hz	-129.0 Hz	-56445.8 Hz
T_2^*	1.22 s	0.66 s	0.63 s	0.61 s
T_2	7.9 s	4.4 s	6.8 s	4.8 s



J. Pan *et al.*, Phys. Rev. A 89, 022313 (2014)

Fig. V.6. Experimental realization of quantum algorithm for solving linear systems of equations, J. Pan *et al.*, Phys. Rev. A **89**, 022313 (2014).

7.4. Experimental realization

1. Experimental Quantum Computing to Solve Systems of Linear Equations, X.-D. Cai *et al.*, Phys. Rev. Lett. **110**, 230501 (2013).
   Problem solved: 2×2 linear equations.
   Qubits: photons (polarization).
2. A two-qubit photonic quantum processor and its application to solving systems of linear equation, S. Barz *et al.*, Sci. Rep. **4**, 6115 (2014).
   Problem solved: 2×2 linear equations.
   Qubits: photons (polarization).
3. Experimental realization of quantum algorithm for solving linear systems of equations, J. Pan *et al.*, Phys. Rev. A **89**, 022313 (2014).
   Problem solved: 2×2 linear equations.
   Qubits: NMR type qubits in a molecule of iodotrifluoroethylene $^{12}C^{13}CF_3I$.

8. Quantum error correction

In classical computing, error correcting codes preserves classical bits. Quantum error correcting code will preserve a qubit $|\psi\rangle$ in quantum computing. Let consider an error affecting one or more qubits is simply an arbitrary (unknown) unitary operator $\hat{N}$ applied to those qubits ($\hat{N}$ is a noise operator). The classical bit-flip is an example, and corresponds to the application of the operator $\hat{X}$.

$$\hat{X}|0\rangle = |1\rangle \text{ and } \hat{X}|1\rangle = |0\rangle.$$

The process of correcting errors in a qubits state $|\psi\rangle$ might be described as shown in Fig. V.7.

— $\hat{E}$ is an **encoding** unitary operator,
— $\hat{N}$ is a **noise** unitary operator,
— $\hat{D}$ is a **decoding** unitary operator,

We encode some qubit state $|\psi\rangle$ in a larger state $|E(\psi)\rangle$ using n ancilla qubits (initially on the state $|0\rangle^{\otimes n}$. Some noise is applied through $\hat{N}$, and later we decode the noisy encoded state to produce a state $|\psi'\rangle$.

Goal of the process: $|\psi'\rangle \approx |\psi\rangle$

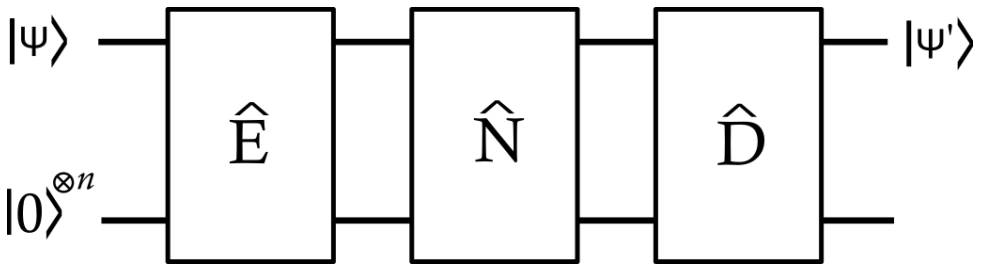


Fig. V.7. Simple description of error correction codes.

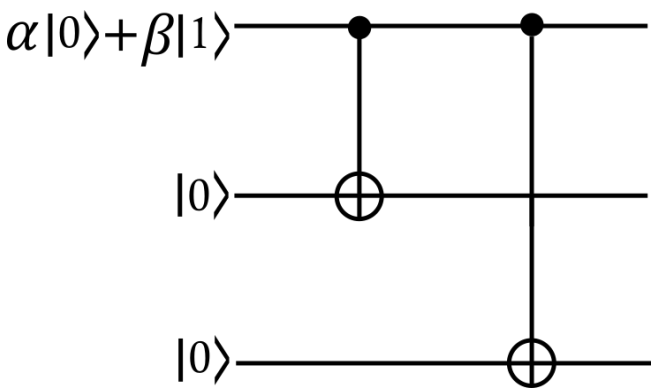


Fig. V.8. Implementation of operator $\hat{E}$.

Non cloning theorem: it is not possible to duplicate a state $|\psi\rangle$ in the general case

$$|\psi\rangle \nrightarrow |\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle \cdots \otimes |\psi\rangle.$$

The error protection can't be performed by cloning the state $|\psi\rangle$.

Principle of error correction code: let consider $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Then, encode it as follow

$$|E(\psi)\rangle = \alpha|000\rangle + \beta|111\rangle.$$

Remark: it is not a cloning !

$$|\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle = (\alpha|0\rangle + \beta|1\rangle)^{\otimes 3}.$$

The operator $\hat{E}$ might be implemented as shown in Fig. **??**. The decoding algorithm for this code will be based on the circuit represented Fig. **??**. The first three qubits are called *input qubits*. The last two qubits are called *output qubits*.

$$|\phi_1\rangle = |x_1 \oplus x_2\rangle,$$
$$|\phi_2\rangle = |x_1 \oplus x_3\rangle.$$

$x_1 \oplus x_2$ and $x_1 \oplus x_3$ are invariant under the flipping of all the bits of x. After the application of $\hat{N}$

$$\hat{N}|E(\psi)\rangle = \alpha|x_1x_2x_3\rangle + \beta|x_1x_2x_3 \oplus 111\rangle.$$

The circuit proposed performs the following map

$$(\alpha|x_1x_2x_3\rangle + \beta|x_1x_2x_3 \oplus 111\rangle) \otimes |0\rangle \otimes |0\rangle$$
$$\longrightarrow (\alpha|x_1x_2x_3\rangle + \beta|x_1x_2x_3 \oplus 111\rangle) \otimes |x_1 \oplus x_2\rangle \otimes |x_1 \oplus x_3\rangle.$$

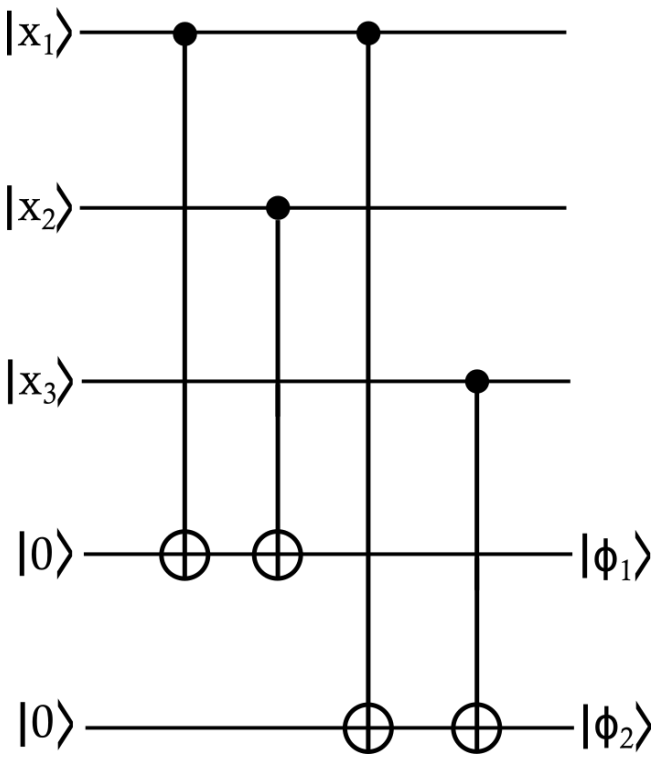


Fig. V.9. Error correcting code.

If one measures the two output qubits, we learn both $x_1 \oplus x_2$ and $x_1 \oplus x_3$ without disturbing the input qubits. The encoded state $|\psi\rangle$ is always of this form, even after arbitrary bit-flip errors are applied to $|E(\psi)\rangle$.

$$|E(\psi)\rangle = \alpha|000\rangle + \beta|111\rangle.$$

Effect of bit-flip on $|E(\psi)\rangle$

$$\left(\hat{X}\otimes\mathbb{I}\otimes\mathbb{I}\right)|E(\psi)\rangle = \alpha|100\rangle + \beta|011\rangle,$$
$$\left(\hat{X}\otimes\hat{X}\otimes\hat{X}\right)|E(\psi)\rangle = \alpha|111\rangle + \beta|000\rangle.$$

The result of measuring the output qubits is known as the **syndrome**. What are the syndromes of different noise operators $\hat{N}$ applied to $|E(\psi)\rangle$? If $\hat{N} = \mathbb{I}$, we always measure 00. If $\hat{N} = \hat{X}\otimes\mathbb{I}\otimes\mathbb{I}$, we always obtain 11.

Syndrome measured for different bit-flip noise is reported in Tab. V.1.

$\hat{N}$	Syndrome
$\mathbb{I}\otimes\mathbb{I}\otimes\mathbb{I}$	00
$\mathbb{I}\otimes\mathbb{I}\otimes\hat{X}$	01
$\mathbb{I}\otimes\hat{X}\otimes\mathbb{I}$	10
$\hat{X}\otimes\mathbb{I}\otimes\mathbb{I}$	11
$\mathbb{I}\otimes\hat{X}\otimes\hat{X}$	11
$\hat{X}\otimes\hat{X}\otimes\mathbb{I}$	01
$\hat{X}\otimes\mathbb{I}\otimes\hat{X}$	10
$\hat{X}\otimes\hat{X}\otimes\hat{X}$	00

Table V.1 – Syndrome measured for different bit-flip noise.

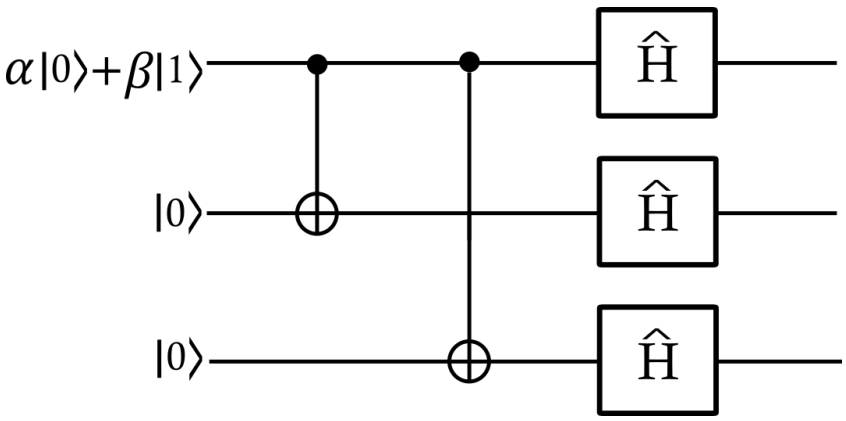


Fig. V.10. More elaborated error correcting code.

If the error occurs on a single qubit, it is possible to detect it, and apply the corresponding bit-flip operation on the corresponding qubit to restore the original encoded state $\alpha|000\rangle + \beta|111\rangle$. On the other hand, if bit flip errors occurs on more than one qubit, one does not detect them. In the case of $\hat{Z}$ noise,

$$\hat{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

the syndrome measurement always return 00, so the error correction operation does nothing and the $\hat{Z}$ error is not corrected. But

$$\hat{Z} = \hat{H}\hat{X}\hat{H},$$

where $\hat{H}$ is the Hadamard gate. Thus $\hat{Z}$ acts in the same way as $\hat{X}$, up to a change of basis. If we use the same code as before, but perform this change of basis for each qubit, we obtain a code which corrects against $\hat{Z}$ errors. In other words, we now encode $|\psi\rangle$ as $\alpha|+++\rangle + \beta|---\rangle$, with

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad \text{and} \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

The new encoding circuit is represented Fig. V.10 and the decoding circuit represented Fig. V.11. But it does no longer protects against $\hat{X}$ errors !!! It is possible to concatenates these two codes. We first encode $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ using the code protecting against phase flips, and then encode each of the resulting qubits using the code that protects against bit flips. In other words, we perform the following map

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

$$\begin{aligned} \longrightarrow \frac{1}{2\sqrt{2}} \quad & (\alpha(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)(|0\rangle + |1\rangle) \\ & + \beta(|0\rangle - |1\rangle)(|0\rangle - |1\rangle)(|0\rangle - |1\rangle)), \end{aligned}$$

$$\begin{aligned} \longrightarrow \frac{1}{2\sqrt{2}} \quad & (\alpha(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \\ & + \beta(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)), \end{aligned}$$

The single qubit $|\psi\rangle$ is now encoded using 9 qubits.

These qubits can naturally be split into three blocks, each of which encodes one qubit of the state

$$\alpha|+++\rangle + \beta|---\rangle.$$

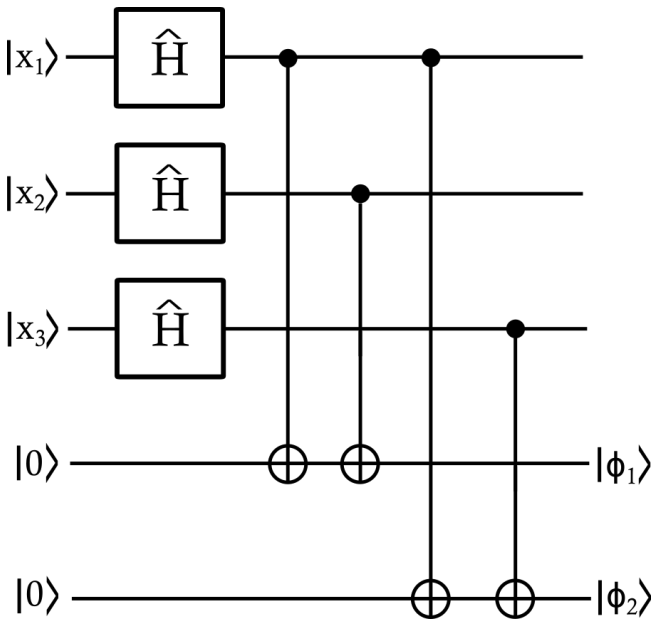


Fig. V.11. Decoding circuit of error correcting code of Fig. V.10.

To decode this encoded state, first the decoding circuit for the bit-flip code is applied to each block. Assuming at most one bit-flip error has occurred in each block, the result will be the state $\alpha|+++\rangle+\beta|---\rangle$, perhaps with a $\hat{Z}$ error applied to one of the qubits. This state can then be mapped back to $\alpha|0\rangle+\beta|1\rangle$ using the decoding algorithm for the phase-flip code.

This quantum error-correcting code was the first such code discovered. It was invented by Peter Shor in 1995, known as Shor's 9 qubit code.

Chapter VI

Decoherence, Noisy Intermediate-Scale Quantum (NISQ) computers and quantum supremacy

Performances of quantum computer are deeply related to the error rate on qubit manipulation. The proper formalism to describe decoherence, source of error, is the density matrix operator. Today's better quantum computers have error rates of typically 0.1% to 1%, with 50 to 100 qubits chips only. As a comparison, a typical classical computer chip holds about $20 \cdot 10^9$ bits (or transistors) while the latest smartphone chips holds about $6 \cdot 10^9$ bits. Classical digital computers are truly reliable at the bit level, with fewer than 1 error in 10^{24} operations. The far more common sources of error are software and mechanical malfunction.

1. Density matrix - T_1 and T_2 times and decoherence

1.1. Definition

A qubit is a two-level system, and quantum technologies mainly exploit qubit is a superposition of state. But such states are inherently fragile as a result of their interaction with the environment: that is *decoherence*. Quantification of decoherence in quantum computers or quantum interferometer is of importance to quantify devices' performances. A commonly used tool in that aim is the **density matrix** $\hat{\rho}$. The later permit to define T_1 and T_2 times. T_1 time refers to state population relaxation toward equilibrium (for example relaxation of spins in the direction of a magnetic field), while T_2 is a dephasing time between particles and refers to decoherence of a quantum superposition toward a classical superposition of states.

For example, if one considers an ensemble of N particles in the state

$$|\psi\rangle = \frac{1}{\sqrt{2^N}} \left(|0\rangle + |1\rangle\right)^{\otimes N},$$

one will measure $N/2$ atoms in state $|0\rangle$ and $N/2$ atoms in state $|1\rangle$. It is a pure quantum state. Now one consider a classical statistical system made of $N/2$ atoms in state $|0\rangle$ and $N/2$ atoms in state $|1\rangle$. This ensemble will result in the same result after measurement but it is not the same quantum state!

One uses the density matrix formalism to describe a statistical ensemble of quantum particles. A pure state is a system without any statistical superposition, that could be described by a single quantum state $|\psi\rangle$. The density matrix operator is then defined as follow

$$\hat{\rho} = |\psi\rangle\langle\psi|.$$

Let's consider an observable $\hat{\mathcal{O}}$. Since one has the following relation

$$\mathrm{Tr}\left(\hat{\rho}\hat{\mathcal{O}}\right) = \mathrm{Tr}\left(\hat{O}|\psi\rangle\langle\psi|\right) = \mathrm{Tr}\left(\langle\psi|\hat{O}|\psi\rangle\right) = \left\langle\hat{\mathcal{O}}\right\rangle.$$

So the density matrix $\hat{\rho}$ permits to evaluate the mean value of any operator

$$\left\langle\hat{\mathcal{O}}\right\rangle = \mathrm{Tr}\left(\hat{\rho}\hat{\mathcal{O}}\right).$$

In the case of a two level particle, the Bloch representation of any state is the following

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

Geometrically, the state is represented on the Bloch sphere with a unitary vector $\vec{n}$ such that

$$\vec{n} = \begin{pmatrix} \sin\theta\cos\varphi \\ \sin\theta\sin\varphi \\ \cos\theta \end{pmatrix}.$$

A straightforward permits to demonstrate easily that in the case of two-level system

$$\boxed{\hat{\rho} = |\psi\rangle\langle\psi| = \frac{\hat{\mathbb{I}} + \vec{n}\cdot\hat{\vec{\sigma}}}{2}},$$

with

$$\hat{\vec{\sigma}} = \hat{\sigma}_x\vec{u}_x + \hat{\sigma}_y\vec{u}_y + \hat{\sigma}_z\vec{u}_z.$$

A mixed state is a classical statistical superposition of an ensemble of orthogonal states. If $\{|\psi_1\rangle, ..., |\psi_N\rangle\}$ is an ensemble of N states, with statistical weight $\{w_i\}$ such that $\sum_i w_i = 1$, then the density matrix $\hat{\rho}$, of such a so-called mixed state, is defined as follow

$$\hat{\rho} = \sum_{i=1}^{N} w_i |\psi_i\rangle\langle\psi_i|.$$

The definition of $\hat{\rho}$ is independent of the choice of the orthogonal basis. Then, for an observable $\hat{\mathscr{O}}$,

$$\langle\hat{\mathscr{O}}\rangle = \mathrm{Tr}\left(\hat{\rho}\hat{\mathscr{O}}\right) = \sum_{i=1}^{N} w_i \mathrm{Tr}\left(|\psi_i\rangle\langle\psi_i|\hat{\mathscr{O}}\right) = \sum_{i=1}^{N} w_i \langle\psi_i|\hat{\mathscr{O}}|\psi_i\rangle,$$

as expected for a classical statistical superposition. Now if one considers the thermal equilibrium and the basis $|i\rangle$ of eigenvectors of the Hamiltonian, the density matrix is provided by Maxwell statistical distribution

$$\hat{\rho} = \sum_{i=1}^{N} \frac{e^{-\beta E_i}}{Z} |i\rangle\langle i|,$$

with Z the partition function, E_i the energy of state $|i\rangle$ and $\beta = \frac{1}{k_B T}$. This state is known as the *Gibbs state* or *thermal state*, describing quantum systems at thermal equilibrium. Since $Z = \mathrm{Tr}\left(e^{-\beta\hat{H}}\right)$ for $\hat{H}$ the Hamiltonian, one may express it as

$$\hat{\rho} = \frac{e^{-\beta\hat{H}}}{\mathrm{Tr}\left(e^{-\beta\hat{H}}\right)}.$$

1.2. The Bloch ball

Now one resists oneself to the case of a statistical ensemble of two level particles, either in pure or mixed states. The state space to describe a particle is an Hermitian space of dimension 2×2. The density matrix $\hat{\rho}$ is an hermitian operator of such a space, and might be decomposed of Pauli's matrices basis. Moreover, it is straightforward to show that

$$\mathrm{Tr}\left(\hat{\rho}\right) = 1.$$

And since $\mathrm{Tr}\,\hat{\sigma}_x = \mathrm{Tr}\,\hat{\sigma}_y = \mathrm{Tr}\,\hat{\sigma}_z = 0$ but $\mathrm{Tr}\,\hat{\mathbb{I}} = 2$, $\exists\vec{n}$ such that

$$\hat{\rho} = \frac{\mathbb{I} + \vec{n}\cdot\hat{\vec{\sigma}}}{2}.$$

It is straightforward that the operator $\hat{\rho}$ is positive, thus with positive eignevalues. The eigenvalues of $\vec{n}\cdot\hat{\vec{\sigma}}$ are $\pm\|\vec{n}\|$. So the positivity of the operator is verified if and only if

$$\|\vec{n}\| \leq 1.$$

So any two level system might be represented in a Bloch ball, *i.e.* described by a vector $\vec{n}$ in a unitary ball (and no longer only a sphere), such that $\|\vec{n}\| \leq 1$.

$$\hat{\rho} \in \left\{ \frac{\mathbb{I} + \vec{n} \cdot \hat{\vec{\sigma}}}{2} \ , \ \|\vec{n}\| \leq 1 \right\}.$$

The vector $\vec{n}$ will be unitary ($\|\vec{n}\| = 1$) only in the case of a pure state.

1.3. Dynamics of density matrices

The Schrödinger equation states that

$$i\hbar \frac{\partial}{\partial t} |\psi\rangle = \hat{H} |\psi\rangle,$$

and by taking the complex conjugate one gets

$$-i\hbar \frac{\partial}{\partial t} \langle\psi| = \langle\psi| \hat{H},$$

such that

$$\frac{\partial}{\partial t} |\psi\rangle\langle\psi| = -\frac{i}{\hbar} \left(\hat{H} |\psi\rangle\langle\psi| - |\psi\rangle\langle\psi| \hat{H} \right) = -\frac{i}{\hbar} \left[\hat{H}, \hat{\rho} \right].$$

So the equation of motion for the density matrix operator evolves in time according to

$$\boxed{i\hbar \frac{\partial \hat{\rho}}{\partial t} = \left[\hat{H}, \hat{\rho} \right]}.$$

1.4. Decoherence

Unitary operations correspond to reversible operations: if $\hat{U}$ is a valid unitary time evolution, then so is $\hat{U}^\dagger$. In terms of Hamiltonians, evolution according to $-\hat{H}$ will reverse evolution according to $\hat{H}$. But other quantum processes cause an irreversible loss of information. Irreversible quantum processes are generally called *decoherence*. This somewhat imprecise term refers to the fact that this information loss is always associated with a loss of *coherence* and with quantum systems becoming more like classical systems.

Let consider a two level system, of basis $\{|0\rangle, |1\rangle\}$ of respecting energies 0 and $\hbar\omega_0$. A thermal equilibrium one has a probability $P_{|0\rangle}$ (resp. $P_{|1\rangle}$) to be in state $|0\rangle$ (resp. $|1\rangle$), with

$$P_{|0\rangle} = \frac{1}{1 + e^{-\beta\hbar\omega}} \ \text{ and } \ P_{|1\rangle} = \frac{e^{-\beta\hbar\omega}}{1 + e^{-\beta\hbar\omega}}.$$

One introduces the density matrix written as

$$\hat{\rho} = \begin{pmatrix} \rho_{11} & \rho_{10} \\ \rho_{01} & \rho_{00} \end{pmatrix}.$$

At thermal equilibrium, the density matrix as the following expression

$$\hat{\rho}_{\text{thermal}} = \frac{1}{1 + e^{-\beta\hbar\omega}} \begin{pmatrix} e^{-\beta\hbar\omega} & 0 \\ 0 & 1 \end{pmatrix},$$

while for a Bell state such as $|\Psi+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$,

$$\hat{\rho}_{|\Psi+\rangle} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

The diagonal terms of a density matrix are called *populations*, and the non-diagonal terms are called *coherences*. A classical state has non coherence, *i.e.* no non-diagonal terms. A quantum superposition will relax to a classical state with a decrease of its non-diagonal terms.

Now if one consider an ensemble of N two level atoms initially in the state $|\Psi+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, there time evolution will be

$$|\psi(t)\rangle = \frac{|0\rangle + e^{i\omega_0 t}|1\rangle}{\sqrt{2}}.$$

But due to the interaction of the environment, the energy of the excited state might be shifted (fluctuation of magnetic field, etc...) from $\hbar\omega_0$ to $\hbar(\omega_0 + \delta\omega)$. It results in time evolutions differents depending on this energy shift. Due to the dispersion in the excited state energy, qubits will progressively undergo relative dephasing between one to each others

$$|\psi(t)\rangle = \frac{|0\rangle + e^{i\delta\omega t}e^{i\omega_0 t}|1\rangle}{\sqrt{2}}.$$

This is responsible of the decoherence, mixing up phases of superpositions and providing collapse of the non-diagonal terms of the density matrix, which becomes

$$\hat{\rho}_{|\Psi+\rangle} = \frac{1}{2}\begin{pmatrix} 1 & e^{-i\omega_0 t}\int p(\delta\omega)e^{-i\delta\omega t}\mathrm{d}\delta\omega \\ e^{i\omega_0 t}\int p(\delta\omega)e^{i\delta\omega t}\mathrm{d}\delta\omega & 1 \end{pmatrix},$$

where $p(\delta\omega)$ is the statistical distribution of the energy shift. Moreover, it is straightforward that

$$\lim_{t\to+\infty}\int p(\delta\omega)e^{i\delta\omega t}\mathrm{d}\delta\omega = 0.$$

Consequently, a pure state progressively collapse to a mixture state. But this simple model only explain the collapse of non-diagonal terms (phase decoherence). The interaction of the environment is also responsible of state rotation so that populations relaxes toward the thermal distribution. The phase collapse and the population relaxation occurs with different timescales. Then, T_1 is the population relaxation time constant on which diagonal terms of the density matrix exponentially relaxes (as e^{-t/T_1}). It might be modelized by an additional term in the equation of evolution of the following form

$$\frac{\partial\hat{\rho}}{\partial t}\Big)_{\text{relaxation}} = -\frac{1}{T_1}\left(\hat{\rho} - \hat{\rho}_{\text{thermal}}\right).$$

T_2 is the coherence relaxation time constant (or dephasing time) on which non-diagonal terms of the density matrix exponentially relaxes (as e^{-t/T_2}). It might be modelized by an additional term in the equation of evolution of the following form

$$\frac{\partial\hat{\rho}}{\partial t}\Big)_{\text{dephasing}} = -\frac{1}{T_2}\begin{pmatrix} 0 & \rho_{10} \\ \rho_{01} & 0 \end{pmatrix}.$$

To model the global dynamic, two additional terms are empirically added to the time evolution equation of the density matrix as follow

$$\boxed{\frac{\partial\hat{\rho}}{\partial t} = -\frac{i}{\hbar}\left[\hat{H},\hat{\rho}\right] - \frac{1}{T_1}\left(\hat{\rho} - \hat{\rho}_{\text{thermal}}\right) - \frac{1}{T_2}\begin{pmatrix} 0 & \rho_{10} \\ \rho_{01} & 0 \end{pmatrix}}.$$

2. Quantum advantage, quantum supremacy

2.1. Definitions

Both quantum supremacy and quantum advantage are conceptual criteria, more related to theoretical computer sciences, regardless to the usefulness of the problem considered.

Quantum advantage consists in demonstrating that a quantum device can solve a problem faster than classical computers.

Quantum supremacy is the demonstration that a programmable quantum device can solve a problem that a classical computer practically can not due to its complexity ("hard problems" with important computation time).

The concept of quantum supremacy has been proposed for the first time by John Preskill in 2012 [21]. Several problems have been proposed to demonstrate quantum supremacy: factoring integers (Shor's algorithm), boson sampling proposed by Aaronson and Arkhipov [2], frustrated cluster loop problems (D-waves) [18] or the sampling of the output of random quantum circuit (Google) [3].

2.2. Demonstration (or not ?) of quantum supremacy by Google in 2019

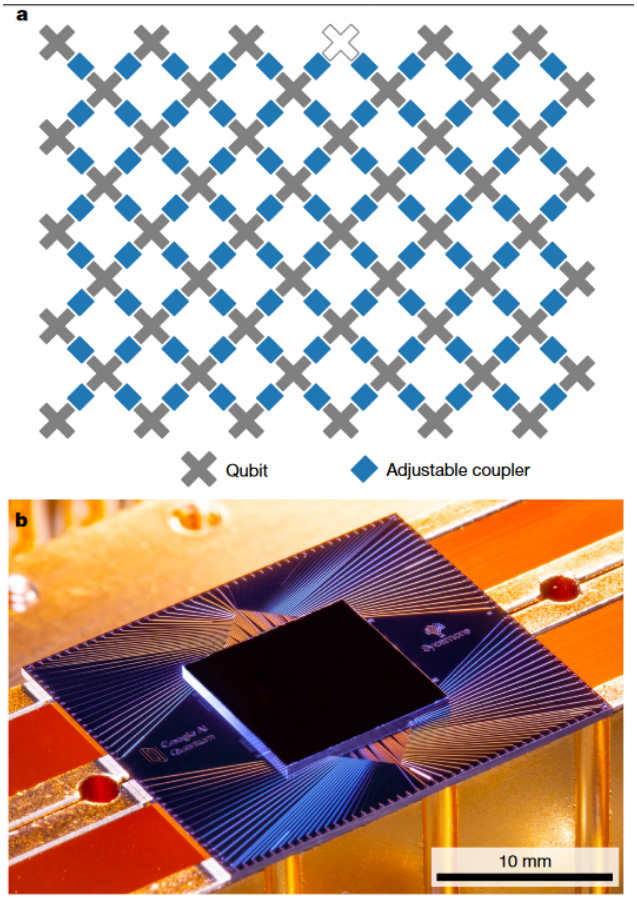


Fig. 1 | The Sycamore processor. a, Layout of processor, showing a rectangular array of 54 qubits (grey), each connected to its four nearest neighbours with couplers (blue). The inoperable qubit is outlined. **b**, Photograph of the Sycamore chip.

Fig. VI.1. The quantum chip used by Google to demonstrate quantum supremacy in October 2019. Extracted from [4].

Sampling the output distributions of random quantum circuits is believed to be hard for classical computers based on reasonable complexity assumptions exposed in reference [3]. Google has announced to work on demonstration of quantum supremacy before the end of 2017 by solving this problem with an array of 49 superconducting qubits [14]. But the main challenge was to develop such a chip, with acceptable error rates. In October 2017, IBM demonstrated the simulation of 56 qubits on a conventional supercomputer, increasing the number of qubits needed for quantum supremacy [19]. Until January 2018, only Intel has officially announced to have produced such a chip, with indeed the goal to demonstrate quantum supremacy [1].

But in October 2019, Google has published an article in Nature [4], claiming its quantum computing research team has demonstrated quantum supremacy with a 53 qubits quantum computer, based on superconducting qubits. On this chip, named *Sycamore*, the typical single qubit gate error rate is about 0.15%. Entanglement in a 2-qubits gates is achieved with 0.6% error rate and is executed in 12 ns. The estimated global circuit fidelity is $\mathcal{F}$=0.2% for circuits with 20 cycles of 2-qubits gates (430 two-qubits gates and 1113 single-qubit gates). However, measurement error rates are up to few percent, typically 3-4%. The quantum processor is qualified to be programmable (in opposition to quantum annealing processors, like D-waves' ones), which means it's a programmable circuit-based quantum computer (like IBM's or Rigetti's ones). With 53 qubits, the computational state-space is of dimension 2^{53} ($\sim 10^{16}$ states). A specific algorithm has been chosen, constructed specifically to demonstrate quantum supremacy and explained in details in reference [6]. The algorithm is based on a circuit with fixed two-qubits gates and randomly-chosen single qubit gates is chosen. The circuit is a sequence of d clock cycles of one- and two-qubits gates, with gates applied to different qubits in the same cycle. On example of such type of circuit is represented Fig. VI.2, in the case of a 1D geometry (linear array) of qubits. Random quantum circuits with gates sampled from a universal gate set are examples of quantum chaotic evolutions that naturally lend themselves to the quantum computational framework [6].

The corresponding circuit is executed millions of times for $d = 20$ cycles. Each time, all qubits are measured, generating a 53-bit string. The collected sample of 53-bit strings is not uniformly distributed. Comparing with classical simulations one can verify "heavy output generation" - that the average probability of strings in the sample is greater than 2^{-n}. Because a random circuit has no structure, and the Hilbert space is exponentially

large in n, simulation using a classical supercomputer is hard. Experiment verifies that the hardware is working well enough to produce meaningful results in a regime where classical simulation is very difficult.

Measurements from repeated experiments sample the resulting probability distribution, which was verified then using classical simulations. **It tooks to the Sycamore processor about 200 seconds to sample one instance of a quantum circuit, while a million times—our benchmarks currently indicate that the equivalent task for a state-of-the-art classical supercomputer would take approximately 10,000 years!** That is why Google claimed to have demonstrated quantum supremacy, demonstration the exponential speed-up of a quantum computer compared to a classical supercomputer, even with only 53 qubits. The strategy of Google consisted in choosing an algorithm adapted to the design of the quantum chip they used. This permits to avoid the use of swap gates to couples non-adjacent qubits and reduce the number of quantum gates requires. The algorithm used was not of practical interest but well adapted to a quantum chip and not at all for a classical calculation.

This results is a milestone is the sense that it demonstrates that a 53 qubits programmable quantum computers might be used to realize a quantum calculation, even if qubits manipulations are not perfect, with good but not negligeable error rates. John Preskill, professor of Theoretical Physics at CalTech, has qualified such type of devices: *Noisy Intermediate-Scale Quantum* (NISQ) [22]. Google's paper has demonstrated that the noisy 50-100 qubits quantum computers has arrived, has an intermediate step. NISQ devices cannot be simulated by brute force using the most powerful currently existing supercomputers. But noise limits the computational power of such NISQ devices. **NISQ devices are not disruptive technologies by themselves**, but rather a step toward more powerful quantum technologies in the future. For Preskill, the next steps consist in finding real world applications and at the same time, one need to dramatically extent qubits lifetimes either with quantum error correction codes or with qubits fabrication process improvements (or technological paradigm shift for qubits implementation). One only need to improve significantly the two-qubits gates fidelity. Of course, one has to develop chips with more qubits and better gates, to enable more complex algorithm than the one used by Google.

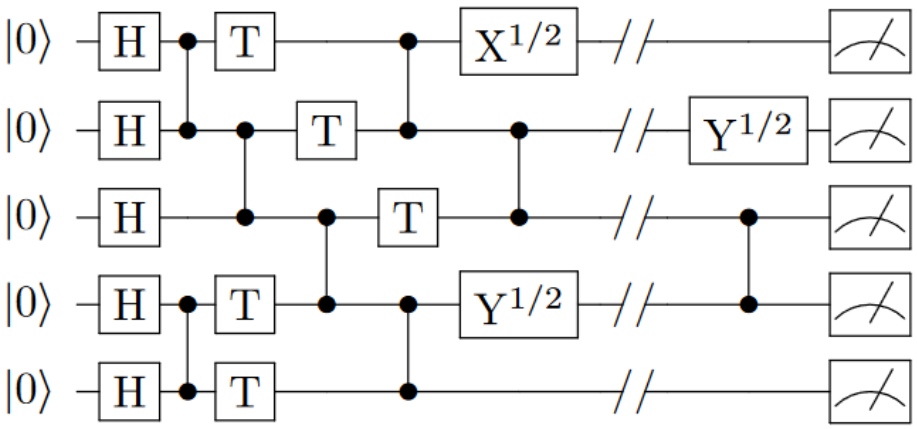


Fig. VI.2. Example of a random quantum circuit in a 1D array of qubits. Vertical lines correspond to controlled-phase (C-Z) gates. Extracted from [6].

2.3. IBM's answer to Google

If Google's results is clearly a milestone as a demonstration of a NISQ quantum computer and a quantum algorithm implementation, the quantum supremacy claimed has been quickly contested by IBM researchers. Indeed, they published a paper on ArXiv [20] as an answer to Google claim. In this article, IBM researchers demonstrate that they have discovered a classical algorithm to simulate the execution of Google's quantum algorithm initially on a 53 qubits quantum computer. They have implemented this classical algorithm on a

classical supercomputer (Summit supercomputer at Oak Ridge National Laboratories). They are able to simulate 53- and 54-qubits Sycamore circuits with high fidelity to arbitrary depth (number of cycle d). They have demonstrated that for 53-qubits with depth of 20 cycles, the result of the quantum calculation can be obtained with their supercomputer in only 2.5 days! While Google claimed classical computers required 10,000 years. Of course, it is still larger than the 200 s obtained with the quantum computer, but it is clearly less obvious to claim quantum supremacy. Moreover, IBM stressed out that it was only a preliminary estimation of the calculation time, with the most pessimist estimation. At the end, it could be possible to obtain the same result than the quantum computer with a calculation time even shorter. But the implementation on the Oak Ridge's supercomputer has not been yet realized. A calculation time of 200 s is certainly shorter than a time of the order of one day, but if IBM is right, quantum supremacy is still not achieved. Moreover, in any case, the conventional computer capable of performing this calculation is more efficient in terms of the precision and fidelity of the results to be achieved. But Google's result with Sycamore remains a milestone: the achievement of a quantum calculation with 53 qubits on a real quantum computer (with noisy qubits).

3. Quantum annealer

3.1. Quantum annealing processor

Quantum annealing processors are chips made out of a lot of qubits but not programmable, in the sense they can't implement a circuit diagram with quantum gates. But a quantum annealing processor naturally returns low-energy solutions of a given potential. Quantum annealing is a method for finding the global minimum of a given function over a given set of candidate solutions, thanks to a process using quantum fluctuations. It is adapted for finding extrema of multidimensional functions [11]. Quantum annealing is used mainly for problems where the search space is discrete (combinatorial optimization problems) with many local minima, such as finding the ground state of a spin system (Ising problem [17]) or the traveling salesman problem.

Quantum annealing starts initializing an ensemble of qubits in a superposition of all possible states (candidate solutions) with equal weights. Then the system evolves, governed by Schrödinger equation.

3.2. Optimization problems

Quantum annealing is a well suited method for optimization problems, where ones searches for the best of many possible combinations. Optimization problems include scheduling challenges, or salesman problem.

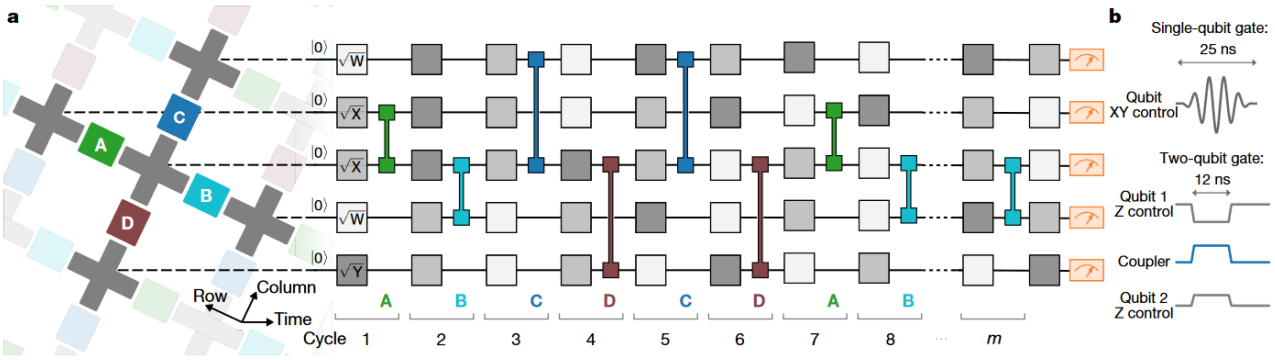


Fig. 3 | Control operations for the quantum supremacy circuits. a, Example quantum circuit instance used in our experiment. Every cycle includes a layer each of single- and two-qubit gates. The single-qubit gates are chosen randomly from $\{\sqrt{X}, \sqrt{Y}, \sqrt{W}\}$, where $W = (X+Y)/\sqrt{2}$ and gates do not repeat sequentially. The sequence of two-qubit gates is chosen according to a tiling pattern, coupling each qubit sequentially to its four nearest-neighbour qubits. The couplers are divided into four subsets (ABCD), each of which is executed simultaneously across the entire array corresponding to shaded colours. Here we show an intractable sequence (repeat ABCDCDAB); we also use different coupler subsets along with a simplifiable sequence (repeat EFGHEFGH, not shown) that can be simulated on a classical computer. **b**, Waveform of control signals for single- and two-qubit gates.

Fig. VI.3. Extracted from [4].

Physics can help solve these sorts of problems because we can frame them as energy minimization problems. That is the key point of a quantum annealer: it converts an optimization problem into a fundamental state search of a quantum system. A fundamental rule of physics is that everything tends to seek a minimum energy state. While objects slide down hills, hot things cool down over time with heat dissipation. A quantum annealer solves optimization problems by evolving a known initial configuration at non-zero temperature towards the ground state of a Hamiltonian encoding a given problem.

Quantum annealing simply uses quantum physics to find low-energy states of a problem and therefore the optimal or near-optimal combination of elements.

3.3. D-wave quantum processors

D-Wave Systems is a Canadian company D-Wave Systems which provide quantum machines dedicated to perform doing quantum annealing. In 2011, Lockheed-Martin purchased a D-Wave One model for about $ 10 million. In May 2013, Google purchased a D-Wave Two with 512 qubits. As of now, the question of whether the D-Wave processors offer a speedup over a classical processor is still unanswered [16]. Tests performed by researchers at Quantum Artificial Intelligence Lab (NASA), USC, ETH Zurich, and Google show that until now there is no evidence of a quantum advantage [7, 23, 28]. Being the only kind of quantum computer available for actual sale (assuming you have $10 million to $15 million to spare) has made D-Wave unique for several years, although the mainstream attention has now shifted away from its approach. Enabling more general operations is the biggest hurdle for quantum annealers going forward. D-Wave's recently launched real-time cloud platform, called Leap, opens up widespread access to its quantum application environment and has the potential to be quickly embraced by the user community.

3.4. The user's view

A D-Wave quantum annealer is constituted by two main elements:

1. a Quantum Processor Unit (QPU) that implement the quantum annealing algorithm;
2. a conventional computer containing a front end server (solver application programming interface, SAPI), and a back end system that communicates with the QPU.

Objective functions

To understand how to express a problem in a form that the D-Wave system can solve, we must first develop an objective function, which is a mathematical expression of the energy of a system as a function of binary variables representing the qubits. In most cases, the lower is the energy of the objective function, the better the solution. Sometimes any low-energy state is an acceptable solution to the original problem; for other problems, only optimal solutions are acceptable. The best solutions typically correspond to the global minimum energy in the solution space.

Ising type problems

The NP problem to solve has to be translated to an input for an equivalent Ising model. Then, this Ising problem is transformed into a so-called *native* problem that matches the qubits connections topology of the D-Wave processor. The transformation into Ising model is done using standard techniques of NP-completeness theory. Binary objective functions can be represented as graphs. The Ising model problem is defined as follows: given a graph $G = (V,E)$ with weights h_i (called *fields*) on vertices and J_{ij} (called *couplers*) on edges, find an assignment of *spins* $S = (s_1,...,s_n)$ to vertices, with $s_i \in \{-1,+1\}$ so they minimize the *energy function* $H(S)$

$$H(S) = \sum_{i \in V} h_i s_i + \sum_{(i,j) \in E} J_{ij} s_i s_j.$$

QUBO type problems

Quadratic unconstrained binary optimization (QUBO) is a pattern matching technique, common in machine learning applications. QUBO is an NP hard problem. Examples of problems that can be formulated as QUBO problems are the maximum cut, graph coloring and the partition problem [13]. For a QUBO problem, variables are TRUE and FALSE (corresponding qubits states correspond to 1 and 0 values). A QUBO problem is defined using an upper-diagonal matrix Q, which is an $N \times N$ upper-triangular matrix of weights, and x a vector of binary variables. This matrix is used to define the function $f(x)$ to be minimized

$$f(x) = \sum_i Q_{ii} x_i + \sum_{i<j} Q_{ij} x_i x_j,$$

where the diagonal terms Q_{ii} are the linear coefficients and the nonzero off-diagonal terms are the quadratic coefficients Q_{ij}. This minimization of $f(x)$ is equivalent to

$$\min_{x \in \{0,1\}^n} x^t Q x.$$

The objective function of the QUBO problem might be reformulate as the following expression in scalar notation

$$H_{\text{QUBO}}(a_i, b_{ij}, q_i) = \sum_i a_i q_i + \sum_{i<j} b_{ij} q_i q_j.$$

A QUBO problem might be transformed to an Ising problem with the following correspondence

$$s = 2q - 1.$$

QPU topology

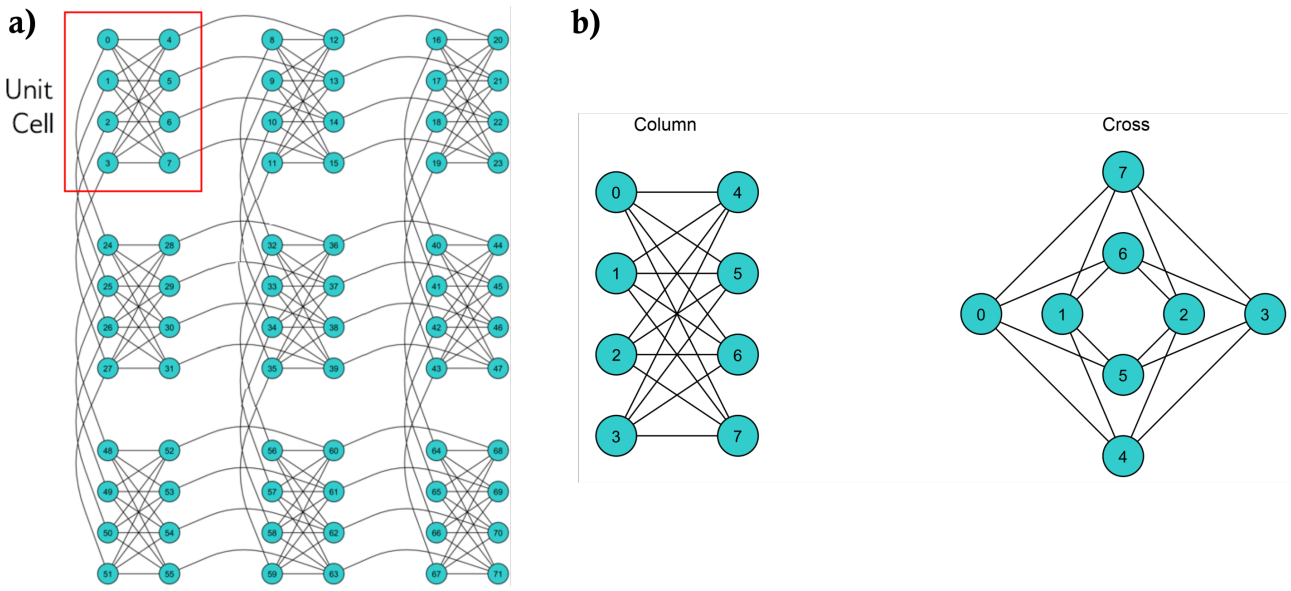


Fig. VI.4. a) A 3×3 Chimera graph. Qubits are arranged in 9 unit cells. b) Cross or column layout of qubits in a unit cell. Figures extracted from `https://docs.dwavesys.com/docs/latest/c_gs_4.html`.

The D-Wave QPU is a lattice of interconnected qubits. While some qubits connect to others via couplers, the D-Wave QPU is not fully connected. Instead, the qubits interconnect in an architecture known as Chimera (Fig. VI.4). The Chimera architecture comprises sets of connected unit cells, each with four horizontal qubits connected to four vertical qubits via couplers. Unit cells are tiled vertically and horizontally with adjacent qubits connected, creating a lattice of sparsely connected qubits. The notation *CN* refers to a Chimera graph

consisting of an $N \times N$ grid of unit cells. D-Wave's most recent QPU, 2000Q, supports a $C16$ Chimera graph, with 2048 qubits logically mapped into a 16×16 matrix of unit cells made of 8 qubits. Within a unit cell, the qubits have bipartite connectivity. The set of qubits and couplers that are available for computation is known as the working graph. The yield of a working graph is typically less than the total number of qubits and couplers that are fabricated and physically present in the QPU.

An Ising Model problem defined on a general graph G must be translated to an equivalent problem that match the *working graph* of the QPU which is a subgraph $\Gamma \subset C$ of a Chimera graph. Translating a problem on G to an identical problem on Γ involves a process called *minor embedding*, provided by SAPI in the case of D-wave quantum computers.

3.5. The quantum annealing algorithm

The Ising model implemented in the QPU is described by the following Hamiltonian

$$\mathscr{H}_p = \sum_i h_i \sigma_i^z + \sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z,$$

where σ_i^z is the Pauli matrix z acting on spin i, h_i is the magnetic field on spin i and J_{ij} the coupling strength between spins i and j. The ground state of $\mathscr{H}_p$ corresponds to a spin configuration $S = (s_1, ..., s_n) \in \{-1, +1\}^n$ that minimises the Ising energy function.

Quantum annealing uses an analog process to find optimal and near-optimal solutions to the energy function $H(S)$. A quantum annealing algorithm consists in four components

— an *initial Hamiltonian* $\mathscr{H}_i$, which describes initial conditions;
— the *problem Hamiltonian* $\mathscr{H}_p$ described above;
— a pair of *path functions* $A(s)$ and $B(s)$ that control the transition from $\mathscr{H}_i$ to $\mathscr{H}_p$ over a time interval $s : 0 \to 1$ (in current D-wave's systems, theses functions are related by $B(s) = 1 - A(s)$;
— a parameter t_a that specifies the total time for the transition (in the microsecond range).

Quantum annealing uses an adiabatic quantum evolution approach to approximate solutions of the energy function $H(S)$. This is done by traversing from the ground state of an initial Hamiltonian $\mathscr{H}_i$ to a ground state of a final Hamiltonian $\mathscr{H}_p$. According to this scheme, a time dependent Hamiltonian is defined as

$$\mathscr{H}(t) = A(\tau)\mathscr{H}_i + B(\tau)\mathscr{H}_p,$$

where $\tau = t/t_a$ for $0 \leq t \leq t_a$ and t_a is the total annealing time. Usually, the ground state of the initial state $\mathscr{H}_i$ is easy to prepare and the ground state of the final Hamiltonian $\mathscr{H}_p$ codifies the solution of our problem. $A(s)$ and $B(s)$ are chosen such that at time $\tau = 0$, $\mathscr{H}_i$ is predominant in $\mathscr{H}(0)$. As time evolution goes from $\tau = 0$ to $\tau = 1$, the influence of $\mathscr{H}_p$ increases while $\mathscr{H}_i$ fades away.

The mean idea of quantum annealing is that it is possible to prepare qubits in the ground state of the known and chosen Hamiltonian $\mathscr{H}_i$. Then, parameters defining $\mathscr{H}(t)$ evolves in time, slow enough so the global wavefunction of the qubits evolves adiabatically, and stay in the ground state of the instantaneous Hamiltonian $\mathscr{H}(t)$. At the end of the process, the wavefunction obtained is expected to be the ground state: one just has to measure the state of each qubit to reconstruct the corresponding state.

An important question is how slow the evolution needs to be in order to assure adiabaticity. According to the adiabatic theorem, a quantum mechanical system subjected to gradually changing external conditions adapts its functional form changes occurs slowly enough. For a non-degenerate spectrum with a gap between the ground state and first excited state, the adiabatic evolution is assured if the evolution time τ satisfies the following condition

$$\tau \gg \frac{\max_{0\leq t\leq\tau}\left[\left|\langle\phi_0(t)\left|\frac{\mathrm{d}\mathscr{H}(t)}{\mathrm{d}t}\right|\phi_1(t)\rangle\right|\right]}{\min_{0\leq t\leq\tau}\left[\Delta^2(t)\right]},$$

where $|\phi_0(t)\rangle$ and $|\phi_1(t)\rangle$ being respectively the instantaneous ground state and the first excited state of the total Hamiltonian $\mathscr{H}(t)$ and $\Delta(t)$ the instantaneous gap between the ground state and the first excited state energies [9].

3.6. Quantum annealing based calculation

Quantum annealing is the key component of a computation made of four steps

Programming/Initialisation The weights (h_i, J_{ij}) are loaded onto the control system and qubits are placed in an initial superposition state according to $\mathscr{H}_i$.

Annealing Adiabatic transition from $\mathscr{H}_i$ to $\mathscr{H}_p$ over a time t_a.

Readout At the end of the transition, qubits have states according to $\mathscr{H}_p$ which matches $E(s)$. Qubits values are read, yielding solution S to the input.

Resampling Since any quantum computation is probabilistic, there is always a non negligible probability that the computation does not finish in the ground state of $\mathscr{H}_p$. Given the relatively high initialisation times, it is cost-effective to repeat the anneal-readout cycle many times per input.

In D-wave systems, the initial Hamiltonian is fixed. The problem Hamiltonian, anneal time t_a and the number of resampling steps R are supplied by the user. Beginning with the 2000Q system, the user may also modify the transition by specifying *anneal path offsets*. These offsets are deviations from the default anneal path determined by $A(s)$ and $B(s)$. The total calculation time $T(R)$ required by the QPU to return a sample of R solutions to one input instance is then

$$T(R) = t_{\text{program}} + R(t_a + t_{\text{read}})$$

For a D-wave 2000Q systems, typical values are $t_{\text{program}} = 9$ ms, $t_a \geq 5\,\mu$s and $t_{\text{read}} = 120\,\mu$s. It results in a calculation time for 1000 solutions of typically 149 ms. The annealing step is just a tiny fraction of $T(R)$.

3.7. Advantages and limitations of quantum annealing

Quantum annealer isn't a general-purpose computer, in that it can only solve a set of problems that can be structured as energy minimizations. And even on those problems, current hardware generally can't outperform algorithms implemented on standard computers. But a key reason D-wave has been selling time on their machines before they have a clear advantage is to give developers the chance to identify the sorts of problems where quantum annealing will prove to be effective. It is quite similar to programmable quantum computers.

Quantum annealing chips are not design to achieve multiple qubits gates manipulations and consequently they are not limited by the corresponding error rate. Moreover, due to its adiabatic nature, and given that a quantum system naturally relaxes towards its ground states, quantum annealing is for more tolerant to noise and errors on qubits. That is why QPU have quite more qubits in the case of quantum annealing, but it is not possible to programm a quantum circuit on such chip.

Quantum annealing based quantum computers have a important number of qubits but are not a programmable computer. They do not permit to implement quantum circuit with multi-qubits quantum gates. Contrary to universal quantum computers, they are limited to specific application, which might be described formally as an Ising problem. It is however useful in many optimization problem of huge interest such as traveling salesman problem, the maxcut problem or even for machine-learning. However quantum advantage has not been yet demonstrated with such annealing-based quantum computers.

By the end of 2019, the most powerful system commercially available from D-wave is the D-wave 2000Q system, released in January 2017. The QPU is made of 2048 qubits (while programmable computers are still in the 50-100 qubits range), and 6,016 couplers. It results in 128,472 Josephson junctions in total on the chip, with 200 I/O lines. It operates in a dilution He cryostat, down to 15 mK, with a total power consumption of 25 kW.

In February 2019 D-Wave announced their next-generation Pegasus P16 quantum processor chip, announcing that it would be "the world's most connected commercial quantum system," with 15 connections per qubit instead of 6 [10]. The next-generation system would use the Pegasus P16 chip. It would have 5,640 qubits and reduced noise, with 40,484 couplers and 1,030,000 Josephson junctions. It is announced to be available in mid-2020.

3.8. The traveling Salesman Problem

In the traveling salesman problem (TSP), there are N cities placed randomly in a country having a definite metric to calculate the inter-city distances. A salesman has to make a tour to cover every city and finally come back to the starting point. The problem is to find a tour of minimum length. An instance of the problem is given by a set $\{d_{ij};\ i,j \in [\![1,N]\!]\}$, where d_{ij} corresponds to the distance between the i-th and the j-th city, or equivalently, the cost for going from the former to the later. We mainly focus on the results of symmetric case, where $d_{ij} = d_{ji}$. The problem can be cast into the form where one minimizes an Ising Hamiltonian under some constraints, as shown below. A tour can be represented by an $N \times N$ matrix $\mathscr{T}$ with elements either 0 or 1. In a given tour, if the city j is visited immediately after visiting city i, then $\mathscr{T}_{ij} = 1$, otherwise $\mathscr{T}_{ij} = 0$. Generally an additional constraint is imposed that one city has to be visited once and only once in a tour. Any valid tour with the above restriction may be represented by a $\mathscr{T}$ matrix whose each row and each column has one and only one element equal to 1 and rest all are 0s. For a symmetric metric, a tour and its reverse tour have the same length, and it is more convenient to work with an undirected tour matrix

$$\mathscr{U} = \frac{1}{2}\left(\mathscr{T} + \mathscr{T}^T\right)$$

where $\mathscr{T}^T$, the transpose of $\mathscr{T}$, represents the reverse of the tour given by $\mathscr{T}$. Clearly, $\mathscr{U}$ must be a symmetric matrix having two and only two distinct entries equal to 1 in every row and every column, no two rows being identical, and so is not any two columns. In terms of $\mathscr{U}_{ij}$s, the length of a tour can be represented by

$$\mathscr{H} = \frac{1}{2}\sum_{i,j=1}^{N} d_{ij}\mathscr{U}_{ij}.$$

One can rewrite the above Hamiltonian in terms of Ising spins $\mathscr{S}_{ij}$s as

$$\mathscr{H}_{\text{TSP}} = \frac{1}{2}\sum_{i,j=1}^{N} d_{ij}\frac{1+\mathscr{S}_{ij}}{2}.$$

where $\mathscr{S}_{ij} = 2\mathscr{U}_{ij} - 1$ are the Ising spins. The Hamiltonian is similar to that of a non-interacting Ising spins on a $N \times N$ lattice, with random fields d_{ij} on the lattice points $\{i,j\}$. The problem is to find the ground state of this Hamiltonian subjected to these constraints. There are N^2 Ising spins, which can assume 2^{N^2} configurations in absence of any constraint, but the constraint here reduces the number of valid configurations to that of the number of distinct tours, which is $\frac{N!}{2N}$.

Mainly two distinct classes of TSP are studied: one with an Euclidean d_{ij} in finite dimension (where d_{ij} are strongly correlated through triangle inequalities, which means, for any three cities A, B and C, the sum of any two of the side AB, BC and CA must be greater than the remaining one), and the other with random d_{ij} in infinite dimension.

Further analysis of this problem might be found in reference [9].

Appendix A

IBM's Q experience

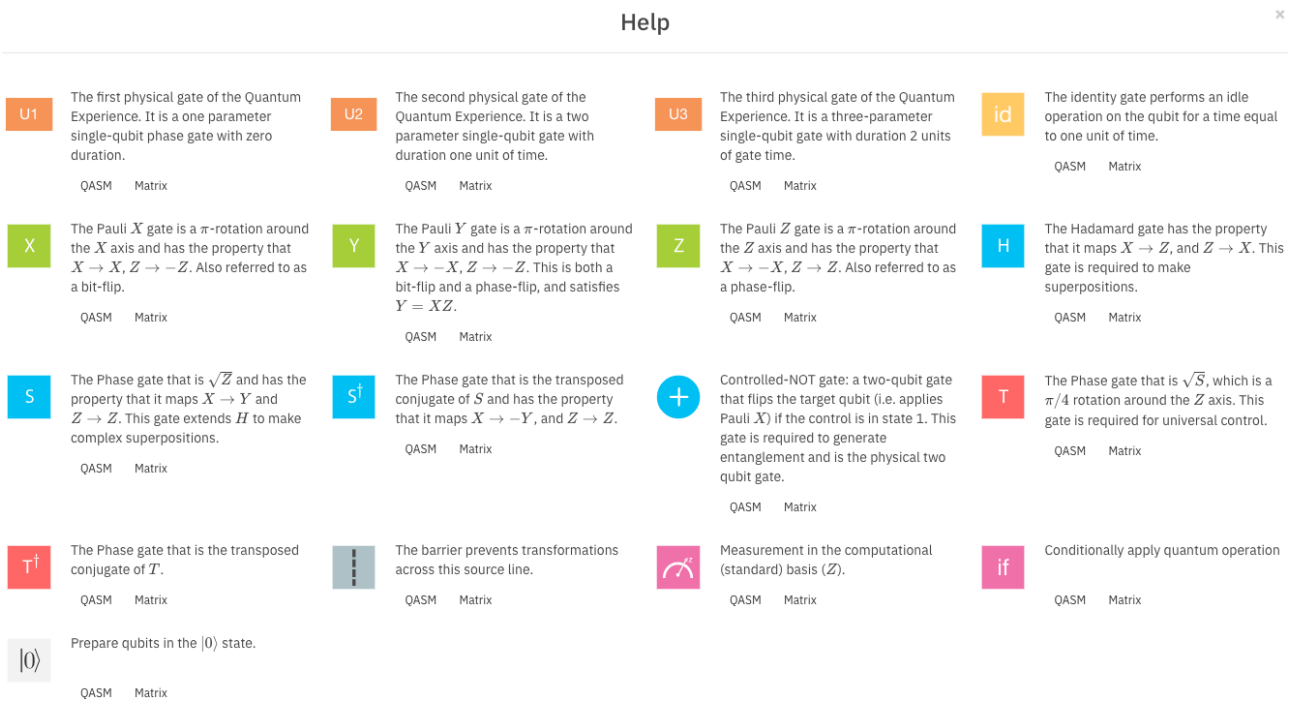


Fig. A.1. Quantum gates available on IBM's Q experience quantum computers.

A. IBM's Q experience

Bibliography

[1] CES 2018: Intel's 49-Qubit Chip Shoots for Quantum Supremacy - IEEE Spectrum.

[2] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics, 2010.

[3] Scott Aaronson and Lijie Chen. Complexity-theoretic foundations of quantum supremacy experiments, 2016.

[4] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, October 2019.

[5] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, 52:3457–3467, Nov 1995.

[6] Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J. Bremner, John M. Martinis, and Hartmut Neven. Characterizing quantum supremacy in near-term devices. *Nature Physics*, 14(6):595–600, June 2018.

[7] Sergio Boixo, Troels F. Rønnow, Sergei V. Isakov, Zhihui Wang, David Wecker, Daniel A. Lidar, John M. Martinis, and Matthias Troyer. Evidence for quantum annealing with more than one hundred qubits. *Nature Physics*, 10(3):218–224, March 2014.

[8] Michael J. Bremner, Christopher M. Dawson, Jennifer L. Dodd, Alexei Gilchrist, Aram W. Harrow, Duncan Mortimer, Michael A. Nielsen, and Tobias J. Osborne. Practical scheme for quantum computation with any two-qubit entangling gate. *Phys. Rev. Lett.*, 89:247902, Nov 2002.

[9] Arnab Das and Bikas K. Chakrabarti. Quantum Annealing and Analog Quantum Computation. *Reviews of Modern Physics*, 80(3):1061–1081, September 2008. arXiv: 0801.2193.

[10] Nike Dattani, Szilard Szalay, and Nick Chancellor. Pegasus: The second connectivity graph for large-scale quantum annealing hardware. *arXiv:1901.07636 [quant-ph]*, January 2019. arXiv: 1901.07636.

[11] A. B. Finnila, M. A. Gomez, C. Sebenik, C. Stenson, and J. D. Doll. Quantum Annealing: A New Method for Minimizing Multidimensional Functions. *Chemical Physics Letters*, 219(5-6):343–348, March 1994. arXiv: chem-ph/9404003.

[12] Neil A. Gershenfeld and Isaac L. Chuang. Bulk spin-resonance quantum computation. *Science*, 275(5298):350–356, 1997.

[13] Fred Glover, Gary Kochenberger, and Yu Du. A Tutorial on Formulating and Using QUBO Models. *arXiv:1811.11538 [quant-ph]*, November 2019. arXiv: 1811.11538.

[14] Posted 24 May 2017 | 15:00 GMT. Google Plans to Demonstrate the Supremacy of Quantum Computing - IEEE Spectrum.

[15] S. M. Hamdi, S. T. Zuhori, F. Mahmud, and B. Pal. A compare between shor's quantum factoring algorithm and general number field sieve. In *2014 International Conference on Electrical Engineering and Information Communication Technology*, pages 1–6, April 2014.

[16] Nicola Jones. Computing: The quantum company. *Nature News*, 498(7454):286, June 2013.

[17] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum Annealing in the Transverse Ising Model. *Physical Review E*, 58(5):5355–5363, November 1998. arXiv: cond-mat/9804280.

[18] James King, Sheir Yarkoni, Jack Raymond, Isil Ozfidan, Andrew D. King, Mayssam Mohammadi Nevisi, Jeremy P. Hilton, and Catherine C. McGeoch. Quantum annealing amid local ruggedness and global frustration, 2017.

[19] Edwin Pednault, John A. Gunnels, Giacomo Nannicini, Lior Horesh, Thomas Magerlein, Edgar Solomonik, Erik W. Draeger, Eric T. Holland, and Robert Wisnieff. Breaking the 49-qubit barrier in the simulation of quantum circuits, 2017.

[20] Edwin Pednault, John A. Gunnels, Giacomo Nannicini, Lior Horesh, and Robert Wisnieff. Leveraging Secondary Storage to Simulate Deep 54-qubit Sycamore Circuits. *arXiv:1910.09534 [quant-ph]*, October 2019. arXiv: 1910.09534.

[21] John Preskill. Quantum computing and the entanglement frontier. *arXiv:1203.5813 [cond-mat, physics:quant-ph]*, November 2012. arXiv: 1203.5813.

[22] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018. arXiv: 1801.00862.

[23] Troels F. Rønnow, Zhihui Wang, Joshua Job, Sergio Boixo, Sergei V. Isakov, David Wecker, John M. Martinis, Daniel A. Lidar, and Matthias Troyer. Defining and detecting quantum speedup. *Science*, 345(6195):420–424, July 2014.

[24] Yaoyun Shi. Both toffoli and controlled-not need little help to do universal quantum computing. *Quantum Information & Computation*, 3:84–92, 01 2003.

[25] Tommaso Toffoli. Reversible computing. In Jaco de Bakker and Jan van Leeuwen, editors, *Automata, Languages and Programming*, pages 632–644, Berlin, Heidelberg, 1980. Springer Berlin Heidelberg.

[26] L. M. K. Vandersypen and I. L. Chuang. Nmr techniques for quantum control and computation. *Reviews of Modern Physics*, 76(4):1037–1069, Jan 2005.

[27] Lieven M. K. Vandersypen, Matthias Steffen, Gregory Breyta, Costantino S. Yannoni, Mark H. Sherwood, and Isaac L. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414(6866):883–887, December 2001.

[28] Davide Venturelli, Salvatore Mandrà, Sergey Knysh, Bryan O'Gorman, Rupak Biswas, and Vadim Smelyanskiy. Quantum optimization of fully connected spin glasses. *Phys. Rev. X*, 5:031040, Sep 2015.