SAPIENZA
UNIVERSITÀ DI ROMA

"Sapienza" University of Rome
Faculty of Information Engineering,
Informatics and Statistics
Department of Computer Science

Quantum Computing

Author
Alessio Bandiera

October 4, 2025

Contents

Information and Contacts

Personal notes and summaries collected as part of the *Quantum Computing* course offered by the degree in Computer Science of the University of Rome "La Sapienza".

Further information and notes can be found at the following link:
https://github.com/aflaag-notes. Anyone can feel free to report inaccuracies, improvements or requests through the Issue system provided by GitHub itself or by contacting the author privately:

- Email: **alessio.bandiera02@gmail.com**
- LinkedIn: **Alessio Bandiera**

The notes are constantly being updated, so please check if the changes have already been made in the most recent version.

Suggested prerequisites:

TODO

Licence:

1

Introduction on Quantum Computation

1.1 The Qubit

Quantum computing is a rapidly developing discipline that explores how the laws of quantum mechanics can be used to *process information*. While classical computation is based on *bits* that take values of either 0 or 1, quantum computation relies on quantum bits, or **qubits**. A qubit can exist in a "superposition" of classical states, allowing it to encode richer information than a single bit. Furthermore, qubits can exhibit particular properties that enable forms of information processing with no classical counterpart. Such properties provide the foundation for algorithms that promise to solve certain problems more efficiently than their classical analogues.

The design of quantum algorithms requires a different perspective from that of classical computation. In classical computer science, the majority of widely studied algorithms are *deterministic*, meaning that for a given input they will always produce the *same output*. Some algorithms are *randomized*, making use of probability to achieve efficiency or simplicity, yet even in those cases the computation itself is ultimately classical in nature. In fact, to achieve such *randomness* classical algorithms employ **pseudo-random number generation**, which must ultimately produce finite sequences.

Quantum computation, by contrast, *incorporates probability* at its core. The act of measuring a quantum system does not reveal a single, predetermined result, but rather yields one outcome from a distribution of possible outcomes, with probabilities governed by the system's quantum state. This fundamental probabilistic characteristic distinguishes quantum algorithms from their classical counterparts.

In fact, in the context of quantum computing we are often interested in **probabilistic algorithms**: for such algorithms, a given input i can lead to a finite set of possible outputs $o_1, \dots, o_N$, each occurring with an associated probability $p_1, \dots, p_N$ — where $\sum_{i=1}^{n} p_i = 1$.

As previously mentioned, the quantum equivalent of the classical bits are the **qubit**, but define the qubits we first need to define some preliminary concepts. The following vectors

are called **basis states**

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

and they represent the classical bits 0 and 1 respectively — the notation above is called "braket" notation and it will be explored in greater detail in later sections.

So what is a qubit? A qubit is the basic unit of information in quantum computing, which represents a **superposition** of states simultaneously — note that we will refer to qubits and their states interchangeably, since the only thing that we care about a qubit is its own state

In practice, the state of a qubit is a vector

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

where $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$ are called **probability amplitudes**. But why are we talking about probabilities in the first place? The "true" state of a qubit **cannot be observed**, and we say that the qubit is in a *superpotion* of $|0\rangle$ and $|1\rangle$ in the sense that α and β describe the probabilities of getting either states once the qubit is measured. This is because to know the value of a qubit we have to *measure it*, and the measurement operation itself will make the qubit *collapse* into either $|0\rangle$ or $|1\rangle$ with probabilities $|\alpha|^2$ and $|\beta|^2$ respectively, i.e.

$$\Pr[\text{measured qubit is } |0\rangle] = |\alpha|^2 \qquad \Pr[\text{measured qubit is } |1\rangle] = |\beta|^2$$

To use a more compact notation, we will denote this property as follows:

$$\alpha\,|0\rangle + \beta\,|1\rangle \begin{cases} |0\rangle & @\ |\alpha|^2 \\ |1\rangle & @\ |\beta|^2 \end{cases}$$

where the @ notation (read as "at") denotes the probabilty of the corresponding outcome. Note that if we measure a collapsed qubit we will keep observing the same state indefinitely.

In reality, to be precise qubits actually collapse into any multiple $z\,|0\rangle$ or $z\,|1\rangle$, where $z \in \mathbb{C}$ is a complex number such that $|z| = 1$, but this is not relevant from a physical point of view. In fact, for any θ physicists treat $|\psi\rangle = |0\rangle$ and $|\psi'\rangle = e^{i\theta}\,|0\rangle$ as the *same physical state*, because probabilities depend on squared magnitudes and thus

$$\left|e^{i\theta}\alpha\right|^2 = |\alpha|^2$$

(and the same applies for β too) even though $|\psi\rangle$ and $|\psi'\rangle$ are different vectors mathematically.

1.2 Qubit operations

What can we do with qubits other then *measure them*? The operations that can be applied on qubits are restricted to **unitary transformations**.

Definition 1.1: Unitary transformation

A transformation U is said to be **unitary** if it preserves the norm of its input vector, i.e.

$$\forall v \quad ||Uv|| = ||v||$$

For instance, the identity matrix I is an example of trivial unitary transformation, but also the NOT matrix, which is the following

$$\text{NOT} := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

which has the effect of *swapping* the input basis state

$$\text{NOT}\,|0\rangle = |1\rangle \qquad \text{NOT}\,|1\rangle = |0\rangle$$

This matrix behaves as the classical NOT gate with the usual bits in classical computing, in fact will refer to *transformations* and *gates* interchangeably.

Another very important transformation is represented by the **Hadamard gate**, which is the following matrix

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

This matrix has the effect of "mapping" classical states into superpositions:

$$H\,|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \begin{cases} |0\rangle & @\ \frac{1}{2} \\ |1\rangle & @\ \frac{1}{2} \end{cases}$$

For instance, in this example given $|0\rangle$ which represents the classical bit 0, we get a qubit as output of the linear transformation. In general, the operation performed by the Hadamard gate can be represented as follows:

$$\forall a \in \{0,1\} \quad \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^a\,|1\rangle\right)$$

As a side note, as we mentioned at the beginning of the chapter quantum mechanics has randomness intrinsically, and since the operation $H\,|0\rangle$ returns a qubit that has 50% of probability of being either $|0\rangle$ or $|1\rangle$ once measured, this operation provides a true random number generator.

side note on Hadamard gate really implemented?

TODO

bloch sphere too bored to do it

1.2.1 The tensor product

So far we have dealt with only one qubit at a time, but what if we have two qubits? First, let's look at the classical counterpart. If we take two bits $a, b \in \{0,1\}$, we can represent 4 possible binary numbers, namely 00, 01, 10 and 11, which we can algebraically obtain by computing the usual cartesian product

$$\{0,1\}^2 = \{0,1\} \times \{0,1\} = \{(0,0),(0,1),(1,0),(1,1)\}$$

Note that in the cartesian products it holds that:

- the length of the tuples of the product is linear w.r.t. the number of factors of the cartesian products — in this case, 2
- each element of a tuple is *independent* from the other elements of the tuple

How can we evaluate all the possible states that two qubits can represent, instead? To answer this question, we need to introduce a new operator, which is called **tensor product**. Given two vectors $\begin{pmatrix} a \\ b \end{pmatrix}$ and $\begin{pmatrix} c \\ d \end{pmatrix}$, their tensor product is defined as follows

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} := \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix}$$

Hence, consider two qubits

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \qquad |\phi\rangle = \beta |0\rangle + \beta_1 |1\rangle = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$$

To obtain all the possible states of $|\psi\rangle$ and $|\phi\rangle$ we just have to compute the tensor product between them, which is

$$\begin{aligned} |\psi\rangle \otimes |\phi\rangle &= \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \otimes \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \\ &= \alpha_0\beta_0 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \alpha_0\beta_1 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \alpha_1\beta_0 \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \alpha_1\beta_1 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \end{aligned}$$

At the beginning of the chapter we defined $|0\rangle$ and $|1\rangle$ to be $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ without providing an explaination; now that we are dealing with more than 2 dimensions we can show why such names are used. In fact, we will use the following naming convention

$$|00\rangle := \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad |01\rangle := \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad |10\rangle := \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad |11\rangle := \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

and in general it holds that

$$|\text{bin}(i)\rangle = e_i$$

where $\text{bin}(i)$ represents for the binary representation of i, and e_i is the i-th vector of the canonical basis. This implies that we can rewrite the previous tensor product as follows:

$$|\psi\rangle \otimes |\phi\rangle = \alpha_0\beta_0 |00\rangle + \alpha_0\beta_1 |01\rangle + \alpha_1\beta_0 |10\rangle + \alpha_1\beta_1 |11\rangle = \sum_{i,j\in\{0,1\}} \alpha_i\beta_j |ij\rangle$$

As a final note, it can be easily proven that

$$\forall i, j \in \{0, 1\} \quad |i\rangle \otimes |j\rangle = |ij\rangle$$

For example, given two qubits

$$|\phi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad |\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

we get that

$$\begin{aligned} |\psi\rangle \otimes |\phi\rangle &= \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \otimes \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \\ &= \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \\ &\begin{cases} |0\rangle \text{ and } |0\rangle & @ \ \frac{1}{4} \\ |0\rangle \text{ and } |1\rangle & @ \ \frac{1}{4} \\ |1\rangle \text{ and } |0\rangle & @ \ \frac{1}{4} \\ |1\rangle \text{ and } |1\rangle & @ \ \frac{1}{4} \end{cases} \end{aligned}$$

where the probabilities at the end refer to the two individual qubits. To recap, in general the tensor product $|\psi\rangle \otimes |\phi\rangle$ of two qubits encodes the superposition of 4 basis states, namely $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$.

Moreover, the following property can be proved easily.

Proposition 1.1: Distributive property of $\otimes$

Given three qubits $|\psi\rangle, |\phi\rangle$ and $|\chi\rangle$, it holds that

$$(|\psi\rangle + |\phi\rangle) \otimes |\chi\rangle = |\psi\rangle \otimes + |\phi\rangle \otimes |\chi\rangle$$

Another very important gate in quantum computing is the so called **Controlled NOT (CNOT)** gate, which is defined as follows:

a	b	CNOT(a, b)
0	0	0
0	1	1
1	0	1
1	1	0

In fact, the names comes from the fact that the first input a is called *control bit*, which if set to 1 will flip the *target bit* b — in fact, in its implementation what actually happens

is that b's wire itself is flipped. Therefore, in general we will write that

$$\text{CNOT}(a, b) = (a, a \oplus b)$$

First, we observe that this function is clearly not invertible, since for instance if we know that the output is 0 we still need the input a to evaluate if b was 0 or 1. Hence, to solve this issue we usually pair the output of CNOT with a itself, so that we can actually invert the computation.

Moreover, so far we only dealt with transformation that only expected one qubit argument as input, but the CNOT gate would certainly need 2 inputs to perform any computation, so how do we provide two inputs to it? As we showed before, we konw that

$$\forall i, j \in \{0, 1\} \quad |i\rangle \otimes |j\rangle = |ij\rangle$$

which directly implies that the vector $|ij\rangle$ encapsulated two qubits at once without ambiguity. Hence, we can actually leverage the tensor product to provide the input to the CNOT matrix, such that the quantum CNOT will behave as follows

$$\text{CNOT}(|a\rangle \otimes |b\rangle) = |a\rangle \otimes |a \oplus b\rangle$$

Hence, the matrix that behaves as such is the following

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

which expects a 4×1 input vector, and outputs a 4×1 output vector as well.

1.2.2 Quantum circuits

Now that we introduced a couple of quantum gates, we can show how computation is actually represented in quantum computing. For instance, consider the following picture:

Figure 1.1: The NOT gate.

In this example, we have 1 single input qubit, namely q, and the box labeled with an X represents the NOT gate. We observe that, by convetion, all qubits in quantum circuits are assumed to be set to $|0\rangle$.

In the following example, instead, it is represented how the Hadamard gate looks like in quantum circuits.

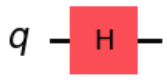

Figure 1.2: The Hadamard gate.

Moreover, if we consider two qubits as inputs q_0 and q_1, we can represent the CNOT operator as follows:

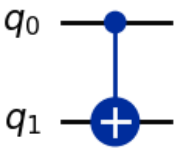

Figure 1.3: The CNOT gate.

We observe that q_1 then becomes the output of the CNOT operation, and q_0 remains unchanged. Lastly, the measurement operation is represented with the following picure:

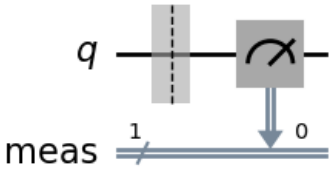

Figure 1.4: The measure operation.

In particular, in this cirtuit we see that:

- the vertical "double lines" reprents *classical bits*
- the number 1 next to the label "meas" indicates the number of qubits that have been measured
- the number 0 is the index of the measured qubit

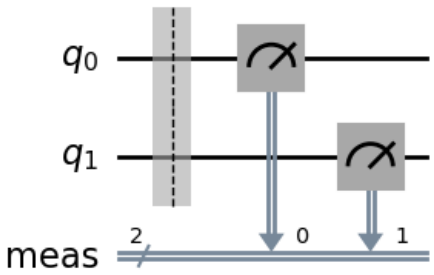

Figure 1.5: An exmaple of measurement of 2 qubits

1.3 EPR

TODO

talk about it

1.4 No-cloning theorem and quantum teleportation

1.4.1 No-cloning theorem

So far we described gates that enable to do stuff and to EPR, but what if we want to copy a qubit? can we do it like we can in normal bits? In 1982 Wootters and Zurek [WZ82] proved the so called **no-cloning theorem**, which states that it is impossible to create an independent and identical copy of an arbitrary *unknown* quantum state.

write this better after epr

Theorem 1.1: No-cloning theorem

There is no quantum transformation that copies an unknown quantum state.

Proof. by way of contradiction, suppose that there exists such a transformation CP that is able to copy an unknown quantum state — and in particular, we observe that such transformation would have to be linear. But clearly, in order to have a copy we need to actually *store* it somewhere, so we can assume that CP has to take two inputs, one being the state that we want to copy and the other one being the state that we want to replace with the copy of the first one. In other words, we are assuming that

$$\exists y \forall x \quad \mathrm{CP}(x \otimes y) = x \otimes x$$

Now, through some algebraic manipulation we get that

$$
\begin{aligned}
& \exists y \forall x \quad \mathrm{CP}(x \otimes y) = x \times x \\
\equiv & \exists y \forall x, a \quad \mathrm{CP}((x+a) \otimes y) = (x+a) \otimes (x+a) \\
\equiv & \exists y \forall x, a \quad \mathrm{CP}(x \otimes y + a \otimes y) = (x+a) \otimes (x+a) && \text{(by distributivity of } \otimes) \\
\equiv & \exists y \forall x, a \quad \mathrm{CP}(x \otimes y) + \mathrm{CP}(a \otimes y) = (x+a) \otimes (x+a) && \text{(by linearity of CP)} \\
\equiv & \exists y \forall x, a \quad x \otimes x + a \otimes a = x \otimes x + x \otimes a + a \otimes x + a \otimes a && \text{(by definition of CP)} \\
\equiv & \exists y \forall x, a \quad \mathbf{0} = x \otimes a + a \times x
\end{aligned}
$$

which should be true for every x and every a, however it does not hold for $x = |0\rangle$ and $a = |1\rangle$, thus raising a contradiction $\lightning$. □

The no-cloning theorem represents an inherent limitation of quantum computation, and has direct impacts on **quantum cryptography** and **quantum error correction**, but must importantly it directly impacts a phenomenon called **quantum teleportation**

1.4.2 Quantum teleportation

So far we saw that quantum states cannot be cloned, but can we at least *send* them? Suppose that Alice wants to send Bob $|\psi\rangle$, described by some α and β. Clearly, the only thing that Bob has to receive are indeed the probability amplitudes of $|\psi\rangle$, so even if Alice cannot clone her quantum state, nothing prevents her to build a quantum circuit which *destroys* her $|\psi\rangle$ but does allow Bob to receive α and β.

The following is the circuit that allows **quantum teleportation**:

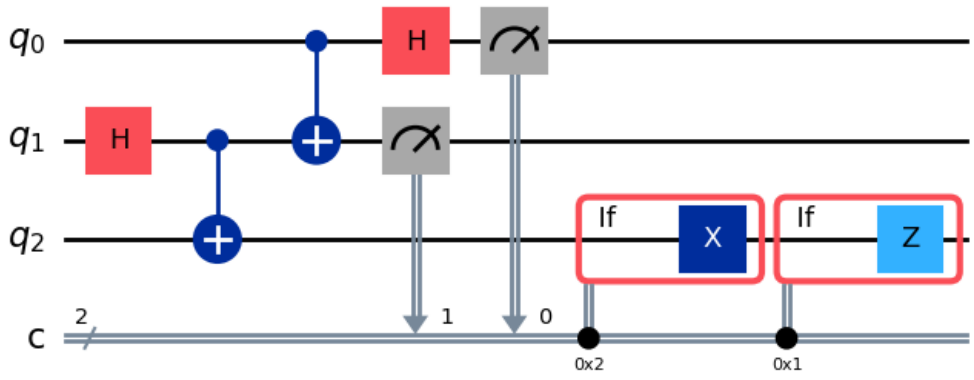

Figure 1.6: The Quantum Teleportation circuit.

There is quite a lot to unpack in this diagram. First, the quantum state that we want to teleport is q_0 in this diagram, and it will be teleported in q_2 at the end of the quantum computation.

In the first part of the circuit, we see that q_1 and q_2 are entangled in a Bell state thanks to the Hadamard and the CNOT gates. In a real-world scenario, we will assume that q_1

parlane nell'EPR

and q_2 are given to Alice and Bob respectively, and quantum mechanics will guarantee that the teleportation will work even our two protagonists are thousands of kilometers away from each other.

After entangling q_1 and q_2, we have the part of circuit that concerns Alice: in fact, she must apply a CNOT to her entangled qubit q_1, controlled by q_0, and then apply a Hadamard transformation to q_0. At this point, the circuit must apply a measurement to both q_0 and q_1 — and in particular, this operation will *destroy* the original state as previously anticipated.

Finally, it's Bob's turn: to obtain the original quantum state of q_0, the only thing he needs to do is first apply a CNOT to his entangled qubit q_2, controlled by q_1's outcome, followed by an application of a CZ , controlled by q_0's outcome instead — we observe that this part is indicated in the diagram through the 0×2 and 0×1 labels respectively. In fact, in the label $0 \times X$ the number X represents the hexadecimal representation of the binary number obtained by joining the classical bits all together — for instance, in this circuit we have that 2 represents 10, meaning that only q_1 will be checked in the condition, and 1 represents 01, which means that only q_0 will be the control bit.

spiega le pauli matrices

Bibliography

[WZ82] W. K. Wootters and W. H. Zurek. “A single quantum cannot be cloned”. In: *Nature* 299.5886 (Oct. 1982), 802–803. ISSN: 1476-4687. DOI: 10.1038/299802a0. URL: http://dx.doi.org/10.1038/299802a0.