

# Quantum Computing

## Notes of the course

Luca Leoni

CHAPTER 1	INTRODUCTION TO QUANTUM COMPUTATIONS	PAGE 2
1.1	Postulates of quantum mechanics Postulate 1 — 2 • Postulate 2 — 5 • Postulate 3 — 5	2
1.2	Quantum gates Single qubit gates — 8 • Two qubits gates — 9 • N qubits gates — 11	7
1.3	Performing classical computations NAND operation — 12 • Coping information — 13	12
1.4	Measurements Projection valued measurement — 14 • Positive operator valued measurement — 16 • Measurements and circuits — 19	14
1.5	Entanglement Quantum teleportation — 22 • EPR paradox and Bell's inequality — 24	21
1.6	Quantum algorithms Quantum Fourier Transform — 26 • Phase evaluation — 29 • Grover algorithm — 32 • Hamiltonian simulation — 36	26
1.7	Universality of quantum computation Gates decomposition — 39 • Single qubit gates approximation — 42	38
1.8	Quantum error coorection Single qubit flip errors — 45 • Shor code — 46	43
1.9	Hybrid algorithm Quantum Approximation Optimizzation Algorithm — 48	48

# Introduction to quantum computations

## 1.1 Postulates of quantum mechanics

The course has the aim of introducing the follower to the working principles of quantum computing and how they work on a fundamental physical and mathematical level. Therefore, will come to the mind of the reader that all the theory ahead of us will rely on quantum mechanical principles that we will need to introduce first. This will be useful also to the experienced reader since a refresh can is always good, and also because we may want to restate the fundamental of quantum mechanics specifically for application to quantum calculators.

Quantum mechanics(QM) is the modern fundamental theory that describes reality and relies on a series of assumptions, or postulates, that allow us to describe the physical systems. There is not a standard version of the postulates, can vary in numbers, but we are going to use the description of QM that uses three postulates to define: states, evolution and mesuraments.

### Postulate 1

#### Definition 1.1.1: States

The set containing the possible states of the system is a Hilbert space  $\mathcal{H}$  and of the vectors  $|\psi\rangle \in \mathcal{H}$  only the normalized ones can describe the real state of the system.

This definition of the notion of state is really different from the simple coordinate couples of position and momenta ( $\mathbf{r}, \mathbf{p}$ ) that we had in classical mechanics. In this situation the states are pure vectors inside a vector space that is the Hilbert space itself, meaning that all the properties known from linear algebra theory apply to them. In particular, we have the closure property of sum and multiplication with a scalar, or the presence of a scalar product  $\langle \cdot, \cdot \rangle: \mathcal{H} \otimes \mathcal{H} \rightarrow \mathbb{C}$ . Those are all things that need to be present inside  $\mathcal{H}$  to be an Hilbert space, and define those inside our restricted series of systems that describe quantum computer is simple. In fact, the main quantum computers rely on systems with a finite number of levels which posses a **finite dimensional**  $\mathcal{H}$  which allow us to highly simplify all the computations thanks to the following

### Theorem 1.1.1

If we have that  $\dim \mathcal{H} = N$  then the space is holomorphic to  $\mathbb{C}^N$ .

That is a known fact indeed, but allow us to directly describe all the states of the system as complex vectors, along with the possible bases and the inner product. In fact, if we have a space with dimension  $N$  we can write down directly an orthonormal base for the system as the canonical one and write whatever vector  $|\psi\rangle$  as a superposition of those

$$|i\rangle = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \quad |\psi\rangle = \sum_i \lambda_i |i\rangle.$$

The other part is that we can imagine the inner product inside this space to be exactly the one of the complex space, having so that we can simply define it by first define the raw vectors as follows

$$\langle\phi| = (|\phi\rangle)^\dagger = (\alpha_0^* \quad \cdots \quad \alpha_i^* \quad \cdots \quad \alpha_N^*).$$

Then, we can use the normal algebra to define the following inner product inside the space

$$\langle\phi|\psi\rangle = \alpha_0^* \lambda_0 + \cdots + \alpha_N^* \lambda_N = \sum_i \alpha_i^* \lambda_i.$$

Knowing it allow also to define to norm inside to space simply as  $\|\psi\|^2 = \langle\psi|\psi\rangle$ , which tells us that for a state to be normalized means that the following must be true, always

$$\langle\psi|\psi\rangle = \sum_i |\lambda_i|^2 = 1. \quad (1.1)$$

Which also show us how a state of a system doesn't care about the phase since even if I redefine it to be  $|\phi\rangle = \exp(i\theta) |\psi\rangle$  will still be normalized, but one can also see how also observables are not touched.

We now have the full construction of the space we are going to use during our computations, and would be kind of interesting to use it to see how  $\mathcal{H}$  will look like in some simple cases. In particular, we want to show how the states of a **qubit** are formed. The latter is non-other than a two level system, which so has  $\dim \mathcal{H} = 2$ , meaning that the base can be seen simply as

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Which we will call as **computational base**, and all the state can be represented with them using a general linear combination that is normalized, so

$$|\psi\rangle = \begin{pmatrix} a \\ b \end{pmatrix}, \quad |a|^2 + |b|^2 = 1.$$

This is effectively simple to use not only for its mathematical simplicity but also because a really simple geometric representation for those state can be used. In fact, one can see the following

### Theorem 1.1.2

A state  $|\psi\rangle$  in a qubit can be represented, instead of using the  $a$  and  $b$  couple, using two angles  $\theta \in [0, \pi]$  and  $\phi \in [0, 2\pi]$  using the following form

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle. \quad (1.2)$$

**Proof:** Devo pensare alla dimostrazione. ☺

This allows us to see a state of the system as a unitary vector on a circumference, referenced by the angles and so the direction in which is pointing. This is indeed remarkable as we will see, since this anticipate the fact that every possible modification of a state is non-other than a rotation, as we will see.

In this construction we can also easily create systems composed by a series of qubit together, since the Hilbert space generated will be the tensor product of the single qubit ones as

$$\mathcal{H}_N = \mathcal{H} \otimes \cdots \otimes \mathcal{H} = \mathbb{C}^{2^N}.$$

Since we are working with a known operation, that is the tensor product, we can also readily create basis functions for the new states by simply make the tensor product of the ones of the older states. For example, calling  $\{|0\rangle_1, |1\rangle_1\}$  the base of the first qubit and  $\{|0\rangle_2, |1\rangle_2\}$  the base of the second one can easily see that the base for the combined system will be

$$\{|0\rangle_1 |0\rangle_2, |0\rangle_1 |1\rangle_2, |1\rangle_1 |0\rangle_2, |1\rangle_1 |1\rangle_2\}.$$

Most of the time the subscripts are omitted, having so that  $|0\rangle_1 |0\rangle_2 = |00\rangle$ , but in this case the order in which you write the basis is important and needs to be remembered! The importance of order is also noticeable in the creation of a two-qubit state by using single qubit ones, if taken  $|\psi\rangle = (a, b)$  and  $|\phi\rangle = (cd)$  one can see how

$$|\psi\phi\rangle = (a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle) \neq (c|0\rangle + d|1\rangle) \otimes (a|0\rangle + b|1\rangle) = |\phi\psi\rangle.$$

In fact the coefficients on the bases  $|01\rangle$  and  $|10\rangle$  are different, and the order here is important, so the states are different.

#### Note

*It's interesting to note how the number of basis elements increase a lot as the number of qubits are inserted inside the system. In particular, the dimension of the space scales as  $2^N$ . This is in contrast with what happens for the phase space in classical dynamics, where to describe a particle I need 6 coordinates and as the particle increase the number rise as  $6N$ . Basically the quantum systems become more complex much quicker and that hints us why is difficult simulate quantum systems on classical computers.*

## Postulate 2

### Definition 1.1.2: Evolution

The evolution of a closed system can be represented using a unitary operator  $\mathcal{U}(t_1, t_2)$  that acts on the state making it evolve in time as follows

$$|\psi(t_2)\rangle = \mathcal{U}(t_1, t_2) |\psi(t_1)\rangle. \quad (1.3)$$

This is obviously a simplified version of the evolution postulate, which in reality should refer to the evolution of the state based on the Schrödinger equation (SE). In fact, we can also see how this principle can be derived directly from the latter. Therefore, let's start from the SE recalling its form

$$i\hbar \frac{\partial |\psi(t)\rangle}{\partial t} = \hat{\mathcal{H}} |\psi(t)\rangle, \quad (1.4)$$

where, being in a closed system, the Hamiltonian will be time independent. Under those conditions we can easily integrate the equation finding out the following general form for the evolution of the system

$$|\psi(t)\rangle = \exp\left(-i\frac{\hat{\mathcal{H}}}{\hbar}(t - t')\right) |\psi(t')\rangle = \mathcal{U}(t', t) |\psi(t')\rangle. \quad (1.5)$$

In this way we have found out an analytic form for the evolution operator that we were describing. Also, one can actually see how since  $\hat{\mathcal{H}}$  is assumed to be selfadjoint,  $\hat{\mathcal{H}}^\dagger = \hat{\mathcal{H}}$ , then the following holds true

$$\mathcal{U}^\dagger(t', t) = \mathcal{U}(t, t') = \mathcal{U}^{-1}. \quad (1.6)$$

Which is telling us that  $\mathcal{U}$  is effectively unitary as required by the postulate, but not only that. In fact inside this construction we will also have that the operator  $\mathcal{U}$  will be also **linear** and **invertible**, giving always rise to reversible operations.

#### Note

*The properties of  $\mathcal{U}$  of being linear and invertible are something that was not really present inside classical computers, in fact we will use that operator to manipulate qubits and every operation can be inverted. That didn't hold true for classical bits, where operations were often irreversible, or neither linear.*

## Postulate 3

So far all the description of QM relies on principles that are totally deterministic, solving the SE does not lead to probabilistic results. Nevertheless, the theory is renown to give a probabilistic interpretation of reality, and this comes from the definition of how we measure physical quantities. To understand it we shall write down the definition in the mathematical terms of linear algebra, where measuring means apply a linear operator to the state of the system and evaluate averages on them as follows.

### Definition 1.1.3: Measure

Let  $\Lambda$  be a measurable quantity with  $\{\lambda_n\}_{n \in \mathcal{I}}$  its possible values,  $\mathcal{I}$  is a set of indices, a measurement of this quantity is defined by a set of operator  $\{\hat{\Pi}_n\}_{n \in \mathcal{I}}$  that respect the following relations:

1. If the state of the system is a generic  $|\psi\rangle$  then the probability of measuring  $\lambda_n$  is given by

$$p_n = \langle \psi | \hat{\Pi}_n^\dagger \hat{\Pi}_n | \psi \rangle. \quad (1.7)$$

2. After the measurement the state collapse into an eigenstate of  $\lambda_n$  written as

$$|\psi_n\rangle = \frac{\hat{\Pi}_n |\psi\rangle}{\sqrt{\langle \psi | \hat{\Pi}_n^\dagger \hat{\Pi}_n | \psi \rangle}}. \quad (1.8)$$

Therefore, a measure is simply a set of operators that is able to describe the probability of having a certain outcome from an experiment. Therefore, in general we can have  $\Pi_n$  to be whatever type of operator with restrictions only on how it acts onto the states. Nevertheless, it's easy to understand that from the definition of measure another restriction on their form needs to be done. In particular, we can use the properties of probability to see how the following needs to be true

$$\sum_n p_n = \sum_n \langle \psi | \hat{\Pi}_n^\dagger \hat{\Pi}_n | \psi \rangle = 1, \quad \forall |\psi\rangle \in \mathcal{H}. \quad (1.9)$$

It's easy to understand that this condition can be translated to a condition on the set of operators per se, requiring that the following is true in order to have a proper measure

$$\sum_n \hat{\Pi}_n^\dagger \hat{\Pi}_n = \mathbb{1}. \quad (1.10)$$

A condition that we will need to keep in mind when we will need to create a measure further in the course.

Using this definition the reader can understand how in QM I really can't know which value  $\lambda_n$  of a certain quantity the experiment will give in output, but only the probability of having it. This may arise the question to the reader of how we can predict something out of this theory if we can't know the final outcome. The answer is that, it's true, no single outcome can be predicted exactly, but the average quantity can be estimated without any problem. In fact, we can simply use the mathematical definition of average to see that

$$\langle \Lambda \rangle = \sum_n \lambda_n p_n = \sum_n \lambda_n \langle \psi | \hat{\Pi}_n^\dagger \hat{\Pi}_n | \psi \rangle. \quad (1.11)$$

Which shows how the average observable for a state  $|\psi\rangle$  is a totally deterministic quantity. From this expression we can also see how different states can give also the same average, in fact it's easy to see how for  $|\psi'\rangle = \exp(i\theta) |\psi\rangle$  with  $\theta \in \mathbb{R}$  leads to

$$\langle \Lambda' \rangle = |e^{i\theta}|^2 \sum_n \lambda_n \langle \psi | \hat{\Pi}_n^\dagger \hat{\Pi}_n | \psi \rangle = \sum_n \lambda_n \langle \psi | \hat{\Pi}_n^\dagger \hat{\Pi}_n | \psi \rangle = \langle \Lambda \rangle. \quad (1.12)$$

This means that basically a system in state  $|\psi\rangle$  or  $|\psi'\rangle$  have same observable and, therefore, are indistinguishable giving us the reason why in the definition of the first postulate we have said that no difference is present between states that differ only by a complex phase.

## 1.2 Quantum gates

After recalling the postulates of QM we want to focus a little on the second one and understanding how we can manipulate the states of qubits using it. In particular, the second postulate tells us that the evolution of a state in time is described by a unitary operator  $\mathcal{U}$ . Normally that operator should have infinite dimensions, nevertheless the qubits possess a finite dimensional Hilbert space meaning that also the operators inside it can be associated to finite matrices. To understand this we can take a system composed by  $N$  qubits, so that  $\dim \mathcal{H} = 2^N$ , the operator could then be described as follows

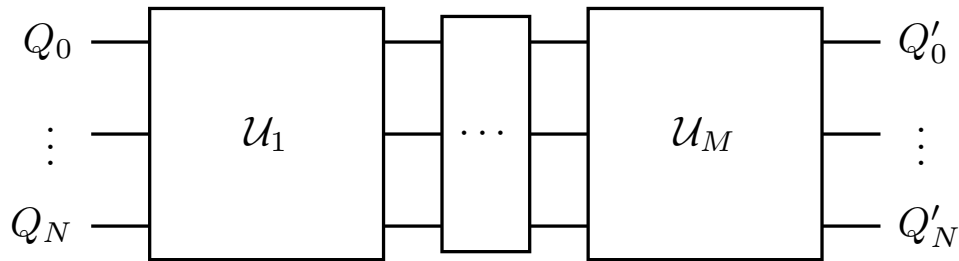
$$\mathcal{U} : \mathbb{C}^{2^N} \rightarrow \mathbb{C}^{2^N}, \quad \mathcal{U} = \begin{pmatrix} a_{11} & \cdots & a_{2^N 1} \\ \vdots & \ddots & \vdots \\ a_{1 2^N} & \cdots & a_{2^N 2^N} \end{pmatrix}. \quad (1.13)$$

This representation is incredibly useful on a mathematical level, in fact manipulating a simple qubit state will become as doing a matrix multiplication. In particular, we know that the state of a qubit can be written generally as  $|\psi\rangle = a|0\rangle + b|1\rangle = (a, b)$ , so applying a generic state manipulation  $\mathcal{U}$  will simply mean doing the following

$$\mathcal{U}|\psi\rangle = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} Aa + Bb \\ Ca + Db \end{pmatrix} = |\psi'\rangle. \quad (1.14)$$

This matrix operations are called **quantum gates**, and we want to try to unveil some possible interesting forms that they can have starting to understand how effectively a so-called **quantum circuits** works. In fact, a quantum circuits is non-other than a series of quantum gates acting on the available qubits in order to obtain a specific state in the end, as depicted in Fig. (1.1).

Thus, what we want to do now is look deeper in the form that those gates can have, and try to identify some important operations that will follow us during the whole course. Therefore, I suggest to the reader (especially my future self) to look into them and try getting attune to their form and their actions on the qubits.



**Figure 1.1:** Sample image of a general quantum circuits where all the manipulations on the qubits are done using quantum gates, matrices, represented as blocks.



## Single qubit gates

The first type of gates that we want to analyze are the one that acts a single qubits, which are also the simplest ones. As we have pointed out previously an operation of this type can be written as a matrix like in Eq. (1.14). In particular, it's easy to understand that every  $2 \times 2$  matrix that is unitary satisfy the second postulate being a valid quantum gate for the manipulation of a qubit state. That let us have an infinite amount of possibilities for the operations that we can do on a single unit of information, setting another difference with the classical bits. In fact, a normal bit could be manipulated through the application of the identity operation, leaving as it is, or the NOT one, changing its state. So, not only the qubits allow for an infinite number of possible states to be represented using superposition, but also allow for an infinite number of operations to be done on it showing a clear superiority.

This simple observation has already shown to us that we have a lot of room in which we can move in order to work with qubit, and what we need to do now is to point out the most significant and important operations that are present inside this really large space. The first, thing that is important to point out is that we know how to generate every  $2 \times 2$  unitary matrix using four of them. Thus, mathematic has already given us a way to write down all the possible gates using a simple universal relation.

### Theorem 1.2.1: Universal single gate

Every quantum gate  $\mathcal{U}$  can be associated to a rotation of the state on the Bloch sphere, and therefore can be written as

$$\mathcal{U} = e^{i\lambda \mathbb{1}} \exp(i\alpha \mathbf{n} \cdot \boldsymbol{\sigma}), \quad (1.15)$$

where  $\lambda, \alpha \in \mathbb{R}$ ,  $\mathbb{1}$  is the identity matrix,  $\mathbf{n}$  is an appropriate unitary vector, and  $\boldsymbol{\sigma}$  is a vector containing the three Pauli matrices.

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (1.16)$$

This theorem tells us that only 4 parameters are needed in order to define every possible gates, in fact the parameter  $\alpha$  is not really needed can be associated to the velocity of the rotation not really on the rotation itself. Therefore, we know how to write every possible operation, still the one that are interesting to us are limited, and so we will report them briefly discussing the form and the action of those specific gates.

**NOT.** The NOT operation is the equivalent of the one in the classical case, meaning that the effect is the one of negating the current state. That can be easily done by recalling that the 0 and 1 of our quantum logic are the state  $|0\rangle$  and  $|1\rangle$ , leading us to the possibility of constructing the truth table and matrix associated to this operation.

IN	OUT
$ 0\rangle$	$ 1\rangle$
$ 1\rangle$	$ 0\rangle$

$$\text{NOT} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (1.17)$$

One can also notice that the final matrix obtained is exactly the X Pauli matrix, with assure to us that is unitary since every one of them has the property  $X^2 = Y^2 = Z^2 = \mathbb{1}$ . This matrix is, in fact, drawn as  $\oplus$  symbol inside circuits to recall that the X matrix is getting used on a certain qubit.

**Hadamart.** This is the first real quantum gate that was thought of, the idea is the one of creating superpositions of the normal states. This can be done really easily since the gate that we are interested in, and it's truth table, is

IN	OUT
$ 0\rangle$	$ +\rangle$
$ 1\rangle$	$ -\rangle$

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (1.18)$$

Where we shall recall how  $|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$  are interesting states that forms the eigenvectors of the  $X$  Pauli matrix, basically being the states in the  $x$  direction on the Bloch sphere while the computational base is on the  $z$  one. This gate has also really important properties that can be interesting such as

$$H^2 = \mathbb{1}, \quad H^\dagger = H = H^{-1}, \quad HZH = X. \quad (1.19)$$

**Phase.** This is another type of gate that wants to do a purely quantum operation, that is the one of adding a phase shift to the two components of the state. Basically we want a gate that starting from a state  $|\psi\rangle = a|0\rangle + b|1\rangle$  is able to add a phase shift to the two bases that can be used to enhance interference effects. The idea is so the following

$$\Phi(\theta)|\psi\rangle = a|0\rangle + be^{i\theta}|1\rangle, \quad \Phi(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}. \quad (1.20)$$

These types of gates can be used in a lot of situations since phase shift are really common inside quantum mechanical application, and some of them are most commonly used than the others and so a specific name was given to them

$$\Phi(\pi/2) = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = S, \quad \Phi(\pi/4) = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1+i}{\sqrt{2}} \end{pmatrix} = T, \quad T^2 = S. \quad (1.21)$$

#### Note

*I want to stress out how the idea of the qubit is basically creating a new type of logic. In the classical computers the boolean logic was the only possible thing with the bits that could be only 0 and 1 along with two possible operations: identity or NOT. Now, the qubit can have infinite states and infinite operations can be done on it having so an incredible much richer logic that allow things obviously impossible before. An example of it is the  $\sqrt{\text{NOT}}$  operation which was demonstrated impossible to define since no operation applied to times could bring to the NOT one, even inside the context of fuzzy logic. In quantum logic we can do it simply by taking the square root of the Pauli matrix*

$$\sqrt{X} = \frac{1}{2} \begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix}. \quad (1.22)$$

## Two qubits gates

The next step is adding a second qubit to our system and see how the possible gates changes. On a mathematical level the answer is really simple since we are simply modifying  $\mathcal{H}$  to have two more dimensions, having so that now  $\mathcal{U} : \mathbb{C}^4 \rightarrow \mathbb{C}^4$  being represented by a  $4 \times 4$  unitary matrix. Thus, the

possibilities for the usable gates have increased respect to the single qubit once. Nevertheless, also in this case we want to make some order and explicitly write down the ones that are used the most and that we will see more frequently.

**Control.** The control gates, in reality, are a class of two qubits gates that one of the two as the control one, not being modified, and perform operations on the other. To understand the concept we can have a look at the most important control gate that is the **CNOT**. The latter is a gate that takes two qubits and: when the control one is  $|0\rangle$  then nothing is done, if the control is  $|1\rangle$  then the NOT is performed on the other qubit. At first this gate may seem complicated to realize, but that would be a wrong assumption since both the truth table and matrix are simple as

IN	OUT
$ 00\rangle$	$ 00\rangle$
$ 01\rangle$	$ 01\rangle$
$ 10\rangle$	$ 11\rangle$
$ 11\rangle$	$ 10\rangle$

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

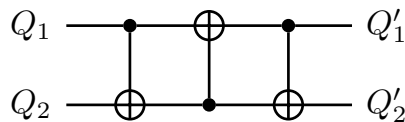
Where we can see how the total matrix is formed by the use of two known single qubit gates on the diagonal, creating the quantum version of the NOR operation in boolean logic. This form, with two gates on the diagonal, allow for a big flexibility inside this category of matrices, in particular it's easy to understand that one can create the C-version of every single qubit operator by simply defining it as



$$C\mathcal{U} = \begin{pmatrix} 1 & 0 \\ 0 & \mathcal{U} \end{pmatrix}. \quad (1.23)$$

Where, in the graphical representation the dot describe the control qubit.

**Swap.** This gate is in reality a simple circuits constructed using three CNOT gates in order, and the reason of the name can be easily seen by the truth table.



IN	OUT
$ 00\rangle$	$ 00\rangle$
$ 01\rangle$	$ 10\rangle$
$ 10\rangle$	$ 01\rangle$
$ 11\rangle$	$ 11\rangle$

Thus, one can see that the effect is literally the one of swapping the states of the qubit. This is a simple and clever operation that can be written in a simple matrix form thanks to the fact that we know the matrix describing the gate of the circuit of which is formed. Therefore, we can write down the final gate by simply matrix multiplying the gates of which is composed in the right order and then having the following one described by its own simbol

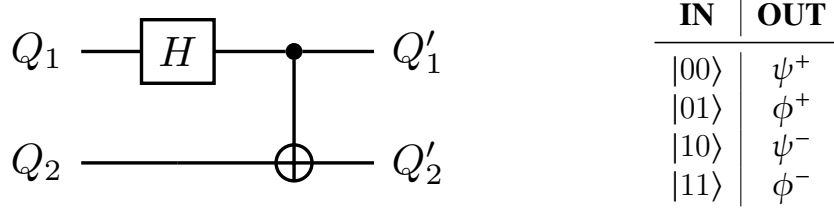


$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (1.24)$$

**Bell.** This is a simple circuit that nevertheless is really important since allow for the creation of particular states of major interest in physics, the Bell's states. The latter are four quantum mechanical states defined as follows

$$\psi^{\pm} = \frac{1}{\sqrt{2}} (|00\rangle \pm |11\rangle), \quad \phi^{\pm} = \frac{1}{\sqrt{2}} (|01\rangle \pm |10\rangle). \quad (1.25)$$

They are mostly important in the theoretical study of spin states, but they appear also in other areas of QM. Thus, we want to describe a circuit that is able to prepare the system in those states and the way in which this can be done is the following



Basically, based on the initial state of the system we are able to generate one of the four Bell state and then study their behavior in specific circuits.

#### Example 1.2.1 (Circuit solving)

As a scrupulous, I want to see the computation of the Bell's gate output for the case  $|10\rangle$  just to show the reader how effectively solve the circuit. At first the Hadamart gate is used on the first qubit

$$H |1\rangle |0\rangle = \frac{1}{\sqrt{2}} (|0\rangle |0\rangle - |1\rangle |0\rangle) = \frac{1}{\sqrt{2}} (|00\rangle - |10\rangle), \quad (1.26)$$

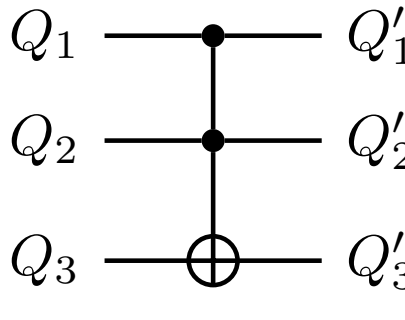
then the CNOT operation is applied using the first qubit as control, having the final result

$$\psi^+ = \frac{1}{\sqrt{2}} = (|00\rangle - |11\rangle) \quad (1.27)$$

## N qubits gates

We can now step into assuming to have a general number of qubits to work with and see how some general gates can effectively be created also in this case, in particular two main types of gates play a huge role in the whole theory of quantum computers.

**Toffoli.** This is a specific 3-qubit gate that has the aim of making a control NOT gate using two different controls. Basically, having two controls the idea is to apply the NOT to the target if and only if both the two controls are in state  $|1\rangle$ . For this reason the gate is also called CCNOT and has the following matrix and circuit representation



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

The task of writing down the truth table from the matrix representation is left to the reader to effectively see how the aim of applying a NOT to the last qubit only if the first two are in  $|1\rangle$  is accomplished.

**N control.** These gates are a generalization of the Toffoli one to an N number of control and an M number of targets. In general the idea is to apply a general M-qubit gate  $\mathcal{U}$  to the targets qubits if the N controls are in the state  $|1\rangle$ , the general form of the matrix is analogous to the one already seen for the CU gates and the circuit is also totally analogous to the Toffoli and CU one.

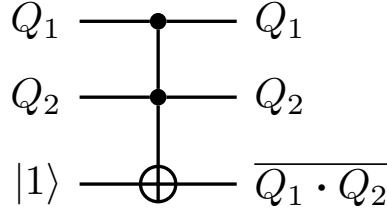
### 1.3 Performing classical computations

The quantum computer seems to have all the potential of performing computations that normal computers can't do, but it's not obvious that they can still perform the task of a normal one. That is a problem, in fact creating computers that are not able to perform simple tasks would be useless for us, and so we want to demonstrate that this isn't the case.

The question "can quantum computers do classical computations?" comes from the doubt that in the possible operations that can be done on the qubits may not be able to simulate the two general operations that give rise to all the classical logic. These operations are the NAND gate and the COPY one, which together forms the building block of all the other possible operations. The first one is the **universal operation** of classical logic meaning that every circuit can be constructed using only NAND operations, while the latter constitute a problem due to some properties of QM that makes coping a non-trivial matter. Here we want to take the two properties one at a time and demonstrate that QC can perform both using the **Toffoli gate**.

#### NAND operation

The possibility of a QC to perform the NAND operation is not obvious, and the reason for that is the fact that all the quantum operations that we can perform on a qubit are invertible which the NAND is not. Basically it's not sure that a non-invertible operation can be constructed using invertible ones, or better if we can represent it using an invertible operation on a qubit. Fortunately turns out that performing the NAND on qubit is not only possible, but simple. The idea is using a Toffoli gate and setting one of the three qubits to  $|1\rangle$ , obtaining as output the NAND of the other two as we can see in the following circuit.



IN	OUT
001>	001>
011>	011>
101>	101>
111>	110>

Which clearly reconstruct the NAND as wanted, meaning that using QC we are able to recreate every possible classical circuit.

### Coping information

First, we shall clarify why this operation represent a problem inside quantum circuits, and the reason is intrinsic inside QM and in the known result called **non-cloning theorem**, which states the following.

#### Theorem 1.3.1: Non-cloning

Taken a state  $|\psi\rangle$  of a system of qubit, it does not exist a unitary operation  $\mathcal{U}$  with the following property

$$\mathcal{U}(|\psi\rangle \otimes |s\rangle) = |\psi\rangle \otimes |\psi\rangle, \quad (1.28)$$

where  $|s\rangle$  is another arbitrary state.

**Proof:** Let's imagine that such operation exist, and take three arbitrary states  $|\psi\rangle$ ,  $|\phi\rangle$  and  $|s\rangle$ . Since  $\mathcal{U}$  is unitary the following relation must hold true

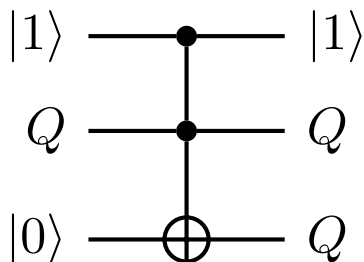
$$\langle s\phi|\psi s\rangle = \langle s|s\rangle \langle \phi|\psi\rangle = \langle s\phi|\mathcal{U}^\dagger\mathcal{U}|\psi s\rangle, \quad (1.29)$$

but using relation Eq. (1.28) and the normalization of the states it's possible to see that the latter equation becomes

$$\langle \phi|\psi\rangle = \langle \phi|\psi\rangle^2. \quad (1.30)$$

This is not true in general, but only if we select carefully the two states in order for them to respect this relation. Otherwise, the operation  $\mathcal{U}$  cannot be unitary and therefore is not a viable quantum gate. ☺

This result may seem a problem by all means, but already in the proof is hidden the answer to overcome it. In fact, we have said that a cloning operation can exist if the starting states are selected carefully so that the norm of the states is conserved. In practice this means that we can use a Toffoli gate with two entries selected in a specific way to obtain the following copy gate.



IN	OUT
100>	000>
110>	111>

In this way also coping information to one qubit to another can be done always using the Toffoli gate, making it probably the most important of all, and demonstrating that QC can perform all classical computations along with the unexplored quantum ones.

## 1.4 Measurements

During the first section we have introduced the concept of measure as a set  $\{\hat{\Pi}_n\}_{n \in I}$  of operators that allow us to evaluate the probability of having a certain outcome  $\lambda_n$  for the measure of a physical quantity  $\Lambda$ . Nevertheless, even if we have listed the main properties that such operators should have we didn't specify the forms that they can possess. It is now time to introduce the main way in which such set of operators can present themselves and how they will work inside quantum circuits.

### Projection valued measurement

If we take a look back to the different properties stated in Def. (1.1.3) an idea should arise, especially by looking at the collapse condition. The latter, in fact, can be thought as the state of the system gets projected into a certain state that posses a definite value of the physical quantity we are interested in measuring. In fact, it's easy to see how the following is true

$$\hat{\Pi}_n |\psi_n\rangle = |\psi_n\rangle, \quad p_n = \langle \psi_n | \hat{\Pi}_n^\dagger \hat{\Pi}_n | \psi_n \rangle = 1. \quad (1.31)$$

Operators that posses this effect of taking a general state  $|\psi\rangle$  and taking it into another are really known in mathematics and are called **projector operators**, represented as  $\hat{P}$ . This type of operator is defined, in particular, by two main conditions

$$\hat{P}^\dagger = \hat{P}, \quad \hat{P}^2 = \hat{P}, \quad (1.32)$$

these two properties are enough to give  $\hat{P}$  a lot of power allowing it to define alone a subspace of the vector space in which is working. In our case this means

$$\mathcal{P} = \{ |\psi\rangle | \exists |\phi\rangle \in \mathcal{H} : |\psi\rangle = \hat{P} |\phi\rangle \}, \quad (1.33)$$

basically we are aiming to create a set  $\{\hat{P}_n\}$  so that  $\mathcal{P}_n$  is the eigenspace of  $\lambda_n$ . Therefore, it's easy to understand that on a general ground we are searching for a specific type of measure constructed using projectors, that will take the following specific name and definition.

#### Definition 1.4.1: Projection valued measurement

We will call projection valued measurement, or PVM, a set of operators  $\{\hat{P}_n\}_{n \in I}$  so that  $\hat{P}_n$  is a projector and the whole set is a measure.

To find out the right projectors to create the complete set we can simply use the properties of the observable in QM. An observable quantity is represented in QM using an operator  $\hat{\Lambda}$  and the possible values that  $\Lambda$  can take, the outcomes, are the eigenvalues of that operator  $\lambda_n$ . Nevertheless, for a general operator  $\lambda_n$  can be complex, or worst they can not exist, that is a problem since is absurd to measure a complex number in an experiment, therefore we will make a further assumption giving out the following definition.

### Definition 1.4.2: Observables

An observable  $\Lambda$  in QM is represented by a hermitian operator  $\hat{\Lambda}$ , so that

$$\hat{\Lambda}^\dagger = \hat{\Lambda}. \quad (1.34)$$

A really important theorem of linear algebra called **spectral theorem** allow us to say that, with this further assumption of hermiticity, the operator can be diagonalized and the eigenvalues are all real. Along with that, the spectral theorem also allow us to say that a set of eigenstates  $\{|\psi_n\rangle\}$  with the following properties exist

$$\hat{\Lambda} |\psi_n\rangle = \lambda_n |\psi_n\rangle, \quad \langle \psi_n | \psi_m \rangle = \delta_{nm}, \quad \sum_n |\psi_n\rangle \langle \psi_n| = \mathbb{1}, \quad (1.35)$$

forming an orthonormal base for  $\mathcal{H}$ . These properties should hint us that the set of projectors that we want to create to measure  $\Lambda$  may be the ones that project onto the eigenspace generated by  $|\psi_n\rangle$ , and we can easily see how that is the case.

#### Theorem 1.4.1: Observable PVM

Taken  $\hat{\Lambda}$  an observable the set of projectors  $\{\hat{P}_n\}_{n \in \mathcal{I}}$  onto the orthonormal base of eigenstate  $\{|\psi_n\rangle\}_{n \in \mathcal{I}}$  of  $\hat{\Lambda}$  forms a measure for the observable itself called **Projection valued measurement**.

**Proof:** We are going to define the projector  $\hat{P}_n$  as follows

$$\hat{P}_n = |\psi_n\rangle \langle \psi_n|, \quad (1.36)$$

which can be easily seen it's a projector, the demonstration is left to the reader. We want to see how all the requirement in Def. (1.1.3) are verified, we can start from the probability by taking a general state  $|\psi\rangle$  and writing

$$|\psi\rangle = \sum_n c_n |\psi_n\rangle, \quad p_n = \langle \psi | \hat{P}_n^\dagger \hat{P}_n | \psi \rangle = \langle \psi | \hat{P}_n | \psi \rangle = |c_n|^2. \quad (1.37)$$

Where I have first written the expansion of  $|\psi\rangle$  on the orthonormal base and then used the properties in Eq. (1.32) and Eq. (1.35) to obtain the probability. It's possible to see how the values of  $|c_n|^2$  effectively represents probabilities since also the condition Eq. (1.10) is respected by the set of operators defined thanks to the completeness condition of the orthonormal base

$$\sum_n \hat{P}_n^\dagger \hat{P}_n = \sum_n \hat{P}_n = \sum_n |\psi_n\rangle \langle \psi_n| = \mathbb{1}. \quad (1.38)$$

Therefore, the first condition for having a measure is respected. We can now see how also the second one is obtained as we want since we can write

$$|\psi_n\rangle = \frac{\hat{P}_n |\psi\rangle}{\sqrt{\langle \psi | \hat{P}_n^\dagger \hat{P}_n | \psi \rangle}} = \frac{c_n}{\sqrt{|c_n|^2}} |\psi_n\rangle = e^{i\theta} |\psi_n\rangle, \quad (1.39)$$

and since a complex phase doesn't change the physical state of the system the wanted result is achieved.

☺



### Example 1.4.1 (Qubit PVM)

To make an example we can use a qubit, where we can imagine measuring the state as  $|1\rangle$  or  $|0\rangle$ , so a computational base measurement. We can also easily imagine what is the form of the operator, the observable, that has them as eigenstates

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad Z|i\rangle = \lambda_i|i\rangle, \quad (1.40)$$

where it's easy to see, by using the  $\mathbb{C}^2$  representation of the states, how  $\lambda_0 = 1$  and  $\lambda_1 = -1$ . This means that for every state  $|\psi\rangle$  we can measure the probability of a certain outcome, state  $|1\rangle$  or  $|0\rangle$ , to appear by using the set of operators defined by  $\hat{P}_i = |i\rangle\langle i|$  given as matrix by

$$P_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad P_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \quad (1.41)$$

Applied to a general vector they will give in general the first or second component, respectively.

## Positive operator valued measurement

In the PVM description of measure that we have given just now we use projector operators to predict the probabilities, which possess a series of properties that makes them really well suited for the task. Nevertheless, they are not the only possible choice, in particular such projectors possess the property of being hermitian that general measures can totally not have. We want so to see a case where another type of measure respect to the PVM can be a better choice to study the system.

Let's imagine having a qubit and a quantum circuit that is able to prepare it in two states given by

$$|\psi_1\rangle = |0\rangle, \quad |\psi_2\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}. \quad (1.42)$$

We want to see if we are able to understand which state after a measure. Using the PVM measure of a qubit using  $Z$  as an observable we can easily see how  $|\psi_1\rangle$  possesses a probability equal to 1 of being in state  $|0\rangle$ , while  $|\psi_2\rangle$  has  $p_i = 1/2$  for both states. This means that if I take a measure and the outcome is  $\lambda = -1$  I know that the only state that can have that outcome is  $|\psi_2\rangle$  since it has a non-zero probability of being in state  $|1\rangle$ . Instead, if the result is  $\lambda = 1$  I can't say which one of the two states is the right one. We can also understand why we are not able to decide, since the two states that we are working with are not orthogonal respect to the computational base and so  $|\psi_2\rangle$  has non-zero probability of being in both states. Therefore, we would like to use a set of operators that instead are able to give us this property, and a possible way of obtaining them is by taking  $\{\hat{\Pi}_n\}$  so that

$$\hat{\Pi}_n^\dagger \hat{\Pi}_n = \hat{E}_n, \quad (1.43)$$

are **positive valued operators**. In this way we are allowed to define as our operators the following objects

$$\hat{E}_1 = |1\rangle\langle 1|, \quad \hat{E}_2 = \left( \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \left( \frac{\langle 0| - \langle 1|}{\sqrt{2}} \right), \quad (1.44)$$

basically the projector on the states that are orthogonal to the ones that we were looking at. Nevertheless, the set composed by  $\hat{E}_1$  and  $\hat{E}_2$  is not sad to respect the relation Eq. (1.10) meaning that we are needed to add a third element in general that allow for the set to be a measure

$$\hat{E}_3 = \mathbb{1} - \hat{E}_2 - \hat{E}_1. \quad (1.45)$$

Then, using this new set we are able to obtain a more satisfying result since the now the possible outcomes of the measurement are three. In fact, we can imagine that doing a measure using  $\{\hat{E}_n\}$  is experimentally equivalent to obtain three possible outcomes  $\lambda_n$  where the probabilities of the outcomes for the two states respectively are

$$\langle \psi_1 | \hat{E}_i | \psi_1 \rangle = \begin{cases} p_1 = 0 \\ p_2 \neq 0 \\ p_3 \neq 0 \end{cases}, \quad \langle \psi_2 | \hat{E}_i | \psi_2 \rangle = \begin{cases} p_1 \neq 0 \\ p_2 = 0 \\ p_3 \neq 0 \end{cases}. \quad (1.46)$$

Therefore, now after a measure we are able to say that if the outcome is  $\lambda_1$  or  $\lambda_2$  the result is state  $|\psi_1\rangle$  or  $|\psi_2\rangle$  respectively, while  $\lambda_3$  is telling us that the measure is not satisfying since we can't distinguish the two states. This may seem not so different respect to the PVM measure since still we have the case where the single measure can distinguish the two cases, but here we have also an outcome that tells us with certainty to have  $|\psi_2\rangle$  that we didn't have before. Also, it is possible to demonstrate that the probability of an not satisfying measure is minimized by this set.

Therefore, with this simple case we have seen how the definition of measure in reality leaves large freedom in the choice of the operator set and not only projectors. In fact, we can now generally define also the positive operator valued measurement as follows.

#### Definition 1.4.3: POVM

We will call positive operator valued measurement a set of operators  $\{\hat{E}_n\}_{n \in \mathcal{I}}$  so that  $\hat{E}_n$  is a positive valued operator and the whole set represent a measure.

This definition is much more general than PVM and can allow for better result in terms of computations as we have seen in the example shown beforehand. Nevertheless, the problem of these measures is that we actually don't know how to represent them in practice. In fact, for PVM the definition was made starting from a physical observable while here the physical representation is much more hidden. Still, doesn't mean that don't exist. Is possible to demonstrate that whatever measure, can be represented into a higher dimensional space through a unitary transformation composed a PVM, in this way everything can be physically interpreted. We can so state the following result in terms related to quantum computers applications as follows.

#### Theorem 1.4.2: Representation of measures

Any generalized measure on a system of  $N$  qubit  $\mathcal{H}_Q$  can be realized as a PVM with the aid of auxiliary qubit and a unitary operator.

**Proof:** Imagine having a general measure  $\{\hat{\Pi}_n\}_{n \in \mathcal{I}}$  acting on  $\mathcal{H}_Q$ , we can take an orthonormal set  $\{|m\rangle\}_{m \in \mathcal{I}'}$ , with  $\#\mathcal{I} = \#\mathcal{I}'$ , of states  $|m\rangle \in \mathcal{H}_Q$  and construct an auxiliary space as

$$\mathcal{H}_{tot} = \mathcal{H}_Q \otimes \mathcal{H}_M, \quad \mathcal{H}_M = \text{span}\{|m\rangle\}. \quad (1.47)$$

So, a general state inside the auxiliary space can be written as  $|\psi\rangle|s\rangle$ , with  $|\psi\rangle \in \mathcal{H}_Q$  and  $|s\rangle \in \mathcal{H}_M$ . Now, we define the transformation  $\mathcal{U}$  as follows

$$\mathcal{U}|\psi\rangle|s\rangle = \sum_n \left( \hat{\Pi}_n |\psi\rangle \right) |m\rangle, \quad (1.48)$$

which can be seen to be a unitary operator since the following relation can be obtained

$$\langle s| \langle \psi| \mathcal{U}^\dagger \mathcal{U} |\phi\rangle |s\rangle = \sum_{mm'} \langle \psi| \hat{\Pi}_m^\dagger \hat{\Pi}_m |\phi\rangle \langle m'|m\rangle = \sum_m \langle \psi| \hat{\Pi}_m^\dagger \hat{\Pi}_m |\phi\rangle = \langle \psi|\phi\rangle. \quad (1.49)$$

Where the relation  $\langle m|m'\rangle = \delta_{mm'}$  and  $\sum_n \hat{\Pi}_n^\dagger \hat{\Pi}_n = \mathbb{1}$  have been used during the process, demonstrating that  $\mathcal{U}$  conserve the norm of the space being so unitary. Next we can define the projection operators to be given by the following

$$\hat{P}_m = \mathbb{1}_Q \otimes |m\rangle\langle m|, \quad (1.50)$$

creating a PVM set of measure. Then, we can see how the operator  $\hat{P}_m \mathcal{U}$  act on a general state of the whole space as


$$\hat{P}_m \mathcal{U} |\psi\rangle |s\rangle = \sum_{m'} \left( \hat{\Pi}_n |\psi\rangle \right) \langle m|m'\rangle |m\rangle = \hat{\Pi}_m |\psi\rangle |m\rangle. \quad (1.51)$$

This is a powerful result since we can already see how this set of operations is able to reproduce the general measure in all of it's fascion. We can start by seeing how the probabilities that predicts are the same

$$p_m = \langle s| \langle \psi| \mathcal{U}^\dagger \hat{P}_m^\dagger \hat{P}_m \mathcal{U} |\psi\rangle |s\rangle = \langle \psi| \hat{\Pi}_n^\dagger \hat{\Pi}_n |\psi\rangle. \quad (1.52)$$

Thus, also the projection remains unchanged under the effect of this operator since

$$\frac{\hat{P}_m \mathcal{U} |\psi\rangle |s\rangle}{\sqrt{\langle s| \langle \psi| \mathcal{U}^\dagger \hat{P}_m^\dagger \hat{P}_m \mathcal{U} |\psi\rangle |s\rangle}} = \frac{\hat{\Pi}_m |\psi\rangle}{\sqrt{\langle \psi| \hat{\Pi}_n^\dagger \hat{\Pi}_n |\psi\rangle}} |m\rangle = |\psi_m\rangle |m\rangle. \quad (1.53)$$

Therefore, all the properties of the general measure are conserved, meaning that  $\{\hat{P}_m \mathcal{U}\}$  successfully represent it. 

This result is telling is that doesn't matter with measure we are using we can always add a number of qubit so that the space becomes larger by a  $\mathcal{H}_M$  with dimensions equal to the number of outcomes you want to measure and use a normal PVM along with a unitary transformation defined by the measure itself. In this way we can always use the same  $Z$  gate seen in the PVM example to measure the auxiliary qubit and based on the value of their state, which gives  $|m\rangle$ , we can know the outcomes of the others, since  $|m\rangle$  correspond to a collapsed  $|\psi_m\rangle$  and so  $\lambda_m$  as outcome.

#### Note

*This whole theory was though first by Von Neumann which tried to create it to describe the collapse on the wavefunction in a rigorous way. In particular He though that an experimental apparatus in general was the auxiliary system  $\mathcal{H}_M$  that taking the measure made the system collapse in the way we have seen.*

## Measurements and circuits

Measuring is an important feature inside a quantum circuits since at the end of the manipulations we also need to know the result of the operations contained in the qubits states. In general, we are going to see how the measurement works inside a quantum circuit and their properties. First, we shall keep in mind that usually all the quantum computers are able to perform measurements mainly in the computational basis, meaning of the observable  $Z$ . This is due to a matter of experimental convenience, but lead us to try to reconduce every possible measure to that one. For example, we may work for a complex quantum circuit in a base like  $|\pm\rangle$  so that a state is given by

$$|\psi\rangle = a|+\rangle + b|-\rangle, \quad (1.54)$$

but we can't use the measure given in that base since the QC only measure in computational base. The answer to this problem is in reality incredibly simple, and we can state it really quickly as

### Theorem 1.4.3: Base change

Let's imagine having a circuit that prepare a qubit state  $|\psi\rangle$  described on an orthonormal base  $|e_1\rangle$  and  $|e_2\rangle$ . The measurement of the state can be performed still on the computational base without losing information by applying an appropriate rotation before the measure.

**Proof:** Let  $\mathcal{U}$  be the unitary transformation that acts as follows

$$\mathcal{U}|e_1\rangle = |0\rangle, \quad \mathcal{U}|e_2\rangle = |1\rangle, \quad (1.55)$$

we can easily see how the state  $|\psi\rangle$  can be written in a really simple way by using the following reasoning

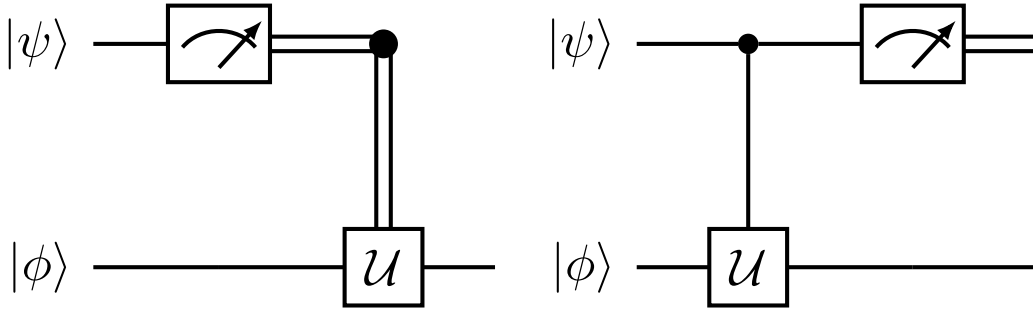
$$|\psi\rangle = a|e_1\rangle + b|e_2\rangle = \mathcal{U}(a|0\rangle + b|1\rangle). \quad (1.56)$$

Meaning that the state  $\mathcal{U}^\dagger|\psi\rangle$  will be a state written in the computational base with the same coefficients, therefore measuring it gives out the same probabilities of before we don't lose any informations. 😊

This little result is really helpful in practice, since basically means that we can perform a measurement using the same base in every possible situation by using circuit like in Fig. (1.2). Where we can see



**Figure 1.2:** Simple quantum circuit for the measure of a qubit in a general base, composed by a rotation and the measure.



**Figure 1.3:** Representation of two equivalent circuits that uses classical and quantum control to apply an operation to the second qubit. In the first case the measurement on the first qubit is done before the control while in the other is done after, still the result will not change having that if the bit is 1 the operation is applied, while if is 0 will remain untouched.

how the measure is represented by the symbol with the arrow which take a quantum state and returns the outcome of the measure on a classical bit 1 or 0, seen as the double line.

The other important thing that we can say about measure inside circuits is that they can be applied in every point of the circuit and the final result of the computation will not change. To understand this we shall make an example using control gates as depicted in Fig. (1.3). We can see how the two circuits differ based on where the measurement is placed in the circuit. Nevertheless, if we analyze the system we can simply see how the outcome of the state  $|\phi\rangle$  will be the same based on the value of the measurement. In fact, take the first circuit: the outcomes were

$$|\phi\rangle, \quad \text{if } c = 1; \quad \mathcal{U}|\phi\rangle, \quad \text{if } c = 0, \quad (1.57)$$

where  $c$  represent the value of the bit. Now, if we take the second one, it's easy to see that if the measure gives 1 meant that the that  $|\psi\rangle$  had a component along  $|1\rangle$  passing through the control so that the rotation was applied to  $|\phi\rangle$ . Instead, if the value was 0 the opposite happens having so that the outcome at the end will be the exact same. This property of measurement takes the name of **principle of deferred measurement**, and allow us to place the measurement procedure always at the end of our circuits without the worries of modifying the outcome of the algorithm.

### Attention

The deferred measurement principle it's not based on a mathematical result, since for it work in a general case we shall need that the measure operator  $\hat{\Pi}_m$  commute with every possible gate  $\mathcal{U}$ . In this way we would have that

$$\hat{\Pi}_m \mathcal{U} |\psi\rangle = \mathcal{U} \hat{\Pi}_m |\psi\rangle, \quad (1.58)$$

which means that the final state at the end of the circuit remains the same. Nevertheless, it's easy to find a counter example by using  $\mathcal{U} = X$  and  $\hat{\Pi}_m = |1\rangle\langle 1|$ , meaning that in general may exist a situation where the final outcome changes. Still, the principle could still hold since our aim is not really find out the exact value of the final state, but to understand it by using a single measure and the connection between the probabilities of certain outcomes and possible prepared states.

Therefore, even if mathematically the result can be different physically specking the result is the same.

#### Note

*It's possible to find that in several circuits the final measures are not explicitly written or are written only on certain qubit, that is because only the necessary measurements to understand the states of the qubits are shown, the non-necessary ones are skipped. This idea is also called **principle of implicit measurements** and will be more clear as we will talk about entanglement.*

## 1.5 Entanglement

The concept of entanglement comes from the need of describing the difference of certain type of states respect to others. In particular, we can understand it by first defining a really simple type of state called separable that has the following form.

### Definition 1.5.1: Separable state

Taken a state  $|\psi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$  it's called separable if  $\exists |\phi_1\rangle \in \mathcal{H}_1, |\phi_2\rangle \in \mathcal{H}_2$  so that we can write

$$|\psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle. \quad (1.59)$$

This type of states are really simple since we can decompose them into simpler once and work with them. To make an example we can see how the state  $(|01\rangle + |00\rangle)/\sqrt{2}$  is a separable one since

$$\frac{|01\rangle + |00\rangle}{\sqrt{2}} = |0\rangle \left( \frac{|0\rangle + |1\rangle}{\sqrt{2}} \right), \quad (1.60)$$

holds true and so can be decomposed. Nevertheless, we shall not go much further to find out more complex states that cannot be decomposed for which we need to work in the higher dimensional space like the Bell's state  $(|01\rangle + |10\rangle)/\sqrt{2}$ . Those are entangled states, whose definition is therefore really simple as.

### Definition 1.5.2: Entangled states

A state  $|\psi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$  it's called entangled if it's not separable.

Therefore, the mathematical definition of entanglement seems really simple, but the physical consequences are not, and we shall see together why this is one of the main powers of QC.

### Example 1.5.1

To make an example of how powerful the entanglement can be we can see how in theory we can transport the information of two bits, so 00, 01, 10 or 11, transmitting only one qubit. The idea is that two people, Alice and Bob, posses two qubit that are entangled in the following Bell state

$$|\psi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \quad (1.61)$$

then Alice can perform one of four operations on it's qubit based on what wants to transmit to

Bob. In particular the following choices can be made

Operation	Final state
$ 00\rangle \rightarrow \mathbb{I}$	$ \psi^+\rangle$
$ 01\rangle \rightarrow Z$	$ \psi^-\rangle$
$ 10\rangle \rightarrow X$	$ \phi^+\rangle$
$ 11\rangle \rightarrow iY$	$ \phi^-\rangle$

So, based on the information that we want to send we have performed an operation of the Alice qubit to obtain a different Bell state. Then, Alice only need to send the qubit to Bob which need to perform a two qubit measure using the Bell's base and the state that will obtain will correspond to the selected two bit information obtained with the transport of only one qubit.

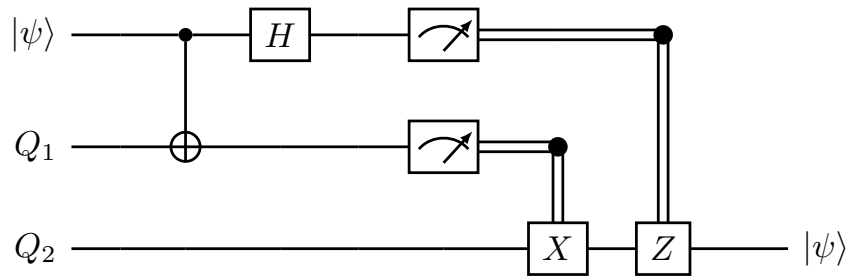
#### Note

*This definition of Entanglement is not really the most general one that can be used. In fact, the Def. (1.5) it's a specific one called bi-entanglement since only two Hilbert spaces are counted, but the most general called n-entanglement count for  $|\psi\rangle \in \bigotimes_{i=1}^n \mathcal{H}_i$ . Nevertheless, for QC application only the bi-entanglement plays an important role.*

## Quantum teleportation

The first real important application of entanglement that we will see it's the quantum protocol for the quantum teleportation of information. Basically we want to teleport a quantum state from one qubit to another, so that no matter is transported to one place to another but only the state of the qubits is changed transporting information. Also, at first site one can also think that the cloning theorem would be violated in a situation of this type, but that is not the case, and we will see why.

The circuit that is used in order to perform the teleportation protocol is reported in Fig. (1.4), and consist in the use of two qubit in order to transport a general state  $|\psi\rangle = a|0\rangle + b|1\rangle$  into one of those qubit. Thus, we want to understand how this is possible in general, and to do that let's pretend that the



**Figure 1.4:** Quantum teleportation circuit for a quantum computer, see how it involves the measuring of two qubit to control two operations.

two qubit  $|\psi\rangle$  and  $Q_1$  are in possession of Alice, while  $Q_2$  is with Bob. We will need that the two qubit  $Q_1, Q_2$  are entangled in a  $|\psi^+\rangle$  Bell state, then we can place them at whatever distance, so that Alice and Bob can be also on different planets. In this situation Alice will perform a series of operation on her qubit starting with a CNOT, to understand what this operation will do on the system we need first to look at the state of the hole three qubit, which will be

$$|\psi Q_1 Q_2\rangle = (a|0\rangle + b|1\rangle) \left( \frac{|00\rangle + |11\rangle}{\sqrt{2}} \right) = \frac{1}{\sqrt{2}} [a|000\rangle + a|011\rangle + b|100\rangle + b|111\rangle]. \quad (1.62)$$

Then we can easily perform the CNOT on the first two qubit ending up in the following state

$$\frac{1}{\sqrt{2}} [a|000\rangle + a|011\rangle + b|110\rangle + b|101\rangle], \quad (1.63)$$

on which the Hadamart tranformation is applied on the first qubit, which we shall remember that transforms  $|0\rangle$  in  $|+\rangle$  and  $|1\rangle$  in  $|-\rangle$ . Therefore, we will have the following form

$$\frac{1}{2} [a|000\rangle + a|100\rangle + a|011\rangle + a|111\rangle + b|010\rangle - b|110\rangle + b|001\rangle - b|101\rangle], \quad (1.64)$$

which can be rewritten by seeing how inside it four separable states can be found out, having

$$\frac{1}{2} [|00\rangle (a|0\rangle + b|1\rangle) + |01\rangle (a|1\rangle + b|0\rangle) + |10\rangle (a|0\rangle - b|1\rangle) + |11\rangle (a|1\rangle - b|0\rangle)]. \quad (1.65)$$

At first sight this state may seem nothing special, but if the states inside parentheses are inspected one could find out different forms of  $|\psi\rangle$  obtaining

$$\frac{1}{2} [|00\rangle (|\psi\rangle) + |01\rangle (X|\psi\rangle) + |10\rangle (Z|\psi\rangle) + |11\rangle (XZ|\psi\rangle)]. \quad (1.66)$$

We have so created a state where the two qubit state given by Alice's couple is entangled to the qubit of Bob, which will poses the  $|\psi\rangle$  state up to two transformation based on the Alice's values. Therefore, if Alice measure it's two qubit and refers the outcomes to Bob, he can perform the right operations to obtain at the end the final state  $|\psi\rangle$  on his qubit. An operation that inside the circuit is described by the two classical controls and can be seen gives the right result keeping in mind that  $X^2 = Z^2 = \mathbb{1}$ .

Thus, using this protocol a general state  $|\psi\rangle$  can be transferred into another qubit at whatever distance without limitations. In fact, the circuit has been performed in several occasions starting from a teleportation distance of some centimeters to the experiment of Anton Zeilinger that performed it at nearly one kilometer of distance inside Vienna. Today we are able to transfer states from hearth to satellite, but still this phenomenon result in being quite strange at first sight. Perhaps, as was told earlier, one can say that this **phenomenon contradicts Thm. (1.3.2)** with the coping of a general state into another one. This is a wrong affirmation, since the non-cloning theorem tells that the following operation doesn't exist

$$\mathcal{U}(|\psi\rangle \otimes |s\rangle) = |\psi\rangle \otimes |\psi\rangle, \quad (1.67)$$

while here the starting states have been measured by Alice, meaning that they have collapsed having a result more similar to

$$\mathcal{U}(|\psi\rangle \otimes |Q_1\rangle \otimes |Q_2\rangle) = |00\rangle \otimes |\psi\rangle. \quad (1.68)$$

Where  $|00\rangle$  is only one of the four possible final result obtained. Another apparent contradiction that can be found out inside the teleportation protocol is the fact that **information seem to pass from one qubit to another faster than the speed of light**. That can't obviously be possible, since relativity forbid it, and in fact that is not the case. The two classical controls that are used inside the circuit solve the problem since in order to obtain the copied state inside  $Q_2$  I first need Alice to transmit the result of the measurements to Bob and that classical exchange of information take times.



## EPR paradox and Bell's inequality

Since in 2022, the year I'm writing these notes, the Nobel Prize in physics was awarded to Alain Aspect, John Clauser and Anton Zeilinger for their experiments on entangled photons, a little part of the course was devoted to the description of the theory they allowed to uncover. To understand it we shall understand that in the first half of 1900 the idea of entanglement was difficult to accept. Especially Einstein was really skeptical about it since at first sight looked like a threat to its restricted relativity with information apparently traveling faster than light. Remained in history Einstein's definition of entanglement as "spooky action at a distance", showing how much he hated it. In this context, great work was putted by Einstein and others physicist in order to see if effectively QM was wrong or, better, not entirely right, meaning that we were still missing something to understand it at its fullest.

In 1935 Einstein, Podolsky and Rosen (EPR) delineated a theory trying to show that QM was indeed not complete, starting by defining what it means for a physical theory to be complete. The line of reasoning starts with the definition of the so-called elements of reality.

### Definition 1.5.3: Elements of reality

A quantity is called element of reality if it has a value that can be predicted before experiment, basically it's already perfectly known already before taking the measure. A property of this type, for logic, must be owned by the system regardless of what you are making so that exist also before taking the measurement with a precise value.

This definition is purely based on logic, assuming that if I'm able to know the exact outcome of an experiment before taking the measure then the property that I'm measuring needs to have that value defined intrinsically inside it. Then, the next move is to use this definition to define when a theory is complete.

### Definition 1.5.4: Complete theory

A physical theory is complete if it contains all elements of reality.

Using these two definitions, the three theoreticians demonstrated that QM is not complete if we assume the notion of locality. Where, locality was also defined by them as follows.

### Definition 1.5.5: Locality

Measurements are independent if done in position and time that are not causally related.

With this they were able to create a thought experiment that demonstrated how some elements of reality were missing inside QM. The experiment that E. P. R. proposed is actually old, today the best way to understand this is by looking at the one proposed by David Bohm in 1951 which shows how spin is what is missing.

#### Theorem 1.5.1: Incompleteness of QM

Spin is an element of reality that is not contained entirely inside quantum mechanics.

**Proof:** We can imagine taking a singlet state of spin which is given by the following Bell state

$$|\phi^-\rangle = \frac{|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle}{\sqrt{2}}. \quad (1.69)$$

This state is a peculiar one, since it's possible to demonstrate that has the same form in every possible base. Meaning that if I change the base of representation to an orthonormal couple  $\{|e_1\rangle, |e_2\rangle\}$  the state become

$$|\phi^-\rangle = \frac{|e_1e_2\rangle - |e_2e_1\rangle}{\sqrt{2}}. \quad (1.70)$$

Thus, if one measures the spin of the system in any direction, so using the right base, the same result will be found out meaning that the spin is an element of reality with a defined value. Nevertheless, in QM we know that the spin operators do not commute, meaning that we can't know both components of the spin at the same time. Basically, even if the whole spin vector is an element of reality quantum mechanics only allows for the partial knowledge of it, meaning that QM does not contain all elements of reality. 😊

This result showed, in the minds of Einstein and colleagues, that the QM is not a complete theory needing still work to find out certain **hidden variables** whose treated statistically will recreate quantum mechanics but used normally will give deterministic results. Basically he was searching for the equivalent of classical mechanics starting from statistical mechanics, where now statistical mechanics is quantum mechanics itself.

This discussion was incredibly interesting at the time, but understanding what the hidden variables were remained more of a philosophical question for a lot of times until the idea of Bell. In 1964, he published a paper showing how whatever deterministic real local theory using hidden variables is doomed to fail in reproducing the quantum mechanical result using the following reasoning. He focused his studies on a particular observable defined using four different ones  $Q, S, R$  and  $T$ , where all of them could only have two possible outcomes  $\pm 1$ . The observable that we are interested in is the following

$$\mathcal{L} = QS + RS + RT - QT, \quad (1.71)$$

which possesses some interesting properties such as it can have only two possible outcomes  $\pm 2$ . Bell showed that inside a local real deterministic theory the following result must hold.

#### Theorem 1.5.2: Bell's inequality

Assuming locality and reality of the four observable composing  $\mathcal{L}$  we have that the following relation hold

$$\langle \mathcal{L} \rangle \leq 2. \quad (1.72)$$

**Proof:** We will assume that the four observables will be evaluated separately in four different measurements, having so that every draw forms the outcome  $(q, r, s, t)$  with a probability  $P(q, r, s, t)$ . We can so see how the following relation is true since we will have

$$\langle \mathcal{L} \rangle = \sum_{qrst} (qs + rs + rt - qt)P(q, r, s, t) \leq 2 \sum_{qrst} P(q, r, s, t), \quad (1.73)$$

but the sum of the probability of all possible outcomes need to be normalized at 1 having so that the wanted relation holds. 😊

This result allowed to set a condition for the existence of the local hidden variables, the only thing remained to do is to show that quantum mechanics do not respect it. We can so take the following

example

$$Q = Z_1, \quad S = -\frac{Z_2 + X_2}{\sqrt{2}}, \quad (1.74)$$

$$R = X_1, \quad T = \frac{Z_2 - X_2}{\sqrt{2}}, \quad (1.75)$$

and evaluate the average of  $\mathcal{L}$  in the singlet state. From locality, we can say that no correlation exist from the outcomes of the different measurements, so that their probability are independent of each other having

$$P(q, r, s, t) = P(q)P(r)P(s)P(t). \quad (1.76)$$

Meaning that the following is true

$$\langle \phi^- | \mathcal{L} | \phi^- \rangle = \langle QR \rangle + \langle RS \rangle + \langle RT \rangle - \langle QT \rangle = 2\sqrt{2} > 2. \quad (1.77)$$

Meaning that QM doesn't respect Bell's inequality and so can't be reconstructed using local hidden variables. This line of reasoning was also proven experimentally by Cluaser first, and then also by Specter with a more refined experiment that reproduced accurately the  $2\sqrt{2}$  result using photons and not spins. Therefore, local hidden variables will not be the answer to find out a deterministic quantum theory.

From that point on quantum information evolved a lot with the creation of quantum optics and entangled photons, in particular the last part of the Nobel Prize was given to Zeller because he was the first to perform quantum teleportation in open air at a far distance in Vienna. He teleported a state using entangled photons from one part to the other of the Danube river.

## 1.6 Quantum algorithms

It's now to see some important quantum circuits that can be used in practice in order to obtain interesting results or behaviour inside a quantum computer. A lot of them will be not usefull per se, but are really important for the creation of larger and more powerful schemes.

### Quantum Fourier Transform

The aim is to create an algorithm that bring the computational basis into the Fourier base, defined by the application of the Fourier transform on the vectors of the base itself. Therefore, we need first to recall the definition of discrete fourier transform which is given by.

#### Definition 1.6.1: Discrete Fourier transform

Taken a vector  $(x_1, x_2, \dots, x_N) \in \mathbb{C}^N$  it's Fourier transform is an application  $\mathcal{U} : \mathbb{C}^N \rightarrow \mathbb{C}^N$  that actes in the following way

$$\mathcal{U}\mathbf{x} = \mathbf{y}, \quad y_k = \frac{1}{\sqrt{N}} \sum_{j=1}^N e^{i2\pi \frac{jk}{N}} x_j. \quad (1.78)$$

We know that such an application possesses a lot of properties, in particular we know how the following relations are true

$$\mathcal{U}^{-1} = \mathcal{U}^*, \quad \mathcal{U}^{-1} = \mathcal{U}^\dagger, \quad (1.79)$$

meaning that it's a particular unitary transformation, and so we imagine that a circuit can represent it.

We basically want to create an algorithm that allows us to take the computational basis and perform Eq. (1.79) on it. In order to do that we will use a particular notation to describe the computational base of an  $N$  qubit system, using the binary representation.

### Definition 1.6.2: Binary representation

Taken a system of  $N$  qubit, where the computational base has the form  $\{|00\dots 0\rangle, |00\dots 1\rangle, \dots\}$  we will write down the elements of the basis as  $\{|j\rangle\}_{j=0}^{2^N-1}$  where the following relation is given

$$|j\rangle = |j_1 j_2 \dots j_N\rangle, \quad j = \sum_{i=1}^N j_i 2^{N-i}. \quad (1.80)$$

basically  $j$  is the number described by the system state in binary base.

Using this notation it's really easy to write down the Fourier transform of the base having that

$$\mathcal{U} |j\rangle = \frac{1}{2^{N/2}} \sum_{k=0}^{2^N-1} e^{i2\pi \frac{jk}{2^N}} |k\rangle, \quad (1.81)$$

having that the base is brought into another more complicated one. In fact, we can see how the states transform accordingly to the FT if the transformation is applied to the base. For example take a general state  $|\psi\rangle$  we can see how

$$|\psi\rangle = \sum_j x_j |j\rangle, \quad \mathcal{U} |\psi\rangle = \frac{1}{2^{N/2}} \sum_k \sum_j x_j e^{i2\pi \frac{jk}{2^N}} |k\rangle = \sum_k y_k |k\rangle, \quad (1.82)$$

basically the coefficients defining the state in vector representation gets Fourier transformed. Still, this form of the Fourier transform leaves not too much room for the creation of an algorithm that allow us to perform it in practice, so we need first to work a little on it and see how the following result holds.

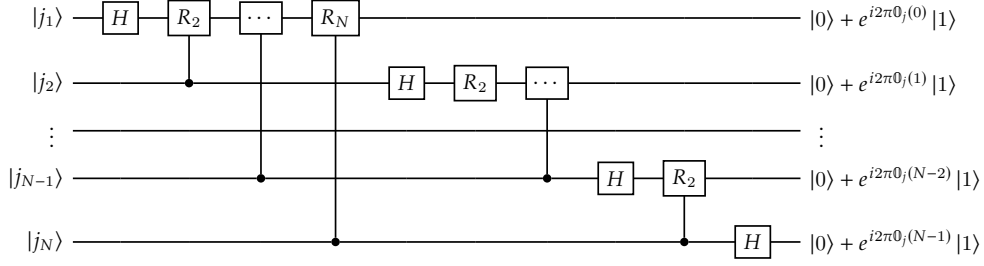
### Theorem 1.6.1: Quantum Fourier Transform

The Fourier transformation of the states in the computational basis can be rewritten as a sum of tensor product given by the following form using the binary representation for the base

$$\mathcal{U} |j\rangle = \frac{1}{2^{N/2}} \bigotimes_{l=1}^N \left[ |0\rangle_l + e^{i2\pi \mathbb{O}_j(N-l)} |1\rangle_l \right]. \quad (1.83)$$

Where the notation  $\mathbb{O}_j(l)$  is defined as the following decimal number

$$\mathbb{O}_j(l) = \sum_{i=l+1}^N j_i 2^{l-i}. \quad (1.84)$$



**Figure 1.5:** Circuits able to perform the quantum fourtranform of the computational basis. It's important to notice how this algorithm is in reality not totally complete since the phases of the final states are inverted, so a series of swap gates needs to be inserted at the end to get the right form.

**Proof:** We can start to work around the expression of Eq. (1.81) by seing how we can rewrite  $k/2^N$  in a more usefull form as

$$\frac{k}{2^N} = \sum_{l=1}^N k_l 2^{-l}. \quad (1.85)$$

We can then substitute it and see how, recalling that  $|k\rangle = |k_1\rangle \otimes |k_2\rangle \cdots \otimes |k_N\rangle$ , the following is true

$$\mathcal{U}|j\rangle = \frac{1}{2^{N/2}} \sum_{k=0}^{2^N-1} e^{i2\pi j \sum_{l=1}^N k_l 2^{-l}} |k\rangle = \frac{1}{2^{N/2}} \sum_{k=0}^{2^N-1} \bigotimes_{l=1}^N e^{i2\pi j k_l 2^{-l}} |k_l\rangle. \quad (1.86)$$

Now, the order of the operation can be inverted by taking some care anbd see how the expression become

$$\mathcal{U}|j\rangle = \frac{1}{2^{N/2}} \bigotimes_{l=1}^N \sum_{k_l=0,1} e^{i2\pi j k_l 2^{-l}} |k_l\rangle = \frac{1}{2^{N/2}} \bigotimes_{l=1}^N \left[ |0\rangle_l + e^{i2\pi j 2^{-l}} |1\rangle_l \right]. \quad (1.87)$$

Now, we can see how the exponent can be simplified. In fact, if you use the definition of  $j$  we can see how

$$j 2^{-l} = \sum_i j_i 2^{N-i-l} = \sum_{i \leq N-l} j_i 2^{N-i-l} + \sum_{i=N-l+1}^N j_i 2^{N-i-l} = \text{integer} + \mathbb{0}_j(N-l). \quad (1.88)$$

Obviously the integer part gives no contribution to the phase since multiply  $2\pi$  at the exponent, so only the  $\mathbb{0}_j(N-l)$  can be written obtaining the final result. ☺

Now, this form of the QFT is much more usefull on an algorithmic point of view since it's now easy per us to see the operations that needs to be done on the single qubits. In particular, to reproduce the result we will need two types of gates that are also easy to understand that are

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad R_k = \begin{pmatrix} 1 & 0 \\ 1 & e^{i \frac{2\pi}{2^k}} \end{pmatrix}, \quad (1.89)$$

basically the Hadamart and phase gates. The final form of the circuit is described in Fig. (1.5), where is possible to see how we can evaluate the QFT by the cimple use of controlled phase gates and swaps ones at teh end of the proces. This particular circuit is interesting since shows how quantum computers are able to perform fuorier transforms easily respect to classical algorithm. In fact, the number of gates required to perform such an operation scales as  $O(n^2)$  while the number of logic gates needed to perform the computation of a fast fourier transform inside a classical computer scales as  $O(n2^n)$ . Quantum algorithm beats the classical one by an exponential factor. Nevertheless, the quantum algorithm has the big problem that the information are present in the phase of the states, and we are not able to evaluate them, basically meaning that the algorithm alone is useless. Still, we will see that QFT will still be used inside larger algorithm giving a lot of benefits.

## Phase evaluation

The evaluation of phases inside quantum systems is always a troublesome problem, to the point that most of the time is thought as an impossible task. Nevertheless, is possible to see how the QFT algorithm is able to give us the possibility of experimentally evaluate a certain type of phase inside the system.

Let's imagine having a unitary operator  $\mathcal{U}$  and wanting to evaluate the eigenvalues  $\lambda_n$ . We know from linear algebra that  $\mathcal{U}\mathcal{U}^\dagger = \mathbb{1}$ , so that from linear algebra we have the following information on the eigenvalues

$$\lambda_n \lambda_n^* = |\lambda_n|^2 = 1. \quad (1.90)$$

This relation tells us that the eigenvalues of such opeartors needs to be some kind of complex phases writable as

$$\lambda_n = e^{2\pi i \phi_n}, \quad \phi_n \in [0, 1[. \quad (1.91)$$

Normally we would think that evaluating  $\phi_n$  would be impossible since phases cannot be observed axperimentally. Still, we will show how using the eigenstate of the phases  $|u_n\rangle$  and a simple operator  $\mathcal{U}^{2^j}$  defined as follows

$$\mathcal{U}^{2^j} : |u_n\rangle \mapsto e^{2\pi i \phi_n 2^j} |u_n\rangle, \quad (1.92)$$

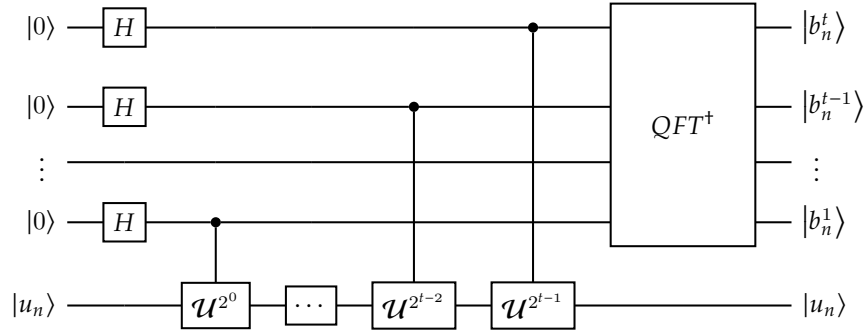
we are able to approximate its value with a wanted precision. In particular, we are aiming to approximate  $\phi_n$  using a decimal form like the one in Eq. (1.84), so that we want to find  $(b_n^1, b_n^2, \dots, b_n^t)$  so that

$$b_n = \frac{b_n^1}{2} + \frac{b_n^2}{4} + \dots + \frac{b_n^t}{2^t}, \quad (1.93)$$

is the decimal number  $b < \phi_n$  closer to it as possible. Therefore, we are going to demonstrate that the algorithm showed in Fig. (1.6) is able to do exactly that, find out the values of  $b_n^i$  approximating the phase up to a wanted accuracy.

### Theorem 1.6.2: Phase evaluation

Taken a unitary operator  $\mathcal{U}$  with eigenvalues defined by the phases  $\phi_n$ , the circuit Fig. (1.6) is able to give as output its decimal approximation  $b_n$  with a wanted accuracy  $\alpha > 0$ , and precision



**Figure 1.6:** Quantum circuit for the approximation of the eigenvalues of a unitary operator up to a wanted accuracy based on the number of qubit used for the decimal approximation of the phase itself.

$\epsilon > 0$  if the number of qubit used for the operation is

$$t > \alpha + \ln \left( 2 + \frac{1}{2\epsilon} \right). \quad (1.94)$$

**Proof:** First we need to look at what the circuit does effectively to the states, and it's easy to see that by looking at the case where  $t = 1$  and see how the first part of the circuit works as follows

$$\left( C\mathcal{U}^{2^0} \right) (H \otimes \mathbb{1}) |0\rangle \otimes |u_n\rangle = C\mathcal{U}^{2^0} \left( \frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \otimes |u_n\rangle = \left( \frac{|0\rangle + e^{2\pi i \phi_n} |1\rangle}{\sqrt{2}} \right) \otimes |u_n\rangle. \quad (1.95)$$

Which can be easily generalized to all the applications on the general  $t$ -dimensional circuit having a final form of the state given by

$$|\psi_f\rangle = \left[ \bigotimes_{j=0}^{t-1} \left( \frac{|0\rangle + e^{2\pi i \phi_n 2^j} |1\rangle}{\sqrt{2}} \right) \right] \otimes |u_n\rangle \quad (1.96)$$

Now, we know how every real number can be written as a converging series of rational numbers such as Eq. (1.84) so that we are going to work with

$$\phi_n = \frac{\phi_n^1}{2} + \frac{\phi_n^2}{4} + \dots = \sum_{i=1}^{\infty} \frac{\phi_i}{2^i} \approx \sum_{i=1}^t \frac{\phi_i}{2^i}. \quad (1.97)$$

Meaning that can be approximated using the decimal representation described before, having that the phases can be rewritten using the following form

$$\exp \left( 2\pi i \phi_n 2^j \right) = \exp \left( 2\pi i \left\{ \left[ \phi_n^1 2^{j-1} + \dots \right] + \left[ \phi_n^{j+1} 2^{-1} + \dots \right] \right\} \right) = \exp \left( 2\pi i \mathbb{0}_\phi(j) \right), \quad (1.98)$$

where the first part was an integer giving no contribution. In this way one can understand that in the first part of the  $|\psi_f\rangle$  state we end up having the QFT of the initial states, allowing us to write it in binary representation as

$$|\psi_f\rangle \approx \left( \frac{1}{2^{t/2}} \sum_{k=0}^{2^t-1} e^{2\pi i \phi_n k} |k\rangle \right) \otimes |u_n\rangle. \quad (1.99)$$

In this way we can go and apply the inverse of the QFT and using the binary representation of the base to see what happens exactly in the transformation

$$QFT^\dagger |\psi_f\rangle = \frac{1}{2^t} \sum_{k=0}^{2^t-1} e^{2\pi i \phi_n k} \sum_{l=0}^{2^t-1} e^{-2\pi i k l / 2^t} |l\rangle = \sum_l \left( \sum_{k=0}^{2^t-1} \frac{1}{2^t} e^{2\pi i (\phi_n - l/2^t) k} \right) |l\rangle = \sum_l c_l |l\rangle. \quad (1.100)$$

The state becomes a superposition of states in the binary computational base of the  $t$  qubits, and we know also the form of the coefficients. In fact, we can easily see how  $c_l$  are composed by a geometric series giving rise to the following result

$$c_l = \frac{1}{2^t} \sum_{k=0}^{2^t-1} \left( e^{2\pi i (\phi_n - l/2^t)} \right)^k = \frac{1}{2^t} \frac{1 - e^{2\pi i (\phi_n - l/2^t) 2^t}}{1 - e^{2\pi i (\phi_n - l/2^t)}}. \quad (1.101)$$

It's interesting to see how this result was obtained by truncating the real form of  $\phi_n$  to a certain decimal order, but the procedure is general and works also in the case where the total series is retained.

Therefore, in Eq. (1.101) the value of  $\phi_n$  can be a rational or irrational number, and we can see what happens to  $c_l$  in the two cases. In the case  $\phi_n$  is rational we have that the approximation Eq. (1.97) is exact, and we can write how

$$(\phi_n - l/2^t) 2^t = \sum_{i=1}^t \phi_n^i 2^{t-i} - l = int. \quad (1.102)$$

This means that the phase on the numerator of  $c_l$  is a multiple of  $2\pi$  and therefore  $c_l = 0$  for every value of  $l$  except for  $l = \phi_n 2^t$ , where

$$c_{\phi_n 2^t} = \frac{1}{2^t} \sum_{k=0}^{2^t-1} \left( e^{2\pi i (\phi_n - \phi_n)} \right)^k = \frac{2^t}{2^t} = 1. \quad (1.103)$$

This means that  $c_l = \delta_{l\phi_n 2^t}$  which basically becomes a condition on the values of the single qubit that will take the following forms

$$\sum_{l=0}^{2^t-1} \delta_{l\phi_n 2^t} |l\rangle = |\phi_n 2^t\rangle = |\phi_n^1\rangle \otimes |\phi_n^2\rangle \otimes \cdots \otimes |\phi_n^t\rangle, \quad (1.104)$$

where the binary representation was used to write down the single qubits. Now, we can simply evaluate all the  $t$  qubits and have our approximation of the phase  $\phi_n$  that is exact in this case, having that  $b_n = \phi_n$ .

In the case  $\phi_n$  is irrational, the truncation in the series generate an error having also that  $(\phi_n - l/2^t) 2^t$  is not an integer anymore and  $c_l$  can be different from 0 for other  $l$  values. In this case we are going to approximate the value of  $\phi_n$  using the values of  $b_n$  that we obtain from the evaluation, and we



want to know the error we are generating from such an approximation along with the accuracy of the result. Therefore, we select an error value  $\omega$  and compute the probability of performing the measure

$$\tilde{P} = P(\bar{l} \in [-\omega - 1, \omega + 1]), \quad (1.105)$$

where  $\bar{l} = l - \phi_n$  so that  $\bar{l} = 0$  correspond to the real  $\phi_n$ . This evaluation can be performed using as  $\omega$  the value  $2^{t-\alpha} - 1$ , where  $\alpha$  is called accuracy, obtaining the result of

$$1 - \tilde{P} \leq \frac{1}{2(2^{t-\alpha} - 2)}. \quad (1.106)$$

Which identify the probability of not obtaining the wanted result, that we are going to set below a certain precision  $\epsilon$  having  $1 - \tilde{P} < \epsilon$  so that the following relations is found

$$t > \alpha + \ln \left( 2 + \frac{1}{2\epsilon} \right). \quad (1.107)$$

☺

Therefore, this algorithm is able to give us a really greate approximation for the phase that increase in accuracy as we scale up the number of qubit used in order to perform the operation. Also, it's possible to see how we from the inequality we can actually controll also the accuracy and precision of the prediction by setting them prior and then choose  $t$  in order to satisfy Eq. (1.94).

## Grover algorithm

Search algorithm are one of the most important type of algorithm that exist, since the aim is to find out specific elements inside a set that satisfy some requirements defined by ceratin function  $f(x)$ . Normally this kind of algorithms are really expensive scaling up real quick with both dimensions of the set and complexity of the function to use, but inside a quantum computer we can try to create a really efficient version of them.

Let's imagine to work with  $n$  qubits, so that the computational base is composed by  $N = 2^n$  elements that compose our set  $\mathcal{N}$  in which only  $1 \leq M \leq N$  elements are solutions of  $f(x)$ . Where, being solutions means that the state  $|x\rangle^1$  that represent that element is so that

$$f(x) = \begin{cases} 1 & x \in \text{solution} \\ 0 & x \notin \text{solution} \end{cases}, \quad (1.108)$$

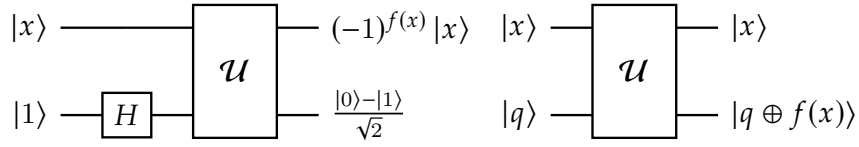
basically  $f : \mathcal{S} \rightarrow \{0, 1\}$  is a boolean function that gives true if it's solution and false otherwise. Then, we are going to need two types of gates that will generate the building blocks of our algorithm.

### Theorem 1.6.3: Oracle gate

The circuit shown in Fig. (1.7) is called oracle and is able to perform an inversion of the states that are not solution by using an ausiliary qubit in the  $|1\rangle$  state.

$$O |x\rangle = (-1)^{f(x)} |x\rangle, \quad (1.109)$$

<sup>1</sup>We are using the binary representation for the state of the computational basis.



**Figure 1.7:** Quantum circuit for the oracle gate used inside the algorithm to perform an inversion of the state about the components that are not solution to the function that we are interested in.

so that if  $|x\rangle$  is solution a  $-1$  pops up, and remains equal otherwise.

**Proof:** We can simply go over the different steps of the circuit to see how the following happens to the state

$$\mathcal{U}(\mathbb{1} \otimes H) |x\rangle \otimes |1\rangle = \mathcal{U} |x\rangle \otimes \left( \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) = |x\rangle \otimes \frac{1}{\sqrt{2}} \begin{cases} |0\rangle - |1\rangle & x \in \mathcal{S}, \\ |1\rangle - |0\rangle & x \in \mathcal{E} \end{cases}. \quad (1.110)$$

Where  $\mathcal{S}$  and  $\mathcal{E}$  are the subset of  $\mathcal{N}$  that contains the states that are solutions and not, respectively. Now, we can see how the difference between the two is simply a minus sign that we can imagine factorizing so that  $|x\rangle$  takes it changing its phase as wanted. ☺

The next step is to define another type of gate that will be needed to perform another, more complex, type of reflection about the average state. Such gate is actually defined through also the use of a **conditional phase operator** defined as

$$\mathcal{U}_x = 2|x\rangle\langle x| - \mathbb{1}, \quad (1.111)$$

which has the effect of changing the phase to all state in the base that are not  $|x\rangle$ , and it's really easy to see since

$$\mathcal{U}_x |k\rangle = \begin{cases} |k\rangle & k = x, \\ -|k\rangle & k \neq x \end{cases}. \quad (1.112)$$

Using this idea we can define the wanted inversion operator in this particular way.

#### Theorem 1.6.4: Inversion about the average

We can define the inversion about the average operator inside the space  $\mathcal{N}$  as  $\tilde{\mathcal{U}} = H^{\otimes n} \mathcal{U}_0 H^{\otimes n}$  and see how it's effect on a general state is

$$|\phi\rangle = \sum_{k \in \mathcal{N}} \alpha_k |k\rangle, \quad \tilde{\mathcal{U}} |\phi\rangle = \sum_{k \in \mathcal{N}} (2\langle\alpha\rangle - \alpha_k) |k\rangle. \quad (1.113)$$

Which defines an inversion about the average state  $\langle\alpha\rangle$ .

**Proof:** We can first have a look at the form of the operator itself, which can be rewritten in a compact way as follows

$$\tilde{\mathcal{U}} = 2H^{\otimes n} |0\rangle\langle 0| H^{\otimes n} - \mathbb{1} = 2|\psi\rangle\langle\psi| - \mathbb{1}, \quad |\psi\rangle = \frac{1}{\sqrt{N}} \sum_{k \in \mathcal{N}} |k\rangle. \quad (1.114)$$

Then, we can apply it to a general state to see how we retain the wanted result. In particular, we are going to write that

$$\tilde{\mathcal{U}} \sum_{k \in \mathcal{N}} \alpha_k |k\rangle = 2 \sum_{k \in \mathcal{N}} \alpha_k \langle\psi|k\rangle |\psi\rangle - \sum_{k \in \mathcal{N}} \alpha_k |k\rangle = 2 \sum_{k \in \mathcal{N}} \sum_{l \in \mathcal{N}} \sum_{x \in \mathcal{N}} \frac{\alpha_k}{N} \langle l|k\rangle |x\rangle - \sum_{k \in \mathcal{N}} \alpha_k |k\rangle, \quad (1.115)$$

where we can use the fact that  $\langle l|k\rangle = \delta_{lk}$  to eliminate a summation and then see how  $\sum_{k \in \mathcal{N}} \alpha_k / N = \langle\alpha\rangle$  so that the final result is obtained as

$$\tilde{\mathcal{U}} \sum_{k \in \mathcal{N}} \alpha_k |k\rangle = 2 \sum_{x \in \mathcal{N}} \langle\alpha\rangle |x\rangle - \sum_{k \in \mathcal{N}} \alpha_k |k\rangle = \sum_{k \in \mathcal{N}} (2\langle\alpha\rangle - \alpha_k) |k\rangle. \quad (1.116)$$

⊕

These two gates are the building blocks that will allow us to perform the quantum search and the idea is to actually define an operator that will allow us to transform the state into a linear combination of the ones that are solutions to our function. This can be done easily, since the following result holds, giving us the final form for the quantum search algorithm.

#### Theorem 1.6.5: Quantum search

A general state  $H^{\otimes n} |0\rangle$  can be decomposed in two main parts inside  $\mathcal{N}$  as

$$|\psi\rangle = H^{\otimes n} |0\rangle = \sin \frac{\theta}{2} |\beta\rangle + \cos \frac{\theta}{2} |\alpha\rangle, \quad (1.117)$$

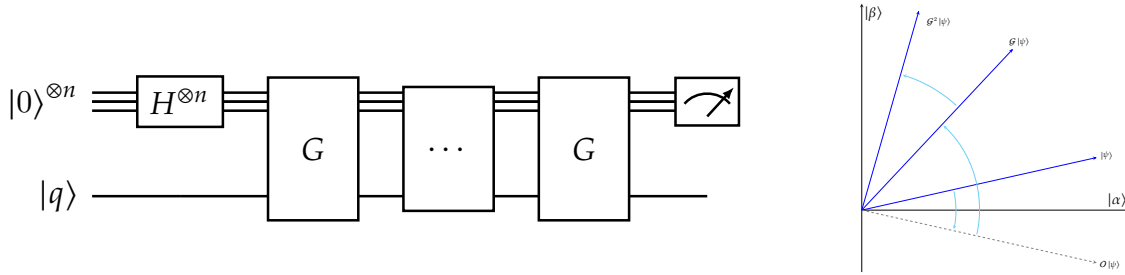
where  $|\beta\rangle$  is a linear combination of all the solutions of  $f(x)$ , while  $|\alpha\rangle$  a combination of all the non-solutions. The **Groover opearator**  $\mathcal{G} = \tilde{\mathcal{U}}\mathcal{O}$  acts on this state by rotating the angle  $\theta$  in the direction of  $|\beta\rangle$ , so that if applied  $R$  times we get

$$\mathcal{G}^R |\psi\rangle = \sin \left( \frac{2R+1}{2} \theta \right) |\beta\rangle + \cos \left( \frac{2R+1}{2} \theta \right) |\alpha\rangle. \quad (1.118)$$

Therefore, we can select the right number of  $\mathcal{G}$  to perform in order to obtain the maximum likelihood to obtain as a result of the measurement the state  $|\beta\rangle$  with the solutions of the function.

**Proof:** We can start by looking at the general state  $|\psi\rangle$  and see how can be rewritten as linear combination of solution and non-solution as follows

$$|\psi\rangle = \frac{1}{\sqrt{N}} \left( \sum_{k \in \mathcal{S}} |k\rangle + \sum_{k \in \mathcal{E}} |k\rangle \right) = \sqrt{\frac{M}{N}} \left( \frac{1}{\sqrt{M}} \sum_{k \in \mathcal{S}} |k\rangle \right) + \sqrt{\frac{N-M}{N}} \left( \frac{1}{\sqrt{N-M}} \sum_{k \in \mathcal{E}} |k\rangle \right), \quad (1.119)$$



**Figure 1.8:** Graphical representation of the Grover's circuit, to the left, with transformations that happens to the state in the quantum search algorithm, to the right, in order to understand how the two reflection transforms into a rotation.

now we can see how  $\sqrt{M/N}$  and  $\sqrt{N - M/N}$  can be setted as the sine and cosine, while the superpositions that are multiplying are exactly the linear combinations of all solutions and non-solutions respectively. Therefore, we effectively have that the following is true

$$H^{\otimes n} |0\rangle = \sin \frac{\theta}{2} |\beta\rangle + \cos \frac{\theta}{2} |\alpha\rangle, \quad (1.120)$$

then we shall see what the operator  $\mathcal{G}$  does on it, and that can be done by looking at  $\mathcal{O}$  and  $\tilde{\mathcal{U}}$  separately. First, it's easy to understand how the oracle gate simply changes the sign of the  $|\beta\rangle$  coefficient, meaning that using it simply performs a reflection of the state respect to the  $|\alpha\rangle$  one. Then, the inversion by the average operator instead perform another inversion in this case that is respect to the state  $|\psi\rangle$ , so that two inversions apply one after giving a final rotation of the state as

$$\mathcal{G} |\psi\rangle = \sin \left( \frac{3}{2} \theta \right) |\beta\rangle + \cos \left( \frac{3}{2} \theta \right) |\alpha\rangle, \quad (1.121)$$

which is also depicted in Fig. (1.8). Meaning that the rotation allowed for the state to become closer to the  $|\beta\rangle$  state, and if we continue to perform it we will get as a state

$$\mathcal{G}^R |\psi\rangle = \sin \left( \frac{2R+1}{2} \theta \right) |\beta\rangle + \cos \left( \frac{2R+1}{2} \theta \right) |\alpha\rangle. \quad (1.122)$$

In this way we can see how the probabilities of having as result  $|\beta\rangle$  scales up since the angle gets closer to  $\pi/2$ , which gives 1 as coefficient for that state. Still, it's possible to do too many rotations and ending up to go past the perfect value and increasing the coefficients for the  $|\alpha\rangle$  state, still one can see how if  $R$  is choosen as

$$R = \left\lceil \frac{\arccos \sqrt{M/N}}{\phi} \right\rceil, \quad \phi \leq 4, \quad (1.123)$$

where  $[x]$  is the integer part function, we will have probabilities that looks like

$$P_\beta > 1/2, \quad P_\alpha < 1/2. \quad (1.124)$$

Basically, we can select  $\phi$  to obtain the most optimized result possible, and already we know that by default the probabilities are in favor of the right result that we are serching for. ☺

Therefore, this algorithm allows us to search inside a set of states of dimensions  $2^n$  the ones that satisfy the condition imposed by a wanted function  $f$ . Also, all of this is done by using a minimal number of operations that scales linearly with the number of qubit that we are using, beating in an incredible way the classical search algorithm that scales exponentially.

## Hamiltonian simulation

One of the main goals that quantum computer scientists want to achieve is the possibility of simulating complex quantum systems efficiently thanks to the intrinsic quantum nature of QCs themselves. Basically, the goal is to find a way to solve the SE inside a quantum computer by finding out the time evolution operator of the system, which we recall in a time-independent  $\mathcal{H}$  is

$$|\psi(t)\rangle = \mathcal{U}(t) |\psi_0\rangle, \quad \mathcal{U}(t) = \exp\left(-i \frac{\mathcal{H}t}{\hbar}\right). \quad (1.125)$$

In order to do that we shall approximate the form of  $\mathcal{U}(t)$  somehow using the qubit that we have available, and the idea is the one of discretize it along with also  $|\psi\rangle$ . Basically,  $|\psi\rangle$  will be thought of as a function  $\psi(\mathbf{x})$  where  $\mathbf{x} \in \mathbb{R}^d$ , and we are going to store only a set of values, of that function defined by the position on a lattice  $\{x_j\}_j$ , inside the qubit. Such discretization, then, needs to be used also for the Hamiltonian, having an approximate form also for it, which defines the evolution operator that we can then use to evolve the state.

To understand better how this is done we will assume that  $\mathcal{H}$  is composed by a series of smaller components  $\mathcal{H}_k$  containing only short range interactions, like

$$\mathcal{H} = \sum_{k=1}^L \mathcal{H}_k = \sum_{\langle nm \rangle} Z_n Z_m + h \sum_l (Z_l + X_l). \quad (1.126)$$

This is an Ising Hamiltonian that has only on-site terms with longitudinal and transversal field, and an interaction term that still applies only on nearest neighbours. Basically, we assume that  $\mathcal{H}_k$  has interaction that at most accounts for a finite number of elements near a particular qubit. Inside these assumptions we can easily go and find out a way to approximate the evolution operator by using a Taylor expansion as

$$\mathcal{U}(\Delta t) = \sum_{k=0}^{\infty} \frac{1}{k!} \left(-i \frac{\mathcal{H} \Delta t}{\hbar}\right)^k = \mathbb{1} - i \frac{\mathcal{H} \Delta t}{\hbar} + \mathcal{O}(\Delta t^2), \quad (1.127)$$

where  $\Delta t$  is the time discretization, assumed to be small. Also, from this point on we will assume  $\hbar = 1$  for simplicity of notation. We can see how the sum can be truncated to obtain an approximation of  $\mathcal{U}$  that becomes better as the time discretization goes down, which is great but not too great. A better approach would be taking advantage of the decomposition of the Hamiltonian in pieces in order to rewrite the exponential as a product of simpler operators that we know how to work with. Nevertheless, it's easy to see how also in the simple case we have looked at the  $\mathcal{H}_k$  does not commute  $[\mathcal{H}_k, \mathcal{H}_m] \neq 0$ , meaning that

$$\exp\left(-it \sum_k \mathcal{H}_k\right) \neq \prod_k \exp(-it \mathcal{H}_k). \quad (1.128)$$

This seems to set the sloppy Eq. (1.127) as the only possible way to approximate the operator, but it's not completely true. In fact, it's possible to demonstrate a result that saves the day.

### Theorem 1.6.6: Trotter decomposition

Taken two non commuting operators  $A, B$  it's possible to decompose the exponential in the fol-

lowing way

$$\exp[i(A+B)t] = \lim_{N \rightarrow \infty} \left( e^{i\frac{t}{N}A} e^{i\frac{t}{N}B} \right)^N. \quad (1.129)$$

**Proof:** We can use Eq. (1.127) to write down the following form

$$e^{i\frac{t}{N}A} e^{i\frac{t}{N}B} = \left[ \mathbb{1} - i\frac{\Delta t}{N}A + \mathcal{O}(1/N^2) \right] \left[ \mathbb{1} - i\frac{\Delta t}{N}B + \mathcal{O}(1/N^2) \right] = \mathbb{1} + i\frac{t}{N}(A+B) + \mathcal{O}(1/N^2), \quad (1.130)$$

which elevated to the power of  $N$  allow for the following result to show up

$$\left( e^{i\frac{t}{N}A} e^{i\frac{t}{N}B} \right)^N = \mathbb{1} + \sum_{k=1}^N \binom{N}{k} \frac{[it(A+B)]^k}{N^k} + \mathcal{O}(1/N). \quad (1.131)$$

Here, is possible also to see how the binomial coefficients can be approximated in the following way

$$\binom{N}{k} \frac{1}{N^k} \sim \frac{1}{k!} + \mathcal{O}(1/N), \quad (1.132)$$

having that the final form becomes exactly the sum of the Taylor expansion wanted in the limit of  $N \rightarrow \infty$ . ☺

Such a result allows for a better way to approximate the evolution operator by setting a certain discretization defined by  $N$  and then perform several steps in the middle. This can be also done analitically to find out some important closed formulas in order to have a better computational approach, and the most used one is the **Baker Hausdorff formula**

$$e^{i(A+B)\Delta t} = e^{iA\Delta t} e^{iB\Delta t} e^{-\frac{i}{2}[A,B]\Delta t^2} + \mathcal{O}(\Delta t^3). \quad (1.133)$$

Which is a good improvment of the previous naive approximation.

Let's now imagine that we want to work in order to simulate a 1D particle inside a potential  $V(x)$ , we will have that the system is described as

$$\mathcal{H} = \frac{\hat{p}^2}{2m} + V(\hat{x}), \quad x \in [-d, d]. \quad (1.134)$$

In this situation we discretize the space to have a spacing  $a = 2d/N$  and the points on the lattice that we are working with are described by the values

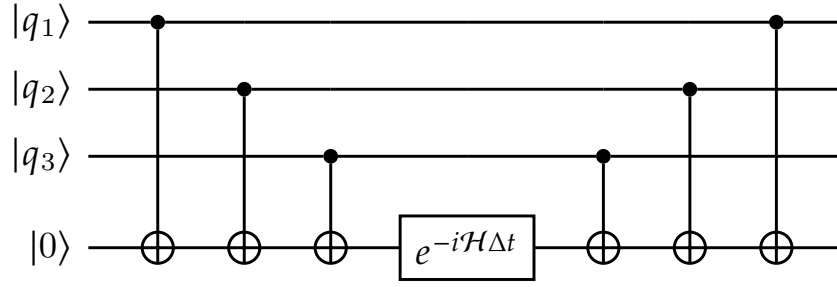
$$x_j = aj, \quad j \in [-N/2, N/2]. \quad (1.135)$$

In this way we will define the position basis as the set  $\{x_j\}$  so that the following is true

$$V(\hat{x})|x_j\rangle = V(x_j)|x_j\rangle, \quad (1.136)$$

having that they diagonalize the potential part of the Hamiltonian. Using this base also the state of the particle can be written as a linear combination of the type

$$|\psi\rangle = \sum_{j=-N/2}^{N/2} \alpha_j |x_j\rangle. \quad (1.137)$$



**Figure 1.9:** Quantum circuit used in order to simulate the simple system of the example, if the circuit is followed it's effectively possible to see how the final form is exactly the initial state plus the wanted phase and the ancilla qubit.

Therefore, this set allows for the discretization of both the state of the particle and a part of the Hamiltonian, to complete our needs we will need also to store the momentum part of the Hamiltonian, but it's not diagonal in such base. For this reason, the idea is to write  $\hat{p}$  as the Fourier transform of the position operator  $\mathcal{U}_{FT} \hat{x} \mathcal{U}_{FT}^\dagger$  since we know to perform the Fourier transform of the state, meaning that we can effectively evaluate also  $\hat{p} |x_j\rangle$  having all that we need.

#### Example 1.6.1 (Simple case)

The prof showed a really simple example of system simulation on a quantum computer, where the Hamiltonian is simply

$$\mathcal{H} = Z_1 \otimes Z_2 \otimes Z_3, \quad (1.138)$$

meaning that three qubit are studied and the base is so  $|q_1 q_2 q_3\rangle$ . We shall also remember that the effect of  $Z$  is changing sign to the state if it's value is 1 or not touching it otherwise. Knowing it allows us to find out how the Hamiltonian act on a general state having

$$\mathcal{H} |q_1 q_2 q_3\rangle = \delta_{q_1 q_2 q_3} |q_1 q_2 q_3\rangle, \quad \delta_{q_1 q_2 q_3} = \begin{cases} 1 & \#1 \text{ is even,} \\ -1 & \#1 \text{ is odd} \end{cases}. \quad (1.139)$$

Using this we can implement a quantum circuit that simulates the evolution by adding the correct phase to a general state of the base every time is applied to it. The circuit we are talking about is represented in Fig. (1.9), where one can see how the ancilla qubit will be 1 before the phase gate if an odd number of 1 are present in the initial state, and then will return to the initial state. Therefore, the final result is simply  $e^{i\delta\Delta t} |q_1 q_2 q_3\rangle \otimes |0\rangle$ , the state has effectively evolved as we wanted.

## 1.7 Universality of quantum computation

An important property that we want to obtain inside our computer is the universality of the logic we are using. Meaning that we aim in being able to represent all the possible function using only a restricted number of operations in order to implement only those inside our machine, making the implementation

much easier. In order to demonstrate it inside quantum computers we need first to make this concept more formal, aiming so to a definition like the following.

### Definition 1.7.1: Universality

A type of computation is universal if exist a small set of gates that can be used to write any algorithm.

We know how classical computations have such properties since it's possible to demonstrate how all the ingredients that are needed to compose every circuit are the following:

- 1 Wires, connecting different gates conserving states;
- 2 Ancilla bits, in order to do more complex operations;
- 3 FANOUT to duplicate the state of a bit;
- 4 CROSSOVER swapping states between bits;
- 5 AND, XOR and NOT gates able to generate all the others.

It's also possible to use only the NAND gate to simulate all of AND, XOR and NOT being the universal gate of classical logic.

What we want to do is see how the following components are also what we need inside quantum computations to be universal, and components from 1 to 4 are already given to us by discussions in precedent sections. Therefore, all remains is to see if exist a set of gates that allow us to write down all possible unitary operations doable on a set of qubit.

## Gates decomposition

To see how to recreate universality we need to see if a general gate  $\mathcal{U}$  acting on  $n$  qubit, so a general  $2^n \times 2^n$  unitary matrix, can be decomposed in several similar operations. This is duable using the concept of two-level gate defined as.

### Definition 1.7.2: Two-level gates

A gate is defined to be two-level if leaves invariant  $2^n - 2$  states and modify in a non-trivial way the remaining two. Basically, acts only on two specific states.

Using such gates we can write down all possible unitary matrices and this can be seen in the following result

#### Theorem 1.7.1: Two-level universality

Every unitary matrix  $\mathcal{U}$  with dimension  $d \times d$  can be decomposed using at most  $O(2^{2n})$  two-level gates.

**Proof:** We haven't really seen a rigourus proof of such a statement we have only worked out the case with  $d = 3$ . Let  $\mathcal{U}$  be a generally  $3 \times 3$  unitary matrix as

$$\mathcal{U} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}, \quad (1.140)$$



we can choose some two-level operations to perform depending on its values. In particular, we choose

$$\text{if } d = 0 \quad \mathcal{U}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{if } d \neq 0 \quad \mathcal{U}_1 = \begin{pmatrix} \frac{a^*}{\sqrt{|a|^2+|b|^2}} & \frac{b^*}{\sqrt{|a|^2+|b|^2}} & 0 \\ \frac{b}{\sqrt{|a|^2+|b|^2}} & -\frac{a}{\sqrt{|a|^2+|b|^2}} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (1.141)$$

In this way it's possible to see how multiplying the two unitary matrices one can obtain the following result

$$\mathcal{U}_1 \mathcal{U} = \begin{pmatrix} a' & b' & c' \\ 0 & e' & f' \\ g' & h' & i' \end{pmatrix}, \quad (1.142)$$

an element of the matrix was setted to 0. We can then go on and use another transformation in a way similar to the one before as

$$\text{if } c' = 0 \quad \mathcal{U}_2 = \begin{pmatrix} a'^* & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{if } c' \neq 0 \quad \mathcal{U}_2 = \begin{pmatrix} \frac{a'^*}{\sqrt{|a'|^2+|c'|^2}} & 0 & \frac{c'^*}{\sqrt{|a'|^2+|c'|^2}} \\ 0 & 1 & 0 \\ \frac{c'}{\sqrt{|a'|^2+|c'|^2}} & 0 & -\frac{a'}{\sqrt{|a'|^2+|c'|^2}} \end{pmatrix}. \quad (1.143)$$

Applied it we can see how the matrix simply further becoming simply

$$\mathcal{U}_2 \mathcal{U}_1 \mathcal{U} = \begin{pmatrix} 1 & b'' & c'' \\ 0 & e'' & f'' \\ 0 & h'' & i'' \end{pmatrix}, \quad (1.144)$$

but due to unitary constrain also  $b'' = c'' = 0$  meaning that we can simply finish the work by using the inverse  $\mathcal{U}_3$  that is a two-level unitary matrix and have that

$$\mathcal{U}_3 \mathcal{U}_2 \mathcal{U}_1 \mathcal{U} = \mathbb{1}, \quad (1.145)$$

leaving us with the fact that  $\mathcal{U}_3 \mathcal{U}_2 \mathcal{U}_1 = \mathcal{U}^\dagger$ , reconstructing the full matrix using simple two-levels operations. It's easy to understand how the process can be scaled up to whatever dimension of the matrix allowing for the decomposition in two-level operations of all possible gates. We can also have an estimate of the number of operations inside such a decomposition and the idea is that for a  $d$  dimensional matrix you need at most  $d - 1$  operations to eliminate the first row and have a  $d - 1$  matrix, then  $d - 2$  operations needs to be done for reducing it further and so on having at most a complexity of

$$(d - 1) + (d - 2) + \dots + 1 = \frac{d(d - 1)}{2}. \quad (1.146)$$

Taking into account that inside a quantum computer  $d = 2^n$  we will have  $\mathcal{O}(2^{2n})$  as complexity. ☺

Basically we have demonstrated o how two-level gates are universally able to represent every possible gate taht we can imagine inside a quantum computer.

We can then go further, since the set of two-level gates is infinite, and we can't imagine implementing an infinite number of gates inside a software to reproduce all possible gates. In fact, we can restrict our view simply to the set of CNOT, that we will use to implement the SWAP, and of single-qubit gates meaning that we manipulate only a single qubit.

### Theorem 1.7.2: CNOT and single qubit universality

Every two-level gate can be decomposed in a series of CNOT operations followed by a single qubit manipulation.

**Proof:** Basically we can imagine every two-level gate as a particular unitary matrix with all one on the diagonal and four coefficients around inside it needed to manipulate the two-states designed, similar to the one seen in the previous demonstration. We want to rewrite such a transformation simply using swap operations, that can be seen as permutations of the states, and a single qubit operations that can be written as

$$\tilde{\mathcal{U}} = \begin{pmatrix} 1 & \dots & \dots & \dots & \dots & 0 \\ \vdots & \ddots & & & & \vdots \\ \vdots & & a & b & & \vdots \\ \vdots & & c & d & & \vdots \\ \vdots & & & & \ddots & \vdots \\ 0 & \dots & \dots & \dots & \dots & 1 \end{pmatrix}. \quad (1.147)$$

To see this we can make an example with a  $4 \times 4$  matrix acting in the following way. The matrix  $\mathcal{U}$  we are working with is the following

$$\mathcal{U} = \begin{pmatrix} a & 0 & b & 0 \\ 0 & 1 & 0 & 0 \\ c & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (1.148)$$

what we need to do is first see how we can swap the states, meaning that by using a permutation we can change the column and the rows of the matrix as if we are changing the base of computation. In this way by using the permutations that bring  $|10\rangle$  to  $|01\rangle$ , meaning the SWAP gate composed on CNOT in Eq. (1.24), and see how the following it's true

$$SWAP(|01\rangle \rightarrow |10\rangle)^\dagger \mathcal{U} SWAP(|01\rangle \rightarrow |10\rangle) = \begin{pmatrix} a & b & 0 & 0 \\ c & d & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (1.149)$$

which is simply a single qubit gate and inverting the operation we can see how  $\mathcal{U}$  can be written with two swaps and a  $\tilde{\mathcal{U}}$  operation. On a much larger scale the idea is to perform a series of swap to bring the state close together forming a single qubit operation, do that, and then reswap the states to return to the original form. This approach is called **gray code** algorithm and allows to generally write down every two-level operation as one single qubit one and  $2(n - 1)$  swaps at most, where  $n$  is the number of qubit. 😊

Thanks to this last result we can say that the all set of the two-level operation can be reduced further to the one comprising CNOT and single qubit operations, much smaller and simpler to implement inside a real quantum computer. Also, we can make an estimate of the total number of operations needed in order to recreate exactly a particular gate  $\mathcal{U}$  inside a quantum computer composed by  $n$  qubit. In

fact, we have that every operation can be written as, at most,  $2^{2n}$  two-level operation and every one of them need a  $\tilde{\mathcal{U}}$  and  $2(n - 1)$  swaps giving a total complexity of

$$\mathcal{O}(n^2 2^{2n}), \quad (1.150)$$

which is quite high but allows to have an exact result for every possible operation.

### Single qubit gates approximation

We have seen how in order to decompose every possible gate inside a quantum computer we need the CNOT one and all the single qubit operation that we can perform. Still, the latter have infinite possibilities inside quantum computers, and we can't implement all of that on a hardware level. For this reason, we want to create a small finite set of operation that can reproduce all of them also in an approximate way.

The idea is to find out a set of operations that allow us to find out a  $\mathcal{V}$  that minimize the error in the estimate of the real  $\mathcal{U}$ , where the error is defined as

$$E[\mathcal{U}, \mathcal{V}] = \max_{|\psi\rangle} \|(\mathcal{U} - \mathcal{V})|\psi\rangle\|. \quad (1.151)$$

There are a several way in which such a minimization can be done using various set of operator but the most used one is the standard composed by the following

$$\mathcal{S} = \{H, CNOT, T\}. \quad (1.152)$$

Only those three gates will be able to create every possible operation inside a quantum computer. In particular, we need to demonstrate that we are able to generate all the single qubit gate up to a certain error using them, and can be seen as follows.

#### Theorem 1.7.3: $T$ and $H$ universality

Let  $R_{\mathbf{n}}(\theta)$  be a generic rotation of the single qubit state, where  $\mathbf{n}$  is the versor that defines the rotation and  $\theta$  the angle of the rotation. Then for every  $\epsilon > 0$  exist a  $\mathbf{m} \in \mathbb{N}^3$  so that by defining  $R_T = THTH$  we will have

$$E[R_{\mathbf{n}}(\theta), R_T^{m_1} H R_T^{m_2} H R_T^{m_3}] < \epsilon. \quad (1.153)$$

**Proof:** Also here we haven't done a real demonstration of the result, but still we can see the main reasons why that works. The idea is to first recall that every rotation can be written as

$$R_{\mathbf{n}}(\theta) = \exp\left(-i\frac{\theta}{2}\mathbf{n} \cdot \boldsymbol{\sigma}\right) = \cos\left(\frac{\theta}{2}\right)\mathbb{1} - i\sin\left(\frac{\theta}{2}\right)\mathbf{n} \cdot \boldsymbol{\sigma}, \quad (1.154)$$

and then look at how the operator  $R_T$  look like by writing it down explicitly. To do that we need first to recall two things

$$T = \exp\left(-i\frac{\pi/4}{2}Z\right), \quad HTH = \exp\left(-i\frac{\pi/4}{2}X\right), \quad (1.155)$$

so that we can easily write down the wanted rotation as

$$R_T = \exp\left(-i\frac{\pi/4}{2}Z\right) \exp\left(-i\frac{\pi/4}{2}X\right) = \cos^2\left(\frac{\pi}{8}\right) \mathbb{1} - i \left[ \cos\frac{\pi}{8}(X+Z) + \sin\frac{\pi}{8}Y \right] \sin\left(\frac{\pi}{8}\right). \quad (1.156)$$

Such thanks to the properties of the special angle  $\pi/8$  such operator defines exactly a rotation respect to a particular versor  $\tilde{\mathbf{n}}$  and angle  $\tilde{\theta}$  defined by

$$\tilde{\mathbf{n}} = \frac{(\cos\frac{\pi}{8}, \sin\frac{\pi}{8}, \cos\frac{\pi}{8})}{\sqrt{2\cos^2\frac{\pi}{8} + \sin^2\frac{\pi}{8}}}, \quad \cos\frac{\tilde{\theta}}{2} = \cos^2\frac{\pi}{8}. \quad (1.157)$$

Now, this is really good since the angle  $\tilde{\theta}$  is irrationale meaning that for every angle  $\phi \in [0, 2\pi]$  and  $\epsilon > 0$  exist an integer  $m$  so that

$$|\phi - \|m\tilde{\theta}\|_{2\pi}| < \epsilon, \quad (1.158)$$

where here  $\|\cdot\|_{2\pi}$  indicates the module of  $2\pi$ . Meaning that we are able to come close to every possible angle of rotation that we want in this contest. In this way we can understand how the relation in Eq. (1.154) holds true in general so that a single qubit gate can be approximated using in general a  $O(m^2/\epsilon)$  number of gates. ☺

That's it, we have demonstrated how by using a set composed by only three gates we are able to generate universally every possible operation that we can imagine, and that is exactly what happens inside a quantum computer. If you give as input to the computer to perform a rotation of  $\pi/9$  in reality he is decomposing it in a series of  $\pi/8$  and hadamart gate to approximate it to a ceratin precision.

In reality exist a much more rich theory that allows us to give a better approximation that we have only cited during the course and is the following.

#### Theorem 1.7.4: Solovay-Kitaev theorem

An arbitrary single qubit gate can be approximated to an accuracy  $\epsilon$  using at most

$$O\left(\log^c \frac{1}{\epsilon}\right), \quad c \sim 2. \quad (1.159)$$

Meaning that if we need  $m$  CNOT and  $m$  single qubit gates we will have a final complexity for the operation given by  $O(m \log^c(m/\epsilon))$ , not bad at all.

## 1.8 Quantum error coorection

It's perfectly known how computers are affected from machine errors, where bits can possibly change states due to various external factors. The same thing happens to quantum computers, in paticular qubits are much more delicate having a greate number of possible fluctuations that needs to be taken into account when we run an algorithm. Therefore, we need to create protocols that allow us to correct such variations of the state during the computations as we have inside classical computers. In order to understand how this can be done on a quantum level is really usefull to see its classical version and grasp the major concepts to use also on quantum computers.

On classical computers a bit can change state randomically going from 0 to 1 or viceversa with a certain probability  $p$  inside the circuit. Such a situation define a particular type of transformation that we will define as follows.

### Definition 1.8.1: Binary symmetric channel

When we will have a transformation  $X : \{0, 1\} \rightarrow \{0, 1\}$  acting on the state of a logic variable so that  $X(0) = 1$  with a certain probability  $p$  and  $X(0) = 0$  with  $1 - p$  it's called binary symmetric channel.

Such are the randomic transformation that acts on the single bits inside a normal computer, having that they can be overcome using a regular routine that has become standard over the years. The idea is the following, we are going to **encode** the state of a single bit using three of them such as follows

$$0 \mapsto 000, \quad 1 \mapsto 111. \quad (1.160)$$

In this way we will have that an error on one the bits defining the state can be **dyagnosed** by looking at the other two. In particular, we will have that if we start from 000 and on the line we see a 100 we know that an error has occoured, and we are able to correct it. We know how the possible flips that could have happened bringing to the 100 are: a flip to the first bit in the state 000, and two bits flips for second and third in the 111 state. In this context, in order to **recover** the state, the idea is to return to the state that has the higher probability of being the right one, in the situation we have discussed is the 000 since we have a probability  $p$  of flipping to that state while 111 has a probability  $p^2$ . Such a protocol, called **majority rate**, allows to brign the possibility of the error to propagate inside the circuit to the one given by the following result.

#### Theorem 1.8.1: Majority rate error probability

The probability of committing an error by using the majority rate protocol is given by

$$p_{err} = 3p^2 - 2p^3. \quad (1.161)$$

**Proof:** In the protocol we are basically ignoring the possibilities of having more than one flip, so tha the error is given by the possibility of having every two flip or a three bit flip, giving

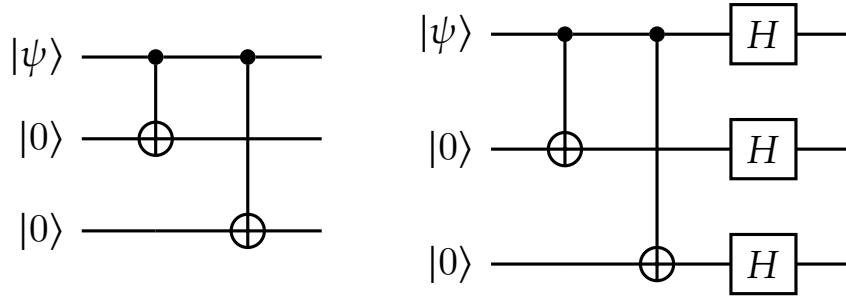
$$p_{err} = 3p^2(1 - p) + p^3 = 3p^2 - 2p^3. \quad (1.162)$$

☺

It's also interesting to see how such a simple protocol is also not always usefull. In fact, the inequality  $3p^2 - 2p^3 < p$  is true only if  $p < 1/2$  or the opposite is real, having that such an error correction can be used only if the probability of error of a single bit is already bellow that threshold.

This was a simple protocol, but it still shows the main important features that a general error correction code should have in principle. We can sintetyze what is needed in order to correct the error on an abstract level as: encoding of the state, dyagnosis of the error committed, and recover of the original state. All of this is not difficult to imagine done inside a normal computer, still working with quantum information makes thing a lot harder due to the presence of some complications:

- 1 The no cloning theorem makes diffucult coping the state, having that encoding is not too much straightforward;
- 2 The types of errors inside a qubit are larger than the simple flip of a bit, in fact we have the whole range of continuos unitary transformation to span;



**Figure 1.10:** Encoder circuit for the simple single qubit error correction routines, in particular the one on the left is the one for the state flip while on the right there's the one for the phase flip.

- 3 Recover a state it's not easy to do especially remembering that if we measure a state we are going to lose it forever.

Nevertheless, we will show step by step how such a task can still be brought up generating an error correction protocol that allows to save qubits information to be corrupted in the process.

### Single qubit flip errors

To start this journey we shall look into the simplest case of quantum error that we can imagine, the quantum version of the bit flip. Assume that there's a probability  $p$  that a gate  $X$  is applied to the state  $|\psi\rangle$  of the qubit inside the circuit. One shall recall how the  $X$  gate acts as a NOT inside a quantum circuit generating a flip of the  $|0\rangle$  and  $|1\rangle$  gates. In this way we can imagine to work exactly as in the classical case and start by encoding the states using three qubits so that a general state becomes

$$|\psi\rangle = a|0\rangle + b|1\rangle \mapsto a|000\rangle + b|111\rangle = |\psi_0\rangle. \quad (1.163)$$

Therefore, we shall start by understanding how to effectively do this, and the answer is easier than it seems thanks to the use of two CNOT as described in Fig. (1.10). The first circuit reported on the left of that figure creates a gate, that we will call  $E$ , that perform exactly the wanted operation. In this way the first step is performed, now we shall be able to understand if a state flip error happens by the fact that  $|\psi_0\rangle$  will change state going from Eq. (1.163) to one of the following

$$|\psi_1\rangle = a|100\rangle + b|011\rangle, \quad |\psi_2\rangle = a|010\rangle + b|101\rangle, \quad |\psi_3\rangle = a|001\rangle + b|110\rangle. \quad (1.164)$$

The last thing that we need to understand is how we can understand in which state is the system in order to then correct it. The idea is so to measure the state using a particular measure defined as

$$\{\hat{P}_k\}_{k=0}^3, \quad \hat{P}_k = |\psi_k\rangle\langle\psi_k|, \quad (1.165)$$

which will allow us to measure exactly which state we have without making it collapse. In fact, if we are in a state  $|\psi_j\rangle$  we will have that

$$p_k = \langle\psi_j|\hat{P}_k|\psi_j\rangle = \begin{cases} 0 & j \neq k \\ |a|^2 + |b|^2 & j = k \end{cases} = \delta_{jk}, \quad (1.166)$$

meaning that the only possible state in which  $|\psi_j\rangle$  can collapse into is itself. This will allow us to evaluate the state perfectly giving a good diagnose of the error that has occurred, a measure that allow us to do this is called **syndrome operator**. Subsequently, after understand the state of the system, we will be able to correct the error by applying an  $X$  operator on the flipped qubit.

To make an example of such measure different from the projector theoretical one, we can simply define it by using a series of  $Z$  operators. We remember how  $|0\rangle$  and  $|1\rangle$  are the eigenstates of the operator so that  $Z|0/1\rangle = +1/-1|0/1\rangle$ . Knowing it we can define the operators

$$Z_1Z_2 = Z_1 \otimes Z_2 \otimes \mathbb{1}_3 = [+1(|00\rangle\langle 00| + |11\rangle\langle 11|) - 1(|10\rangle\langle 10| + |01\rangle\langle 01|)] \otimes \mathbb{1}_3, \quad (1.167)$$

which shows how it returns  $+1$  if the first and second qubit are equal, and  $-1$  otherwise. In this way we can define a really clever syndrome operator measure by using only two operators  $Z_1Z_2$  and  $Z_2Z_3$ , having that by their value we can understand the error exactly. In fact, by knowing which qubits are equal or not allows us to tell exactly which of the four state we have in that specific moment.

This method is so an exact copy of the classical one setted in a quantum environment, and we can see how it is the same as before since the probability of an error to propagate using this routine is given exactly by Eq. (1.161). Meaning that this can be implemented only if the error rate is already below the 50%, but still this is a good simple algorithm since can be applied not only to the case of the state flip but also to other type of errors. For example, for a qubit not only the state can be change by external factor, but also the phase is vulnerable. In particular, we can imagine that now the operator that is applied with a probability  $p$  is  $Z$ , which applied on  $|\psi\rangle$  will flip the phase of  $|1\rangle$ . One may think that such an error needs to be treated differently from the one seen before, but that would be not true. In fact, we can see how if instead of the computational base we use  $|\pm\rangle$  to represent the states the problem returns to a flip of the state since  $Z|\pm\rangle = |\mp\rangle$ . Therefore, if we encode the states by using the circuit in Fig. (1.10) that performs the following transformation

$$|0\rangle \mapsto |+++\rangle, \quad |1\rangle \mapsto |--\rangle, \quad (1.168)$$

we can then define the syndrome measure by also changin base as the normal one and obtaining

$$H^{\otimes 3}Z_1Z_2H^{\otimes 3} = X_1X_2. \quad (1.169)$$

This will allow us to understand if a phase flip happens allowing us to recover the initial state without any problem, reducing the probability of error as seen before.

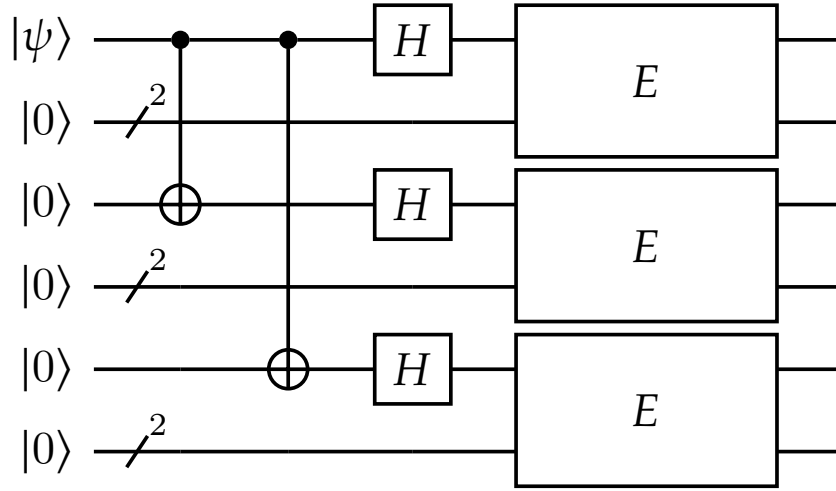
## Shor code

We have seen how we are able to correct both states flips and phase flips in a separate way, but inside a quantum computer we are interested in the full correction of the qubits aiming to a protocol allowing for the correction of both at the same time. Such a routine exist and is called **Shor code** that we shall see next.

The idea starts with the encoding of the states by using a total of 9 qubits mapping the computational base to the following monsters

$$|0\rangle = \left( \frac{|000\rangle + |111\rangle}{\sqrt{2}} \right)_{B1} \left( \frac{|000\rangle + |111\rangle}{\sqrt{2}} \right)_{B2} \left( \frac{|000\rangle + |111\rangle}{\sqrt{2}} \right)_{B3} = |+\rangle_{B1} |+\rangle_{B2} |+\rangle_{B3}, \quad (1.170)$$

$$|1\rangle = \left( \frac{|000\rangle - |111\rangle}{\sqrt{2}} \right)_{B1} \left( \frac{|000\rangle - |111\rangle}{\sqrt{2}} \right)_{B2} \left( \frac{|000\rangle - |111\rangle}{\sqrt{2}} \right)_{B3} = |-\rangle_{B1} |-\rangle_{B2} |-\rangle_{B3}. \quad (1.171)$$



**Figure 1.11:** Encoder for the Shor code, where the gates defined by  $E$  are encoders for the simple case of the state flip correction.

Basically defining three blocks with inside the same encoded state of before, having in fact an encoder circuit really close to the two seen previously as depicted in Fig. (1.11). Now, we can go and see what happens if we have a state or phase flip inside one of the qubit. In the case of the state flip, we can see how one of the blocks will change the state as in Eq. (1.164) so that we can use a generalization of the  $Z_1 Z_2$  measure to find out the error

$$Z_1 Z_2 = (Z_1 Z_2)_{B1} \otimes \mathbb{1}_{B2} \otimes \mathbb{1}_{B3}. \quad (1.172)$$

We have so an efficient way to spot the state errors and correct them by simply apply a NOT where needed. Then, for the phase flips instead we will have that one of the blocks will change having that  $|\pm\rangle$  will fo into  $|\mp\rangle$ . To spot which one of the blocks we need to correct we can use the same idea seen previously for the phase flip, where we can use the following operators

$$S_1 = X_{B1} X_{B2} = (X_1 X_2 X_3) \otimes (X_4 X_5 X_6) \otimes \mathbb{1}_{B3}, \quad (1.173)$$

$$S_2 = X_{B2} X_{B3} = \mathbb{1}_{B1} \otimes (X_4 X_5 X_6) \otimes (X_7 X_8 X_9). \quad (1.174)$$

In this way we can be sure to see every possible state and phase flip inside our circuit and be able to correct them properly. That was all we needed in our machines, in fact we can see how every possible error can be described in terms of those two primitive problems as follows.

#### Theorem 1.8.2: Universality of Shor code

Every possible error inside quantum computers can be decomposed in phase and state flips, so that Shor code forms a universal error correcting procedure.

**Proof:** In order to prove that we will need some notions from the theory of open systems. Meaning that inside the context of SE evolution the state is preserved, and no random error should occur all



is deterministic. Still, when the system is coupled with environment random change in the state can appear and  $|\psi\rangle$  evolves from a pure state to a mixed one. Such evolution can be described by a more general version of the dynamic equation called **Lindblad equation** which is well known, and shows an evolution that can be described by a **quantum channel** acting as follows

$$|\psi\rangle \mapsto \sum_i E_i^\dagger |\psi\rangle \langle\psi| E_i. \quad (1.175)$$

Where,  $E_i$  is a set of operators with the property of being complete meaning  $\sum_i E_i^\dagger E_i = \mathbb{1}$ , condition needed in order to conserve the norm of the state. Now, we can imagine having a 1-qubit situation so that  $E_i$  are  $2 \times 2$  matrices and therefore can be written as expansion of the Pauli matrices base as

$$E_i = e_0 \mathbb{1} + e_1 X + e_2 Y + e_3 Z. \quad (1.176)$$

We can interpret this result by seeing how the part with  $\mathbb{1}$  gives out the normal evolution, the SE like one with no errors, while all the others describe possible evolution when **decoherence**, errors, occurs. In particular, we can see that the possible errors are given by the state flip, the  $X$  part, phase flip, the  $Z$ , and the  $Y$  one that can be written as  $ZX$  giving rise to a situation when both occurs. Therefore, all possible errors are linear combinations of the primitive ones that Shor code can predict and cure. ☺

## 1.9 Hybrid algorithm

In the last moment of the course the prof just introduced the concept of hybrid algorithms, which is a way of incorporating quantum routines inside classical algorithms in order to speed up computations. The main idea is to use quantum computers in order to evaluate complex functions inside classical algorithms, like minimization ones, since they can do it faster.

Therefore, I will write down the simple example that she showed during class and stop there, since the course has effectively stopped at that moment.

### Quantum Approximation Optimization Algorithm

The idea is that we want to minimize the energy of a system finding the ground state of a Hamiltonian  $\hat{\mathcal{H}}_c$ . To do that an ansatz for the GS  $|\psi\rangle$  is created depending on some parameters  $|\psi(\theta)\rangle$  and the set  $\theta_0$  that minimize the function

$$f(\theta) = \langle\psi(\theta)|\hat{\mathcal{H}}_c|\psi(\theta)\rangle, \quad (1.177)$$

is found out by minimization. What we want to do is evaluate  $f$  by using quantum algorithm and find the minimum using the classical approach since QC still are not good at it. Therefore, the general outline of the algorithm should be the following:

- 1] Prepare the ansatz  $|\psi(\theta)\rangle$  through the use of a series of operator  $\{\mathcal{U}_{\theta_1}, \mathcal{U}_{\theta_2}, \dots, \mathcal{U}_{\theta_p}\}$ , where  $p$  is called depth of the algorithm;
- 2] Use  $\hat{\mathcal{H}}_c$  to measure the state obtained state  $|\psi(\theta)\rangle$   $N$  times, so that every time we will obtain a different weigenvalue  $E_\lambda$  with probability  $p_\lambda$  that we can use to approximate  $f$  as follows

$$f(\theta) = \sum_{\lambda} E_{\lambda} p_{\lambda} \approx \sum_{\lambda} E_{\lambda} \frac{n_{\lambda}}{N}. \quad (1.178)$$

- 3 Pass the result to the classical computer to perform the minimization algorithm.

One can easily understand how perform the computation Eq. (1.178) is able to approximate the value of  $f$  in a sufficiently accurate way that is easier to achieve than perform the integration of the high dimensional  $|\psi\rangle$ . Such an algorithm seems already good to work, but in reality problems are present on both first and third point of the whole construction. We will focus on the first step since it's more interesting as a problem how to effectively construct the  $\mathcal{U}_\theta$  operators, and we will only give a glimpse of the problem in the third point.

To construct the ansatz the main approach used is to work with evolution in the space of parameters guided by two Hamiltonians  $\hat{\mathcal{H}}_1$  and  $\hat{\mathcal{H}}_2$  so that the operators can be written as

$$\mathcal{U}_{\theta_i} = e^{i\theta_i \hat{\mathcal{H}}_1} e^{i\theta_{i+1} \hat{\mathcal{H}}_2}, \quad (1.179)$$

meaning that now, the parameters grows from  $p$  to  $2p$ . Using this forms allows for a good mobility inside the state space only if the two operators are not close together, meaning that if  $\hat{\mathcal{H}}_1 = \lambda \hat{\mathcal{H}}_2$  a problem would occur since the evolution will only span the state of  $\hat{\mathcal{H}}_1$  and all the possible ones. For this reason we need to take Hamiltonians that are different from each other's having

$$[\hat{\mathcal{H}}_1, \hat{\mathcal{H}}_2] \neq 0. \quad (1.180)$$

That is a requirement, but it's not sad that is the only one needed but still we don't know for certain, as we don't know if exist a possible best choice for the two operator in order to arrive at the result quicker than all the others way. Nevertheless, what we can do now is show the main tricks that are used in order to construct the state:

**Trotter.** One possible situation is when we have  $\hat{\mathcal{H}}_c = \hat{\mathcal{H}}_1 + \hat{\mathcal{H}}_2$  with the two pieces that do not commute. In that case we can use the Trotter result in order to approximate the evolution that brings to the ground state as

$$e^{it\hat{\mathcal{H}}_c} \approx \left( e^{i\epsilon\hat{\mathcal{H}}_1} e^{i\epsilon\hat{\mathcal{H}}_2} \right)^N. \quad (1.181)$$

**Exact Hamiltonian.** In the case we know the exact Hamiltonian of the system is quite helpful instead of going trotter using it so that  $\hat{\mathcal{H}}_1 = \hat{\mathcal{H}}_c$  and choose the second simply in order to not commute.

In both ways we should have at the end a series of operations that can span all the phase space depending on the parameters  $\theta$  that needs to be optimized. And the problem in the third point is exactly there, the optimization of  $\theta$  is a pain for several reasons. The main problem is that the energetic landscape inside parameter space looks really like a plateau with a big minimum in a certain point, is just like that we need to accept that. Nevertheless, such a situation is a complicated one since means that  $\nabla_\theta f(\theta) \approx 0$  in a wide range. Therefore, the simple gradient descend method can't be used efficiently we need more complex algorithms but find the right one is a challenge that we are still facing.

#### Example 1.9.1 (Maximal independent set)

At a last the prof made a simple example of minimization problem that can be tackled using those hybrid algorithms. The problem is to find out a particular set of nodes inside a network and to do that you describe the nodes using a state like

$$|0110011 \dots\rangle, \quad (1.182)$$

where every qubit represents a specific node and 1 means that it is in the wanted set, while 0 otherwise. The only thing that remains to do is to engineer a Hamiltonian that has the right state as the ground state, but that can be done by using an appropriate Ising model

$$\hat{\mathcal{H}}_c = \sum_i Z_i + P \sum_{\langle ij \rangle} Z_i Z_j, \quad f_c = \langle \psi | \hat{\mathcal{H}}_c | \psi \rangle. \quad (1.183)$$

In this way you only need to decide the right nearest neighbours and the value of the penalty  $P$  to obtain the wanted result. Also, this approach allows for the use of the second kind of approach to create the ansatz really easily by using as  $\hat{\mathcal{H}}_1 = \hat{\mathcal{H}}_c$  and  $\hat{\mathcal{H}}_2 = \sum_i X_i$  that usually can make the algorithm efficient.

#### Note

*It's interesting to understand also how in the approach described for the creation of the ansatz it's obvious that we will have only an approximation of the expected  $|\psi(\theta)\rangle$  state, but still that is not a problem. In fact, one can see how defining the fidelity of the procedure as*

$$Fid = \langle \psi_{exact} | \psi(\theta) \rangle,$$

*we have that a fidelity of 0.8 is already good to have values of energies precise up to six or seven decimals since the energy converges much faster than the eigenstate.*