Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN

Science, Cognition and Technology

Ciclo 27

Settore Concorsuale di afferenza: 11/C2

Settore Scientifico disciplinare: M-FIL/02

Exploring Quantum Speed-up Through Cluster-state Computers

Presentata da: Filippo Annovi

Coordinatore Dottorato	**Relatore**
Prof. Giuliano Pancaldi	**Prof.ssa Rossella Lupacchini**

Esame finale anno 2015

Abstract

The aim of this thesis is to investigate the nature of quantum computation and the question of the quantum speed-up over classical computation by comparing two different quantum computational frameworks, the traditional quantum circuit model and the cluster-state quantum computer. After an introductory survey of the theoretical and epistemological questions concerning quantum computation, the first part of this thesis provides a presentation of cluster-state computation suitable for a philosophical audience. In spite of the computational equivalence between the two frameworks, their differences can be considered as structural. Entanglement is shown to play a fundamental role in both quantum circuits and cluster-state computers; this supports, from a new perspective, the argument that entanglement can reasonably explain the quantum speed-up over classical computation. However, quantum circuits and cluster-state computers diverge with regard to one of the explanations of quantum computation that actually accords a central role to entanglement, i.e. the Everett interpretation. It is argued that, while cluster-state quantum computation does not show an Everettian failure in accounting for the computational processes, it threatens that interpretation

of being not-explanatory. This analysis presented here should be integrated in a more general work in order to include also further frameworks of quantum computation, e.g. topological quantum computation. However, what is revealed by this work is that the speed-up question does not capture all that is at stake: both quantum circuits and cluster-state computers achieve the speed-up, but the challenges that they posit go besides that specific question. Then, the existence of alternative equivalent quantum computational models suggests that the ultimate question should be moved from the speed-up to a sort of "representation theorem" for quantum computation, to be meant as the general goal of identifying the physical features underlying these alternative frameworks that allow for labelling those frameworks as "quantum computation".

Contents

Chapter 1

Overview

The aim of this thesis is to investigate the nature of quantum computation and the question of the quantum speed-up over classical computation by comparing two different quantum computational frameworks, the traditional quantum circuit model and the cluster-state quantum computer.

While the quantum circuit model is well-known, cluster-state quantum computation has not received broad attention within the community of philosophers of physics (with some exceptions mentioned within the thesis), primarily because of the recent appearance of the topic in the literature and the very technical nature of the publications, especially if compared to the more intuitive structure of the circuit model.

After an introductory survey of the theoretical and epistemological questions concerning quantum computation, the first part of this thesis will provide a presentation of cluster-state computation suitable for a philosophical audience, by integrating the technicalities with a general description that

enlightens the structural features of that framework. This presentation is accompanied by the illustration of some algorithms for cluster-state computers, that show how the features of cluster-state computers are used "at work", and how differently the same tasks are managed and solved by quantum circuits.

The comparison between quantum circuits and cluster-state computers, and their theoretical consequences, is then tackled in the rest of the thesis. In particular, it will be shown that the two frameworks have convergences and divergences, both on a physical and on an epistemological level.

On the physical level, it is shown that, in spite of quantum circuits and cluster-state computers being computationally equivalent, their differences can be considered as structural. While this claim could appear immediate from the fact that quantum circuits have a reversible dynamics and cluster-state computers do not, it is reinforced by sketching a fully-unitary, i.e. reversible, dynamical account of cluster-state computation, where each computational step is treated as an unitary transformation that entangles the measured qubit with its "measuring apparatus". This twisted, but consistent, account of cluster-state computation shows that the differences with quantum circuits can be considered as structural.

A second aspect revealed by this analysis is that entanglement plays a fundamental role in both quantum circuits and cluster-state computers. This supports, from a new perspective, the long-debated argument according to which entanglement can reasonably explain the quantum speed-up over classical computation.

In spite of both supporting entanglement as playing a decisive role in

the quantum speed-up, quantum circuits and cluster-state computers diverge with regard to one of the explanations of quantum computation that does actually accord a central role to entangling operations, i.e. the so-called Everett interpretation. This is tackled by accounting for cluster-state computation in "Everettian terms": this account makes use of the fully-linear dynamical picture previously sketched and thus can be made consistent, but turns out to be unable to provide an explanation for the quantum speed-up. Hence, while cluster-state quantum computation does not show an Everettian failure in accounting for the computational processes, it threatens that interpretation of being not-explanatory, which is the main thing it would be required of.

This thesis shows some convergences and divergences between quantum circuits and cluster-state computers that are valuable by themselves, but would necessitate to be integrated in a more general work meant to include also further frameworks of quantum computation, e.g. topological quantum computation.

However, what is revealed by this thesis is that the where-does-the-quantum-speed-up-come-from question, which has driven the epistemological research over quantum computation, is not well positioned to capture all that is at stake: both quantum circuits and cluster-state computers achieve the speed-up, but the challenges that they posit go besides that specific question.

Then, the existence of alternative equivalent quantum computational models suggests that the ultimate question should be moved from the speed-up to a sort of "representation theorem" for quantum computation. Here "representation theorem" should not be meant in a strict mathematical sense, but

just as the general goal of identifying which are the physical features underlying these alternative frameworks that allow for labelling those frameworks as "quantum computation".

This would be very fruitful for the understanding of quantum computation itself and for the more general debate over the foundations of quantum mechanics.

Chapter 2

Introduction

Among the several areas of research that have been originated by the connection between quantum physics and information theory, quantum computation is one of the most original and fascinating, both because of its practical applications and of its theoretical relevance.

The focus of this thesis will be on the second of these aspects.

Usually, the first thing one learns about quantum computers is that they are more powerful than classical ones. This suggests that there is some structural difference between quantum and classical computers, and that intuitively such difference should depend on which physical theory they are grounded upon, i.e. the principles of quantum mechanics allow for building up computational devices whose performances are better than those of classical devices.

Three main points are in need for clarification:

- What does it mean that a computational device is built upon the prin-

ciples of quantum mechanics?

- What does the better performances of quantum computers consist of?
- Which kind of theoretical relevance should quantum computation have?

The first two questions will be tackled in the following of this chapter, after a brief overview of the basics of quantum computation. The third question will be the subject of the rest of this thesis.

Quantum mechanics is undoubtedly the best available theory for describing and predicting physical phenomena. Hence, in principle there should be no difference between classical and quantum computational devices: all the physical systems corresponding to these computers ultimately obey to the laws of quantum mechanics, i.e. their dynamical evolution can be described by the Schrödinger equation. It is not possible to deny this without at the same time denying the universality of quantum mechanics and no one would seriously take such position.

When considering classical and quantum computation, however, a different level needs to be taken into account. In classical computation, the "classicality" refers to the symbolic representation of information and to the kind of operations allowed as computational steps. For instance, a (classical) unit of information can be seen as a two-states system which can never enter a superposition state, while two-states quantum systems are naturally allowed to be in a superposition (even if that cannot be observed); but this does not

mean that the particles of matter that constitute a classical computer cannot be in a superposition.

It is worth to recall the pioneering paper of Deutsch (1985):

> Intuitively, a computing machine is any physical system whose dynamical evolution takes it from one of a set of 'input' states to one of a set of 'output' states. The states are labelled in some canonical way, the machine is prepared in a state with a given input label and then, following some motion, the output state is measured. For a classical deterministic system the measured output label is a definite function f of the prepared input label; moreover the value of that label can in principle be measured by an outside observer (the 'user') and the machine is said to 'compute' the function f.

This is a very "physically-oriented" description of how computational devices work, but useful for the purposes of discussing classical and quantum computation: classical computation requires the dynamical evolution taking the computing machine form the set of input states to the set of output states to obey to classical physics, while quantum computation requires that evolution to obey to the laws of quantum physics. As will be shown in the following chapters, the laws of quantum physics allow for a very different range of computational frameworks to drive computational devices.

Quantum computers are usually known for outperforming their classical

counterparts on a number of relevant tasks, among which factorization is the one with the most important practical consequences. Hence, the question of providing an explanation for such quantum speed-up has been the driving question for all research efforts devoted to understanding and clarifying the relationship between quantum and classical computation.

However, when focusing on the relationship between quantum and classical computation, the ultimate issue should be whether there exists a fundamental divide between the two, and what should these divide consist of. This question really provides the deepest insights into the nature of quantum computation, since it is explicitly concerns what do quantum and classical fundamentally differ in and how this is related to the features of the corresponding physical theories.

In order to investigate where the divide between quantum and classical computation lies, a number of issues require to be taken into account that are not directly related to the different amount of resources necessary for quantum and classical computers to solve some given class of problems. What is suggested in this work is then that the speed-up question does not seem to capture all what is at stake concerning foundational issues relative to quantum computation; however, this is far from claiming that the relevance of the speed-up question is non-fundamental or overrated. It will be argued that it would make sense to discuss a quantum-classical computational divide even if the quantum speed-up were to be not in place.

Moreover, there exist alternative quantum computational frameworks, all achieving the speed-up over classical computers but with different structural

features and making a different use of quantum and classical resources. Quantum circuits and cluster-state quantum computers will be taken into account here. From this follows that the real foundational insight does not consist in showing how the speed-up is obtained, but in explaining if and why all these frameworks "share" a quantum-classical boundary, while at the same time they provide radically different pictures of the physical processes going on during a computation.

In order to investigate the divide between quantum and classical computation, the first step consists in acknowledging that such divide does actually exist. Where can a quantum-classical divide in computation be found?

Undoubtedly, it does not concern the tasks that quantum and classical computers are able to perform: it is well known their mutual capability of simulating each other, regardless of the amount of resources (time and size) required for the computation to be successfully executed. However, when the amount of computational resources is taken into account, we immediately see that a boundary between quantum and classical computation is in place, which concerns computational complexity. In order to draw this "complexity boundary" we could make use of Nielsen & Chuang's (2000) representation of quantum and classical complexity classes (see Figure 1), where:

- **P** is the class of computational problems that can be solved by a deterministic Turing Machine in a polynomial time;

- **NP** is the class of computational problems whose positive solutions can be verified by a non-deterministic Turing Machine in a polynomial

time;

- **PSPACE** is the class of computational problems that can be solved by a deterministic Turing Machine with a polynomial size (but not necessarily in a polynomial time);

- **BQP** is the class of computational problems that can be solved by a quantum computer in a polynomial time with a bounded probability of error (it is the quantum counterpart of the classical **BPP**).

The exact relation between **BQP** and the other three classes is not known yet, as is the relation between **P**, **NP** and **PSPACE**: it is certainly an inclusion relation, but it is not known whether it is a strict inclusion or not.

Assuming that sooner or later the relation between the complexity classes will be fully clarified, could the dotted line of Figure 1 be considered as the fundamental divide standing between quantum and classical computation? This claim should face at least two possible counterarguments:

- Computational complexity is not a physical issue, so it should not be considered as a legitimate ground for a quantum-classical divide, which instead is a physical issue.

- The quantum speed-up is not proven, even if it is very strongly believed to hold. In the highly unlikely case that a fast classical algorithm for factorization were to be discovered, there would be no speed-up and quantum and classical complexity classes would collapse on each other. Thus there would be no quantum-classical boundary at all.

Both of the arguments above suggest that the "complexity boundary" does not have the physical meaning it should possess in order to stand as a foundational issue. Before considering whether these arguments are actually decisive for ruling out computational complexity as the core of the quantum-classical computational divide, a couple of further considerations should be taken into account:

- The physical tools that quantum computers make use of (e.g. entangled states) are not available to classical computers.
- The formal structure on which quantum computers are based is not classical, since it makes use of properties of Hilbert spaces.

What these two last considerations show is that, independently of the existence of the quantum speed-up and of complexity concerns, a fundamental divide between quantum and classical is standing. Even if quantum and classical computers were to perform the same tasks with the same efficiency, it would still be a non-trivial question to explain the reason why computational devices based on the properties of quantum systems are able to solve certain classes of problems with a polynomial amount of resources.

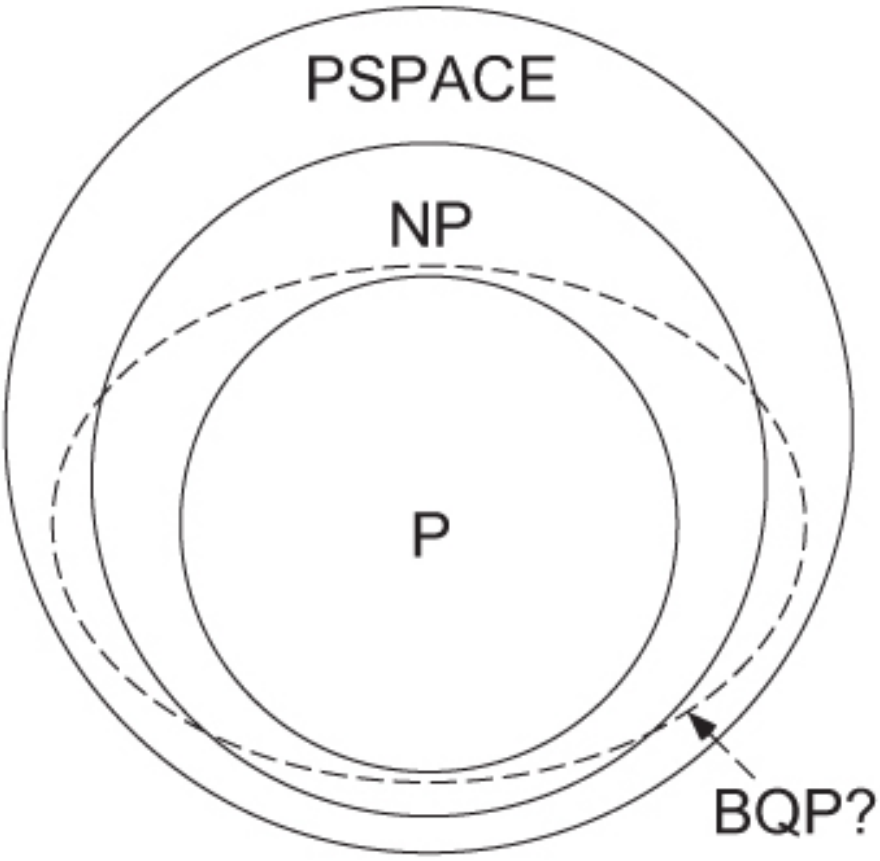

Figure 2.1: Quantum and classical complexity classes (Nielsen & Chuang 2000).

Chapter 3

Cluster-state quantum computation

When taking into account the quantum-classical computational divide, it is worth considering the existence of alternative quantum computational frameworks. The ones which will be taken into account here are the traditional quantum circuit and the so called cluster-state quantum computer. What is interesting with regard to these frameworks is that they are able to efficiently simulate each other, i.e. they can perform the same tasks with the same amount of resources, while at the same time providing a structurally different picture of the physical processes going on during a computation. In particular, as will be clear from the following, those two frameworks make a different use of classical and quantum resources; thus, it would look like that the boundary between quantum and classical moves on different lines according to which type of quantum computer is running. This is puzzling, since

that boundary is supposed to be what ultimately defines quantum (in contrast to classical) computation and consequently should not vary according to the choice of one among several equivalent frameworks. This suggests that a deeper work on the relationship between these two frameworks would help in enlightening the nature of quantum computation and would be valuable from a foundational point of view.

3.1 Quantum circuits

Quantum circuit make use of qubits, i.e. units of information corresponding to the state of a quantum two-state system:

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

where α and β are complex numbers satisfying the normalization condition $|\alpha|^2 + |\beta|^2 = 1$. The basis $\{|0\rangle, |1\rangle\}$ is known as the computational basis. In the usual graphical representation (see Figure 1), qubits are represented by horizontal lines (wires), just like bits in a classical circuit.

Any operation that can be performed on a qubit corresponds to a specific quantum gate, i.e. a bounded linear unitary operator $\mathbf{U}$ such that:

$$\mathbf{U}\mathbf{U}^\dagger = \mathbf{U}^\dagger\mathbf{U} = \mathbf{I},$$

where $\mathbf{U}^\dagger$ is the adjoint of $\mathbf{U}$ and $\mathbf{I}$ denotes the identity operator. Notice that unitarity implies that the transformations corresponding to quantum gates

are reversible.

The most known single-qubit gates are the Pauli operators, represented by the following matrices:

$$\sigma_x \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma_z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Another important single-qubit gate is the Hadamard operator:

$$\mathbf{H} \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

When applied to a computational basis state, this gate creates a superposition $(|0\rangle \pm |1\rangle)/\sqrt{2}$.

A relevant class of multi-qubits unitary gates are the controlled gates. In the simple case of a two-qubits controlled operation **U** (like the one shown in figure 1.1), the upper qubit is labelled as the control qubit, and the lower one as the target qubit. The operation **U** is applied to the target qubit if and only if the control qubit is in a desired state (usually $|0\rangle$ or $|1\rangle$). In general, any unitary operation can be obtained by combining the action of all the

single-qubit gates and of the two-qubit controlled gate:

$$\mathbf{CNOT} \equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

At the end of the computation, the output is read-off by measuring a subset of the qubits in an orthonormal basis (usually in the computational basis). As stated by the *Deferred Measurement Principle*, the measurement can always be postponed at the end of the computation without affecting the result[1].

A class of quantum gates consists of the controlled-phase gates

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix},$$

which will turn out to be relevant for the purposes of this work.

An example of a quantum circuit is shown in figure 2.1.

[1]Cf. Nielsen & Chuang (2000). Another feature of quantum circuits is the *Implicit Measurement Principle*, which states that, at the end of the computation, any unmeasured qubit (graphically represented by an unterminated wire) can be considered to be measured.

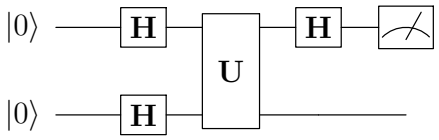

Figure 3.1: Example of a quantum circuit, which starts with two qubits prepared in the state $|0\rangle$. They both undergo a Hadamard transformation **H** and a two-qubits controlled gate **U**. The upper qubit is then subject to a second Hadamard transformation and is finally measured in the computational basis, in order to read-off the output. Since the lower qubit does not play any further role in the circuit after the application of **U**, it can be considered to be measured, by virtue of the Implicit Measurement Principle.

3.2 Cluster-state quantum computers

The picture of computational processes outlined by cluster-state quantum computation differs radically from the one provided by quantum circuits. However, these two models have been proven to be computationally equivalent, i.e. each of them is able to efficiently simulate any computation that can be executed by the other[2].

The essential ingredient of cluster-state quantum computation is the cluster $\mathcal{C}$, an ensemble of highly entangled qubits. This can be illustrated by considering a $n-$vertices graph: a qubit is associated to each vertex $\{a_1, \ldots, a_n\}$ and a neighbourhood relation holds between qubits connected by an edge. The preparation of the cluster consists of two steps:

1. All the n qubits are initialized to the product state

$$|+\rangle_{\mathcal{C}} = |+\rangle_{a_1} \otimes \ldots \otimes |+\rangle_{a_n},$$

[2]See Raussendorf & Briegel (2001b) for a proof. The content of the current and the following sections will also be based on that work.

where $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ [3].

2. The entanglement is generated by controlled-phase unitary operations $\mathbf{S}^{a_i a_j}$ applied to all neighbouring qubits i and j. The global operation on the cluster is the generalized product of all these transformations:

$$\mathbf{S}^{\mathcal{C}} = \prod_{a_i, a_j} \mathbf{S}^{a_i a_j}.$$

Since the operators $\mathbf{S}^{a_i a_j}$ all commute, the order of their application is not relevant[4].

A fundamental point to note is that the preparation procedure is such that the reduced density matrix of each qubit is $\frac{1}{2}\mathbf{I}$. This implies that single-qubit measurements on the cluster will give completely random outcomes.

The cluster is a universal computational resource, i.e. it is independent of the specific algorithm to be performed and of the specific input to be processed. In other words, the previously described preparation procedure can be used for any desired algorithm.

---

[3] This state can be obtained by applying the Hadamard gate to the computational basis state $|0\rangle$.

[4] More formally, the cluster state is defined as the only common eigenstate of the correlation operators

$$K^{(a)} = \sigma_x^{(a)} \bigotimes_{b \in \mathsf{nb}(a)} \sigma_z^{(b)},$$

where $\mathsf{ng}(a)$ is the set of all neighbours of a. The cluster states $\left|\psi_{\{k\}}\right\rangle_{\mathcal{C}}$ must thus obey the set of eigenvalue equations

$$K^{(a)} \left|\psi_{\{k\}}\right\rangle_{\mathcal{C}} = (-1)^{k_a} \left|\psi_{\{k\}}\right\rangle_{\mathcal{C}},$$

where $\{k\} = \{k_a \in \{0,1\} : a \in \mathcal{C}\}$ is a set of binary parameters specifying the cluster state.

The computation proceeds by single-qubit projective measurements on the cluster; the bases of the measurements can be determined by fed-forward operations on the outcomes of previous measurements. More in detail, we can set a distinction between measurements in the Pauli bases, which are immediately performed on the cluster, and measurements in different bases, determined by the results of previous measurements. This allows for an ordering of the cluster qubits by their "measurement priority". Let us define:

- the set Q_0 of the qubits which can be immediately measured on the cluster;
- the set Q_1 of the qubits whose measurements basis is determined by fed-forwarding operations on the outcomes of measurements on qubits belonging to the set Q_0;
- the set Q_n of the qubits whose measurements basis is determined by fed-forwarding operations on the outcomes of measurements on qubits belonging to the set $Q_{(n-1)}$.

It is immediate to see that $Q_0, \ldots, Q_n$ are disjoint subsets whose generalized union corresponds to the cluster. The succession of the measurements can thus be seen as a chain, beginning with measurements on qubits belonging to the Q_0.

Any time a measurement takes place, the measured qubit is disentangled from the cluster and the state of the cluster-state computer becomes a tensor of the (product) states of the measured qubits and the (entangled) sub-cluster

state of the remaining unmeasured qubits. At the end of the chain, the final state of the computation is given by the remaining qubits of the cluster. In order to obtain a string of classical bits, these qubits can be measured in the computational basis[5].

3.2.1 Clusters at work: Simulation of an arbitrary rotation

The behaviour of a cluster-state quantum computer can be illustrated by the following example (taken and simplified from Raussendorf & Briegel 2001b), which shows how a cluster-state computer is capable of simulating any arbitrary rotation $U_{\mathsf{rot}} \in SU(2)$ with a chain of five qubits.

According to Euler's rotation theorem, any rotation in $\mathbb{R}^3$ can be described by means of three real parameters:

$$U_{\mathsf{rot}}(\alpha, \beta, \gamma) = U_x(\gamma) U_z(\beta) U_x(\alpha),$$

where the rotations about $x-$ and $y-$axes are respectively given by

$$U_x(\phi) = \exp\left(-i\phi\frac{\sigma_x}{2}\right)$$

and

$$U_z(\phi) = \exp\left(-i\phi\frac{\sigma_z}{2}\right).$$

[5]The qubits of the cluster not taking part in a specific computation are removed by a measurement in the computational basis.

Here σ_x, σ_z denote the Pauli matrices.

The first qubit of the chain is prepared in a quantum state $|\psi\rangle$, while the other four qubits are initialized to the state $|+\rangle$. According to the cluster preparation procedure illustrated above, an entanglement-generating unitary transformation is applied to the five qubits; the qubit in state $|\psi\rangle$ is then moved to the last slot. Now the first four qubits are measured in the basis:

$$\mathcal{B}_j\left(\varphi_j\right) = \left\{ \frac{|0\rangle_j \pm e^{i\varphi_j}|1\rangle_j}{\sqrt{2}} \right\},$$

where $j \in \{1,2,3,4\}$ denotes the measured qubit. We refer to the outcomes of these measurements as $s_j \in \{0,1\}$, The chain is structured as follows:

1. The first qubit is measured in the basis $\mathcal{B}_1\left(0\right)$.
2. The second qubit is measured in the basis $\mathcal{B}_2\left(-\gamma\left(-1\right)^{s_1}\right)$.
3. The third qubit is measured in the basis $\mathcal{B}_3\left(-\beta\left(-1\right)^{s_2}\right)$.
4. The fourth qubit is measured in the basis $\mathcal{B}_4\left(-\alpha\left(-1\right)^{s_1+s_3}\right)$.

We are now able to define the *random byproduct operator*:

$$\mathbf{U}_\Sigma = \sigma_x^{s_2+s_4}\sigma_x^{s_1+s_3}.$$

By means of this operator it is possible to obtain the desired rotation by applying the transformation:

$$\mathbf{U}'_{\mathsf{rot}}\left(\alpha,\beta,\gamma\right) = \mathbf{U}_\Sigma\mathbf{U}_{\mathsf{rot}}\left(\alpha,\beta,\gamma\right)$$

to the only remaining qubit (the one initialized to the state $|\psi\rangle$).

3.2.2 Clusters at work: Deutsch algorithm

Despite its low practical appeal, the well-known algorithm presented by Deutsch (1985) has been the first example of a genuinely *quantum* algorithm and thus represents one of the milestones in the history of quantum computation theory.

Deutsch's algorithm provides a one-step solution to the problem of determining whether a given valid Boolean function $f : \{0,1\} \longrightarrow \{0,1\}$ is constant or balanced, i.e. whether $f(0) = f(1)$ or $f(0) \neq f(1)$.

Classically, two evaluations of the function are required in order to solve this problem (with probability one); in the quantum case, only one evaluation is needed, as shown by Deutsch's algorithm.

Deutsch's algorithm on a quantum network

Deutsch's algorithm requires to prepare the following composite state of two qubits:

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

which can be easily obtained by applying the Hadamard transform $\mathbf{H}^{\otimes 2}$ to the initial state $|01\rangle$.

The computation proceeds by the application of an unitary operator $\mathbf{U}_f$ such that

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

Note that $f(0) \oplus f(1) = 0$ if the function is constant and $f(0) \oplus f(1) = 1$ if the function is balanced. Since

$$\mathbf{U}_f|x\rangle \otimes |-\rangle = |x\rangle \otimes \frac{|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}} = (-)^{f(x)}|x\rangle \otimes |-\rangle,$$

it follows that

$$\mathbf{U}_f(|+\rangle \otimes |-\rangle) = \frac{(-)^{f(0)}|0\rangle + (-)^{f(1)}|1\rangle}{\sqrt{2}} \otimes |-\rangle,$$

which can be rewritten as

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

With a second application of $\mathbf{H}^{\otimes 2}$, we get the state

$$(-)^{f(0)}|f(0) \oplus f(1)\rangle \otimes |1\rangle.$$

A measurement on the first qubit is now sufficient to determine whether the function is constant or balanced: if f is constant (balanced), we get 0 (1) with probability one.

What do we know about the black-box transformation $\mathbf{U}_f$ that Deutsch's algorithm makes use of? We know that it must correspond to one of the following two-qubit unitary operations:

(i) $\mathbf{U}_f \equiv \mathbf{I} \otimes \mathbf{I}$. Note that:

$$[|x\rangle\,|y \oplus f\,(x)\rangle = (\mathbf{I} \otimes \mathbf{I})\,(|x\rangle|y\rangle)] \Longrightarrow f\,(x) = 0.$$

(ii) $\mathbf{U}_f \equiv \mathbf{I} \otimes \mathbf{X}$. Note that:

$$[|x\rangle\,|y \oplus f\,(x)\rangle = (\mathbf{I} \otimes \mathbf{X})\,(|x\rangle|y\rangle)] \Longrightarrow f\,(x) = 1.$$

(iii) $\mathbf{U}_f \equiv \mathbf{CNOT}$. Note that:

$$[|x\rangle\,|y \oplus f\,(x)\rangle = \mathbf{CNOT}\,(|x\rangle|y\rangle)] \Longrightarrow \begin{cases} \text{if} \quad x = 0, \quad \text{then} \quad f\,(x) = 0; \\ \text{if} \quad x = 1, \quad \text{then} \quad f\,(x) = 1. \end{cases}$$

Thus, $f\,(x) = x \oplus 0$.

(iv) $\mathbf{U}_f \equiv (\mathbf{I} \otimes \mathbf{X})\,\mathbf{CNOT}$. Note that:

$$[|x\rangle\,|y \oplus f\,(x)\rangle = (\mathbf{I} \otimes \mathbf{X})\,\mathbf{CNOT}\,(|x\rangle|y\rangle)] \Longrightarrow \begin{cases} \text{if} \quad x = 0, \quad \text{then} \quad f\,(x) = 1; \\ \text{if} \quad x = 1, \quad \text{then} \quad f\,(x) = 0. \end{cases}$$

Thus, $f\,(x) = x \oplus 1$.

In cases (i) and (ii), the function is constant; in cases (iii) and (iv), it is balanced.

Deutsch's algorithm on a cluster-state quantum computer

The cluster-state version of the Deutsch's algorithm consists in a chain of 4 qubits; thus, the cluster $\mathcal{C}$ will be in the state

$$|\Psi\rangle = \frac{|0000\rangle + |0011\rangle + |1100\rangle - |1111\rangle}{2}.$$

1. Qubit 2 is measured in the computational basis $\{|0\rangle, |1\rangle\}$ and is disentangled from the cluster.

2. The fed-forward operation ($\sigma_x^{s_2}$), with s_2 denoting the result of the measurement on qubit 2, is applied to qubit 1. This means that:

   - if $s_2 = 0$, the cluster becomes

   $$|+\rangle_1 \frac{|0\rangle_3|+\rangle_4 + |1\rangle_3|-\rangle_4}{\sqrt{2}};$$

   - if $s_2 = 1$, the cluster becomes

   $$|-\rangle_1 \frac{|0\rangle_3|+\rangle_4 - |1\rangle_3|-\rangle_4}{\sqrt{2}}.$$

3. Qubit 1 is measured in the basis

   $$\left\{ \frac{|0\rangle + e^{i\pi}|1\rangle}{\sqrt{2}}, \frac{|0\rangle - e^{i\pi}|1\rangle}{\sqrt{2}} \right\}$$

   and is disentangled from the cluster.

4. The fed-forward operation ($\sigma_x^{s_4}$), with s_4 denoting the result of the

measurement on qubit 4, is applied to qubit 3. Note that, for any result of the measurement, the state of qubit 3 becomes $|-\rangle \equiv \frac{|0\rangle-|1\rangle}{\sqrt{2}}$.

The state of the cluster becomes $|\pm\rangle_1|-\rangle_3$, which is equivalent to the state $|x\rangle\,|y \oplus f\,(x)\rangle$. If qubit 1 is in the state $|+\rangle$ (the cluster is in the state $|+\rangle_1|-\rangle_3$), the function is constant; if qubit 1 is in the state $|-\rangle$ (the cluster is in the state $|-\rangle_1|-\rangle_3$), the function is balanced.

3.2.3 Clusters at work: Search algorithm

A further illustration of a working cluster-state computer will regard the computational problem known as search.

By means of an algorithm due to Lev Grover and known as Grover's algorithm, a quantum computer would be able to find a given element within an unstructured database of N elements in $\mathcal{O}\left(\sqrt{N}\right)$ computational steps, i.e. with a quadratic speed-up over its classical counterpart[6].

A practical example of a search problem consists in finding someone's number within a phone book. Classically, the most efficient approach to such a problem is by means of brute force. Select an item in the middle of the list: if this name follows (precedes) the desired one in the alphabetic order, then discard all the names below (above); repeat the procedure until the result is obtained. Since at each time half of the items is discarded, for a list of N elements the average number of steps required for finding a specific name will be $N/2$.

[6]The first version of the algorithm appeared in Grover (1996); its optimality was proven by Zalka (1999).

More formally, the search problem can be formalized as finding a unique element s_v belonging to an unsorted set $S = \{s_1, \ldots, s_N\}$ of cardinality $N = 2^n$. This element being unique means that there exists an "oracle" function f ranging on S such that $f(s_v) = 1$ and $f(s_{i \neq v}) = 0$ (this function is able to recognize the desired result). It has been shown that any classical search algorithm, either deterministic or probabilistic, will require $\mathcal{O}(2^{n-1})$ queries to the oracle in order to obtain the desired solution.

Grover's algorithm

The goal of the search problem is finding a specific element x_0 belonging to an unstructured set $X = \{1, \ldots, N_G\} \in \{0,1\}^{\otimes n}$ whose cardinality is $N_G = 2^n$.

For $x \in X$, let $f(x) : \{0,1\}^{\otimes n} \rightarrow \{0,1\}$ be an oracle function such that $f(x = x_0) = 1$ and $f(x \neq x_0) = 0$. Then:

(1) Initialize the N_G-sized system to the uniform amplitude, unbiased state $|\psi_{in}\rangle$ defined as

$$|\psi_{in}\rangle = \frac{1}{\sqrt{N_G}} \sum_x |x\rangle \equiv \frac{1, \ldots, 1}{N_G}.$$

(2) **Iterator.** Repeat $O\left(\sqrt{N_G}\right)$ times the following operations:

(2a) **Tagging.** Let $|x\rangle$ be any computational basis state. Then:

- if $f(x) = 1$, rotate the state by π;
- if $f(x) = 0$, leave the state unaltered.

The tagging operation corresponds to flipping by π the phase pf the desired state $|x_0\rangle$.

(2b) **Inversion-about-the-mean.** Apply the diffusion transform **D** such that:

- $\mathbf{D_{ij}} = 2/N_G$ if $i \neq j$;
- $\mathbf{D_{ij}} = -1 + 2/N_G$ if $i = j$.

The inversion-about-the-mean operation corresponds to an averaging of the state amplitudes.

(3) Sample the state by a projective measurement.

Since both (2a) and (2b) can be written as unitary operators, respectively $\mathbf{U}_f$ and $\mathbf{U}_{inv}$, the iterator can be written as the matrix product $\mathbf{U}_G = \mathbf{U}_{inv}\mathbf{U}_f$.

Grover's algorithm begins from a uniform input state of size $N_G = 2^n$ and performs a sequence of repeated tagging and inversion-about-the-mean operations $\mathbf{U}_G^k$ until the desired number of steps $k \approx \mathcal{O}\left(\sqrt{N_G}\right)$ has been performed. The state is then measured and the desired result will have an overwhelming likelihood of being measured.

Cluster-state Search Algorithm

The search algorithm for cluster-state quantum computers, (Matthew Smith & al, 2012) works as follows.

1 Initialize a set of $N = n^2$ qubits into a $n \times n$ cluster-state such that

each qubit is in the state $|+\rangle$ and 2^n is the maximum desired number of elements to be searched.

2 An oracle selects a sets of measurements of size $n\,(n-1)$ to be made in either $s = 0$ or $s = \pi$ basis, such that

$$|\pm_s\rangle = \frac{|0\rangle \pm e^{is}|1\rangle}{\sqrt{2}}.$$

(2a) $n-1$ sets of measurements are applied sequentially, in which n qubits are measured simultaneously, moving across the cluster-state in a uniform direction and such that the $(k+1)-$th measurement set depends on the outcomes of the $k-$th measurement set, for all $k = 1, \ldots, n-1$ iterations.

(3) The final state is sampled by rotating the remaining n qubits into the $|\pm\rangle$ basis, followed by a projective measurement.

The most apparent difference between Grover's and cluster-state search algorithms lies in the iterator step:

- in Grover's case, the iterator is a product of two unitary operators which repeatedly implement the same two operations;
- in cluster-state case, the iterator, consisting of the n measurement bases chosen by the oracle, need not (and in general will not) be a constant between iterations.

Oracle tagging

General non-constant tagging operations

In Grover's algorithm, the tagging operation is 1-to-1, i.e. there is a unique tag for each and every element in the search space.
The cluster-state search algorithm is unique only up to overall phases (e.g. $\pm$ signs and factors of i) and the final output states can be degenerate.

While in the special case of $n = 2$ element search there are 4 measurements and 4 unique output states (i.e. tagging operations are $1-$to-1), in the $n = 3$ case the final possible outcomes of the last column are $8-$fold degenerate, i.e. there are 8 tagging operations all pointing to the same final state (up to overall phases). Thus the $1-$to-1 association of the tagging to a unique output does not scale with the size of the cluster-state.

If one ignores (or corrects) the global phase, not all the $n\,(n-1)$ measurements turn out to be strictly necessary. In the $n = 3$ case, the search requires $3\,(3-1) = 6$ measurements chosen by the oracle, with the remaining 3 output qubits rotated to the $|\pm\rangle$ basis via an $\mathbf{H}^{\otimes 3}$ operation. However, it is possible to perform the full $8-$element search (i.e. providing every possible output) by varying only $n = 3$ measurements, with the sequence $(0, 0, s_3; 0, s_5, s_6)$ where 0 denotes a measurement onto $|\pm\rangle$ and s_i a measurement that can be varied ($s = 0$, $s = \pi$) by the oracle according to the desired output. The pattern of measurements $(0, 0, s_3; 0, s_5, s_6)$ has the form

$$\mathbf{U} = \mathbf{H}_1 \otimes \mathbf{H}_2 \otimes \mathbf{B}_3\,(\gamma) \otimes \mathbf{H}_4 \otimes \mathbf{B}_5\,(b) \otimes \mathbf{B}_6\,(c) \otimes \mathbf{H}_7 \otimes \mathbf{H}_8 \otimes \mathbf{H}_9.$$

What precisely is the tagging operation performing on the state? According to the definition above, $\mathbf{U}$ is the Kronecker product of matrices chosen from the set of 0 and π rotations:

$$\mathbf{B}(0) = \mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad \mathbf{B}(\pi) = \mathbf{F} = \frac{i}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}.$$

Single-constant iterator for any outcome

The above description of cluster-state search algorithm has un undesirable feature: if we consider the measurement of the vertical columns to be a "step" in the iteration, then it turn out that the algorithm does not have a fixed, constant iterator. The iterator will vary between steps and is neither sequential nor random, so that the various iterators must be carefully chosen by the oracle at each step.

It is possible to modify the algorithm such that the iterator is constant not only between steps, but also constant for any desired output. Let us consider, for the $n = 3$ case, the matrices producing the outputs $(1,0,0,0,0,0,0,0)^T$ and $(0,0,0,1,0,0,0,0)^T$ (up to a global phase):

$$\mathbf{U}_1 = \mathbf{H}_1 \otimes \mathbf{H}_2 \otimes \mathbf{H}_3 \otimes \mathbf{H}_4 \otimes \mathbf{H}_5 \otimes \mathbf{H}_6 \otimes \mathbf{H}_7 \otimes \mathbf{H}_8 \otimes \mathbf{H}_9$$

and

$$\mathbf{U}_1 = \mathbf{H}_1 \otimes \mathbf{H}_2 \otimes \mathbf{F}_3 \otimes \mathbf{H}_4 \otimes \mathbf{F}_5 \otimes \mathbf{F}_6 \otimes \mathbf{H}_7 \otimes \mathbf{H}_8 \otimes \mathbf{H}_9.$$

Notice that in the first case, the iterator is always a constant.

To determine the effect pf the tagging operation, we compare two different $\mathbf{U}$ matrices:

$$\Delta\mathbf{U} = \mathbf{U}_1 - \mathbf{U}_4 = \mathbf{H}^{\otimes 9} - \mathbf{H}_1 \otimes \mathbf{H}_2 \otimes \mathbf{F}_3 \otimes \mathbf{H}_4 \otimes \mathbf{F}_5 \otimes \mathbf{F}_6 \otimes \mathbf{H}_7 \otimes \mathbf{H}_8 \otimes \mathbf{H}_9.$$

Many of the columns are identically 0, while the remaining ones is $\Delta\mathbf{U}$ have an absolute value twice that in any given element of $\mathbf{U}_i$. Thus, the two differently tagged matrices are identical except for the sign of the elements in several vertical columns. This implies that

$$\mathbf{U}_1 - \mathbf{U}_4\mathbf{T}_4 = \mathbf{H}^{\otimes 9} - (\mathbf{H}_1 \otimes \mathbf{H}_2 \otimes \mathbf{F}_3 \otimes \mathbf{H}_4 \otimes \mathbf{F}_5 \otimes \mathbf{F}_6 \otimes \mathbf{H}_7 \otimes \mathbf{H}_8 \otimes \mathbf{H}_9)\,\mathbf{T}_8 = 0.$$

Since $\mathbf{F}\mathbf{F}^\dagger = 1$ and $\mathbf{F}^\dagger\mathbf{H} = i\,(-\mathbf{Z})$, it follows that

$$\mathbf{T}_4 = \mathbf{I}_1 \otimes \mathbf{I}_2 \otimes (-\mathbf{Z}_3) \otimes \mathbf{I}_4 \otimes (-\mathbf{Z}_5) \otimes (-\mathbf{Z}_6) \otimes \mathbf{I}_7 \otimes \mathbf{I}_8 \otimes \mathbf{I}_9.$$

Since i is an overall phase, it can be factored out at any desired point. Thus $\mathbf{T}_4$ becomes a real diagonal tagging matrix with elements ± 1. The $+1$s correspond to the columns already identical (thus subtract to 0), while the -1s correspond to the remaining columns. $\mathbf{U}_1$ is identical to the product $\mathbf{U}_4\mathbf{T}_4$. Since $\mathbf{T}_i\mathbf{T}_j = \mathbf{I}$, we can write $\mathbf{U}_1\mathbf{T}_4 = \mathbf{U}_4$. However, the matrices $\mathbf{T}_i$ do not commute with $\mathbf{U}_j$: $[\mathbf{T}_i, \mathbf{U}_j] \neq 0$. A matrix $\mathbf{T}_i$ can be found for any matrix $\mathbf{U}_1$ (or equivalently, for any measurement pattern). In the $\mathbf{U}_1$ transformation, we have that $\mathbf{T}_1 = \mathbf{I}$.

The quantum search equation can now be rewritten as

$$|\Psi_{out,i}\rangle = \mathbf{H}^{\otimes n^2}\mathbf{T}_i|\Psi_{in}\rangle,$$

where $\mathbf{T}_i$ is the tagging operation chosen by the oracle for the desired output $i \in \{1, \ldots, 2^n\}$. Note that $\mathbf{T}_j$ can be applied either to the rhs od to the lhs of the equation above, i.e. we can apply the tag:

1. to the measurement operation, thus changing the iterators between iterations;

2. to the initial cluster-state, prior to the measurement process.

The second choice leaves the measurement pattern (the search iterator) constant for any chosen tag, regardless of the size N of the system or the desired input of the algorithm. The constant iterator is simply $\mathbf{H}^{\otimes n}$ acting on each column of qubits in turn.

Given an input state for the box cluster-state

$$|\Psi_{in}\rangle = \prod_{i,j=nn} \mathbf{CZ}_{i,j}|\psi\rangle = \mathbf{CZ}_{1,2}\mathbf{CZ}_{1,3}\mathbf{CZ}_{2,4}\mathbf{CZ}_{3,4}\prod_{i=1}^{N}|+_i\rangle.$$

The application of the tagging matrix $\mathbf{T}_i$ gives

$$\mathbf{T}_i|\Psi_{in}\rangle = \mathbf{T}_i\mathbf{CZ}_{1,2}\mathbf{CZ}_{1,3}\mathbf{CZ}_{2,4}\mathbf{CZ}_{3,4}\prod_{i=1}^{N}|+_i\rangle;$$

since $\mathbf{T}_i$ and $\mathbf{CZ}_{i,j}$ are diagonal and hence commute, we have

$$\mathbf{T}_i|\Psi_{in}\rangle = \mathbf{CZ}_{1,2}\mathbf{CZ}_{1,3}\mathbf{CZ}_{2,4}\mathbf{CZ}_{3,4}\mathbf{T}_i \prod_{i=1}^{N} |+_i\rangle.$$

This means that the "shape" of the cluster-state, contained within the list of **CZ** gates, remains the same. $\mathbf{T}_i$ only changes the state of the constituent qubits, independently of the pairwise connectivity of the cluster-state.
The effect of $\mathbf{T}_i$ is to change several of the signs of the initial cluster state from $|+\rangle$ to $|-\rangle$. There is a $1-$to-1 correlation between those qubits, which in the standard implementation would be measured in the $\mathbf{B}(\pi)$ basis, and the qubits rotated from $|+\rangle$ to $|-\rangle$.

In the $N = 9$ qubits case, the measurement sequence $(0, 0, \pi; 0, \pi, \pi)$ corresponding to the output state $(0, 0, 0, 1, 0, 0, 0, 0)^T$ requires a search algorithm with initial state:

$$\mathbf{T}_4|\psi_{in}\rangle = \mathbf{T}_4 \prod_{i=1}^{N} |+_i\rangle = |+_1\rangle\otimes|+_2\rangle\otimes|-_3\rangle\otimes|+_4\rangle\otimes|-_5\rangle\otimes|-_6\rangle\otimes|+_7\rangle\otimes|+_8\rangle\otimes|+_9\rangle.$$

Tagging the output state consists in

$$|\Psi_{out,4}\rangle = \mathbf{U}_4|\Psi_{in}\rangle = (\mathbf{H}_1 \otimes \mathbf{H}_2 \otimes \mathbf{H}_3)\otimes(\mathbf{H}_4 \otimes \mathbf{H}_5 \otimes \mathbf{H}_6)\otimes(\mathbf{H}_7 \otimes \mathbf{H}_8 \otimes \mathbf{H}_9) \prod_{i,j=nn} \mathbf{CZ}_{i,j} \prod_{i=1}^{N} |+_i\rangle$$

which becomes

$$
\begin{aligned}
&= (\mathbf{H}_1 \otimes \mathbf{H}_2 \otimes \mathbf{H}_3) \otimes (\mathbf{H}_4 \otimes \mathbf{H}_5 \otimes \mathbf{H}_6) \otimes (\mathbf{H}_7 \otimes \mathbf{H}_8 \otimes \mathbf{H}_9)\, \mathbf{T}_4 \prod_{i,j=nn} \mathbf{CZ}_{i,j} \prod_{i=1}^{N} |+_i\rangle = \\
&= (\mathbf{H}^{\otimes n})^{\otimes n} \prod_{i,j=nn} \mathbf{CZ}_{i,j} \mathbf{T}_4 \prod_{i=1}^{N} |+_i\rangle = \\
&= (\mathbf{H}^{\otimes n})_k \prod_{i,j=nn} \mathbf{CZ}_{i,j} |+_1\rangle \otimes |+_2\rangle \otimes |-_3\rangle \otimes |+_4\rangle \otimes |-_5\rangle \otimes |-_6\rangle \otimes |+_7\rangle \otimes |+_8\rangle \otimes |+_9\rangle.
\end{aligned}
$$

We have defined the cluster-state search in terms of a single tagging operation $\mathbf{T}_i$ acting on a constant input state $\prod_{i=1}^{N} |+_i\rangle$ followed by entangling operations $\mathbf{CZ}_{i,j}$ and a constant iterator defined as $\mathbf{H}^{\otimes n}$ acting on each of the columns in turn (from left to right).

In fully general terms, the equation above becomes

$$
|\Psi_{out,m}\rangle = \prod_{k=1}^{n} \left(\mathbf{H}^{\otimes n}\right)_k \prod_{i,j=nn} \mathbf{CZ}_{i,j} \mathbf{T}_m \prod_{l=1}^{N} |+_l\rangle,
$$

where m is the desired output state (in decimal representation) found by the tagging operation $\mathbf{T}_m$; $N = n^2$ is the total number of qubits; n is the number of iterations (as well as the size of one side of the cluster state) and nn is the set of nearest qubits in the cluster-state.

The comparison between Grover's algorithm and the cluster-state search algorithm shows that:

- Grover's oracle must repeatedly apply a unique tag to the state at each step such that the inversion-about-the-mean operation can be applied correctly (otherwise, the algorithm would fail for converge to a unique answer);

- the cluster-state search algorithm can be performed with only one initial tagging operation to converge to a unique output state.

Amplitude amplification?

In Grover's algorithm, the i-a-t-m operator $\mathbf{U}_{inv}$ has entries $\mathbf{D}_{ij} = 2/N_G$ for $i \neq j$ and $\mathbf{D}_{ij} = -1 + (2/N_G)$ for $i = j$. Thus can be rewritten as $\mathbf{U}_{inv} = 2P' - I$, where $P' = |\psi_{in}\rangle\langle\psi_{in}|$ is the projection matrix on the unbiased state $|\psi_{in}\rangle = \left(1/\sqrt{N_G}\right)\sum_x |x\rangle$ with $P'_{i,j} = 1/N_G$ for all i, j.
The amplitude amplification takes advantage of the sign difference between targeted solution states after the tagging operation, and the re-adjustment of the amplitudes affected by the inversion-about-the-mean operator. Repeated applications of tagging operator $\mathbb{U}_f$ are necessary because of hermiticity of $\mathbf{U}_{inv}$.

In the cluster-state search algorithm, the state is not repeatedly tagged and there is no amplitude amplification.
In spite of the cluster-state iterator not being invertible (since it (implicitly) contains projective measurements), the search algorithm can be treated as effectively implementing a unitary iterator, since projections performed on columns of qubits in the $n \times n$ grid leave the measured qubits in a well defined state and the remaining ones in a pure state.

Following Brassard et al (2000) work, the general amplitude amplification iterator can be defined as

$$\mathbf{Q} = \mathbf{A}\mathbf{U_0}\mathbf{A}^{-1}\mathbf{U}_f,$$

where $\mathbf{U}_f$ is the tagging operator; $\mathbf{U}_0$ flips the sign of the computational basis state $|0\rangle^{\otimes n}$ while leaving all other computational basis states unchanged; $\mathbf{A}$ is the generalized unitary oracle constrained only by the requirement that $\mathbf{A}|0\rangle^{\otimes n} = |\psi_{in}\rangle$.

Since the cluster-state algorithm inherently requires projective measurements to proceed, it cannot implement amplitude amplification because it has no analogue to the inverse $\mathbf{A}^{-1}$. Instead of amplitude amplification, the cluster-state search algorithm seems to be performing a binary search operation. The computational time required for implementation supports this (empirical) statement: exactly $(n-1)$ iterations are required in order to find a solution with unit probability. Each iteration involves a simultaneous measurement of n qubits, for a total of $n\,(n-1)$ measurements. If we include the final read-out measurement, the algorithm has applied n measurements iterations to effect a search of 2^n elements. Note that any $k \leq n-1$ number of iterations will not yield a valid solution, since the system remains in a state unitarily equivalent to the initial cluster-state.

This suggests that the cluster-state search is not amplitude amplification, rather a $1-$to-2^n mapping, and that the oracle is simply selecting a given path. As each (single-qubit) measurement has only two outcomes, the algorithm can be thought as a binary tree structure, through which the oracle directs the search based on the applications of the tagging operation. Each individual measurement in either $0-$ or $\pi-$rotated basis selects a tree to move down. It happens that all s^n outcomes can be found with a minimum of exactly n variable measurements. This is suggestive of the binary

structure, since the total number of possible outcomes is just s^n.

Chapter 4

Circuits vs Clusters: The Role of Entanglement

The circuit and cluster-state frameworks are computationally equivalent. The efficient simulation of a cluster-state computation on a circuit is not surprising, since both the operations involved in the preparation of the cluster (Hadamard and controlled-phase gates) and the single-qubit measurements present in the computation are within the capabilities of quantum circuits. The efficient simulation of a circuit on a cluster-state computer is less immediate to be shown; a complete proof of equivalence is provided by Raussendorf & Briegel (2001b).

4.1 Circuits vs Clusters

In spite of their computational equivalence, quantum circuits and cluster-state computers present very different pictures of computational processes. Here are listed some of the peculiar features of cluster-state quantum computation.

4.1.1 Input and output

In quantum circuits, the input and output of the computation are respectively encoded in the initial and final state of a selected register. For instance, in the case represented in Figure 1, the output is read-off by a final measurement on the first qubit; it is possible to follow the evolution of this qubit during the computation, from the initial state to the final (even if it is forbidden to observe it until the final step).

Noticeably, in cluster-state computation the qubits corresponding to the output register of the circuit can be among the first to be measured, thus it is not appropriate to term them "output qubits". Let it be recalled that the cluster is independent of any specific algorithm to be performed and input value to be processed; moreover, the role of the qubits is just to be measured, without undergoing a unitary evolution like in the circuit framework. This means that it is not possible to describe the computation as a process in which an initial state encoding the input is transformed into a final state encoding the output. In this sense, Raussendorf & Briegel (2001b) claim that the cluster-state quantum computer has no quantum input and output.

4.1.2 Evolution

Since all the operations occurring in a quantum circuit are unitary (except for the final measurement), the evolution is, by definition, reversible. Even if a (non-unitary) measurement is designed to be performed at the middle of the computation, it can always be postponed to the end by virtue of the Deferred Measurement Principle.

On the other hand, cluster-state computations are not reversible[1]. This feature follows from the fact that measurements are not unitary transformations. An appeal to the Deferred Measurement Principle would be unhelpful, because computational steps are just measurements, and thus the Deferred Measurement Principle does not make sense in cluster-state computation. The non-reversibility of the dynamics is often stressed as the most radical departure of cluster-state computation from quantum circuits. This point will be delved into in the following sections.

One further noteworthy difference between the two computational frameworks concerns the ordering of the computational steps. In particular, when a simulation of a quantum circuit on a cluster-state computer is performed, the ordering of the gates in the circuit does not have a counterpart in the ordering of the measurements in the cluster-state computer; the converse (the case of a simulation of a cluster-state computation on a circuit) is also true. In the cluster-state framework, rather than the ordering of the measurements, what turns out to be fundamental is their priority classification.

[1] This is the reason why this computational framework is also known as *one-way* quantum computation.

4.1.3 Ordering

When a simulation of a quantum circuit on a cluster-state computer is performed, the ordering of the gates in the circuit does not have a counterpart in the ordering of the measurements in the cluster-state computer; the converse (the simulation of a cluster-state computation on a circuit) is also true. In cluster-state quantum computation, rather than the ordering of the measurements, what turns out to be relevant is their priority classification.

4.1.4 Parallelism

From the presentation of cluster-state quantum computation sketched so far, it should be clear that within this framework there is nothing resembling the "parallel interfering computational paths" picture, which on the contrary is a hugely popular account of what happens when a circuital quantum computation is running. This notion of parallelism is meant to originate from the superpositions of states, i..e parallelism denotes the possibility of having "several states at the same time". The only form of parallelism present in cluster-state computation is that qubits belonging to the same set Q_i can be measured simultaneously; however, this cannot be seen as a form of quantum parallelism, because no interference takes place during the parallel evolutions of these qubits.

4.1.5 Computational resources

For the sake of completeness, we briefly take into account the definition of the computational resources. The spatial resources of a cluster-state computer are defined as the number of qubits belonging to the cluster; there are here no fundamental differences with the circuit framework, where the spatial resources correspond to the number of qubits entering the circuit.

The temporal resources of a cluster-state computation consist in the number of disjoint subsets representing those qubits whose measurements can be performed in parallel. This definition too is analogous to its circuital counterpart, which is the minimum number of computational steps up to the possible parallelization of elementary operations (quantum gates and final measurements)[2].

Summarizing, it has been noted that the main differences between cluster-state and circuit quantum computation do not lie in the kinds of elementary operations allowed, neither in the computational resources required for the computation. Rather, the comparison of the two frameworks outlined above suggests that what they essentially differ in are the ways they exploit the fundamental features of quantum systems.

[2]Raussendorf & Briegel (2001b) also define the operational resources, which correspond to the total number of elementary operations (single-qubit measurements in cluster-state computation; gates and measurements in circuits.)

4.2 The Role of Entanglement

During the long and articulated debate on the explanation of the quantum speed-up, many different answers have been proposed and discussed (e.g. superposition, parallelism, entanglement, interference, quantum logic). Since a global overview of the issue would go very far beyond the scope of this work, we shall focus on the proposal of entanglement as the explanation of the quantum speed-up[3].

The discussion will here be restricted to *exponential* computational speed-up. There exist many quantum algorithms of undisputed theoretical and practical relevance, like Grover's, which achieve just a sub-exponential speed-up over their classical counterparts. Anyway, the most outstanding consequence of quantum computation is just the exponential speed-up exhibited by some algorithms; since the speed-up question is centred on the power of quantum computers, it seems reasonable, for the purposes of this work, to take into investigation just the exponential case[4].

4.2.1 Quantum circuits: entangling evolution

The analysis of the role of entanglement in quantum computation requires, at first, to investigate whether entanglement can be a sufficient or a necessary

[3]See Jaeger (2009) for a general survey on the explanation of the quantum speed-up.

[4]For analogous reasons, the discussion will be restricted to computation with pure states: the explanation of the quantum speed-up and the comparison between cluster-state and circuit quantum computation simply do not require the mixed states case to be primarily taken into account. A discussion of the role of entanglement in the explanation of the quantum speed-up, where computation with mixed states and sub-exponential speed-up are considered, can be found in Cuffaro (2013).

condition for the quantum speed-up. The first possibility is ruled out by the Gottesman-Knill theorem, which states that an efficient classical simulation is available for any quantum computation which only makes use of gates belonging to the Clifford group (Pauli, Hadamard, **CNOT** gates)[5]. This means that there exists a set of protocols (which comprehends teleportation and dense coding) that both a quantum and a classical computer are able to efficiently perform. If entanglement were a sufficient condition for the quantum speed-up, then such an efficient classical simulation should not be available, because classical computers cannot prepare entangled states and operate on them. As a consequence, entanglement cannot make a sufficient condition for the quantum speed-up.

The claim that entanglement is a necessary condition for the quantum speed-up has many advocates. In this respect, the main result is a theorem proven by Jozsa & Linden (2003, p. 2021):

> ... Multi-partite entanglement with unboundedly many qubits entangled together, is a necessary feature of any quantum algorithm (operating on pure states) if the algorithm is to exhibit an exponential speed-up over classical computation.

A similar argument is claimed by Steane (2003, p. 476), who claims that:

> A quantum computer can be more efficient than a classical one at generating some specific computational results because quantum entanglement offers a way to generate and manipulate a physical

[5]Cf. Nielsen & Chuang (2000).

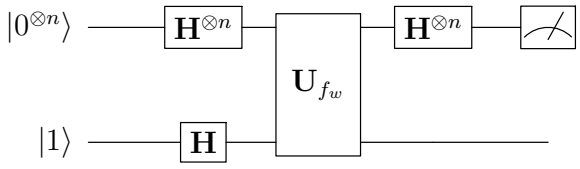

Figure 4.1: Circuit for the Bernstein-Vazirani algorithm.

> representation of the correlations between logical entities without the need to completely represent the logical entities themselves.

A way to object to the thesis of the necessity of entanglement for the quantum speed-up, consists in showing that the speed-up can be achieved without the use of entanglement. An argument along these lines is due to Fortnow (2003), who claims that, instead of entanglement, interference should be regarded as the source of the quantum computational speed-up.

We shall here focus on Fortnow's *pars destruens* contra entanglement, rather than on his *pars construens* pro interference. Fortnow's position appeals to an argument of Meyer (2001), who claims that it is possible to show a counterexample to the thesis of the necessity of entanglement for the quantum speed-up: the quantum algorithm for the Bernstein-Vazirani problem (whose circuit is represented in figure 3.1) can be solved polynomially «without creating any entanglement at any timestep»[6]. This drives Fortnow to discard entanglement as the source of the quantum computational speed-up.

In order to discuss this issue, attention should be paid to what is meant by the necessity of entanglement. Cuffaro (2013) suggests that, if entanglement has to be considered as the explanation of the quantum speed-up, it is too

[6]Meyer (2001), p. 6.

strong a requirement that a quantum computer should be able to achieve an exponential speed-up only when it enters an entangled state. More in detail, two different claims need to be kept separate:

(1) «Any state that displays computational speed-up must be entangled»[7].

(2) «Entanglement must play a role in any explanation of quantum speed-up»[8].

While it is easy to see that the first claim is too strong (this indeed is also the argument of Fortnow), the second looks rather vague. However, it can be made precise by appealing to a result of Abbott (2010), which states that, for any $n \geq 3$, there exists a valid (i.e. either constant or balanced) function $f : \{0,1\}^n \longrightarrow \{0,1\}$ such that an unitary transformation $\mathbf{U}_f$ parametrized by f is an entangling transformation, in the sense that the first n qubits of $\mathbf{U}_f|+\rangle|-\rangle$ are all entangled. Henceforth, the term *entangling evolution* will be used to denote the evolution of a quantum circuit which makes use of a transformation of the kind just described.

It is thus to possible to say that, if a quantum algorithm requires an entangling evolution to take place during the computation, then entanglement does play a role in the explanation of the quantum speed-up, even if entangled states are not displayed by the quantum computer.

According to the previous definition, the operation $\mathbf{U}_{f_w}$ taking part in the Bernstein-Vazirani algorithm is entangling. Hence, it would be hard to

[7] Cuffaro (2011), p. 20.
[8] Cuffaro (2011), p. 20.

deny that entanglement is involved in the explanation of the efficiency of the algorithm: without the entangling operation $\mathbf{U}_{f_w}$, the algorithm would indeed not achieve any speed-up at all[9].

We shall now turn our attention to the role of entanglement in the cluster-state framework.

4.2.2 Cluster-state computation: disentangling evolution?

Entanglement seems to play a fundamental role also in cluster-state quantum computation, since the state of the cluster is, by definition, entangled. Moreover, the cluster is a universal resource, thus entanglement turns out to be an essential ingredient for any computation in this framework (also those computations not achieving a speed-up).

However, the way entanglement is exploited by cluster-state computers is quite different from the circuit case. The main difference is that, as seen above, any computational step of a cluster-state computer consists in a measurement, i.e. a disentanglement of a qubit from the cluster. Thus, at first glance the cluster-state computer seems to be governed by a disentangling evolution; this would be just the opposite of the circuit case.

More in detail, let us consider what happens when the first set of qubits of the cluster is measured in the Pauli bases. The state of each qubit is

[9]It is also interesting to note that Fortnow seems to embrace claim (2), when stating that «entanglement does not *play an important role* in the power of efficient quantum computation» (Fortnow 2003, p. 606); however, Meyer's argument only works if the necessity of entanglement is assumed in the sense of (1).

projected onto an eigenstate of one Pauli basis; consequently, the global state of those qubits is a product state and (by definition) they are no more entangled. After this round of measurements, the global state of the cluster-state computer has thus become the tensor of the (product) state of the measured qubits and of the (entangled) state of the unmeasured ones, which form a subset of the original cluster.

When, at the end of the computation, the qubits have all been measured (recall that the qubits not involved in the computation are assumed to be measured in the computational basis), the initial entanglement of the cluster has completely disappeared. Thus, such an evolution could be labelled as disentangling, in the sense that it starts with a multi-partite entangled state which, step by step, is turned into a product state.

The disentangling evolution picture suggests two consequences:

- A counterargument to the claim that any state displaying computational speed-up must be entangled. If that claim were true, there would be no need of breaking down the initial entanglement of the cluster in order to speed up the computation.

- A reinforcement of the claim that entanglement plays a role in the explanation of the quantum speed-up, even if it is differently exploited by cluster-state computers with respect to quantum circuits.

The most fundamental difference between the two computational frameworks discussed here is the dynamical evolution, which is unitary (thus reversible) in quantum circuits and non-unitary (thus irreversible) in cluster-

state computers. This is due to the presence of single-qubit measurements, which are not unitary operations, in cluster-state computation.

A measurement is usually described as an interaction between an observer system S and an observed system O, such that the evolution of S is governed by the standard unitary quantum dynamics until the interaction with O causes its state to "collapse" into an eigenstate of the measured observable. Nothing prevents the joint system $S + O$ to be considered as the observed system, with respect to a "larger" observer O'. The patent unsatisfactoriness of this account has given rise to what is commonly known as the quantum measurement problem, which is the most hugely debated issue within the foundations of quantum mechanics[10].

Turning the attention back to computation, it is possible, by virtue of the Deferred Measurement Principle, to consider a quantum circuit as an observed system governed by unitary evolution, which gets observed at the end of the algorithm. On the other hand, in a cluster-state computer there are more than one observed system and observer: for any measurement round, the observed system consists of a subset of the cluster (which the qubits to be measured belong to), associated to a measuring system which records the outcomes, thus allowing for fed-forwarding of information.

However, just as seen above, nothing prevents us from treating the entire cluster-state computer as the observed system, which consequently has to undergo a unitary dynamics.[11] Could it turn out that, when considered in

[10]See Wallace (2008) for a survey of the quantum measurement problem.

[11]Let it be remarked that this strategy, while looking gratuitously twisted from a practical point of view, does not represent a commitment to any no-collapse interpretation of

such a perspective, cluster-state computers reveal unexpected properties? This approach is sometimes referred to as the "Church of the Larger Hilbert Space" (henceforth CLHS). See Sec. 5 of Brassard and Raymond-Robichaud (2012) for a brief discussion of the tenets of the CLHS.[12]

4.2.3 Fully-unitary cluster-state computation

In order to describe measurements in a fully-unitary dynamics, we will firstly take into account the most simple case, a single-qubit measurement.

Let us consider a measurement in the computational basis of a qubit prepared in the state:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{4.1}$$

The measurement requires the presence of a "recorder" (an observer), whose initial state will be written as $\left|R^{\{0,1\}}\right\rangle$, where the apex symbols denote the selected measurement basis. The eigenstates of the recorder correspond to the observed eigenstates of the qubit.

The initial state of the joint system formed by the qubit and the recorder

quantum mechanics. No claim is made here about which type of dynamical evolution the universe is subject to, or the ultimate nature of physical systems, or what really happens when an observation on a quantum system takes place. Of course, if a no-collapse interpretation of quantum mechanics is assumed, then the evolution of a cluster-state computer must ultimately be fully-unitary; but a fully-unitary account of cluster-state computation, by itself, can be legitimately provided simply by setting a peculiar observed-observer boundary, without engaging in foundational issues. This does not mean that the foundations of quantum mechanics are not relevant for computation, but simply that arguments against no-collapse interpretations of quantum mechanics would be inadequate here.

[12] No commitment is here made to the main arguments sustained by Brassard and Raymond-Robichaud in that paper.

will thus be:

$$(\alpha\,|0\rangle + \beta\,|1\rangle) \otimes \left|R^{\{0,1\}}\right\rangle$$

According to the standard non-unitary account of measurement, the joint system simply ends up in one of the two following states (where the notation $|R_i\rangle$ denotes that the value i has been recorded):

$$\alpha|0\rangle \otimes |R_0\rangle,$$

with probability $|\alpha|^2$, or

$$\beta|1\rangle \otimes |R_1\rangle,$$

with probability $|\beta|^2$.

On the other hand, a fully-unitary account leaves us with the final state:

$$\alpha|0\rangle \otimes |R_0\rangle + \beta|1\rangle \otimes |R_1\rangle.$$

Now the final state is an entangled superposition, while the initial was a product state. In the following, the tensor product symbol will be omitted when clear from the context.

In order to deal with multi-qubit systems, we introduce the notation $|\psi\rangle_{q_i}$ and $|R\rangle_{r_j}$, where q_i and r_j denote the $i-$th qubit and the $j-$th recorder respectively. If we are going to measure a product state of two qubits $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ and $|\phi\rangle = \gamma\,|+\rangle + \delta\,|-\rangle$, we start with the state:

$$\left(\left(\alpha\,|0\rangle_{q_1} + \beta\,|1\rangle_{q_1}\right) \otimes \left(\gamma\,|+\rangle_{q_2} + \delta\,|-\rangle_{q_2}\right)\right) \otimes \left(\left|R^{\{0,1\}}\right\rangle_{r_1} \otimes \left|R^{\{+,-\}}\right\rangle_{r_2}\right).$$

After the observation of both qubits, the post-measurement state will then be:

$$\left(\alpha|0\rangle_{q_1} \otimes |R_0\rangle_{r_1} + \beta|1\rangle_{q_1} \otimes |R_1\rangle_{r_1}\right) \otimes \left(\gamma|+\rangle_{q_2} \otimes |R_+\rangle_{r_2} + \alpha|-\rangle_{q_2} \otimes |R_-\rangle_{r_2}\right).$$

How can this picture work for cluster-state quantum computation? The transformation on a bi-partite entangled state can be taken as a model of what happens when a single-qubit measurement is performed within a cluster-state quantum computer. Let us take into account the case of a two-qubits entangled state. The initial state is the tensor of the entangled state of the qubits and the product state of the recorders:

$$\frac{|0\rangle_{q_1} \otimes |+\rangle_{q_2} + |1\rangle_{q_1} \otimes |-\rangle_{q_2}}{\sqrt{2}} \otimes \left(\left|R^{\{0,1\}}\right\rangle_{r_1} \otimes \left|R^{\{+,-\}}\right\rangle_{r_2}\right).$$

The observation of one of the qubits yields the state:

$$\frac{\left(|0\rangle_{q_1} \otimes |R_0\rangle_{r_1} \otimes |+\rangle_{q_2} \otimes |R_+\rangle_{r_2}\right) + \left(|1\rangle_{q_1} \otimes |R_1\rangle_{r_1} \otimes |-\rangle_{q_2} \otimes |R_-\rangle_{r_2}\right)}{\sqrt{2}}.$$

It is fundamental to remark that the entanglement of the joint system (qubits and recorders) has increased after the measurement operation, while in the standard "collapse" formulation the analogous operation would have disentangled the two qubits.

In order to account for the fed-forwarding operations present in cluster-state computation, let us take into account the simple case of a two-qubits

chain, where the first qubit is measured in the computational basis and the second one in a basis parametrized by the measurement outcome of the first.

The measurement of the first qubit in the computational basis yields the entangled superposition:

$$|0\rangle_{q_1} \otimes |R_0\rangle_{r_1} + |1\rangle_{q_1} \otimes |R_1\rangle_{r_1} .$$

The second qubit is then measured in a basis $\mathcal{B}$:

$$\mathcal{B} = \left\{ \frac{|0\rangle \pm e^{i(-\theta(-1)^{s_i})}|1\rangle}{\sqrt{2}} \right\},$$

where $s_i \in \{0, 1\}$ is the measurement outcome of the $i-$th qubit. The eigenstates of this basis will be denoted by $\mathcal{B}_{\pm,s_i}$. The post-measurement state then results to be:

$$\begin{aligned} &\left(\left(|0\rangle_{q_1} \otimes |R_0\rangle_{r_1} \right) \otimes \left(|\mathcal{B}_{+,0}\rangle_{q_2} \otimes \left| R_{\mathcal{B}_{+,0}} \right\rangle_{r_2} + |\mathcal{B}_{-,0}\rangle_{q_2} \otimes \left| R_{\mathcal{B}_{-,0}} \right\rangle_{r_2} \right) \right) + \\ &\left(\left(|1\rangle_{q_1} |R_1 \otimes\rangle_{r_1} \right) \otimes \left(|\mathcal{B}_{+,1}\rangle_{q_2} \otimes \left| R_{\mathcal{B}_{+,1}} \right\rangle_{r_2} + |\mathcal{B}_{-,1}\rangle_{q_2} \otimes \left| R_{\mathcal{B}_{-,1}} \right\rangle_{r_2} \right) \right). \end{aligned}$$

Here we have an entangled superposition of the two eigenstates of the first qubit (the one measured in the computational basis); to each of these eigenstates there corresponds a further entangled superposition of the eigenstates of the second qubit, in the basis parametrized by the selected eigenvalue of the first qubit.

This procedure can be repeated for each computational step requiring a measurement basis parametrized by the result of a previous measurement.

Each of these steps will then generate a larger entangled superposition. The final state of the computation will be represented by a superposition analogous to the one represented in equation (20), generalized to the n qubits of the cluster.

Chapter 5

Everettian Clusters: Is the Everett Interpretation Explanatory?

The fully-unitary analysis of cluster-state quantum computation, sketched in the previous chapter, seems to reveal that the evolution of the cluster-state quantum computer could be considered as entangling: at every computational step, an entangled superposition between the measured qubit and its recorder is generated. More precisely, at every computational step, one set of qubits (characterized by the same measurement priority) is entangled to its respective set of recorders. Thus, the cluster-state quantum computer appears to be governed by an entangling evolution, and this could reveal that it is more similar to the quantum circuit framework than it initially appeared.

In spite of this, a deeper glance suggests a fundamental difference between

the two kinds of entangling evolutions here considered. In the circuit case, there is one single entangling transformation, a controlled operation acting on all the qubits at one time; in the cluster-state case, the entanglement is created step by step, because of the different bases of the qubits belonging to different measurement rounds. Thus, the main difference between cluster-state and circuit quantum computation seems to remain the absence of "quantum parallelism", also when cluster-state computation is considered in a fully-unitary dynamical picture.

However, also in cluster-state computation there is an entanglement-generating unitary transformation acting at the same time on all the qubits: the global controlled-phase operation $\mathbf{S}^{\mathcal{C}}$ involved in the preparation of the cluster. Because of the universality of the cluster, such a transformation is a necessary step of cluster-state computation, even if not peculiar to the specific algorithm to be performed; moreover, since this transformation is unitary, it is independent of the description (fully-unitary or not) of the subsequent measurements on the cluster. Another important thing to remark is that the preparation of the cluster is the only part of cluster-state computation which makes use of exclusively quantum features, because the fed-forwarding operations of the measurement outcomes can be considered as classical information processing.

When considering cluster-state computation in a fully-unitary dynamical account, one would be tempted to raise the stake and see how well does cluster-state computation fit with a fully-unitary *interpretation* of quantum computation. This chapter will then consider if the Everett interpretation

can be considered explanatory for the evolution of cluster-state computers.

5.1 Everettian Interpretation of Quantum Computation

Among the proposals which have been advanced for the explanation of quantum computation, one of the most popular is the Everettian one. It makes direct appeal to the Everettian interpretation of quantum mechanics, whose original formulation is attributed to Everett (1957) and whose most recent version is outlined by Wallace (2012). A discussion of this interpretation of quantum mechanics would go very far beyond the aims of this work, which just focuses on what Everettians have to say about quantum computation, and particularly on the challenges they receive from cluster-state computation, which is not the computational framework they usually appeal to.

When taking into account quantum computation, Everettians have an apparently favourable situation, because the "parallel-interfering-computational-paths" picture of quantum circuits seems to fit very well with their interpretation: parallels paths could just be seen as taking place in parallel worlds[1].

How can the existence of many computational worlds be argued for? Wallace (2012) makes use of an epistemic principle, taking the name of Deutsch's criterion (due to David Deutsch), according to which *something is real if it*

[1]While the introduction of many computational worlds is all but unproblematic, its immediate appeal has given popularity to the Everett interpretation within the computationalist community.

plays an indispensable explanatory role in accounting for the behaviour of other real things. Wallace's argument runs as follows:

> ... [the point] is not that there could be no other explanation for the factorization, but that we actually have a very good ... explanation. Namely, it involves simple, well-understood algorithms operating in a massively parallel way, within a single computer. It presumes that each computation happens independently, the empirical prediction is that everything will happen as if each computation is occurring independently, and there is no way of explaining the actual computational processes taking place which does not assume the computations are happening independently. By Deutsch's criterion, then, there is no way of so explaining the algorithm which does not accept the reality of all of the independent computations. At least within the quantum computer, there would be many worlds.[2]

It is relevant to remark that Wallace assumes Deutsch's criterion to be neutral as to whether the entities supposed to be real are ontologically fundamental or emergent; thus, this allows for taking the multiple branches as real even if they are not accorded a fundamental ontological status, once their indispensable explanatory role is established.

Does Wallace's argument work? Before taking into account the case of cluster-state quantum computation, it is necessary to recall two objections

[2]Wallace (2012), p. 390.

which directly hit the Everettian interpretation of quantum computation[3]. The first one regards the fact that quantum algorithms require the "parallel computational paths" to interfere between each other (in order to favour the desired outputs). This is an essential ingredient of quantum circuits, but it gets troublesome when applied to the branching process: that would be a violation of the required independence of the parallel worlds.

The second objections concerns the role of decoherence. Decoherence plays a decisive role in the Everettian interpretation of quantum mechanics; however, quantum circuits need to be kept decoherence-free in order for the unitarity to be preserved. Thus, Everettians look to be forced to introduce a process which could break down the superposition and nullify any chance of getting the speed-up.

Wallace (2012) provides a sort of rejoinder, and it remains an open question whether those attacks are fatal or not; anyway, it will be useful to compare them with the objections raising from the analysis of cluster-state computation. If the remarks against the Everettian interpretation formulated in the circuit and in the cluster-state cases were to be independent from each other, then this situation would supporting the argument that the differences between the two models are structural.

[3]Cf. Cuffaro (2013). Many objections have been raised against the Everettian interpretation of quantum mechanics (the main of them are addressed by Wallace 2012), but the two discussed here are of interest because they directly regard the explanation of computation (and do not apply to the general Everettian interpretation of quantum mechanics).

5.2 Everettian cluster-state computation

As noted in the previous chapter, there is nothing in cluster-state quantum computation resembling the parallel-computational-paths picture. Can the Everettian explanation of quantum computation hold also in that framework? A two-steps way to proceed consists in firstly trying to see whether a branching account of cluster-state computation can be formulated, and then in checking whether or not the Everettian explanation is still working in that context.

The first step requires to formulate a fully-unitary dynamical account of cluster-state computation. On the formal level, it is necessary to describe any measurement as an unitary transformation which creates an entangled superposition between the state of the observed qubit and the state of the measurement apparatus; thus, at each computational step a larger entangled state is generated. This can be simply illustrated by considering a single qubit prepared in the state:

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

A measurement requires the presence of an "observer" (which does not need to be a human being, but can just be considered as a recorder[4]), whose initial state will be written as $\left|R^{\{0,1\}}\right\rangle$, the apex symbols denoting the selected measurement basis. The eigenstates of the recorder correspond to the observed eigenstates of the qubit. The initial state of the joint system formed by the

[4] A "servomechanism", in the language of Everett (1957).

qubit and the recorder is:

$$(\alpha\,|0\rangle + \beta\,|1\rangle) \otimes \left|R^{\{0,1\}}\right\rangle$$

According to the standard non-unitary account of measurement, the joint system simply ends up in one of the two following states (where the notation $|R_i\rangle$ denotes that the value i has been recorded):

$$\alpha|0\rangle \otimes |R_0\rangle,$$

with probability $|\alpha|^2$, or

$$\beta|1\rangle \otimes |R_1\rangle,$$

with probability $|\beta|^2$. On the other hand, a fully-unitary account leaves us with the final state:

$$\alpha|0\rangle \otimes |R_0\rangle + \beta|1\rangle \otimes |R_1\rangle.$$

Now the final state is an entangled superposition, while the initial was a product state (in the following, the tensor product symbol will be omitted when clear from the context).

Such a formulation requires just an enlargement of the Hilbert space, without any commitment to specific interpretations of quantum mechanics; however, in order to treat cluster-state computation in Everettian terms, it would then be sufficient to add the postulate that the universe actually ends up in an entangled post-measurement state, but at the same time there is

a split such that observers in one branch record the outcome value 0 and observers in the other record the value 1.

Can the previous fully-unitary picture work for cluster-state computation? Let us take into account the case of a two-qubits entangled state (for the sake of simplicity, this state is taken to be maximally entangled)[5]. The initial state is the tensor of the entangled state of the qubits and the product state of the recorders:

$$\frac{|0\rangle_{q_1}|+\rangle_{q_2} + |1\rangle_{q_1}|-\rangle_{q_2}}{\sqrt{2}} \otimes \left(\left| R^{\{0,1\}} \right\rangle_{r_1} \otimes \left| R^{\{+,-\}} \right\rangle_{r_2} \right).$$

The observation of one of the qubits yields the state:

$$\frac{\left(|0\rangle_{q_1} |R_0\rangle_{r_1} |+\rangle_{q_2} |R_+\rangle_{r_2} \right) + \left(|1\rangle_{q_1} |R_1\rangle_{r_1} |-\rangle_{q_2} |R_-\rangle_{r_2} \right)}{\sqrt{2}}.$$

It is fundamental to remark that the entanglement of the joint system (qubits and recorders) has increased after the measurement operation, while in the standard "collapse" formulation the analogous operation would have disentangled the two qubits. In order to account for the fed-forwarding operations, let us take into account the simple case of a two-qubits chain, where the first qubit is measured in the computational basis and the second one in a basis parametrized by the measurement outcome of the first. The measurement of the first qubit in the computational basis yields the entangled

[5]In order to deal with multi-qubit systems, we introduce the notation $|\psi\rangle_{q_i}$ and $|R\rangle_{r_j}$, where q_i and r_j denote the $i-$th qubit and the $j-$th recorder respectively.

superposition:

$$|0\rangle_{q_1}|R_0\rangle_{r_1} + |1\rangle_{q_1}|R_1\rangle_{r_1}\,.$$

The second qubit is then measured in a basis $\mathcal{B}$:

$$\mathcal{B} = \left\{ \frac{|0\rangle \pm e^{i(-\theta(-1)^{s_i})}|1\rangle}{\sqrt{2}} \right\},$$

where $s_i \in \{0,1\}$ is the measurement outcome of the $i-$th qubit. The eigenstates of this basis will be denoted by $\mathcal{B}_{\pm,s_i}$. The post-measurement state then results to be:

$$\begin{aligned}&\left(\left(|0\rangle_{q_1}|R_0\rangle_{r_1}\right)\otimes\left(|\mathcal{B}_{+,0}\rangle_{q_2}\left|R_{\mathcal{B}_{+,0}}\right\rangle_{r_2} + |\mathcal{B}_{-,0}\rangle_{q_2}\left|R_{\mathcal{B}_{-,0}}\right\rangle_{r_2}\right)\right) + \\ &\left(\left(|1\rangle_{q_1}|R_1\rangle_{r_1}\right)\otimes\left(|\mathcal{B}_{+,1}\rangle_{q_2}\left|R_{\mathcal{B}_{+,1}}\right\rangle_{r_2} + |\mathcal{B}_{-,1}\rangle_{q_2}\left|R_{\mathcal{B}_{-,1}}\right\rangle_{r_2}\right)\right).\end{aligned}$$

Here we get an entangled superposition of the two eigenstates of the first qubit (the one measured in the computational basis); to each of these eigenstates there corresponds a further entangled superposition of the eigenstates of the second qubit, in the basis parametrized by the selected eigenvalue of the first qubit. This procedure can be repeated for each computational step requiring a measurement basis parametrized by the result of a previous measurement. Each of these steps will then generate a larger entangled superposition, until the final state corresponding to a superposition generalized to the n qubits of the cluster.

In spite of being quite twisted, this fully-unitary account shows that, at least in principle, it is possible to describe cluster-state quantum computation

in Everettian terms[6]. However, the biggest hurdle is still to be leapt: to show that the Everettian interpretation allows for an explanation which would not be achievable otherwise.

5.3 Is the Everett explanation explanatory?

Summarizing, it is possible to provide an Everettian account of the behaviour of a cluster-state quantum computer. However, this is not enough: according to Deutsch's criterion, the branching process should be playing an indispensable explanatory role. Could such an indispensable explanatory role be claimed for? The analysis of cluster-state quantum computation suggests that this claim is unsupported.

The main problem for Everettians is what, at first sight, looked like an achievement regarding one of the objections listed above: the fact that, when considering cluster-state computation in a branching dynamics, there is no interference between the branches. This could be seen as an achievement because interference between computational paths is a necessary ingredient of quantum circuits, but at the same time it is at odds with the requirement that branches should be mutually independent. Thus, cluster-state quantum computation could provide Everettians with a defence from a serious challenge. But at what price?

[6]This should come as a surprise, since Everettians just take the Hilbert space formalism as universal and without adding any further element. Thus, it would have been noticeable if it were *not* possible for Everettian to describe the behaviour of a cluster-state quantum computer

If no interference is assumed to take place between different branches, it immediately follows that the speed-up gained in a specific branch does not depend on what processes are going on within the other ones; but this entails that parallelism simply does not explain the quantum speed-up. In other words, since computational paths are completely independent from each other, quantum parallelism turns out to be unhelpful in accounting for the reasons why the computation on a specific branch involves a polynomial amount of resources.

It is thus possible to describe cluster-state quantum computation in Everettian terms, but consequently leaving no room for the "indispensable explanatory role" required by Deutsch's criterion. Since Wallace's argument is entirely built upon that principle, it seems that Everettians should build up a completely new argument if they wish to explain quantum computation.

If the Everettian explanation does not work for cluster-state quantum computers, where could an account for the speed-up be looked for? Could entanglement, which undoubtedly plays a fundamental role both in quantum circuits and in cluster-state computers, be a good candidate?

In the quantum circuit framework, there is one single entangling transformation; a controlled operation acting on all the qubits at one time. In the fully-unitary cluster-state framework, the entanglement is created step by step, because each measurement round requires a change of basis. It follows that, even if the classical part of cluster-state quantum computation is replaced by quantum operations doing the same work, the "structural" differences between quantum circuits and cluster-state computers remain in place.

Then, this does not rule out entanglement as an explanation of the quantum speed-up; rather, it makes it as an even stronger candidate.

It is also noteworthy that distinct objections are formulated against the Everettian explanation in the circuit and in the cluster-state contexts. In the first case, Everettians do have an explanation for quantum computation, but based on an account not fully compatible with the tenets of their interpretation of quantum mechanics; in the second case, they have an account that works well, but which is not explanatory. The mutual independence of these arguments could provide indirect support to the thesis that the differences between the two computational frameworks are structural.

Chapter 6

Besides the Quantum Speed-up

It is very interesting to notice that quantum circuits and cluster-state quantum computers have radically different features, the main of which regards dynamical evolution: quantum circuits have a reversible dynamics consisting of invertible unitary operations, while cluster-state computers have a non-reversible dynamics because of the non-unitarity of measurements.

Another relevant point, of particular concern for the quantum-classical divide, is that quantum circuits and cluster-state computers make a different use of quantum and classical resources.

Quantum circuits have a fully-quantum unitary evolution and the only classical operation is the final measurement. Cluster-state computers have a quantum part consisting of the preparation of the cluster (which requires the use of controlled-phase unitary operations in order to generate entanglement) and a classical part consisting in the measurements over qubits and in the fed-forwarding of information.

It is worth to be noticed that classical operations play an essential role in cluster-state computation, while having a marginal role in quantum circuits. Does this entail that, when considering quantum circuits and cluster-state computers, different quantum-classical divides need to be taken into the picture? This is not convincing, since both quantum circuits and cluster-state computers are built upon the principles of the same physical theory, namely quantum mechanics.

Turning then to the question pointed out in the introductory chapter, what kind of theoretical relevance does the match between quantum circuits and cluster-state computers bring?

There are divergences between these two frameworks, both on the physical-structural level (they have incompatible dynamics) and from the theoretical-epistemological one (they fit very differently with the Everett interpretation of quantum computation); at the same time, they also have convergences on both those levels, since they are computationally equivalent and both suggest a prominent role of entanglement in the explanation of computational processes.

This thesis has shown some of these divergences and converges, and investigated their consequences for the epistemological understanding of quantum computation. However, it would be very interesting to take into account further frameworks of quantum computation (i.e. topological quantum computation) and verify whether those convergences would still be in place. This would provide for an even deeper understanding of the nature of quantum computation.

However, the arguments presented here suggest one conclusion that could be achieved without enlarging the investigation to further quantum computational frameworks: the quantum speed-up question does not seem to capture all what is at stake concerning foundational issues relative to quantum computation.

First of all, large part of the work done in looking for an answer to the quantum speed-up question would be valuable even in case the speed-up were discovered not to be in place: for instance, insights on the role played by entanglement in quantum computation would retain importance even if classical computers, which make no use of entangled states, were discovered to be faster than they were supposed to be. Then, the definition of what a quantum-classical divide consists of goes beyond the effective existence of the speed-up.

Then, the existence of alternative equivalent (i.e. all able to achieve the speed-up) quantum computational models seems to suggest that, in order to investigate the quantum-classical computational divide, the ultimate question should be moved from the speed-up to a sort of "representation theorem" for quantum computation. Here the "representation theorem" should not be meant in a strict mathematical sense, but just as the general goal of identifying which are the physical features underlying these alternative frameworks that allow us to list all these frameworks under the label "quantum computation".

This would be very fruitful for the understanding of quantum computa-

tion itself and for the more general debate over the foundations of quantum mechanics.

Chapter 7

Summary and Conclusion

The introductory chapter of this thesis has been devoted to outline the theoretical and epistemological questions that regard quantum computation. Then, in Chapter 3 the cluster-state model of quantum computation has been introduced, enlightening its structural features and showing it "at work" on some well-known algorithms.

The reason why cluster-state computation is interesting from the perspective of philosophy of physics is that it is computationally equivalent to the traditional circuit model, but at same time it provides a structurally different account of the processes going on during a computation. Hence, a comparison between these two frameworks helps in waging an investigation the nature of quantum computation that is not limited to the particular features of one specific model.

In Chapter 4 it is shown that entanglement plays a fundamental role in both quantum circuits and cluster-state computers; as a consequence, the

position of entanglement as candidate for explaining the quantum speed-up over classical computation turns out to get reinforced. This can be considered as a "convergence" between quantum circuits and cluster-state computers, but at the same time also "divergences" can be pointed out: in Chapter 5 it is argued that the Everett explanation of quantum computation, which looks to fit well with the circuit picture, does not have explanatory power in the cluster-state framework.

In Chapter 6 it is suggested that, once the difference between quantum circuits and cluster-state computers is assumed to be structural (as claimed in Chapter 3), then the quantum speed-up question does not seem to capture all that it is at stake regarding the nature of quantum computation. It is then outlined the idea that the philosophical research on quantum computation should be directed to the general goal of a "representation theorem" for quantum computation, and that this would also require to take into account further frameworks of quantum computation, for instance topological quantum computation.

This thesis does not provide the demonstration of new results; rather, it is conceived as an investigation over results that have already been achieved, but whose consequences have not completely received the attention they would have deserved. I think that cluster-state quantum computation is an issue of interest for the community of philosophers of physics, because it concerns the nature of computational processes, the boundary between classical and quantum, and ultimately the meaning of probability. I consider

my work to be a little contribution on that topic, and I hope that further major developments could and will follow.

References

Abbott A.A. (2010), The Deutsch-Jozsa Problem: De-quantisation and Entanglement. *Natural Computing* **11**, 3-11. arXiv:0910.1990v3

Brassard G., Hoyer P., Mosca M., Tapp P. (2000), Quantum Amplitude Amplification and Estimation. arXiv:quant-ph/0005055v1

Cleve R., Ekert A., Macchiavello C., Mosca M. (1997), Quantum Algorithms Revisited. *Proceedings Of The Royal Society Of London* A **454**, 339-354. arXiv:quant-ph/9708016v1

Cuffaro M.E. (2013), *On the Physical Explanation for Quantum Computational Speedup.* PhD Thesis, University of Western Ontario. arXiv:1304.0208v1

Deutsch D. (1985), Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer. *Proceedings of the Royal Society of London* A **400**, 97-117.

Everett H. (1957), Relative state formulation of quantum mechanics. *Reviews of Modern Physics* **29**, 454-462.

Fortnow L. (2003), One Complexity Theorist's View Of Quantum Computing. *Theoretical Computer Science* **292**, 597-610.

Gross D., Flammia S.T., Eisert J. (2009), Most Quantum States are too Entangled to be Useful as Computational Resources. *Physical Review Letters* **102**, 339-354. arXiv:0810.4331v3

Grover L., (1996), A Fast Quantum Mechanical Algorithm for Database Search. *Proceedings of 28th Annual ACM Symposium on Theory of Computing*, 212-219. arXiv:quant-ph/9605043v3

Jaeger G. (2009), *Entanglement, information and the interpretation of quantum mechanics*, Berlin: Springer.

Jozsa R., Linden N. (2003), On the Role of Entanglement in Quantum Computational Speed-up. *Proceedings Of The Royal Society Of London* A **459**, 2011-2032. arXiv:quant-ph/0201143v2

Matthew Smith A., Alsing P.M., (2012), The Measurement Based Quantum Computing Search Algorithm is Faster than Grover's Algorithm. arXiv:1211.3405v3

Matthew Smith A., Alsing P.M., McDonald J.R., Uskov D.B. (2012), Is the Measurement Based Quantum Computing Search Algorithm Really Grover's Algorithm? arXiv:1211.3407v3

Meyer D.A. (2001), Quantum Games and Quantum Algorithms. in Lomonaco S.J., Brandt H.E. (eds., 2002), *Quantum Computation And Information*, Providence (RI): American Mathematical Society; 213-220. arXiv:quant-ph/0004092v2

Nielsen M.A. (2006), Cluster-state Quantum Computation. *Reports on Mathematical Physics* **57**, 147-161. arXiv:quant-ph/0504097v2

Nielsen M.A., Chuang I.L. (2000), *Quantum Computation And Quantum Information*, Cambridge: Cambridge University Press.

Prevedel R., Walther P., Tiefenbacher F., Böhi P., Kaltenbaek R., Jennewein T., Zeilinger A. (2006), High-speed Linear Optics Quantum Computing using Active Feed-forward. *Nature* **445**, 65-69. arXiv:quant-ph/0701017v1

Raussendorf R., Briegel H.J., (2001a), A One-way Quantum Computer. *Physical Review Letters* **86**, 5188-5191.

Raussendorf R., Briegel H.J., (2001b), Computational Model Under-

lying The One-way Quantum Computer. *Quantum Information & Computation* **6**, 443-486. arXiv:quant-ph/0003084v3

Steane A. (2003), A Quantum Computer Needs Only One Universe. *Studies in History and Philosophy of Modern Physics* **34**, 469-478. arXiv:quant-ph/0003084v3

Tame M.S., Prevedel R., Paternostro M., Böhi P., Kim M.S., Zeilinger A. (2007), Experimental Realization of DeutschâĂŹs Algorithm in a One-Way Quantum Computer. *Physical Review Letters* **98**, 140501. arXiv:quant-ph/0611186v2

Timpson C.G. (2008), Philosophical Aspects of Quantum Information Theory. In: Rickles D. (ed., 2008), *The Ashgate Companion to Contemporary Philosophy of Physics*, London: Ashgate, 197-261. arXiv:quant-ph/0611187v1

Van den Nest M., Dür W., Miyake A., Briegel H.J. (2007), Fundamentals of Universality in One-way Quantum Computation. *New Journal of Physics* **9**. arXiv:quant-ph/0702116v2.

Wallace D. (2008), The Quantum Measurement Problem: State of Play. In: Rickles D. (ed., 2008), *The Ashgate Companion to Contemporary Philosophy of Physics*, London: Ashgate, 197-261. arXiv:0712.0149v1

WALLACE D. (2012), *The Emergent Multiverse: Quantum Theory According to the Everett Intepretation*, Oxford: Oxford University Press.

WALTHER, RESCH, RUDOLPH, SCHENCK, WEINFURTER, VEDRAL, ASPELMEYER, ZEILINGER (2005), Experimental One-way Quantum Computing. *Nature* **434**.

ZALKA C. (1999), Grover's quantum searching algorithm is optimal. *Physical Review Letters* **A60**, 2746-2751. arXiv:quant-ph/9711070v2

Acknowledgements

This thesis would not have been possible without the support of many people.

I am sincerely thankful to my supervisor, Professor Rossella Lupacchini, that through the years has given me the chance to transform my interest for philosophy of physics and quantum computation into a structured work, and has helped me with fundamental comments and encouragement in writing this thesis and in all the work that I have been able to perform during my doctorate.

I am indebted to Professor Stefano Mancini from the Department of Physics of the University of Camerino for his advice and support, regarding this thesis and more in general the build-up of my background on quantum computation and information theory.

From January to June 2013, I had the occasion to spend a very fruitful visiting period at the Department of Philosophy at the University of Western Ontario. Of all the people that I had occasion to meet and interact with during that period, I would like to especially thank Professor Wayne Myrvold and Michael Cuffaro for advice and comments that have been decisive in defining the subject of this thesis.

Several people provided a precious help by reading or informally discussing part of this thesis, among which I would like to mention Fabrizio Baldassarri, Enrico Bergianti, Beatrice Collina, Tommaso Fasano, Luca Iori, Lorenzo Medici, Eugenio Orlandelli, Michele Palmira, Valeria Vignudelli and of course my parents.

I am finally indebted to the coordinator of the Science, Cognition and Technology PhD programme Professor Giuliano Pancaldi and to the rest of the students, faculty members and staff of the Department of Philosophy at the University of Bologna for having made the years that I have spent here as a PhD candidate an inestimable and unforgettable experience.