Chapter 15
The Future of Parallel Computation

Selim G. Akl and Marius Nagy

Abstract

As any other scientific discipline, computing science is undergoing a continuous process of transformations and innovations driven by theoretical research and technological advancements. Inspired by physical and biological phenomena occurring in Nature, new computational models are proposed, with the potential to greatly increase the efficiency of computational processes. Another direction of development pertains to the characteristics of the problems tackled by computing science. With the increasingly ubiquitous and pervasive nature of computers in the modern society, the class of problems and applications computing science has to address is continuously expanding.

The importance played by parallelism in each of these two major development trends confirms the fundamental role parallel processing continues to occupy in the theory of computing. The idea of massive parallelism permeates virtually all unconventional models of computation proposed to date and this is shown here through examples like DNA computing, quantum computing or reaction-diffusion computers. Even a model that is mainly of theoretical interest, like the accelerating machine, can be thought of as deriving its power from doubling the number of processing units (operating in parallel) at each step.

The scope of computing science has expanded enormously from its modest boundaries formulated at the inception of the field and many of the unconventional problems we encounter today in this area are inherently parallel. We illustrate this by presenting five examples of tasks in quantum information processing that can only be carried out successfully through a parallel approach. It is one more testimony to

Selim G. Akl
School of Computing, Queen's University, Kingston, Ontario, Canada,
e-mail: akl@cs.queensu.ca

Marius Nagy
School of Computing, Queen's University, Kingston, Ontario, Canada,
e-mail: marius@cs.queensu.ca

the fact that parallelism is universally applicable and that the future of computing cannot be conceived without parallel processing.

15.1 Introduction

The purpose of this final chapter is to glance into the future and sketch the most probable forms parallel computing may take, having as a starting point the trends we can observe today. When it comes to computing in parallel, we can distinguish two major directions heading into the future. The first is strongly related to the continuous development and expansion of the Internet and the improvement in network management schemes. Having better means available for a group of computers to cooperate in solving a computational problem, whether they are homogeneous or heterogeneous, geographically close to each other or distributed over the Internet, small or large in number, will inevitably translate into a higher profile of clusters and grids in the landscape of parallel and distributed computing. We are not going to insist further here on the increasing role played by clusters and grids, especially since we have already discussed them in Chapter 2 in the context of models of parallel computing.

Another, more revolutionary, direction taken by parallel computation challenges the very physical level at which information is stored and manipulated in a computing machine. The electronic computers now in use are based on large scale integration of transistors onto silicon chips, such that a logical bit is physically realized as a voltage level in an electronic circuit. Although this technology was able to sustain a steady increase in the speed of processors over the past decades, its limits are well in sight by now. Consequently, researchers focused on finding alternate ways of encoding and processing information that have the potential to speed up computation beyond what is possible using an electronic computer.

Proposals for an unconventional computing device include, but are not limited to, computing with DNA strands, quantum information processing, membrane computing (P systems) or computations in reaction-diffusion systems. All these alternatives are inspired from natural phenomena and each advances a fundamentally new physical support for information.

DNA Computing

In DNA computing, the computation is performed by synthetically obtained DNA strands. Performing an algorithm in this context amounts to applying some standard lab manipulation techniques (annealing and denaturation, polymerase chain reaction, gel electrophoresis, etc.) to the DNA in a test tube [1]. The strands act both as processors and memory units. Using this "bioware", NP-complete problems can be solved in linear time by covering an exponential search space in parallel [2, 3, 4]. The Watson-Crick complementarity, responsible for the formation of the hydrogen

bonds that allow two strands of DNA to anneal together, is the key mechanism used to explore all possible computational paths simultaneously. Unfortunately, the scalability of this technique is severely restricted by the amount of DNA required to ensure an exhaustive search and by the error rate of the molecular operations involved.

Membrane Computing

Another biologically inspired computational model bears the name of membrane computing or P systems, in honor of its founder, Gheorghe Păun [5]. The model employs a hierarchy of membranes, with each membrane separating a region, just as cell components (the nucleus, the Golgi Apparatus, mitochondria and various vesicles) as well as the whole cell itself are identified by a separating membrane. The mathematical equivalent of molecules (chemicals) inside a cell component are symbols belonging to a certain region. The analogy continues by regarding chemical reactions as production rules.

All membrane regions evolve simultaneously, according to a global clock. During each time unit, in each region, all applicable rules are applied non-deterministically, in a maximally parallel manner. The computation stops when no further rules can be applied and the result (output) is read either from the environment (outside the skin membrane) or as the content of some non-destructible membrane.

Since the inception of the field, a plethora of variants of membrane systems have been defined and their computational power studied. In particular, algorithms have been designed to solve NP-complete problems in polynomial time, with the trade-off of exponential space [6, 7, 8]. Through membrane division, however, this exponential space can be created in polynomial (linear) time.

Quantum Information Processing

A strong candidate for tomorrow's computing paradigm is manipulating information at the quantum level. The idea of harnessing quantum mechanical effects in order to improve the efficiency of computation naturally follows the miniaturization trend witnessed in the computer industry for so many years now. According to this trend, we will soon reach the atomic and sub-atomic level for the embodiment of a logical bit and, inevitably, the laws of quantum mechanics will have to be taken into consideration.

Similar to DNA computing, quantum algorithms attempt to find a solution to a problem through an exhaustive search. The efficiency of the procedure comes again from the fact that an exponential number of computational paths can be pursued in parallel, by preparing a quantum register in a superposition state comprising an exponential number of classical states. Thus, for applications like integer factorization and finding discrete logarithms, a quantum computer offers an exponential speedup over a conventional one [9]. Quantum algorithms acting on small inputs have been

successfully implemented in practical experiments [10, 11, 12], but the main difficulty to overcome remains the scalability of the various techniques proposed to build a quantum computer.

The Reaction-diffusion Computer

Proposals to improve the efficiency of computation can come from any branch of science, not only biology or physics. Our last example draws its inspiration from chemistry: the reaction-diffusion computer. In this truly novel paradigm, both the data and the results of the computation are encoded as concentration profiles of the reagents. The computation itself is performed via the spreading and interaction of wave fronts. Because molecules diffuse and react in parallel, a computer based on reaction-diffusion is endowed with a natural parallelism that allows it to efficiently solve combinatorial problems [13].

Probably the most evident and, at the same time, fundamental observation about the unconventional computing paradigms enumerated above is that they owe their computational power to some form of *massive parallelism*. In a full test tube acting as a DNA computer we may have $10^{15} - 10^{17}$ operations performed in parallel, while a small chemical reactor may host millions of elementary $(2 - 4$ bit) processors operating in parallel through reaction-diffusion means. Similar characteristics empower the other two paradigms mentioned. It is therefore justified to affirm, without the risk of making an overstatement, that parallel processing lies at the heart of the quest for efficiency in computation. As various computing devices are infiltrating every aspect of human life and the pervasive nature of computers is on the rise, the need for a parallel approach comes also from an increasing number of applications dealing with real-time requirements and inherently parallel problems.

Parallel processing may even be the power behind some hyper-computational models credited with capabilities that go beyond those of a Turing machine. Thus, the accelerating machine, a computational model of mainly theoretical interest, can double its speed at each step [14]. More precisely, the time required by an operation at any given step of a computation is only half (or some other constant fraction) of that required to perform the same operation in the previous step. This property allows the accelerating machine to perform any number of iterations of a computational step in a finite amount of time. As a consequence, solving the Halting Problem is within the reach of the accelerating machine. This result is entirely due to the accelerating feature of the model and one way the speed can be doubled each step is by doubling the number of processors operating in parallel at each step.

But the particular form that parallelism will take in the operation of tomorrow's computing machines remains for the future to decide. The few paradigms briefly discussed in this section are representative for the efforts made nowadays towards a radically new computing technology, with important advantages over the electronic computer. As things stand today, we credit quantum information processing with

the highest chances of playing an important role in the way computations are going to be performed in a few decades time. This attitude is encouraged by the tremendous effort put today into finding a viable design for a practical quantum computer and the impressive achievements, already commercially available, in quantum cryptography. Consequently, the remainder of this chapter will focus on uncovering the "secrets" responsible for the potential quantum computation has to offer and the different ways parallelism is encountered in this novel paradigm of computation.

15.2 Quantum Computing

The field of quantum information processing is based on the postulates governing quantum mechanics. The aim of this section is to familiarize the reader with these postulates and the mathematical formalisms required to work with them to the extent needed for quantum computing. Good introductions to quantum mechanics for computing scientists can be found in [15, 16, 17, 18, 19], but for a detailed exposition of the field one should see [20].

15.2.1 Quantum Mechanics

We begin our presentation by describing a few experiments that, in our opinion, best illustrate those features of quantum mechanics that are at the heart of quantum information processing, namely, *superposition*, *measurement* and *interference*.

15.2.1.1 Double-slit Experiment

This experiment was first conducted by Thomas Young in 1801, and it demonstrated that light behaves like waves. Young projected light onto a screen through a barrier pierced with two closely spaced slits (see Fig. 15.1). What he observed on the screen was an *interference* pattern, the hallmark of waves. The importance of modern-day versions of Young's experiment is best illustrated by Richard Feynman in his *Lectures* [20]. He believed that the result of the double-slit experiment was the fundamental mystery of quantum mechanics.

If Young performed his experiment using simple screens and candlelight, the tremendous advances in technology allow us today to repeat the experiment with very weak light, that is, light produced as one photon at a time. Thus, it is very unlikely that several photons would be found within the experimental apparatus at the same time. Surprisingly (and against our intuitions), given that enough time elapses as to allow the photons, arriving one at a time, to accumulate on the screen, the same interference pattern will appear. The obvious question is: what was each photon interfering with, if it was alone in the experimental apparatus?

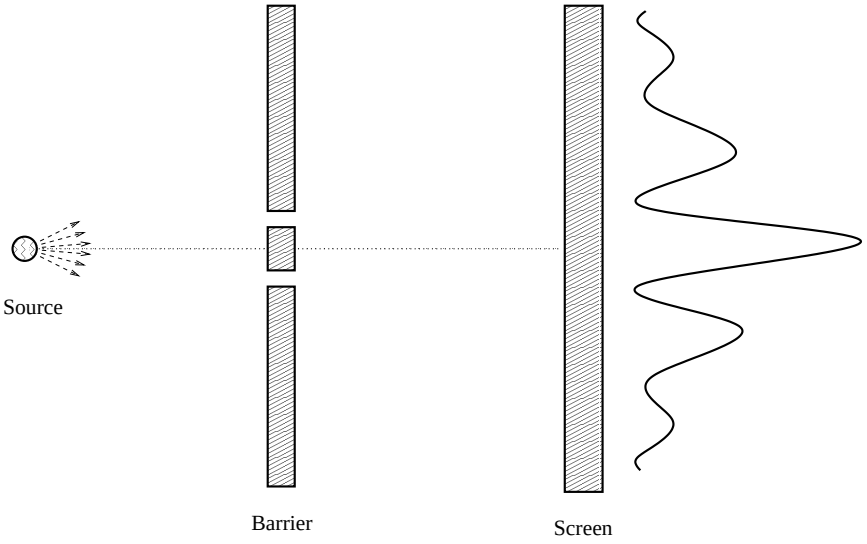

Fig. 15.1 Young's double-slit experiment. Light projected onto a screen through a barrier pierced with two closely spaced slits creates an *interference pattern.*

According to the Copenhagen interpretation (the standard view among many physicists), the only possible answer can be: with itself. In the absence of any observations, it doesn't make sense to associate a specific route to the photon in its way from the light source to the screen. In a sense, each particle went not through one slit, but rather through both slits, and, as it appeared on the other side, it interfered with itself. This behavior is a manifestation of the quantum principle of superposition of states, a principle without which quantum computation and quantum information would be inconceivable.

If we choose to observe the particle as it goes through the experimental apparatus (that is, to measure its state), the wave function describing it will collapse into one of the two possible outcomes and the particle will be detected passing through one of the two slits with equal probability. In either case, the superposition is destroyed and with it any chance of interference. But if the particle is not observed until the end, as it collects on the screen, then the superposition holds through to the end, enabling the interference phenomenon witnessed on the screen. The duality between particles and waves has also been demonstrated for other quanta that can be localized (electrons, neutrons, atoms) and even for larger entities, like complex molecules composed of tens of atoms.

15.2.1.2 Single Photon Interferometry

The Mach-Zehnder interferometer (depicted in Fig. 15.2) is an optical device composed of beam splitters, mirrors and photon detectors carefully placed to bring about

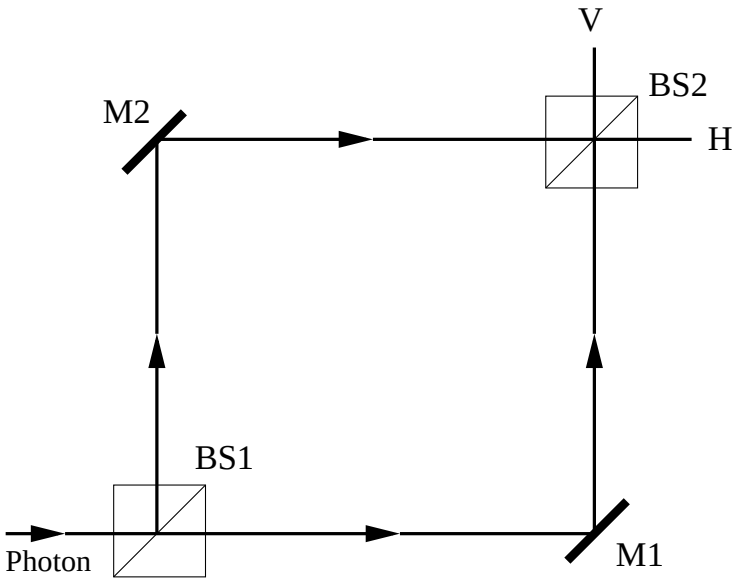

Fig. 15.2 A Mach-Zehnder interferometer (BS=beam splitter; M=mirror). A photon entering the first beam splitter horizontally, will always emerge from the horizontal port of the second beam splitter, due to *self-interference*.

quantum interference when a photon travels through the apparatus. A beam splitter is a half-silvered mirror that will let half of the incident beam pass through and reflect the other half. But when a single photon is confronted with a beam splitter, its state becomes a superposition of being reflected and going through at the same time. Thus, a photon entering the first beam splitter horizontally, will always emerge from the horizontal port of the second beam splitter, provided the two arms of the interferometer have equal lengths. As in the case of Young's two-slit experiment, the reason is *self-interference*.

The probabilities of leaving the interferometer horizontally in the two possible histories (traveling the upper arm and lower arm, respectively) reinforce each other during the interference process that takes place in the second beam splitter. At the same time, the probabilities of leaving the experimental apparatus vertically cancel each other out. Any attempt to find out which way the photon took through the experimental device will collapse the superposition and ruin the interference. In such a case, there will be an equal probability of detecting the photon exiting horizontally or vertically, regardless of the path the photon was observed to take between the beam splitters.

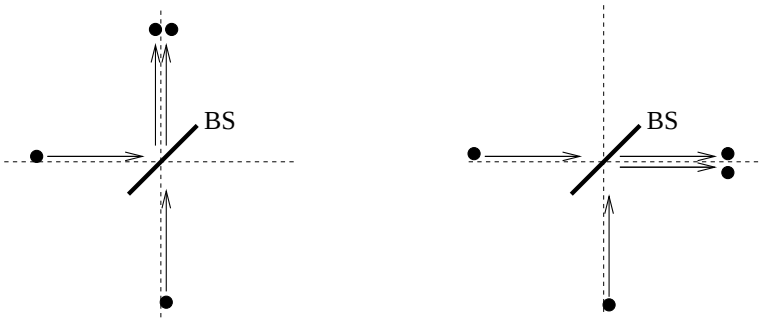

Fig. 15.3 The two photons always emerge from the beam splitter (BS) along the same output, due to a quantum-interference effect.

15.2.1.3 Two-photon Interferometry

The quantum-interference effect witnessed at the second beam splitter in the previous experiment can also occur if two single-mode, but otherwise independent, photons enter a $50-50$ beam splitter, as shown in Fig. 15.3. The "mode" of a photon refers to the physical properties, like frequency and polarization, that together define the electromagnetic field with which a photon is associated. When the two photons are in the same mode, all the properties of the two photons are identical at the beam splitter output, so they become essentially indistinguishable.

As a consequence of this "bosonic" character of photons, the probabilities that both photons will be transmitted or both reflected interfere destructively, canceling each other. As a result, the two photons will always be seen emerging from the beam splitter along the same output, either both horizontally or both vertically. This surprising quantum-interference effect was demonstrated for independent photons, emitted from a single-photon source [21]. Such an experiment is also important from the practical viewpoint of building quantum logic gates for photon-based quantum computing [22].

15.2.2 Mathematical Framework

Quantum mechanics takes place in the framework provided by linear algebra. We can associate to any isolated physical system a complex vector space with an inner product defined on it, known as the state space of the system. Mathematically, such a vector space with an inner product is called a Hilbert space. At any given point in time, the system is completely described by its state vector, which must be a unit vector in the system's state space.

Quantum state spaces and the transformations acting on them are traditionally described in terms of vectors and matrices using the compact *bra/ket* notation introduced by Dirac [23]. According to his conventional notation, for states that corre-

spond to discrete values of an observable, *kets* like $|x\rangle$ are simply column vectors, typically used to describe quantum states. Similarly, the matching *bra* $\langle x|$ is a row vector denoting the conjugate transpose of $|x\rangle$.

15.2.2.1 The Qubit

At an abstract level, the simplest quantum mechanical system is the quantum bit, or *qubit*. A qubit is a unit vector in a two-dimensional state space, for which a particular orthonormal basis, denoted by $\{|0\rangle, |1\rangle\}$ has been fixed. The two basis vectors $|0\rangle$ and $|1\rangle$ correspond to the possible values a classical bit can take. However, unlike classical bits, a qubit can also take many other values. In general, an arbitrary qubit $|\Psi\rangle$ can be written as a linear combination of the computational basis states:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{15.1}$$

where α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$. This is the fundamental difference distinguishing quantum bits from classical ones and is a direct application of the quantum principle of superposition of states. The qubit $|\Psi\rangle$ in Eq. (15.1) is in a superposition of $|0\rangle$ and $|1\rangle$, a state in which it is not possible to say that the qubit is definitely in the state $|0\rangle$, or definitely in the state $|1\rangle$. After all, what better intuition about the superposition principle than the idea (quite old and widely accepted now) that each particle is also a wave.

For a single qubit, there is a very intuitive geometric representation of its state as a point on a sphere. Taking $\alpha = e^{i\gamma}\cos(\theta/2)$ and $\beta = e^{i\gamma}e^{i\varphi}\sin(\theta/2)$ in Eq. (15.1), we can rewrite the state of qubit $|\psi\rangle$ as

$$|\psi\rangle = e^{i\gamma}(\cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle), \tag{15.2}$$

where θ, φ and γ are real numbers. Note that this is always possible since $|\alpha|^2 + |\beta|^2 = 1$. Also, because a global phase factor like $e^{i\gamma}$ has no observable effects (i.e., it does not influence the statistics of measurement predicted for qubit $|\psi\rangle$), we can effectively ignore it. Consequently, the pair (θ, φ) uniquely identifies a point $(\cos\varphi\sin\theta, \sin\varphi\sin\theta, \cos\theta)$ on a unit three-dimensional sphere called the *Bloch sphere* [24, 17].

Fig. 15.4 depicts four possible states of a qubit using the Bloch sphere representation. Note that the states corresponding to the points on the equatorial circle have all equal contributions of 0-ness and 1-ness. What distinguishes them is the *phase*. For example, the two states displayed above, $1/\sqrt{2}(|0\rangle + |1\rangle)$ and $1/\sqrt{2}(|0\rangle - |1\rangle)$ are the same up to a relative phase shift of π, because the $|0\rangle$ amplitudes are identical and the $|1\rangle$ amplitudes differ only by a relative phase factor of $e^{i\pi} = -1$.

We have described qubits as mathematical objects, but there are real physical systems which may be described in terms of qubits. Possible physical realizations of a qubit include two different polarizations of a photon, the alignment of a nuclear spin in a uniform magnetic field or two electronic levels in an atom. In the experi-

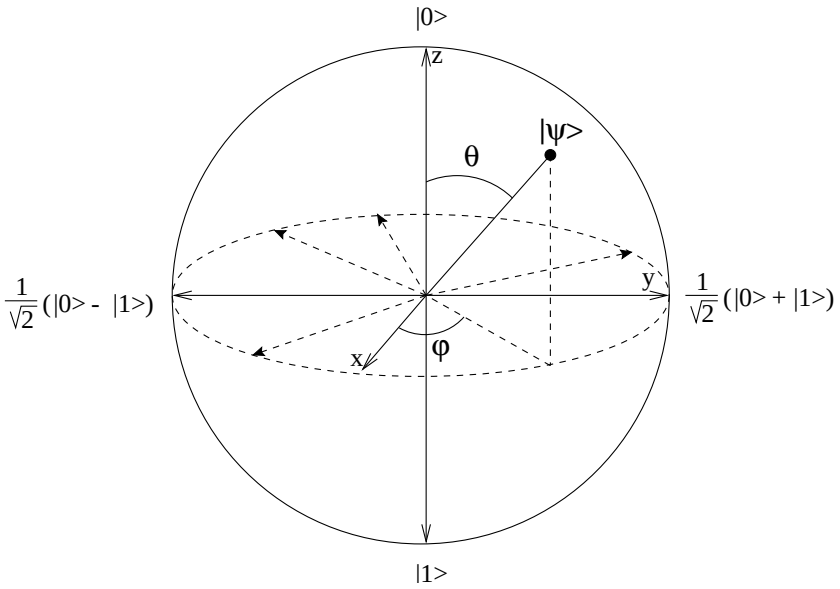

Fig. 15.4 The Bloch sphere representation of a qubit.

ments presented at the beginning of the section, the state of a photon is described in terms of the two possible routes that can be used when traversing the experimental apparatus.

15.2.2.2 Measurements

We now turn our attention on the amount of information that can be stored in a qubit and, respectively, retrieved from a qubit. Since any point on the Bloch sphere can be characterized by a pair of real-valued parameters taking continuous values, it follows that, theoretically, a qubit could hold an infinite amount of information. As it turns out, however, we cannot extract more information from such a qubit than we are able to do it from a classical bit.

The reason is that we have to *measure* the qubit in order to determine which state it is in. And another of the fundamental postulates of quantum mechanics, the one regarding measurements (Postulate 3 in [17]), restricts us in the amount of information that can be gained about a quantum state through measurement. According to this postulate, when we measure a qubit $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ with respect to the standard basis for quantum computation $\{|0\rangle, |1\rangle\}$, we get either the result 0 with probability $|\alpha|^2$, or the result 1 with probability $|\beta|^2$. The condition that the probabilities must sum to one corresponds geometrically to the requirement that the qubit state be normalized to length 1, that is the inner product $\langle\Psi|\Psi\rangle$ equals 1.

Furthermore, measurement alters the state of a qubit, collapsing it from its superposition of $|0\rangle$ and $|1\rangle$ to the specific state consistent with the measurement result.

For example, if we observe $|\Psi\rangle$ to be in state $|0\rangle$ through measurement, then the post-measurement state of the qubit will be $|0\rangle$, and any subsequent measurements (in the same basis) will yield 0 with probability 1. In general, measurement of a state transforms the state into one of the eigenvectors of the observable being measured. The probability that the state is measured as basis vector $|u\rangle$ is the square of the norm of the amplitude of the component of the original state in the direction of the basis vector $|u\rangle$. The implicit assumption we adopt herein is that a measurement is performed in the standard basis for quantum computation, whenever the basis vectors associated with the measurement operation are not stated explicitly.

15.2.2.3 No-clonability

Naturally, measurements in bases other than the computational basis are always possible, but this will not help us in determining α and β from a single measurement. One might think of solving this problem by making multiple copies of the initial qubit $|\Psi\rangle$ and then measuring each of the copies in order to obtain an estimation of α and β. In fact, it turns out to be impossible to make a copy of an unknown quantum state. The *no-cloning* theorem, one of the earliest results of quantum computation and quantum information [25], states that quantum mechanics prevents us from building a quantum cloning device capable of copying non-orthogonal quantum states. The ability to clone orthogonal quantum states translates into the ability to copy classical information, since the different states of classical information can be thought of merely as orthogonal quantum states. So it seems that quantum mechanics places severe limitations on the accessibility of quantum information, but in some circumstances (like devising secure quantum cryptographic protocols, for instance) this can be turned into an advantage.

15.2.2.4 Quantum Registers

Let us examine now more complex quantum systems, composed of multiple qubits. In classical physics, individual two-dimensional state spaces of n particles combine through the Cartesian product to form a vector space of $2n$ dimensions, representing the state space of the ensemble of n particles. However, this is not how a quantum system can be described in terms of its components. Quantum states combine through the tensor product to give a resulting state space of 2^n dimensions, for a system of n qubits. It is this exponential growth of the state space with the number of particles that quantum computers try to exploit in their attempt to achieve exponential speedup of computation over classical computers.

For a system of two qubits, each with basis $\{|0\rangle, |1\rangle\}$, the resulting state space is the set of normalized vectors in the four dimensional space spanned by basis vectors $\{|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle\}$, where $|x\rangle \otimes |y\rangle$ denotes the tensor product between column vectors $|x\rangle$ and $|y\rangle$. It is customary to write the basis in the more

compact notation $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. This generalizes in the obvious way to an n-qubit system with 2^n basis vectors.

15.2.2.5 Quantum Evolution

The next step after laying the mathematical foundation for describing quantum registers is to focus on the "circuits" composing a hypothetical quantum computer. Operating a quantum gate is strongly related to the way an isolated quantum system evolves over time. We already saw what happens when we try to measure such a quantum system. If, for example, we are trying to read the content of a quantum memory register, the system will undergo a sudden, unpredictable jump into one of the eigenvectors associated with the measurement operator. In other words, there will be a discontinuity in the evolution of the quantum memory register. But, if we leave the register unobserved, the system will undergo a smooth, continuous evolution governed by Schrödinger's equation, a deterministic differential equation which enables us to predict the future or uncover the past evolution of the memory register. Consequently, any quantum computation is reversible and therefore quantum gates (the quantum analog of classical gates) must always have as many outputs as they have inputs, in order to avoid any loss of information that would prevent the computation from being undone.

15.2.2.6 Quantum Gates

A quantum NOT gate acting on a single qubit will evolve the initial state $\alpha|0\rangle + \beta|1\rangle$ into the final state $\alpha|1\rangle + \beta|0\rangle$, in which the roles of $|0\rangle$ and $|1\rangle$ have been interchanged. Because every quantum gate acts linearly, the transformation is fully specified by its effect on the basis vectors. Hence, there is a very convenient representation of a quantum gate in matrix form. Starting from the expressions of the two basis vectors in column form:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \; |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{15.3}$$

the matrix X representing the quantum NOT gate is then defined as follows:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{15.4}$$

The first column represents the effect of applying the NOT gate to state $|0\rangle$, while the second column is the result of applying the NOT gate to state $|1\rangle$. We can now describe the operation of the quantum NOT gate, acting on an arbitrary qubit state, through the following equation:

$$X \cdot \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}. \tag{15.5}$$

Other examples of single qubit gates are the Z gate:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{15.6}$$

which leaves $|0\rangle$ unchanged, but introduces a phase shift by flipping the sign of $|1\rangle$, and the Hadamard gate:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{15.7}$$

which is one of the most useful quantum gates, because it creates superpositions of $|0\rangle$ and $|1\rangle$.

Although there are an infinite number of single qubit gates, not any two by two matrix is a legitimate representation of a quantum gate. Schrödinger's equation states that the dynamics of a quantum system must take states to states in a way that preserves orthogonality. In other words, the normalization condition $|\alpha|^2 + |\beta|^2 = 1$ for the initial state $\alpha|0\rangle + \beta|1\rangle$ must also be true for the quantum state after the gate has acted. This translates into the requirement that the matrix U describing the single qubit gate be *unitary*, that is $U^* \cdot U = I$, where U^* is the conjugate transpose of U. Single qubit gates can be conveniently visualized as rotations of the arrow representing the qubit state on the surface of the Bloch sphere.

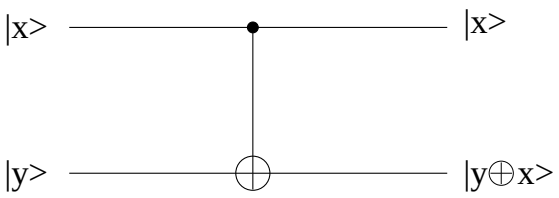

Fig. 15.5 Controlled-NOT quantum gate.

Quantum gates on multiple qubits can also be defined. Fig. 15.5 depicts a controlled-NOT gate, an instance of the more abstract controlled-U gate, where $U = X$. The target bit $|y\rangle$ is flipped if and only if the control bit $|x\rangle$ is set to 1. The matrix describing the operation of the controlled-NOT gate is:

$$CNOT = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,0\,1 \\ 0\,0\,1\,0 \end{bmatrix}. \tag{15.8}$$

Multiple qubit gates must also satisfy the requirement that probability be conserved, so they too must be unitary transformations. Since any unitary matrix is invertible and the inverse is also a unitary matrix, it follows that a quantum gate can always be inverted by another quantum gate. The set of all 1-qubit rotations (gates) together with the controlled-NOT gate is universal for quantum computation. But

finite universal sets of gates exist as well. Two researchers working independently have shown that any imaginable quantum computation can be performed by connecting together multiple copies of a certain 2-qubit gate [26, 27]. Such universal quantum gates are analogous to the NAND gate in classical computation.

15.2.3 Entanglement

Entanglement is probably the strangest and most controversial aspect of quantum mechanics, but at the same time it is credited with the most surprising applications. This section contains a brief discussion of this unusual phenomenon.

Similar to single qubits, multiple-qubit systems can also be in a superposition state. The vector

$$|\Psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \tag{15.9}$$

describes a superposition state of a two-qubit system in which all four components (corresponding to the four basis vectors) have equal amplitudes. What about the two qubits composing the system? Can we characterize their states individually? If we rewrite Eq. (15.9) in order to express $|\Psi\rangle$ as the tensor product

$$|\Psi\rangle = (\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle) \otimes (\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle) \tag{15.10}$$

then we can legitimately assert that each of the component qubits is also in a superposition state, perfectly balanced between $|0\rangle$ and $|1\rangle$. Now let us drop the two middle terms in Eq. (15.9) and consider the superposition state described by

$$|\Phi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \tag{15.11}$$

In this case it is no longer possible to find complex numbers α, β, γ and δ such that

$$(\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle) = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \tag{15.12}$$

The state of the system cannot be decomposed into a product of the states of the constituents. Even though the state of the system is well defined (through the state vector $|\Phi\rangle$), neither of the two component qubits is in a well-defined state. This is again in contrast to classical systems, whose states can always be broken down into the individual states of their components. Furthermore, if we try to measure the two qubits, the superposition will collapse into one of the two basis vectors contributing to the superposition and the outcomes of the two measurements will always coincide. In other words, if one of the qubits is found to be in state $|0\rangle$, then the second one will necessarily be in the same state, while a state $|1\rangle$ assumed after measurement will be shared by both qubits. Therefore, we say that the two qubits

are entangled and $|\Phi\rangle$ describes an entangled state of the system. It was Schrödinger who actually named the phenomenon *entanglement* in 1935 [28].

Entanglement defines the strong correlations exhibited by two or more particles when they are measured and which cannot be explained by classical means. This does not imply that entangled particles will always be observed in the same state, as entangled states like

$$\frac{1}{\sqrt{2}}|01\rangle \pm \frac{1}{\sqrt{2}}|10\rangle \tag{15.13}$$

prove it. States like these or the one in Eq. (15.11) are known as Bell states or EPR pairs after some of the people who pointed out their strange properties [29].

In some sense, we can say that superposition encompasses entanglement, since entanglement can be viewed as a special case of superposition. It is also interesting to make an analogy between entanglement and the concept of primality from number theory. Indeed, an entangled state of the system corresponds to a prime number, since it cannot be factored or decomposed as a product of subsystem states.

15.3 Parallelism in Quantum Computing

We now have the necessary tools to discuss the role of parallelism in quantum computing. When talking about parallelism in the context of quantum computation, the immediate understanding given to the term refers to the ability of a quantum computer to simultaneously evolve (transform) a potentially large number of classical states, by preparing a quantum register in a superposition of all those states and then applying the desired transformation on the quantum register. This form of parallelism is specific to quantum computing because it exploits the quantum mechanical principle of superposition of states and, hence, it is termed *quantum parallelism*. We describe in detail the mechanism of quantum parallelism in the following section and show that it is the key ingredient in obtaining an exponential speedup over a conventional computer, for some applications.

15.3.1 Quantum Parallelism

Suppose we want to evaluate an arbitrary function $f : N \rightarrow N$ for various inputs x. Then we can define a unitary quantum gate U_f, whose action on the inputs x and y is shown in Fig. 15.6. Since U_f must be reversible by definition, we need input y in order to ensure that x is "remembered" at the output (no loss of information). The image of x through f XOR-ed with y is obtained on the bottom output line ($\oplus$ denotes an Exclusive OR operation or, equivalently, addition modulo 2). In general, the input and corresponding output lines depicted in Fig. 15.6 may represent an arbi-

trary number of qubits, such that $\oplus$ is applied bitwise. This construction is possible for any function f.

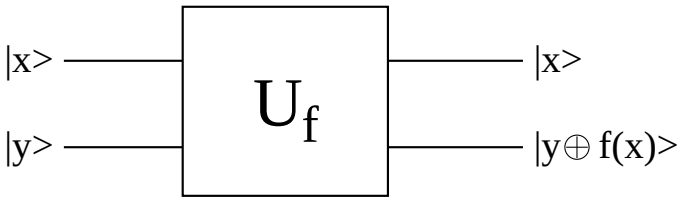

Fig. 15.6 A generic quantum gate designed to compute the values of a function f.

In order to compute $f(x)$, for some input x, we set y to zero and then $f(x)$ can be read from the bottom ouput line(s):

$$U_f(|x\rangle \otimes |0\rangle) = |x\rangle \otimes |f(x)\rangle. \tag{15.14}$$

The advantage of the quantum paradigm of computation now becomes apparent. If we want to compute $f(x)$ for an arbitrary number of inputs x, all we have to do is to prepare the x part of the quantum register as a superposition of all inputs that we want to be evaluated and then apply the gate U_f. The y part of the register, which was initially 0, now stores a superposition of all images $f(x)$ that we sought to compute.

In particular, if we start with n qubits, each in the state $|0\rangle$, and apply a Hadamard gate to each of them, then what we get is a superposition of all inputs from 0 to $2^n - 1$:

$$\frac{1}{2^n} \sum_{i=0}^{2^n-1} |i\rangle. \tag{15.15}$$

Now, with a single application of the gate U_f we obtain all 2^n corresponding images in a superposition:

$$U_f((\frac{1}{2^n} \sum_{i=0}^{2^n-1} |i\rangle) \otimes |0\rangle) = (\frac{1}{2^n} \sum_{i=0}^{2^n-1} |i\rangle) \otimes (\frac{1}{2^n} \sum_{i=0}^{2^n-1} |f(i)\rangle). \tag{15.16}$$

In this way, a quantum computer can evaluate an exponential number of inputs in the time it takes a conventional electronic computer to evaluate just one input. This type of transformation operating in parallel on all inputs is known as *quantum parallelism*. The enormous potential of a quantum computer to outperform a classical machine lies precisely in the massive parallelism it offers "within a single piece of hardware" [30].

This form of parallelism, however, does not automatically translate into an exponential speedup for any computational problem. The difficulty resides in extracting the information we have computed in quantum parallel. In order to see what are the values $f(x)$ obtained, we must read (that is, measure) the quantum register. And we have already seen in Sect. 15.2.2.2 that measuring is a disruptive process implying a

loss of information by collapsing the superposition state of the quantum register to a state compatible with the outcome obtained through measurement. This means that, from the 2^n values encoded in the state of the quantum register before measurement, we can only read out one and worse still, we don't even have control over which one we get, since the measurement process is a probabilistic one.

Nevertheless, an exponential speedup can still be obtained, if the information sought through measurement is a global property of all terms in the superposition and not just one particular term. It is the case, for example, of the quantum algorithm devised by Shor to factorize an integer in polynomial time [9]. Knowing that factoring n is as hard as computing orders modulo n, Shor set out to find the period of a function $f_{x,n}(a) = x^a \bmod n$, for some x chosen to be co-prime with n. Once the period is found, the divisors of n can easily be inferred using standard techniques from number theory.

Classically, in order to find the period of a function, we need to evaluate that function over and over again, for many different inputs. But using quantum parallelism, we only need one evaluation. Furthermore, what we need afterwards is a global property of all images through f (the period) and not a particular image of a particular input. In Shor's algorithm, the quantum Fourier transform is used in order to interfere the computational paths and bring out the period. Some kind of Fourier transform is usually employed in quantum algorithms to constructively recombine different alternatives in a superposition such that the amplitude of solutions is strengthened, while non-solutions interfere destructively, canceling each other.

Thus, factoring integers (and the related problem of finding discrete logarithms) can be solved in quantum polynomial time, while the best known classical technique for factorization (the number field sieve) is super-polynomial or sub-exponential in the size of the integer to be decomposed. On the other hand, for the vast majority of problems in computer science (including NP-complete ones) quantum parallelism is not expected to bring more than a quadratic speedup [31].

But *quantum parallelism* is not the only form of parallelism encountered in the context of quantum information processing. This syntagm is used to denote the ability to perform a certain computation simultaneously on all terms of a quantum superposition, regardless of the number of qubits composing the quantum register whose state is described by that superposition. A different interpretation refers to parallelism as the ability to act simultaneously on a certain number of qubits, whether for the purpose of measuring them or evolving their quantum state.

In the following section, we illustrate this second meaning of the term *parallelism* in quantum computation by presenting five examples in which a parallel computing approach is most appropriate, if not vital, for the success of the computation. The common theme of all these examples, apart from the fact that they are all drawn from the field of quantum information processing, is their *evolving* nature, in the sense that their characteristics vary during the computational process itself. Because of their dynamic nature, these computations may be labeled as *unconventional*, as opposed to the computation performed by a Turing machine, for example. The problems we are about to describe may also be interpreted as quantum mechan-

ical instances of the unconventional computing paradigms introduced in Chapter 2 as computations that cannot be simulated sequentially.

15.4 Examples

In each of the five cases enumerated below, we describe the problem as it is formulated in quantum information processing and emphasize the importance of a parallel approach in order to reach a solution. Furthermore, we identify the characteristics that make it belong to a certain class of unconventional (evolving) computations.

15.4.1 Parallelizing the Quantum Fourier Transform

The Fourier transform is a very useful tool in computer science and it proved of crucial importance for quantum computation as well. Since it can be computed much faster on a quantum computer than on a classical one, the discrete Fourier transform allows for the construction of a whole class of fast quantum algorithms. Shor's quantum algorithms for factoring integers and computing discrete logarithms [9] are the most famous examples in this category.

The quantum Fourier transform (QFT) is a linear operator whose action on any of the computational basis vectors $|0\rangle, |1\rangle, \cdots, |2^n - 1\rangle$ associated with an n-qubit register is described by the following transformation:

$$|j\rangle \longrightarrow \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i jk/2^n} |k\rangle,\ 0 \leq j \leq 2^n - 1. \tag{15.17}$$

However, the essential advantage of quantum computation over classical computation is that the quantum mechanical principle of superposition of states allows all possible inputs to be processed at the same time. Consequently, if the quantum register is in an arbitrary superposition of the basis vectors $\sum_{j=0}^{2^n-1} x_j |j\rangle$, then the quantum Fourier transform will rotate this state into another superposition of the basis vectors $\sum_{k=0}^{2^n-1} y_k |k\rangle$, in which the output amplitudes y_k are the classical discrete Fourier transform of the input amplitudes x_j. Classically, we can compute the numbers y_k from x_j using $\Theta(2^{2n})$ elementary arithmetic operations in a straightforward manner and in $\Theta(n2^n)$ operations by using the Fast Fourier Transform algorithm [32].

In contrast, a circuit implementing the quantum Fourier transform requires only $O(n^2)$ elementary quantum gates, as proved by Coppersmith [33]. Such a circuit can be easily derived if Eq. (15.17) is rewritten as a tensor product of the n qubits involved:

$$|j_1 \cdots j_n\rangle \longrightarrow \frac{(|0\rangle + e^{2\pi i 0.j_n}|1\rangle) \otimes (|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i 0.j_1 \cdots j_n}|1\rangle)}{2^{n/2}}. \tag{15.18}$$

using the binary representation $j_1 j_2 \cdots j_n$ of j and binary fractions in the exponents (for full details see [17]).

Note that each Fourier transformed qubit is in a balanced superposition of $|0\rangle$ and $|1\rangle$. These qubits differ from one another only in the relative phase between the $|0\rangle$ and the $|1\rangle$ components. For the first qubit in the tensor product, j_n will introduce a phase shift of 0 or π, depending on whether its value is 0 or 1, respectively. The phase of the second qubit is determined (controlled) by both j_n and j_{n-1}. It can amount to $\pi + \pi/2$, provided j_{n-1} and j_n are both 1. This dependency on the values of all the previous qubits continues up to (and including) the last term in the tensor product. When $|j_1\rangle$ gets Fourier transformed, the coefficient of $|1\rangle$ in the superposition involves all the digits in the binary expansion of j.

In the case of each qubit, the 0 or π phase induced by its own binary value is implemented through a Hadamard gate. The dependency on the previous qubits is reflected in the use of controlled phase shifts, as depicted in Fig. 15.7. In the figure, H denotes the Hadamard transformation

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{15.19}$$

while the gate R_k implements a $\pi/2^{k-1}$ phase shift of the $|1\rangle$ component, according to the unitary transformation

$$R_k \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix}. \tag{15.20}$$

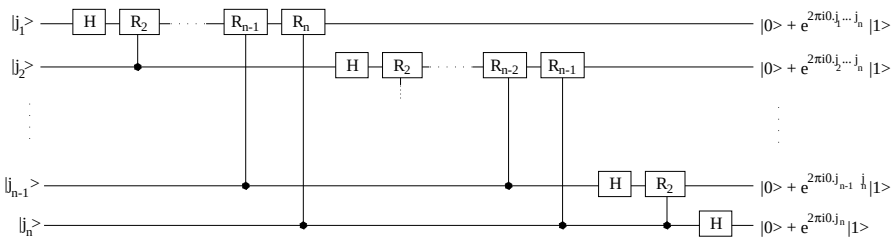

Fig. 15.7 Quantum circuit performing the discrete Fourier transform. The final swapping of qubits was omitted, for simplicity.

15.4.1.1 Rank-varying Complexity

Computing the quantum Fourier transform and its inverse can be viewed as examples of algorithms with rank-varying complexity. According to the quantum circuit above, we need n Hadamard gates and $n-1+n-2+\cdots+1$ conditional rotations, for a total of $n(n+1)/2$ gates required to compute the Fourier transform on n qubits.

But this total amount of work is not evenly distributed over the n qubits. The number of gates a qubit needs to be passed through is in inverse relation with its *rank*. $|j_1\rangle$ is subjected to n elementary quantum gates, $n-1$ elementary unitary transformations are applied to $|j_2\rangle$, and so on, until $|j_n\rangle$, which needs only one basic operation.

If we break down the quantum Fourier transform algorithm into n steps (one for each qubit involved), then its complexity varies with each step. Starting with $|j_1\rangle$, the time needed to complete each step decreases over time. Since the rank of each step dictates its complexity, the circuit implementing the quantum Fourier transform is an example of a rank-varying complexity algorithm.

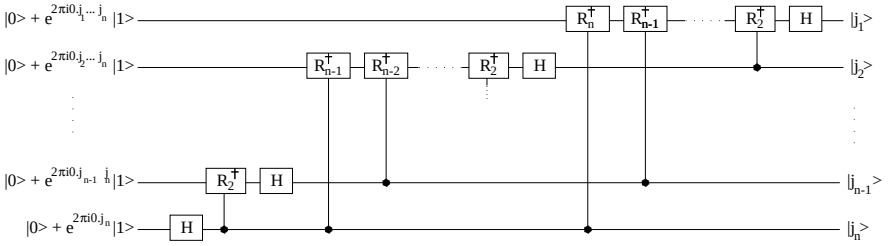

Fig. 15.8 Quantum circuit performing the inverse Fourier transform.

Naturally, the computation of the inverse quantum Fourier transform can also be decomposed into steps of varying complexity. Reversing each gate in Fig. 15.7 gives us an efficient quantum circuit (depicted in Fig. 15.8) for performing the inverse Fourier transform. Note that the Hadamard gate is its own inverse and $R_k^\dagger$ denotes the conjugate transpose of R_k:

$$R_k^\dagger \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{-2\pi i/2^k} \end{bmatrix}. \tag{15.21}$$

Getting back to the original $|j_1 j_2 \cdots j_n\rangle$ from its Fourier transformed expression has a certain particularity however. Because of the interdependencies introduced by the controlled rotations, the procedure must start by computing $|j_n\rangle$ and then work its way up to $|j_1\rangle$. The value of $|j_n\rangle$ is needed in the computation of $|j_{n-1}\rangle$. Both $|j_n\rangle$ and $|j_{n-1}\rangle$ are required in order to obtain $|j_{n-2}\rangle$. Finally, the value of all the higher rank bits are used to determine $|j_1\rangle$ precisely. Thus, computing the inverse Fourier transform by the quantum circuit illustrated in Fig. 15.8 is a procedure the complexity of whose steps increases with their rank.

Certainly, the fact that the total amount of operations (work) is not evenly distributed over the steps composing a certain algorithm does not change the overall complexity of the algorithm in any way. But the study of computations that can be characterized as having rank-varying complexity is important especially in the field of parallel computing. Operations pertaining to the same step, or belonging to distinct steps may be executed in parallel, leading to an important reduction in the overall running time of the respective algorithm. In the particular case of the QFT, the transformation of the first qubit has the highest computational complexity.

However, the use of an appropriate parallel architecture allows us to complete the entire computation during the n time units required just for the first qubit. Since the solution we describe can be characterized as a parallelization of the semiclassical solution due to Griffiths and Niu [34], we analyze the advantages offered by the former with respect to the performance of the latter.

15.4.1.2 Semiclassical (Sequential) Solution

Although the circuits for computing the quantum Fourier transform and its inverse are efficient in terms of the total number of gates employed, the majority of these gates operate on two qubits. This makes a practical implementation difficult, since arranging for one qubit to influence another in a desired way is far greater a challenge than evolving a single-qubit closed quantum system in accordance with any unitary transformation.

A method to replace all the two-qubit gates in the circuit performing the quantum Fourier transform by a smaller number of one-qubit gates controlled by classical signals has been developed by Griffiths and Niu [34] under the assumption that a measurement of the quantum register follows the application of the QFT, as it is usually the case, including in Shor's factoring quantum algorithm. Their approach takes advantage of the fact that the roles of the control and target qubits in any of the two-qubit gates required to carry on the computation of the quantum Fourier transform are interchangeable. Consequently, the quantum circuit in Fig. 15.7 is equivalent to the one depicted in Fig. 15.9 (for inputs restricted to four qubits).

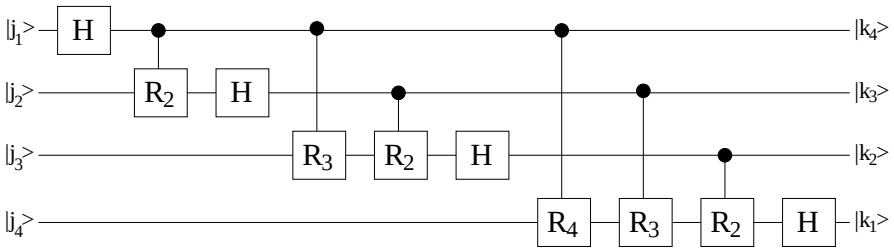

Fig. 15.9 Alternative arrangement of gates in the circuit performing the quantum Fourier transform. The roles of the control and target qubits in the controlled phase shift gates can be switched.

Note that, from this new perspective, the computation of the quantum Fourier transform appears to be a procedure whose steps are of increasing complexity. However, under the assumption that the Fourier transform is immediately followed by a quantum measurement, the complexity of each step in the computation can be made constant. Since a control qubit enters and leaves a two-qubit gate unchanged, it follows that the top qubit in Fig. 15.9 yields the same result regardless of whether it is measured as it exits the circuit or immediately after undergoing the Hadamard transform. In the latter case, the result of the measurement can be used to deter-

mine the phase shift that needs to be applied on the second qubit, before it too is subjected to a Hadamard transform and then measured. The phase computed for the second qubit together with the result of the second measurement are passed down as classical inputs for the rotation applied to the third qubit.

The computation proceeds in this manner all the way down to the last qubit, with a phase rotation, a Hadamard gate and a measurement being performed at each step. The process is illustrated in Fig. 15.10, where double lines have been used to denote a classical signal, according to the usual convention. Although the phase shift applied to each qubit is considered a single operation, conceptually, it is a combination of the gates depicted in the corresponding box, with each component being applied only if the controlling qubit was measured as 1.

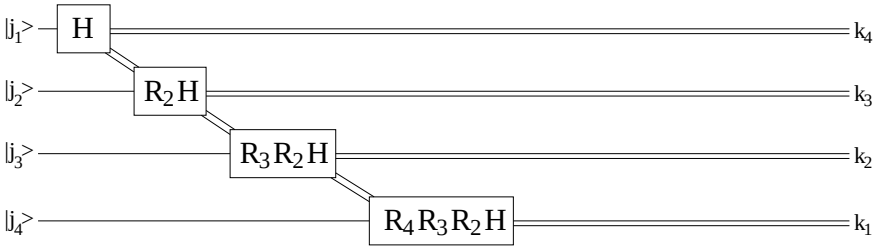

Fig. 15.10 Semiclassical circuit for computing the quantum Fourier transform. Single lines convey quantum information, while double lines carry classical information.

Example:

Here is an example of how the outcome of measurements determines the phase rotation that will be applied to subsequent qubits. If the top qubit in Fig. 15.10 yields a 1 ($k_4 = 1$) when measured, then the second qubit undergoes a $\pi/2$ phase shift before the Hadamard gate and then it is measured. Suppose now that the outcome of this measurement is a 0 ($k_3 = 0$). Then the third qubit is phase shifted by

$$k_4 \frac{\pi}{4} + k_3 \frac{\pi}{2} = \frac{\pi}{4} \tag{15.22}$$

and then the Hadamard gate is applied. Again, without loss of generality, let the measurement yield a 1 ($k_2 = 1$). Then the phase shift applied to the bottom qubit is

$$k_4 \frac{\pi}{8} + k_3 \frac{\pi}{4} + k_2 \frac{\pi}{2} = \frac{5\pi}{8}. \tag{15.23}$$

This semiclassical approach to computing the quantum Fourier transform achieves optimality in terms of the number of elementary unitary transformations that have to be applied. It also has the important advantage of employing only quantum transformations acting on a single qubit at a time. However, there is still room for improve-

ment, as the total time needed to complete the computation can be further squeezed down if parallelism is brought into play. In the next section we show how a quantum pipeline architecture is able to speed up the computation of the Fourier transform.

15.4.1.3 Parallel Approach

The solution developed in [34] to reduce the complexity of the quantum Fourier transform envisages a purely sequential approach, which is motivated by the same data dependency that causes the complexity of a step to vary with its rank. Nevertheless, there is a certain degree of parallelism that is worth exploiting in the computation of the quantum Fourier transform (or its inverse) in order to minimize the overall running time.

Our parallel approach is based on the observation that once a qubit has been measured, all phase shift gates classically controlled by the outcome of that measurement can be applied in parallel. The arrangement, again for just four qubits, is shown in Fig. 15.11. The one-qubit gates are ordered into a linear array having a Hadamard transform at the top and followed by a $\pi/2$ phase shift gate. The phase shift induced by any other gate down the array is just half the rotation performed by the immediately preceding gate.

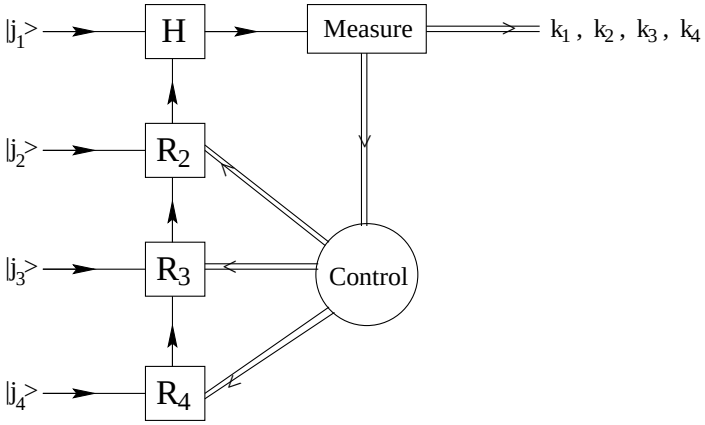

Fig. 15.11 Quantum pipeline array for computing the Fourier transform. The input is quantum, but the output is classical. At each step, qubits move one position up in the array.

This architecture allows R_2, R_3 and R_4 to be performed in parallel during the first cycle. Since each phase shift gate acts on a different qubit, they can all be applied simultaneously, if the top qubit yielded a 1 upon measurement. In the second cycle, each qubit in the array travels up one position, except of course for the top one, which has already been measured. Now, depending on the outcome of the second measurement, R_2 and R_3 can be simultaneously effected on the corresponding qubits. In the third cycle, only R_2 is needed and only if the control is 1. The com-

putation ends with the last qubit reaching the Hadamard gate and being measured afterwards. A formal description of the procedure, in the general case, is given as Algorithm **Parallel_Quantum_Fourier_Transform**.

Algorithm 15.1 Parallel_Quantum_Fourier_Transform

Input: $|j_1 j_2 \cdots j_n\rangle$
Output: $k_1 k_2 \cdots k_n$

for $i = 1$ to n **do**
  $|j_i\rangle \longleftarrow H|j_i\rangle$;
  Measure $|j_i\rangle$ as k_{n-i+1};
  if $k_{n-i+1} = 1$ **then**
    for $l = 2$ to $n-i+1$ in parallel **do**
      $|j_{i+l-1}\rangle \longleftarrow R_l|j_{i+l-1}\rangle$;
      $|j_{i+l-1}\rangle$ moves one position up in the array
    end for
  end if
end for

In the worst case, when all qubits are measured as 1, there is no difference between the parallel algorithm outlined above and the sequential solution envisaged by Griffiths and Niu [34] with respect to the overall running time. Assuming, for analysis purposes, that measuring a qubit, applying a phase shift, and performing a Hadamard transformation, each takes one time unit, then the total time necessary to complete the Fourier transform on a quantum register with n qubits is $3n-1$, as the top qubit in both the sequential circuit of Fig. 15.10 and the parallel circuit of Fig. 15.11 does not require a phase shift.

However, in the average case, some of the classical signals controlling the array of phase shift gates in Fig. 15.11 will have been observed as 0, meaning that no phase shifts have to be performed during those respective cycles. In contrast, the sequential solution depicted in Fig. 15.10 requires the application of a phase shift at every step following the first measurement with outcome 1. If the expected probability of a measurement yielding 0 equals the expected probability to observe a 1 following a measurement, then the running time of the parallel solution is shorter than the sequential running time by a difference proportional to the time it takes to effect a number of $O(n)$ phase shift gates, where n is the size of the input register.

The difference between the sequential running time and the parallel running time is maximum when $|j_1\rangle$ is measured as 1 and all the other qubits are observed in the state 0. In this case, the circuit in Fig. 15.10 still performs $n-1$ phase shifts, for a total running time of $3n-1$ time units, while the circuit in Fig. 15.11 executes all $n-1$ phase shifts in parallel during the first cycle, thus completing the computation in $2n+1$ time units.

The second advantage of the parallel approach is that the phase shift gates that need to be applied during the computation are known at the outset, making it easy to set them up beforehand in order to form the required linear array architecture. In other words, regardless of the initial quantum state of the register on which the

QFT is to be performed, the first gate in the linear array (top gate in Fig. 15.11) will always perform a Hadamard gate, the second gate always performs a $\pi/2$ phase shift, the third gate is "hardwired" to effect a $\pi/4$ phase shift and so on. The systolic mode of operation of the quantum array compensates for the fixed characteristics of each gate, the qubits traversing the array to undergo a specific quantum evolution at each node. In the current context, the attribute "systolic" describes the rhythmic mode in which data travel through the array of gates, much like blood does through the circulatory system.

In the sequential approach, on the other hand, the phase shift applied to each qubit is not known at the outset, as it is computed on the fly based on the information about the measurements performed so far and transmitted as classical signals. This means that the gates effecting the necessary phase shifts in the semiclassical approach of Griffiths and Niu [34] have to be "programmed" or adjusted during the computation, in order to accommodate a discrete set of possible values for the phase shift.

In the example given at the end of previous section, the phase shift applied to the bottom qubit is $5\pi/8$ because the previous measurements yielded $k_4 = 1$, $k_3 = 0$ and $k_2 = 1$. But the phase shift could have been $7\pi/8$ if all the measurements yielded a 1 or just $\pi/8$ if $k_4 = 1$ and $k_3 = k_2 = 0$. Therefore, we don't know at the outset how to "set" the quantum gates responsible with the phase shift performed on each qubit, as this information becomes available only during the computation, depending on the probabilistic results of the measurements. Technologically, this is more difficult to implement than a linear array of gates whose characteristics are fixed for any possible course of the computation.

The semiclassical Fourier transform and its parallelization are applicable to those quantum computations in which the Fourier transform immediately precedes a measurement of the qubits involved in the computation, like in Shor's algorithms for factoring integers and computing discrete logarithms [9]. Furthermore, the quantum systolic array architecture works equally fine if the input is already classical, in which case the restriction to measure the qubits after applying the Fourier transform can be lifted altogether.

When $j_1, j_2, \cdots, j_n$ are classical bits, the topology of the circuit in Fig. 15.11 remains unchanged, except that no measurements are performed and the flow of data through the linear array is reversed, as shown in Fig. 15.12. As more data are fed into the linear array through the Hadamard gate, after having "controlled" the parallel execution of a set of phase shifts, the computational complexity of each step increases with its rank. When j_1 enters the array, only the Hadamard gate is active, but with each consecutive step, a new gate down the array joins the ones above it to operate on the qubits traversing the array. Because these gates operate in parallel, the execution time of each step is maintained constant. Also note that, in this case, all outputs are simultaneously obtained during the last step of the computation.

The overall parallel running time, in the worst case, is therefore $2n-1$ time units, as there are no measurements to perform. The worst case occurs when $j_2, j_3, \ldots, j_n$ have all value 1. For all other inputs (that is, when at least one of $j_2, j_3, \ldots, j_n$ is 0), the parallel running time is smaller than the time needed to complete the com-

putation in a purely sequential manner, where each qubit is dealt with one after the other, in decreasing order of their ranks.

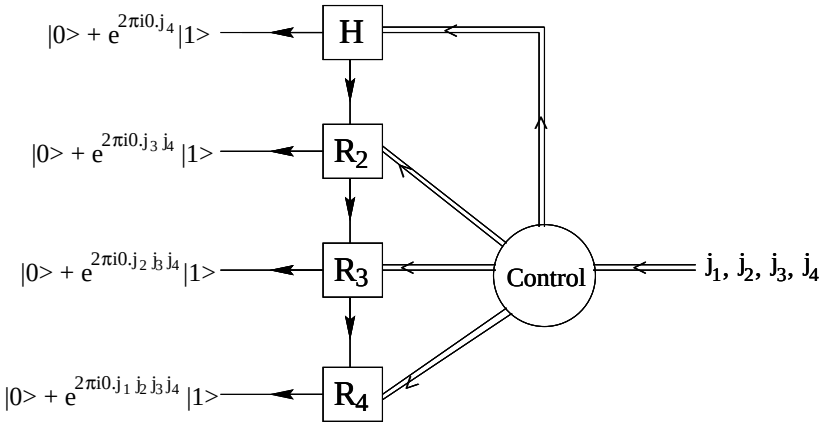

Fig. 15.12 Quantum pipeline array for computing the Fourier transform on classical inputs. The output is now quantum and the flow of qubits in the array is downwards.

Quantum algorithms employ the Fourier transform in order to create an interference among the terms in a superposition. From this point of view, the quantum Fourier transform offers little advantage, if any, when applied to a classical input. However, the situation is different for quantum cryptography. Distributing classical keys through quantum means is a procedure that may use the quantum Fourier transform and its inverse as encoding and decoding algorithms to protect vital information while in transit [35].

Naturally, the parallel approach detailed in this section for the computation of the direct Fourier transform is also applicable, with the same results, to the circuit in Fig. 15.8, performing the inverse Fourier transform. The difference in time complexity between the sequential approach and the parallel one, although seemingly insignificant from a theoretical perspective, may prove essential under practical considerations, as we show in our next example.

15.4.2 Quantum Decoherence

Qubits are fragile entities and one of the major challenges in building a practical quantum computer is to find a physical realization that would allow us to complete a computation before the quantum states we are working with become seriously affected by quantum errors. In an ideal setting, we evolve our qubits in perfect isolation from the outside world. But any practical implementation of a quantum computation will be affected by the interactions taking place between our system and the environment. These interactions cause quantum information to leak out into

the environment, leading to errors in our qubits. Different types of errors may affect an ongoing computation in different ways, but *quantum decoherence*, as defined below, usually occurs extremely rapidly and can seriously interfere with computing the quantum Fourier transform and its inverse.

In the context of a quantum key distribution protocol [35], consider the task of recovering the original (classical) bit string $j = j_1 j_2 \cdots j_n$ from its quantum Fourier transformed form. The circuit performing this computation (see Fig. 15.8) takes as input n qubits. The state of each qubit can be described by the following general equation:

$$|\psi_k\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{e^{i\theta_k}}{\sqrt{2}}|1\rangle,\ 1 \le k \le n \tag{15.24}$$

where the relative phase θ_k, characterizing the qubit of rank k, depends on the values of bits $j_k, j_{k+1}, \cdots, j_n$. The corresponding density operator is given by

$$\rho_k = |\psi_k\rangle\langle\psi_k| = \frac{1}{2}|0\rangle\langle 0| + \frac{e^{-i\theta_k}}{2}|0\rangle\langle 1| + \frac{e^{i\theta_k}}{2}|1\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|, \tag{15.25}$$

or in matrix form

$$\rho_k = \frac{1}{2}\begin{bmatrix} 1 & e^{-i\theta_k} \\ e^{i\theta_k} & 1 \end{bmatrix}. \tag{15.26}$$

The diagonal elements (or the *populations*) measure the probabilities that the qubit is in state $|0\rangle$ or $|1\rangle$, while the off-diagonal components (the *coherences*) measure the amount of interference between $|0\rangle$ and $|1\rangle$ [36]. Decoherence then, resulting from interactions with the environment, causes the off-diagonal elements to disappear. Since that is where the whole information carried by a qubit is stored, the input qubits for computing the inverse Fourier transform are very sensitive to decoherence. When they become entangled with the environment, the interference brought about by the Hadamard gate is no longer possible, as the system becomes effectively a statistical mixture. In other words, decoherence makes a quantum system behave like a classical one.

Naturally, this process is not instantaneous, but it usually occurs extremely rapidly, subject to how well a qubit can be isolated from its environment in a particular physical realization. Because of decoherence, we must obtain the values of $j_1, j_2, \cdots, j_n$ before time limit δ, after which the errors introduced by the coupling with the environment are too serious to still allow the recovery of the binary digits of j.

The precise value of δ will certainly depend on the particular way chosen to embody quantum information. If the qubits are implemented as trapped ions, then usually such a physical system is relatively well-isolated and decoherence is not a major concern. Nevertheless, other impediments make the design of a scalable quantum architecture a very challenging task. As a consequence, current experiments are only able to manipulate a handful of qubits. An illustrative example is a result from

2005 reporting the implementation of the semiclassical QFT on a "quantum register" composed of three beryllium ion qubits [12].

At the other end of the spectrum, we have attempts to implement the QFT, and quantum algorithms in general, using the well-established technology of nuclear magnetic resonance (NMR). In this case, decoherence plays a much more important role, directly affecting the accuracy of the results and placing a serious limitation on the scalability of this type of quantum computing architecture.

Experimental arrangements to compute the QFT on a 3-qubit NMR quantum information processor are reported by Weinstein et al. [37, 11]. Also, a 7-qubit experiment to implement the simplest meaningful instance of Shor's algorithm for factoring the number 15 uses the QFT as an important step of the computation [10]. Again we can see that scalability is the main obstacle towards building a practical quantum computer.

Of course, one of the possibilities to cope with the errors introduced by quantum decoherence is to use quantum codes to correct them. But here too, there are limitations. The more serious the errors are, the more ancillary qubits are required to correct them and consequently, the higher the probability of an error occurring in the correcting circuit itself. Therefore, we can only use that many auxiliary qubits to correct quantum errors before no advantage whatsoever in the accuracy of the solution is gained. From this point of view, parallelism offers a "clean" solution, avoiding the errors caused by quantum decoherence altogether and completing the computation before the entanglement with the surrounding environment seriously affects the ongoing quantum transformations.

The point we wish to make here is that when all other means have been used, a parallel approach may be the only way to further improve scalability by reducing the running time of the quantum algorithm and keep it below the decoherence threshold. In the particular case of computing the QFT and its inverse for cryptographic purposes, when δ lies between the parallel completion time and the sequential completion time, then the quantum pipeline array may be the only architecture capable to precisely recover all digits in the binary expansion of j. From a different perspective, the parallel solution allows for longer bit strings to be transmitted between the communicating parties, thus achieving better scalability over the purely sequential approach. With respect to scalability, it is also important to note that the parallel solution scales up linearly in the number of quantum gates employed, when the number of qubits on which the QFT is performed increases.

15.4.2.1 Time-varying Variables

We have already seen that the computation of the Fourier transform by quantum means belongs to the class of computations in which the complexity of each step depends on its rank. In addition, if we also take into consideration the properties of the computational environment, we are faced with the negative effects caused by quantum decoherence. Formally, the data stored in the quantum register before time limit δ is significantly different from what the same qubits encode after the deco-

herence threshold δ. The coupling between our qubits and their surrounding environment effectively places a hard deadline on the computation. After this deadline, the input data (variables) will have changed and if the computation is not yet complete, it has inevitably failed. From this perspective, the computation of the quantum Fourier transform (whether direct or inverse) in the presence of decoherence is an example of the paradigm dealing with *time-varying variables*.

As we have demonstrated above, parallelism can help us cope with variables whose values change over time. The use of a parallel approach becomes critical when the solution to a certain problem must accommodate a deadline. In our case, quantum decoherence places an upper bound on the scalability of computing the quantum Fourier transform or its inverse, and the only chance to reach beyond that limit is through a parallel solution.

15.4.3 Quantum Error-correction

Parallel processing is often the best alternative to avoid quantum errors in general and not just decoherence. The following examples on correcting quantum errors using specialized *quantum codes* or via *symmetrization* clearly show this.

In the computation of the QFT and its inverse the complexity of each step evolves with its rank. The more steps are executed before the current one, the higher the computational resources required to complete it. In this section, we still focus on steps of variable complexity, but in this case the variation is *time driven* rather than *rank driven*. In other words, we can have a high computational complexity even for the first step, if we allow some time to pass before starting the computation. The amount of computational resources required to successfully carry out a certain step are directly proportional with the amount of time elapsed since the beginning of the computation. We illustrate this paradigm through the use of error-correcting codes employed to maintain a quantum computation error-free.

The laws of quantum mechanics prevent, in general, a direct application of the classical error-correction techniques. We cannot inspect (measure) at leisure the state of a quantum memory register to check whether an ongoing computation is not off track without the risk of altering the intended course of the computation. Moreover, because of the no-cloning theorem, quantum information cannot be amplified in the same way digital signals can. Correcting quantum errors certainly requires much more ingenuity than fixing classical bits, but the basic idea of using redundancy is still useful.

Like in the classical case, the information contained in a qubit is spread out over several qubits so that damage to any one of them will not influence the outcome of the computation. In the quantum case, though, the encoding of the logical qubit is achieved through the use of specific resources, by entangling the logical qubit with several ancilla qubits. In this way, the information in the state of the qubit to be protected is spread among the correlations characterizing an entangled state.

Paradoxically enough, entanglement with the environment can be fought back using quantum error-correcting codes based on entanglement [38].

15.4.3.1 Quantum Codes

The construction of all quantum error-correcting codes is based on the surprising, yet beautiful idea of *digitizing the errors*. Any possible error affecting a single qubit can be expressed as a linear combination of no errors (I), bit flip errors (X), phase errors (Z) and bit flip phase errors (Y), where I, X, Z and Y are the Pauli operators describing the effect of the respective errors. Generalizing to the case of a quantum register, an error can be written as $\sum_i e_i E_i$ for some error operators E_i and coefficients e_i. The error operators can be tensor products of the single-bit error transformations or more general multibit transformations. An error correcting code that can undo the effect of any error belonging to a set of correctable errors E_i will embed n data qubits (logical qubits) in $n+k$ code qubits (physical qubits). The joint state of the ensemble of code qubits is subject to an arbitrary error, mathematically expressed as a linear combination of the correctable error operators E_i.

To recover the original encoded state, a syndrome extraction operator has to be applied that uses some ancilla qubits to create a superposition of the error indices i corresponding to those correctable error operators E_i that have transformed the encoded state. Measuring only the ancilla qubits will collapse the superposition of errors, yielding only one index k. But because the ancilla qubits were entangled with the code qubits through the application of the syndrome extraction operator, the side effect of the measurement is that the corruption caused by all error transformations will be undone, save for the one corresponding to index k. Consequently, only one inverse error transformation is required in order to complete the recovery process. In essence, knowing how to deal with a set of fundamental error transformations allows us to tackle any linear combination of them by projecting it to one of the basis components. This process is referred to as *digitizing* or *discretizing* the errors.

Peter Shor's second major contribution to the advancement of quantum computation was the creation in 1995 of an algorithm that could correct any kind of error (amplitude and/or phase errors) affecting a single qubit in a 9-qubit code [39]. In a different approach, Steane studied the interference properties of multiple particle entangled states and managed to devise a shorter, 7-qubit code [40]. The number of qubits necessary for a perfect recovery from a single error was later squeezed down to a minimum of five [41, 42].

Naturally, in order to cope with more than one error at a time, it is necessary to use larger and more elaborate codes. The book of Nielsen and Chuang [17] offers a detailed treatment of quantum codes, explaining how ideas from classical linear codes can be used to construct large classes of quantum codes, as the Calderbank-Shor-Steane (CSS) codes [43, 44], or the stabilizer codes (also known as additive quantum codes), which are even more general than the CSS codes and are based on the stabilizer formalism developed by Gottesman [45].

The major drawback in using large and intricate quantum codes is that the corrective circuit itself is as much prone to errors as the quantum circuit responsible for the main computation. The more errors we are attempting to rectify, the more the complexity and length of the recovery procedure will increase (see [46] for some theoretical bounds on the relationship between the number of data qubits, the total number of entangled qubits and the maximal number of errors that can be tolerated). Thus, we can only increase the size of the error correction codes up to a certain cut-off point, past which no further gains in accuracy can be made.

One attempt to overcome this limitation are the *concatenated* codes. If a certain code uses n physical qubits to encode one logical qubit, a concatenated version of that code is obtained by further encoding each of the n qubits in another block of n. This hierarchical structure (tree) can be further expanded to accommodate as many levels as desired. By adding more levels of concatenation, the overall chance for an error can be made arbitrarily small, provided that the probability of an individual error is kept below a certain critical threshold [47]. Of course, the high cost of using concatenated codes lies in the exponential increase in the number of qubits with the number of levels added.

15.4.3.2 Time-varying Complexity

This short exposition of the various quantum error-correcting codes devised to maintain the coherence of fragile quantum states and to protect them from dissipative errors caused by spontaneous emissions, for example, clearly shows one thing. The more time it takes to complete a quantum computation, the more errors are introduced in the process, and consequently, the more time, number of ancilla qubits and higher complexity error-correcting schemes that need to be employed. Correcting quantum errors is an important task executed alongside the mainstream computation and its complexity is heavily dependent on time. Steps executed soon after the initialization of the quantum register will require none or low complexity recovery techniques, while steps executed long after the initialization time may require complicated schemes and heavy resources allocated to deal with quantum errors.

Again, parallelism can help avoid this increase in the complexity of the recovery procedure and ultimately ensure the success of the computation. If the steps of the algorithm are independent of one another and can be executed in any order, then the most straightforward application of parallelism is to execute all steps simultaneously and thus complete the computation before any serious errors can accumulate over time. In this way we try to avoid or elude quantum errors rather than deal with them. But parallelism, in the form of redundancy, can also be used to correct quantum errors.

15.4.3.3 Error Correction via Symmetrization

The technique called *error correction via symmetrization* [48, 49] is yet another example of how the duality of quantum-mechanical laws can be exploited for the benefit of quantum computation. Although the measurement postulate severely restricts us in recycling techniques from classical error correction, it can still offer conceptually new ways of achieving error correction that are simply unavailable to classical computers. Error correction via symmetrization relies on the projective effect of measurements to do the job. The technique uses n quantum computers, each performing the same computation. Provided no errors occur, the joint state of the n computers is a symmetric one, lying somewhere in the small symmetric subspace of the entire possible Hilbert space. Devising a clever measurement that projects the joint state back into the symmetric subspace should be able to undo possible errors, without even knowing what the error is.

To achieve this, the n quantum computers need to be carefully entangled with a set of ancilla qubits placed in a superposition representing all possible permutations of n objects. In this way, the computation can be performed over all permutations of the computers simultaneously. Then, by measuring the ancilla qubits, the joint state of the n computers can be projected back into just the symmetric computational subspace, without the errors being measured explicitly. Peres has shown that this technique is most appropriate for correcting several qubits that are slightly wrong, rather than correcting a single qubit that is terribly wrong [50]. Error correction via symmetrization can be applied repeatedly, at regular time intervals, to avoid the accumulation of large errors and continually project the computation back into its symmetric subspace.

No matter which parallel approach is employed, if the required number of quantum processing units is provided, then the algorithm is successful. Simulating the same solution on an insufficient number of quantum computers will lead to a gradual accumulation of the quantum errors up to the point where the results of the computation are compromised.

15.4.4 Quantum Distinguishability

The problem of distinguishing among entangled quantum states is a quantum mechanical instance of the interacting variables paradigm (Sect. 7.2.4 in Chapter 2). Suppose we have a fixed set of quantum states described using the usual Dirac notation $|\Psi_i\rangle$ $(1 \leq i \leq n)$ known to both Alice and Bob. Alice randomly chooses a state from the set and prepares a qubit (or set of qubits) in that particular state. She then gives the qubit(s) to Bob who is free to measure them in any way he likes, without applying any quantum operation on them (Bob lacks the power of a quantum computer). To be more specific, Bob can apply any kind of measurement on the qubit(s) and possibly process and/or interpret the classical information acquired through measurement, but he cannot manipulate quantum information using unitary

evolution. In the end, his task is to identify the index i of the state characterizing the qubit(s) Alice has given him. The only case in which a set of quantum states can be reliably (that is, 100% of the time) distinguished from one another is if they are pairwise orthogonal.

Now consider the case in which we try to distinguish among the four Bell states $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$, $\frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle$, $\frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle$, $\frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle$. by resorting only to direct quantum measurements (in other words, no quantum transformations are possible before a measurement). In these circumstances, any sequential approach (that is, measuring the qubits one after the other) will be of no help here, regardless of the basis in which the measurements are performed. By measuring the two qubits, in sequence, in the computational basis, Bob can distinguish the states $\frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle)$ from $\frac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle)$. He does this by checking if the outcomes of the two measurements are the same or not. But this kind of measurement makes it impossible to differentiate between $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and $\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$, or between $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ and $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$.

Alternatively, Bob can decide to perform his measurements in a different basis, like $(|+\rangle, |-\rangle)$, where the basis vectors are

$$|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle, \tag{15.27}$$

$$|-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle. \tag{15.28}$$

Due to the fact that

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} = \frac{|++\rangle + |--\rangle}{\sqrt{2}} \tag{15.29}$$

and

$$\frac{|00\rangle - |11\rangle}{\sqrt{2}} = \frac{|+-\rangle + |-+\rangle}{\sqrt{2}}, \tag{15.30}$$

Bob can now reliably distinguish the quantum state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ from $\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$. Indeed, if the two qubits yield identical outcomes when measured in this new basis, then we can assert with certainty that the state was not $\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$. Similarly, if the measurement outcomes for the qubits are different, the original state could not have been $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Unfortunately, in this new setup, the quantum states $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ become indistinguishable and the same is true about $\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$ and $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$.

The computational bases $(|0\rangle, |1\rangle)$ and $(|+\rangle, |-\rangle)$ are, respectively, the two extremities of an (theoretically) infinite number of choices for the basis relative to which the quantum measurements are to be performed. But even though the separation line between the four Bell states will drift with the choice of the basis vectors, the two extreme cases discussed above offer the best possible distinguishability.

Intuitively, this is due to the entanglement exhibited between the two qubits in all four states. As soon as the first qubit is measured (regardless of the basis), the superposition describing the entangled state collapses to the specific state consistent with the measurement result. In this process, some of the information originally encapsulated in the entangled state is irremediably lost. Consequently, measuring the second qubit cannot give a complete separation of the four EPR states. But the Bell states do form an orthonormal basis, which means that (at least theoretically) they can be distinguished by an appropriate quantum measurement. However, this measurement must be a *joint* measurement of both qubits simultaneously, in order to achieve the desired distinguishability.

15.4.4.1 Generalization

A more compact representation of the Bell basis is through a square matrix where each column is a vector describing one of the Bell states:

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{pmatrix} \tag{15.31}$$

The elements of each column are the amplitudes or proportions in which the computational basis states $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$ are present in the respective EPR state.

This scenario can be extended to ensembles of more than two qubits. The following matrix describes eight different entangled states that cannot be reliably distinguished unless a joint measurement of all three qubits involved is performed:

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \tag{15.32}$$

In general, for a quantum system composed of n qubits, one can define the following 2^n entangled states of the system:

$$\begin{gathered}\frac{1}{\sqrt{2}}(|000\cdots0\rangle \pm |111\cdots1\rangle)\\ \frac{1}{\sqrt{2}}(|000\cdots1\rangle \pm |111\cdots0\rangle)\\ \vdots\\ \frac{1}{\sqrt{2}}(|011\cdots1\rangle \pm |100\cdots0\rangle)\end{gathered} \tag{15.33}$$

These vectors form an orthonormal basis for the state space corresponding to the n-qubit system. The only chance to differentiate among these 2^n states using quantum measurement(s) is to observe the n qubits simultaneously, that is, perform a single joint measurement of the entire system. In the given context, *joint* is really just a synonym for *parallel*. Indeed, the device in charge of performing the joint measurement must posses the ability to "read" the information stored in each qubit, in parallel, in a perfectly synchronized manner. In this sense, at an abstract level, and just for the sake of offering a more intuitive understanding of the process, the measuring apparatus can be viewed as having n probes. With all probes operating in parallel, each probe can "peek" inside the state of one qubit, in a perfectly synchronous operation. The information gathered by the n probes is seen by the measuring device as a single, indivisible chunk of data, which is then interpreted to give one the 2^n entangled states as the measurement outcome.

From a mathematical (theoretical) point of view, such a measurement operator can be easily constructed by defining each of the 2^n states that are to be distinguished to be a projector associated with the measurement operation. We are well aware though, that a physical realization of this mathematical construction is extremely difficult, if not impossible to achieve in practice, with today's technology. Yet, if there is any hope to see a joint measurement performed in the future, then only a device operating in a parallel synchronous fashion on all n qubits (as explained above) would succeed.

It is perhaps worth emphasizing that if such a measurement cannot be applied then the desired distinguishability can no longer be achieved regardless of how many other measuring operations we are allowed to perform. In other words, even an infinite sequence of measurements touching at most $n-1$ qubits at the same time cannot equal a single joint measurement involving all n qubits.

Furthermore, with respect to the particular distinguishability problem that we have to solve, a single joint measurement capable of observing $n-1$ qubits simultaneously offers no advantage whatsoever over a sequence of $n-1$ consecutive *single* qubit measurements. This is due to the fact that an entangled state like

$$\frac{1}{\sqrt{2}}(|000\cdots0\rangle + |111\cdots1\rangle) \tag{15.34}$$

cannot be decomposed neither as a product of $n-1$ individual states nor as a product of two states (one describing a single qubit and the other describing the subsystem composed of the remaining $n-1$ qubits). Any other intermediate decomposition is also impossible.

Overall, our distinguishability problem can only be tackled successfully within a parallel approach, where we can measure all qubits simultaneously. Conceptually, distinguishing among entangled quantum states is a quantum example of measuring interdependent variables. In this particular quantum instance, the interdependence between variables takes the form of entanglement between qubits, the phenomenon ultimately responsible for making a parallel approach imperative. But not only measuring entangled states requires a parallel solution, quantum evolutions that have to maintain a certain entangled state may also resort to parallelism in order to achieve their goal. In our last example, we investigate entanglement as a global mathematical constraint that has to be satisfied throughout a quantum computation.

15.4.5 Transformations Obeying a Global Condition

Some computational problems require the transformation of a mathematical object in such a way that a property characterizing the original object is to be maintained at all times throughout the computation. This property is a global condition on the variables describing the input state and it must be obeyed at every intermediate step in the computation, as well as for the final state. Geometric flips, map recoloring and rewriting systems are three examples of transformations that can be constrained by a global mathematical condition [51].

Here, we show that some quantum transformations acting on entangled states may also be perceived as computations obeying a global mathematical constraint. Consider, for example, an ensemble of n qubits sharing the following entangled state:

$$\frac{1}{\sqrt{2}}|000\cdots0\rangle + \frac{1}{\sqrt{2}}|111\cdots1\rangle. \tag{15.35}$$

The entanglement characterizing the above state determines a strict correlation between the values observed in case of a measurement: either all qubits are detected in the state 0 or they are all seen as 1. Suppose that this correlation has to be maintained unaltered, regardless of the local transformations each of the qubits may undergo. Such a transformation may be the application of a *NOT* quantum gate to any of the qubits forming the ensemble. After such an event, the particular entangled state given in Eq. (15.35) is no longer preserved and as a consequence, the correlation between the qubits will be altered. The qubit whose state was "flipped" will be observed in the complementary state, with respect to the other qubits. The global mathematical constraint is no longer satisfied.

Parallelism can once again make the difference and help maintain the required entangled state. If, at the same time one or more of the qubits are "flipped", we

also apply a *NOT* gate to all remaining qubits, simultaneously, then the final state coincides with the initial one. In this way, although the value of each qubit has been switched, the correlation we were interested to maintain remains the same. Also note that any attempt to act on less than n qubits simultaneously is doomed to failure.

The state given in Eq. (15.35) is not the only one with this property. Any entangled state from the orthonormal basis set (15.33) could have been used in the example presented above. The correlation among the qubits would have been different, but the fact that applying a *NOT* gate, in parallel, to all qubits does not change the quantum state of the ensemble is true for each entangled state appearing in system (15.33).

Perhaps the scenario described above can be extended to other quantum transformations beside the *NOT* gate. Another, perhaps more interesting generalization would be a quantum computation that has to maintain entanglement as a generic, global mathematical constraint and not a specific type of entanglement with a particular correlation among the qubits involved. Such a computation would allow entanglement to change form, but the mathematical definition of entanglement would still have to be obeyed at each step, with each transformation.

15.5 Looking Ahead

In this final chapter, we have reviewed some of the most promising computing paradigms that have emerged from our relentless quest to make computation more efficient in terms of speed or accuracy of the result obtained. A special attention was given to the quantum computing paradigm, which still has the potential to radically transform the field of computer science, provided that experimentalists will eventually find a viable design for a practical quantum computer. However, the essential observation that can be formulated seeing all these efforts is that parallel processing and the future of computation go hand in hand. Whether we are discussing conventional computing architectures (like clusters and grids) or more exotic proposals (DNA computing, quantum computing, etc.), they all draw their power from some form of parallelism and they all can be considered as massively parallel computing devices.

Moreover, the polymorphic nature of parallelism becomes evident by surveying all these different ways to envisage computation. In most paradigms, parallelism refers to a large (and sometimes huge) number of processing elements operating simultaneously, whether these are conventional electronic processors or DNA molecules. But in quantum information processing, for instance, massive parallelism is an attribute characterizing the "software" rather than the "hardware", since it refers to how a huge computational space can be explored in parallel by manipulating only a relatively small number of qubits. Furthermore, quantum computing is also the perfect example of how different instances of parallelism can be encountered within the same computational paradigm, as we showed in this chapter.

It is difficult to foretell, at this point, what will be the dominant computing technology in a few decades time. It could be one of the alternatives described at the beginning of the chapter or a hybrid solution involving a combination of two or more paradigms. Yet another possibility would be the emergence of a totally new and revolutionary way to perform computations. What we can say for sure, though, is that parallel processing has re-affirmed its importance with every novel model of computation proposed over time and it will continue to do so. Its capital role in the theory of computing will not change, regardless of the physical layer used to represent and manipulate information. In this respect, the universal attribute of parallelism becomes apparent, ensuring its perennity.

References

1. G. Păun, G. Rozenberg, A. Salomaa, DNA Computing - New Computing Paradigms, Springer, 1998.
2. L. Adleman, Molecular computation of solutions to combinatorial problems, Science 266 (1994) 1021–1024.
3. R. J. Lipton, DNA solution of hard computational problems, Science 268 (5210) (1995) 542–545.
4. W.-L. Chang, M. Guo, J. Wu, Solving the independent-set problem in a DNA-based supercomputer model, Parallel Processing Letters 15 (4) (2005) 469–479.
5. G. Păun, Computing with membranes, Journal of Computer and System Sciences 61 (1) (2000) 108–143.
6. M. Pérez-Jiménez, A. Riscos-Núñez, A linear solution for the knapsack problem using active membranes, in: Membrane Computing. Lecture Notes in Computer Science, Vol. 2933, Springer, 2004, pp. 250–268.
7. G. Păun, P systems with active membranes: Attacking NP-complete problems, Journal of Automata, Languages, Combinatorics 6 (1) (2001) 5–90.
8. C. Zandron, C. Ferretti, G. Mauri, Solving NP-complete problems using P systems with active membranes, in: I. Antoniou, C. Calude, M. Dinneen (Eds.), Unconventional Models of Computation, Springer, London, 2000, pp. 289–301, dISCO - Universita di Milano-Bicocca, Italy.
9. P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, Special issue on Quantum Computation of the SIAM Journal on Computing 26 (5) (1997) 1484–1509.
10. L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, I. L. Chuang, Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance, Nature 414 (2001) 829–938.
11. Y. S. Weinstein, et al., Quantum process tomography of the quantum fourier transform, Journal of Chemical Physics 121 (13) (2004) 6117–6133, http://arxiv.org/abs/quant-ph/0406239v1.
12. J. Chiaverini, et al., Implementation of the semiclassical quantum fourier transform in a scalable system, Science 308 (5724) (2005) 997–1000.
13. A. Adamatzky, B. D. L. Costello, T. Asai, Reaction-Diffusion Computers, Elsevier, 2005.
14. R. Fraser, S. G. Akl, Accelerating machines: a review, International Journal of Parallel, Emergent and Distributed Systems 23 (1) (2008) 81–104.
15. N. D. Mermin, From Cbits to Qbits: Teaching computer scientists quantum mechanics, http://arxiv.org/abs/quant-ph/0207118 (July 2002).
16. E. Rieffel, W. Polak, An introduction to quantum computing for non-physicists, ACM Computing Surveys 32 (3) (2000) 300–335.

17. M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2000.
18. M. Hirvensalo, Quantum Computing, Springer-Verlag, 2001.
19. A. Berthiaume, Quantum computation, in: L. A. Hemaspaandra, A. L. Selman (Eds.), Complexity Theory Retrospective II, Springer-Verlag, New York, 1997, pp. 23–51.
20. R. Feynman, R. B. Leighton, M. Sands, The Feynman Lectures on Physics, Vol. III, Addison-Wesley, Reading, Mass., 1965.
21. C. Santori, et al., Indistinguishable photons from a single-photon device, Nature 419 (2002) 594–597.
22. E. H. Knill, R. Laflamme, G. J. Milburn, A scheme for efficient quantum computation with linear optics, Nature 409 (2001) 46–52.
23. P. Dirac, The Principles of Quantum Mechanics, 4th Edition, Oxford University Press, 1958.
24. E. W. Weisstein, et al., Bloch sphere, From *MathWorld*–A Wolfram Web Resource, `http://mathworld.wolfram.com/BlochSphere.html`.
25. W. K. Wootters, W. H. Zurek, A single quantum cannot be cloned, Nature 299 (1982) 802–803.
26. A. Barenco, A universal two-bit gate for quantum computation, Proceedings of the Royal Society of London A 449 (1995) 679–683.
27. D. DiVincenzo, Two-bit gates are universal for quantum computation, Physical Review A 51 (1995) 1015–1022.
28. E. Schrödinger, Discussion of probability relations between separated systems, Proceedings of the Cambridge Philosophical Society 31 (1935) 555–563.
29. A. Einstein, B. Podolsky, N. Rosen, Can quantum-mechanical description of physical reality be considered complete?, Physical Review 47 (1935) 777–780.
30. A. Berthiaume, G. Brassard, Oracle quantum computing, Journal of Modern Optics 41 (12) (1994) 2521–2535.
31. S. Robinson, Emerging insights on limitations of quantum computing shape quest for fast algorithms, SIAM News 36 (1).
32. J. W. Cooley, J. Tukey, An algorithm for the machine calculation of complex fourier series, Mathematics of Computation 19 (1965) 297–301.
33. D. Coppersmith, An approximate fourier transform useful in quantum factoring, Tech. Rep. RC19642, IBM (1994).
34. R. Griffiths, C.-S. Niu, Semiclassical Fourier transform for quantum computation, Physical Review Letters 76 (1996) 3228–3231.
35. M. Nagy, S. G. Akl, S. Kershaw, Key distribution based on the quantum Fourier transform, in: Proceedings of the International Conference on Security and Cryptography (SECRYPT 2008), Porto, Portugal, 2008, pp. 263–269.
36. C. Cohen-Tannoudji, B. Diu, F. Laloe, Quantum Mechanics, Vol. 1 and 2, Wiley, New York, 1977.
37. Y. S. Weinstein, et al., Implementation of the quantum fourier transform, Physical Review Letters 86 (9) (2001) 1889–1891.
38. J. Preskill, Fault-tolerant quantum computation, in: H.-K. Lo, S. Popescu, T. Spiller (Eds.), Introduction to quantum computation and information, World Scientific, 1998, pp. 213–269, `http://xxx.lanl.gov/abs/quant-ph/9712048`.
39. P. W. Shor, Scheme for reducing decoherence in quantum computer memory, Physical Review A 52 (1995) 2493–2496.
40. A. M. Steane, Error correcting codes in quantum theory, Physical Review Letters 77 (5) (1996) 793–797.
41. C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, W. K. Wootters, Mixed state entanglement and quantum error correction, Physical Review A 54 (1996) 3824–3851, `http://arxiv.org/abs/quant-ph/9604024`.
42. R. Laflamme, C. Miquel, J. P. Paz, W. H. Zurek, Perfect quantum error correction code, `http://arxiv.org/abs/quant-ph/9602019` (February 1996).
43. A. R. Calderbank, P. W. Shor, Good quantum error-correcting codes exist, Physical Review A 54 (2) (1996) 1098–1106, `http://arxiv.org/abs/quant-ph/9512032`.

44. A. M. Steane, Multiple particle interference and quantum error correction, Proceedings of the Royal Society of London A 452 (1996) 2551–2576.
45. D. Gottesman, Class of quantum error-correcting codes saturating the quantum hamming bound, Physical Review A 54 (1996) 1862–1868, http://arxiv.org/abs/quant-ph/9604038.
46. A. Ekert, C. Macchiavello, Quantum error correction for communication, Physical Review Letters 77 (1996) 2585–2588.
47. J. Preskill, Reliable quantum computers, Proceedings of the Royal Society of London A 454 (1998) 385–410, http://xxx.lanl.gov/abs/quant-ph/9705031.
48. A. Berthiaume, D. Deutsch, R. Jozsa, The stabilization of quantum computation, in: Proceedings of the Workshop on Physics and Computation: PhysComp '94, IEEE Computer Society Press, Los Alamitos, CA, 1994, 1994, pp. 60–62.
49. A. Barenco, A. Berthiaume, D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, Stabilization of quantum computations by symmetrization, http://xxx.lanl.gov/abs/quant-ph/9604028 (April 1996).
50. A. Peres, Error symmetrization in quantum computers, http://xxx.lanl.gov/abs/quant-ph/9605009 (May 1996).
51. S. G. Akl, Evolving computational systems, in: S. Rajasekaran, J. H. Reif (Eds.), Parallel Computing: Models, Algorithms, and Applications, CRC Press, 2007, a modified version is available as Technical Report No. 2006-526, School of Computing, Queen's University, Kingston, Ontario, Canada.