Measurement-based quantum computation with cluster states

Robert Raußendorf

München 2003

Messungsbasiertes Quantenrechnen mit Clusterzuständen

Dissertation
an der Fakultät für Physik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Robert Raußendorf
aus Dresden

München, den 20. Juni 2003

First referee: Dr. Hans-Jürgen Briegel
Second referee: Prof. Dr. Herbert Wagner
Day of exam: 06.10.2003

Contents

List of Figures

Zusammenfassung

In dieser Dissertation beschreiben wir den Einweg-Quantenrechner ($\mathrm{QC}_\mathcal{C}$), ein Schema zum universellen Quantenrechnen, das allein aus Einteilchenmessungen an einem hochgradig verschränkten Vielteilchenzustand, dem Clusterzustand, besteht. Wir beweisen die Universalität des $\mathrm{QC}_\mathcal{C}$, beschreiben das zugrunde liegende Rechnermodell und zeigen, dass der $\mathrm{QC}_\mathcal{C}$ fehlertolerantes Quantenrechnen erlaubt.

In Kapitel 2 zeigen wir, dass der $\mathrm{QC}_\mathcal{C}$ als ein Simulator quantenlogischer Netzwerke aufgefasst werden kann. Damit beweisen wir dessen Universalität und stellen den Zusammenhang zum Netzwerkmodel her, welches das verbreitete Model eines Quantenrechners darstellt. Wir weisen auch darauf hin, dass die Beschreibung des $\mathrm{QC}_\mathcal{C}$ als Netzwerksimulator nicht in jeder Hinsicht passend ist.

In Kapitel 3 leiten wir das dem Einweg-Quantenrechner zugrunde liegende Rechnermodell her. Es ist sehr verschieden vom Netzwerkmodell des Quantenrechners. Der $\mathrm{QC}_\mathcal{C}$ besitzt keinen Quanten-Input, keinen Quanten-Output und kein Quantenregister. Unitäre Quantengatter aus einem universellen Satz sind nicht die elementaren Bestandteile von $\mathrm{QC}_\mathcal{C}$-Quantenalgorithmen. Darüber hinaus sind die Messergebnisse aus den Einteilchenmessungen die einzige Information, die vom $\mathrm{QC}_\mathcal{C}$ verarbeitet wird, und somit existiert Informationsverarbeitung beim $\mathrm{QC}_\mathcal{C}$ nur auf klassischem Niveau. Dennoch arbeitet der $\mathrm{QC}_\mathcal{C}$ fundamental quantenmechanisch, da er den hochverschränkten Clusterzustand als zentrale physikalische Resource nutzt.

In Kapitel 4 zeigen wir, dass positive Fehlerschranken für das fehlertolerante Quantenrechnen mit dem $\mathrm{QC}_\mathcal{C}$ existieren. Desweiteren skizzieren wir das Konzept der Prüfsummen im Zusammenhang mit dem $\mathrm{QC}_\mathcal{C}$, das ein Element zukünfitiger praktikabler und zweckmäßiger Methoden für fehlertolerantes $\mathrm{QC}_\mathcal{C}$-Quantenrechnen werden kann.

Abstract

In this thesis we describe the one-way quantum computer ($QC_{\mathcal{C}}$), a scheme of universal quantum computation that consists entirely of one-qubit measurements on a highly entangled multi-particle state, the cluster state. We prove universality of the $QC_{\mathcal{C}}$, describe the underlying computational model and demonstrate that the $QC_{\mathcal{C}}$ can be operated fault-tolerantly.

In Chapter 2 we show that the $QC_{\mathcal{C}}$ can be regarded as a simulator of quantum logic networks. In this way, we give the universality proof and establish the link to the network model, the common model of quantum computation. We also indicate that the description of the $QC_{\mathcal{C}}$ as a network simulator is not adequate in every respect.

In Chapter 3 we derive the computational model underlying the one-way quantum computer, which is very different from the quantum logic network model. The $QC_{\mathcal{C}}$ has no quantum input, no quantum output and no quantum register, and the unitary gates from some universal set are not the elementary building blocks of $QC_{\mathcal{C}}$-quantum algorithms. Further, all information that is processed with the $QC_{\mathcal{C}}$ are the outcomes of one-qubit measurements and thus processing of information exists only at the classical level. The $QC_{\mathcal{C}}$ is nevertheless quantum mechanical as it uses a highly entangled cluster state as the central physical resource.

In Chapter 4 we show that there exist nonzero error thresholds for fault-tolerant quantum computation with the $QC_{\mathcal{C}}$. Further, we outline the concept of checksums in the context of the $QC_{\mathcal{C}}$ which may become an element in future practicable and adequate methods for fault-tolerant $QC_{\mathcal{C}}$-computation.

Chapter 1

Introduction

Quantum computation has come into the focus of physics and information science largely due to Peter Shor's discovery of a quantum algorithm for factoring large numbers [1]. This algorithm demonstrated that a quantum computer is capable of solving a problem for whose solution no efficient classical algorithm is known, namely to break the RSA crypto-system. It thus became apparent that a quantum computer, once it can be built, is something that needs to be reckoned with. The existence of the factoring algorithm also gives hope that quantum algorithms with equal power and more universal benefit can be found. Among the further applications of a quantum computer which have so far been envisioned are a data base with faster access to unsorted data [2], and a universal simulator of quantum systems [3, 4]. Shor's algorithm, one of the early quantum algorithms being described and certainly the most striking, may prove to be a profound trigger in leading the quantum computer from a mere "Gedankenexperiment" to a physical device.

Physics plays a fundamental role in computation. As David Deutsch writes in [5]: "[It is not] obvious *a priori* that any of the familiar recursive functions is in physical reality computable. The reason why we find it possible to construct, say, electronic calculators, and indeed why we can perform mental arithmetic, cannot be found in mathematics or logic. *The reason is that the laws of physics 'happen to' permit the existence of physical models for arithmetic* such as addition, subtraction and multiplication. If they did not, these familiar operations would be non-computable functions. We might still know *of* them and invoke them in mathematical proofs (which would presumably be called 'non-constructive') but we could not perform them.".

However, quantum computation is not merely an application of physics, in this case quantum mechanics. As draftsmen of quantum computers and quantum algorithms we may easily find ourselves quoting Richard Feynman: "One feels like Cavalieri must have felt calculating the volume of a pyramid before the invention of calculus."[1]. One is faced with questions like "What feature of quantum mechanics makes the quantum computer powerful?" and "What are the basic design principles for effective quantum algorithms?". Various explanations for the origin of the speedup in a quantum computer have been

[1]R.P. Feynman, on developing the path integral formalism, taken from: M. Kaku, *Quantum Field Theory*, Oxford University Press (1993).

proposed and construction techniques for quantum algorithms, such as period finding [1], amplitude amplification [2, 6], and quantum random walks [7, 8] have been identified. Nevertheless, it appears that the two questions –in which information science and physics are intertwined– are to a large extent still open. It will certainly require elements from both physics and information science to answer them.

Quantum information science (QI), with quantum communication, quantum computation and quantum information theory as its subfields, is at the junction between information science and physics. Well established and cherished concepts from both these disciplines contribute to the vocabulary of QI. For the present it very much seems, at least for quantum computation, that the identification of the appropriate terminology is still in progress.

This thesis is about the one-way quantum computer ($\mathrm{QC}_{\mathcal{C}}$) [9], a universal scheme of quantum computation. Therein, the entire quantum computation consists of a spatio-temporal pattern of one-qubit measurements on an entangled multi-qubit quantum state, the cluster state. This state forms a universal resource for quantum computation, i.e. any quantum circuit may be imprinted on the cluster state via the measurements. The result of the computation is derived from all the obtained measurement outcomes. In the process of the computation, all the entanglement of the cluster state is destroyed, and thus the state can be used only once. Therefore we call the scheme the "one-way quantum computer".

A proper quantum computer needs to be universal, scalable and fault-tolerant. Therefore, a considerable part of this thesis is devoted to demonstrating that the $\mathrm{QC}_{\mathcal{C}}$ meets these criteria. Specifically, we prove universality in Chapter 2 and fault-tolerance in Chapter 4. Scalability is discussed in Section 2.2.8. The resource cluster state can be created via a tunable Ising interaction in a single time step irrespective of the number of qubits involved. Also, the one-qubit measurements do not become more complicated if the size of the system is scaled up. The $\mathrm{QC}_{\mathcal{C}}$ avoids by construction all long-range qubit-selective interactions.

The $\mathrm{QC}_{\mathcal{C}}$ does not only fulfill the above essential requirements for quantum computation. It also turns out that the $\mathrm{QC}_{\mathcal{C}}$ is based upon a computational model [10, 11] in which the physics and the logic of quantum computation are clearly separated. This model is described in Chapter 3. It is very different from the network model, the most widely used model of quantum computation. To better illustrate the difference, let us first explain the network model and how it emerged.

Before Landauer investigated the question of physical irreversibility of logical operations in the 1960's, the commonly accepted opinion was that a computer operating at temperature T needs to dissipate at least an amount $\ln 2\, kT$ of energy per elementary act of operation processing, [12]. In the attempt [13] to prove this assertion he realized that it is not the processing of information which requires energy dissipation, but instead the erasure of information. Specifically, he demonstrated that the physical realization of a logically irreversible operation generates an amount of entropy equivalent to the erased information. Later it was realized by Charles Bennett that any computation could be performed logically reversible [14], such that no information erasure is required.

Beyond showing that logically irreversible operations can be avoided in classical compu-

tation, a proof of thermodynamic reversibility of the computational process requires some physically reasonable theoretical model in which the sequence of logically reversible operations can be shown to be thermodynamically reversible. Such a model [15] was constructed by Paul Benioff, who used a quantum system to implement reversible logic.

Benioff's model was, in spite of the fact that a quantum system was proposed for its realization, a classical model from the computational perspective, i.e. it could be simulated efficiently by a Turing machine. With his 'universal quantum simulator' [3] Feynman demonstrated that the use of a quantum system for a computation or a similar task may actually be an advantage. He pointed out that, as a consequence of the fact that the size of the state space increases exponentially with the number of particles, the simulation of quantum systems on classical computers is generally inefficient. On the other hand, quantum systems can be simulated efficiently by other quantum systems. David Deutsch went a significant step further and constructed a quantum version of a Turing machine [5] which could exert an arbitrary unitary transformation on an arbitrary state.

Deutsch also introduced the network model of quantum computation [16]. It emerged from the combination of classical reversible networks with the unitary evolution known in quantum mechanics. In this model, with some simplification, the process of computation consists of the initialization, processing and readout of the quantum register, the quantum counterpart of the register of a classical reversible computer. In the initialization, the quantum register is prepared in some (generally fixed) quantum state, the quantum input. Subsequently, this register state is acted upon by a unitary transformation consisting of a sequence of quantum gates, creating the quantum output. Finally, the output is read by measurements performed on the quantum register.

The Hilbert space of the quantum register usually comes with a natural tensor product structure. That is, the quantum register is composed of a number of d-level subsystems. Most commonly, two-level systems are chosen as these subsystems of the quantum register. They are, in the context of quantum information, called *quantum bits*, or, in short, *qubits*. The notion of the "qubit" makes the intertwining of information theory and physics very apparent. The qubit lives in a two-dimensional Hilbert space whose basis vectors are conveniently denoted as $|0\rangle$ and $|1\rangle$. They form the counterpart to the states of a classical bit, 0 and 1. While the classical bit may only be in either of the two states 0 or 1, for the quantum bit any linear combination $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, with $\alpha, \beta \in \mathbb{C}$, $|\alpha|^2 + |\beta|^2 = 1$, represents a legitimate state. In this way, the qubit is the merger of the concept of binary choice from information theory with the superposition principle of quantum mechanics. One may say that the qubit has become the mascot of quantum information.

The quantum gates are chosen from a set of gates which the quantum computer in question must be able to perform. Such a set of gates is called universal if an arbitrary unitary transformation can be approximated with arbitrary accuracy by sequences of quantum gates out of this set. Various sets of universal gates have been proposed. For example, the set of all one-qubit rotations $U_R \in SU(2)$ together with the CNOT-gate $\sum_{i,j\in\{0,1\}} |i\rangle_c\langle i| \otimes |i + j \bmod 2\rangle_t\langle j|$ is universal [17]. There also exist finite universal sets of gates. As an example, the set consisting of the Toffoli- and the Hadamard gate is universal [18] as well as the set consisting of the CNOT-, the Hadamard- and the $\pi/4$-phase gate

$\exp\left(-i\frac{\pi}{8}\sigma_z\right)$ [19]. Quite interestingly, if in the latter set of gates we double the rotation angle of the phase gate from $\pi/4$ to $\pi/2$ we change an extremely powerful set of gates into one with rather limited capabilities. The CNOT-, the Hadamard- and the $\pi/2$-phase gate generate the so-called Clifford group which is the normalizer of the Pauli group. Quantum circuits consisting only of these gates can be simulated efficiently classically, as is stated by the Gottesman-Knill theorem [21].

To mention a non-standard variant of the network model, in [22] it was shown that universal quantum computation can be performed using Bell measurements, local unitary transformations and certain entangled quantum states such as the Greenberger-Horne-Zeilinger (GHZ) state. Like in teleportation, a quantum register state is read in via Bell measurements and thereby transferred to another physical carrier. But this time it is not Bell states which are used as the nonlocal resource in teleportation but instead more complicated quantum states encoding quantum gates. In this way, the quantum register state is both processed and teleported at the same time. The gate teleportation technique has successfully been applied in many settings. For example, in [23] it was shown that universal quantum computation is possible only using linear optical elements and photon number measurement.

In the quantum logic network model, the physics and the logic of computation go very much in parallel. The quantum register represents, from the viewpoint of information theory, the processed information and, from the perspective of physics, the state of the system. Likewise, the quantum gates represent the logical operations carried out, and at the same time the unitary transformations according to which the quantum system evolves.

For the $\mathrm{QC}_\mathcal{C}$ the situation is very different. As will be explained in detail in Chapter 3, processing of information exists only at the classical level, namely as the processing of the one-qubit measurements. The $\mathrm{QC}_\mathcal{C}$ has no quantum input, no quantum output and no quantum register, and it does not consist of quantum gates. Nevertheless, the $\mathrm{QC}_\mathcal{C}$ is genuinely quantum mechanical as it uses the entangled cluster state as its central physical resource. The $\mathrm{QC}_\mathcal{C}$ works by measuring a subset of the cluster state quantum correlations in several rounds of one-qubit measurements.

It should be noted that if the $\mathrm{QC}_\mathcal{C}$ is used as a simulator of quantum logic networks, the simulation of the network algorithm requires in general much fewer time steps than the original network itself. For example, the entire Clifford part of a quantum logic network, i.e. all the CNOT-, Hadamard- and $\pi/2$-phase gates, as well as the readout measurements (!), can be performed simultaneously in the first measurement round, as explained in Chapter 2. It shall be pointed out, however, that the $\mathrm{QC}_\mathcal{C}$ is equivalent to the network model with regard to computational power. This can easily be seen by the facts that one-qubit measurements are within the standard repertoire of network computation and that the resource cluster state can be created efficiently within the network model via conditional phase gates. Nevertheless, it may be concluded that quantum circuits which mainly rely on unitary evolution are in general not time-optimal.

Let us now return to the initially posed question of what gives a quantum computer its computational power. Various viewpoints are taken to elucidate this question; superposi-

tion and interference, entanglement and measurement are all argued for being at the heart of the quantum speedup. Deutsch [5] identifies the capability of forming state superpositions as the cause for the quantum speedup, and coins the term of 'quantum parallelism'. Cleve, Ekert, Macchiavello and Mosca [24] find that a common pattern underpinning quantum algorithms can be identified when quantum computation is viewed as a multi-particle interference. In this picture, the computation starts with the preparation of a superposition of different classical inputs. Upon this superposition state, representing the quantum input, unitary transformations inducing phase shifts are applied, and by a final Fourier transformation the different computational paths are brought to interference. In this way, superposition and interference are seen as the basic ingredients for the quantum speedup. In the conclusion of the same paper, [24], the authors make a statement which may be read independently of the discussion on the role of superposition and interference in quantum computation: "We believe that the paradigm of estimating (or determining exactly) the eigenvalues of operators on eigenstates gives helpful insight into the nature of quantum algorithms and may prove useful in constructing new and improving existing algorithms."

The feature that sets quantum systems apart from classical systems is epitomized by entanglement. It has therefore been expected since the early days of quantum computation that entanglement or the ability to generate it are responsible for the quantum speedup. This "working hypothesis" has, for quantum computation with pure states, been given a rigorous basis by Jozsa and Linden [25] and recently by Vidal [26]. In [25] it is shown that, for a quantum computation to offer an exponential speedup, it requires multi-particle entanglement across a number of subsystems (e.g. qubits) of the quantum register which increases unboundedly with the input size. In [26] it is demonstrated that the evolution of a pure state of n qubits can be simulated classically with resources that grow linearly with n and exponentially with the amount of entanglement. These two works make it clear that entanglement indeed is an important ingredient in pure state quantum computation.

Finally, it was shown that projective quantum measurements are a sufficient resource for universal quantum computation. Specifically, in [27] (see also [28] and [29]) it was shown that preparation of the state $|0\rangle$ and four-qubit measurements are sufficient for universal quantum computation. Here, quantum computation gets by without unitary evolution altogether.

Can the one-way quantum computer contribute to the debate of what the origin of the speedup in quantum computation is? As for now, it does not come up with definite answers, but it shades light on the matter form a different perspective. Gathering all the facts learned in the universality proof for the $\mathrm{QC}_{\mathcal{C}}$ and the derivation of the underlying computational model, we may say that the $\mathrm{QC}_{\mathcal{C}}$ works –as stated before– by measuring a subset of the cluster state quantum correlation operators in one-qubit measurements. More precisely, the cluster state is an eigenstate of these correlation operators and what one measures is whether the corresponding eigenvalues are $+1$ or -1. We thus arrive at a statement quite similar to the one in [24] quoted above. At any rate, it is precisely the structure of these cluster state quantum correlations what requires further analysis in trying to better understand the $\mathrm{QC}_{\mathcal{C}}$ and the implications it may have for the design of future quantum algorithms.

'Quantum parallelism' does not seem to be a sufficient explanation for why quantum computation on the $\mathrm{QC}_{\mathcal{C}}$ is fast. For the $\mathrm{QC}_{\mathcal{C}}$ there is no superposition of classical inputs which are processed in parallel and are finally brought to interference. The structure to which the concept of 'quantum parallelism' is applied in the network model, the quantum register, has been removed from the description of the $\mathrm{QC}_{\mathcal{C}}$. The structures for processing of information which emerge are classical, and could therefore hardly exhibit quantum parallelism.

What turned out to become the $\mathrm{QC}_{\mathcal{C}}$, started off with the question "What can one do with ultra-cold atoms in optical lattices where one has only global control over the atom-atom interaction?" The inability to perform qubit-selective interaction must a priori seem as a severe limitation. Therefore, it came quite as a surprise that such a system suffices for universal quantum computation.

With the observation that the cold controlled collisions used to generate the universal entanglement resource simulate the Ising interaction, it became apparent that the potential realization of the $\mathrm{QC}_{\mathcal{C}}$ is not restricted to optical lattices and that there may exist a variety of other suitable systems. In this way, the scope has widened from a specific system to a system class with certain general properties. A more drastic change has occurred with the finding of the computational model underlying the $\mathrm{QC}_{\mathcal{C}}$. Thereby, the physics and the logic of quantum computation are divided, and the focus is shifted from physics to logic.

Nevertheless, the realization of the $\mathrm{QC}_{\mathcal{C}}$ is physical, and therefore let us conclude this introduction with a few notes on the possible future implementation of the $\mathrm{QC}_{\mathcal{C}}$, recent advances in experiment and fault-tolerant quantum computation. From the present perspective, the most promising candidate for a physical system to implement the $\mathrm{QC}_{\mathcal{C}}$ is that of an optical lattice loaded with ultra-cold atoms. In such a device the globally tunable Ising interaction used to generate the cluster state can be realized e.g. via state-selective displacement of the atoms and cold controlled collisions [30] or via tunneling [31]. Measurements can in principle be performed via the techniques of Raman spectroscopy.

Towards an experimental realization of the $\mathrm{QC}_{\mathcal{C}}$ it has so far been demonstrated that an optical lattice can be loaded with ultra-cold atoms in such a way that over extended regions of the lattice each lattice site is occupied with exactly one atom. This is done by driving the system from a superfluid to a Mott insulator phase [32]. Coherent state-dependent transport of the atoms has been demonstrated over a distance of up to seven lattice sites [33]. Further, Bose-Einstein condensates have been trapped in an optical lattice and their evolution has been investigated in interference experiments. It was found that the matter wave field of the Bose-Einstein condensate undergoes a periodic series of collapses and revivals [34], attributed to the quantized structure of the matter wave field of the Bose-Einstein condensate and the collisions between individual atoms. This experiment shows in the passing that coherent cold controlled collisions between atoms, proposed for the creation of cluster states, are possible. What has so far not been demonstrated experimentally are the subsequent measurements. These need to address the qubits individually (the operations to create the cluster state are all global), which is difficult with present set-ups in optical lattices. Except for the measurements, all techniques required for $\mathrm{QC}_{\mathcal{C}}$-computation

do, though probably not with the required accuracy, exist.

What is, in fact, the accuracy required for fault-tolerant quantum computation? This question needs to be asked not only for the $\mathrm{QC}_{\mathcal{C}}$ but for every type of a quantum computer. A priori, the regimes of quantum computation with perfect and imperfect means are very different. For the latter, quantum coherence will decay if the computation only is long enough, no matter how small the error rates are. This observation may lead one to expect that the requirements on the accuracy of elementary quantum operations become more and more stringent with increasing size of the computation. Fortunately, this is not the case. The degradation of quantum coherence, due to both interaction with environmental degrees of freedom and imprecise gate operation, can be counteracted by techniques of quantum coding and error correction [35] - [39] and, comprising these, fault-tolerant gate operation [40] - [42]. Using these techniques, arbitrary long quantum computations can be performed at a moderate increase in the size requirements for the quantum computer. There exist fixed error thresholds which have to be met.

Between the accuracy of present-day technology for the manipulation of quantum systems and the accuracy required for fault-tolerant quantum computation there is a gap, though, and this gap is not even small. Proven bounds on the error thresholds are of the order of 10^{-6} [43, 44] which appears extremely hard to achieve. Experiments in quantum information are often designed to first of all observe and verify quantum mechanical behavior. To mention a few highlights, coherent oscillations have been demonstrated for qubits in Josephson tunnel junctions [45, 46], and, in systems of trapped ions, four qubits have been entangled [47].

Very recently, two-qubit gates have been realized in systems of trapped ions [48, 49], with gate error probabilities down to a few percent. And the gap is narrowed from the side of theory, too. Refinements in gate construction, syndrome extraction and -processing have led to estimates for the error thresholds of about 10^{-3} [50]. The advances in experiment and theory make the realization of a quantum computer seem much more likely today than it appeared, say, in 1994 when Peter Shor discovered the factorization algorithm.

Chapter 2

The one-way quantum computer

2.1 General picture of the $\mathrm{QC}_{\mathcal{C}}$

Most of the current experiments are designed to implement sequences of highly controlled interactions between selected particles (qubits), thereby following models of a quantum computer as a (sequential) network of quantum logic gates [16, 17].

Here we describe a different model of universal and scalable quantum computation, the one-way quantum computer ($\mathrm{QC}_{\mathcal{C}}$). In our model, the entire resource for the quantum computation is provided initially in the form of a specific entangled state, a so-called cluster state $|\phi\rangle_{\mathcal{C}}$ of a large number of qubits. We will give a definition of cluster states below. Information is then written onto the cluster, processed, and read out from the cluster by one-particle measurements only. The entangled state of the cluster thereby serves as a universal "substrate" for any quantum computation. It provides in advance all entanglement that is involved in the subsequent quantum computation. In the process of computation, i.e. during the rounds of one-qubit measurements, all entanglement in the cluster state is destroyed such that the cluster state can be used only once. Therefore, we call this scheme the one-way quantum computer.

Cluster states can be created efficiently in any system with a quantum Ising-type interaction (at very low temperatures) between two-state particles in a lattice configuration. We consider two and three-dimensional arrays, or clusters, of qubits. To create a cluster state $|\phi\rangle_{\mathcal{C}}$ on the cluster $\mathcal{C}$ from a product state $\bigotimes_{a\in\mathcal{C}}|+\rangle_a$, (where $\sigma_x^{(a)}|\pm\rangle_a = \pm|\pm\rangle_a$), the Ising-interaction is switched on for an appropriately chosen finite time interval T, and is switched off afterwards. Since the Ising Hamiltonian acts uniformly on the lattice, an entire cluster of neighboring particles becomes entangled in a single step.

To process quantum information with the cluster $\mathcal{C}$ it suffices to measure its particles in a certain order and in a certain basis, as illustrated in Fig. 2.1. This figure shows, in a way, the physical and the logical layer of the $\mathrm{QC}_{\mathcal{C}}$. The physical part is represented by the entangled cluster qubits and the measurements performed on them. The cluster qubits are displayed as dots "$\odot$" or as arrows "$\uparrow$", "$\nearrow$", depending on the respective measured observable (see caption). These measurements induce a quantum processing of logical

qubits. The horizontal spatial axis on the cluster can be associated with the time axis of the implemented quantum circuit, i.e. with the direction of the "information flow". As will be explained, measurements of observables σ_z effectively remove the respective lattice qubit from the cluster. This property allows one to structure the cluster state on the lattice and imprint a network-like structure on it (displayed in Fig. 2.1 in gray underlay). More precisely, the σ_z-measurements project the cluster state $|\phi\rangle_{\mathcal{C}}$ into the tensor product $|\mu\rangle_{\mathcal{C}\setminus\mathcal{C}_N} \otimes |\tilde{\phi}\rangle_{\mathcal{C}_N}$. Therein, $|\mu\rangle_{\mathcal{C}\setminus\mathcal{C}_N}$ is a product state in the computational basis, and $|\tilde{\phi}\rangle_{\mathcal{C}_N}$ the state of the so far unmeasured qubits. It is again a cluster state on a network-shaped sub-cluster $\mathcal{C}_N$. On this sub-cluster quantum gates can be implemented via measurements of observables σ_x, σ_y, and linear combinations thereof. Measurements of σ_x and σ_y are used for "wires", i.e. to propagate logical quantum bits across the cluster, and for CNOT gates between two logical qubits. Observables of the form $\cos(\varphi)\,\sigma_x \pm \sin(\varphi)\,\sigma_y$ are measured to realize arbitrary rotations of logical qubits. For the cluster qubits of which a linear combination of σ_x and σ_y is measured, the measurement basis depends on the results of measurements at other cluster qubits. This introduces a temporal order among the measurements. The processing is finished once all qubits except a last one on each wire have been measured. The remaining unmeasured qubits form the quantum register which is now ready to be read out. At this point, the results of previous measurements determine in which basis these "output" qubits need to be measured for the final readout, or, if the readout measurements are in the σ_x-, σ_y- or σ_z-eigenbasis, how the readout measurements have to be interpreted.

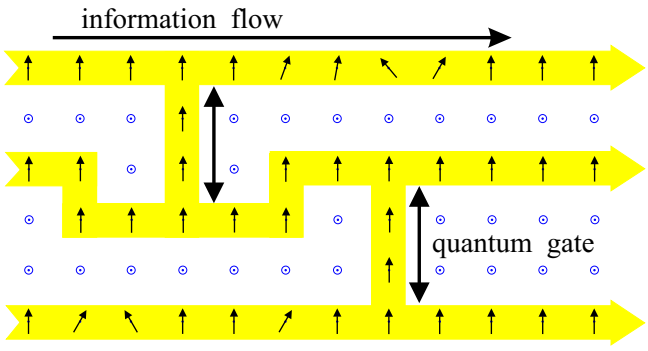


Figure 2.1: Simulation of a quantum logic network by measuring two-state particles on a lattice. Before the measurements the qubits are in the cluster state $|\phi\rangle_{\mathcal{C}}$ of (2.1). Circles $\odot$ symbolize measurements of σ_z, vertical arrows are measurements of σ_x, while tilted arrows refer to measurements in the x-y-plane.

The purpose of this chapter is twofold. First, it is to introduce the $\mathrm{QC}_{\mathcal{C}}$ and to give the proof for its universality. We do this by showing that the $\mathrm{QC}_{\mathcal{C}}$ may be regarded as a simulator of network quantum computers. In this way, we clarify the relation between the $\mathrm{QC}_{\mathcal{C}}$ and the network model which is the most widely used model of a quantum computer. Second, we provide a number of examples for $\mathrm{QC}_{\mathcal{C}}$-circuits which are characteristic and of practical interest.

Specifically, in Section 2.2 we give the universality proof for the described scheme of computation in a complete and detailed form. We provide an analytic explanation for the functioning of the gate simulations on the $\mathrm{QC}_{\mathcal{C}}$ in Section 2.2.6 and apply it to both the gates of a universal set in Section 2.2.7 and to the more complicated circuits in Section 2.3. In Section 2.2.8 we discuss the spatial, temporal and operational resources required in $\mathrm{QC}_{\mathcal{C}}$-computations in relation to the resources needed for the corresponding quantum logic networks. We find that overheads are at most polynomial. In Section 2.3 we give examples of larger gates and sub-circuits which may be of practical relevance, among them the $\mathrm{QC}_{\mathcal{C}}$-circuit for quantum Fourier transformation and for the n-qubit adder. In Section 2.4 we discuss the $\mathrm{QC}_{\mathcal{C}}$ computations on finite (small) clusters. We describe a variant of the scheme consisting of repeated steps of (re-)entangling a cluster via the Ising interaction, alternating with rounds of one-qubit measurements. Using this modified scheme it is possible to split long computations such that they fit piecewise on a small cluster.

2.2 Universality of the $\mathrm{QC}_{\mathcal{C}}$

In this section we prove that the $\mathrm{QC}_{\mathcal{C}}$ is a universal quantum computer. The technique to accomplish this is to show that any quantum logic network can be simulated efficiently on the $\mathrm{QC}_{\mathcal{C}}$.

2.2.1 Cluster states and their quantum correlations

Cluster states are pure quantum states of two-level systems (qubits) located on a cluster $\mathcal{C}$. This cluster is a connected subset of a simple cubic lattice $\mathbb{Z}^d$ in $d \geq 1$ dimensions. The cluster states $|\phi_{\{\kappa\}}\rangle_{\mathcal{C}}$ obey the set of eigenvalue equations

$$K^{(a)}|\phi_{\{\kappa\}}\rangle_{\mathcal{C}} = (-1)^{\kappa_a}|\phi_{\{\kappa\}}\rangle_{\mathcal{C}}, \tag{2.1}$$

with the correlation operators

$$K^{(a)} = \sigma_x^{(a)} \bigotimes_{b\in\mathrm{nbgh}(a)} \sigma_z^{(b)}. \tag{2.2}$$

Therein, $\{\kappa\} := \{\kappa_a \in \{0,1\} |\, a \in \mathcal{C}\}$ is a set of binary parameters which specify the cluster state and $\mathrm{nbgh}(a)$ is the set of all neighboring lattice sites of a.

A cluster state $|\phi_{\{\kappa\}}\rangle_{\mathcal{C}}$ is completely specified by the eigenvalue equations (2.1) since the $K^{(a)}$, $a \in \mathcal{C}$, form a complete set of $|\mathcal{C}|$ independent and commuting observables for the system of qubits on the cluster $\mathcal{C}$. This can most easily be seen from the fact that $K^{(a)}$ is obtained from $\sigma_x^{(a)}$ under conjugation with a unitary transformation, as shown below (2.11). For a set of eigenvalues specified by $\{\kappa\}$ the corresponding eigenspace is thus one-dimensional, i.e. $|\phi_{\{\kappa\}}\rangle_{\mathcal{C}}$ is determined modulo an irrelevant phase factor. There are $2^{|\mathcal{C}|}$ different choices for $\{\kappa\} \in \{0,1\}^{|\mathcal{C}|}$, and since the $K^{(a)}$ are hermitian operators,

the associated common eigenstates, the cluster states, are mutually orthogonal and form a basis in the Hilbert space of the cluster.

The discussion in this thesis will be based entirely on the eigenvalue equations (2.1) and we will never need to work out some cluster state in any specific basis. In fact, to write down a cluster state in its explicit form would be quite space-consuming since the minimum number of required terms scales exponentially with the number of qubits [51], and for computation we will be going to consider rather large cluster states. Nevertheless, for illustration we give a few examples for cluster states of a small number of qubits. The cluster states on a chain of 2, 3 and 4 qubits, fulfilling the eigenvalue equations (2.1) with all $\kappa_a = 0$, are

$$\begin{aligned} |\phi\rangle_{\mathcal{C}_2} &= \frac{1}{\sqrt{2}}\left(|0\rangle_1|+\rangle_2 + |1\rangle_1|-\rangle_2\right), \\ |\phi\rangle_{\mathcal{C}_3} &= \frac{1}{\sqrt{2}}\left(|+\rangle_1|0\rangle_2|+\rangle_3 + |-\rangle_1|1\rangle_2|-\rangle_3\right), \\ |\phi\rangle_{\mathcal{C}_4} &= \frac{1}{2}|+\rangle_1|0\rangle_2|+\rangle_3|0\rangle_4 + \frac{1}{2}|+\rangle_1|0\rangle_2|-\rangle_3|1\rangle_4, \\ &\quad + \frac{1}{2}|-\rangle_1|1\rangle_2|-\rangle_3|0\rangle_4 + \frac{1}{2}|-\rangle_1|1\rangle_2|+\rangle_3|1\rangle_4, \end{aligned} \tag{2.3}$$

with the notation

$$\begin{aligned} |0\rangle_a &:= |0\rangle_{a,z} = \sigma_z^{(a)}|0\rangle_{a,z}, \\ |1\rangle_a &:= |1\rangle_{a,z} = -\sigma_z^{(a)}|1\rangle_{a,z}, \\ |\pm\rangle_a &:= \tfrac{1}{\sqrt{2}}(|0\rangle_a \pm |1\rangle_a). \end{aligned} \tag{2.4}$$

The state $|\phi\rangle_{\mathcal{C}_2}$ is local unitary equivalent to a Bell state and $|\phi\rangle_{\mathcal{C}_3}$ to the Greenberger-Horne-Zeilinger (GHZ) state. $|\phi\rangle_{\mathcal{C}_4}$, however, is not equivalent to a 4-particle GHZ state. In particular, the entanglement in $|\phi\rangle_{\mathcal{C}_4}$ cannot be destroyed by a single local operation [51].

Ways to create a cluster state in principle are to measure all the correlation operators $K^{(a)}$, $a \in \mathcal{C}$ of (2.2) on an arbitrary $|\mathcal{C}|$-qubit state or to cool into the ground state of a Hamiltonian $H_K = -\hbar g \sum_{a\in\mathcal{C}} \kappa_a K^{(a)}$.

Another way –likely to be more suitable for realization in the lab– is as follows. First, a product state $|+\rangle_{\mathcal{C}} = \bigotimes_{a\in\mathcal{C}} |+\rangle_a$ is prepared. Second, the unitary transformation $S^{(\mathcal{C})}$,

$$S^{(\mathcal{C})} = \prod_{a,b\in\mathcal{C}|b-a\in\gamma_d} S^{ab}, \tag{2.5}$$

is applied to the state $|+\rangle$. Often we will write S in short for $S^{(\mathcal{C})}$. In (2.5), for the cases of dimension $d = 1, 2, 3$, we have $\gamma_1 = \{1\}$, $\gamma_2 = \{(1,0)^T, (0,1)^T\}$ and $\gamma_3 = \{(1,0,0)^T, (0,1,0)^T, (0,0,1)^T\}$, and the two-qubit transformation S^{ab} is such that the state $|1\rangle_a \otimes |1\rangle_b$ acquires a phase of π under its action while the remaining states $|0\rangle_a \otimes |0\rangle_b$, $|0\rangle_a \otimes |1\rangle_b$ and $|1\rangle_a \otimes |0\rangle_b$ acquire no phase. Thus, S^{ab} has the form

$$S^{ab} = |0\rangle_a\langle 0| \otimes \mathbb{1}^{(b)} + |1\rangle_a\langle 1| \otimes \sigma_z^{(b)}, \tag{2.6}$$

i.e. is a conditional phase gate between a and b. Note that all operations S^{ab} in S mutually commute and that they can therefore be carried out at the same time. Initial individual preparation of the cluster qubits in $|+\rangle_{a\in\mathcal{C}}$ can also be done in parallel. Thus, the creation of the cluster state is a two step process. *The temporal resources to create the cluster state are constant in the size of the cluster.*

The state $|+\rangle_{\mathcal{C}}$ obviously obeys the eigenvalue equations $\sigma_x^{(a)}|+\rangle_{\mathcal{C}} = |+\rangle_{\mathcal{C}}\ \forall a \in \mathcal{C}$ and thus the cluster state $|\phi\rangle_{\mathcal{C}}$ generated via S obeys

$$|\phi\rangle_{\mathcal{C}} = S\sigma_x^{(a)}S^{\dagger}\,|\phi\rangle_{\mathcal{C}}\ ,\quad \forall a \in \mathcal{C}. \tag{2.7}$$

To obtain $S\sigma_x^{(a)}S^{\dagger}$, we use the transformation relations for the stabilizer of a state under action of a phase gate [41]. We observe that

$$\begin{array}{rcl} S^{ab}\sigma_x^{(a)}S^{ab\dagger} &=& \sigma_x^{(a)}\otimes\sigma_z^{(b)}, \\ S^{ab}\sigma_x^{(b)}S^{ab\dagger} &=& \sigma_z^{(a)}\otimes\sigma_x^{(b)}, \end{array} \tag{2.8}$$

and

$$S^{ab}\sigma_x^{(c)}S^{ab\dagger} = \sigma_x^{(c)},\quad \forall c \in \mathcal{C}\backslash\{a,b\}. \tag{2.9}$$

Further, the Pauli phase flip operators $\sigma_z^{(d)}$ commute with all S^{ab}, i.e.

$$S^{ab}\sigma_z^{(d)}S^{ab\dagger} = \sigma_z^{(d)},\quad \forall d \in \mathcal{C}. \tag{2.10}$$

Now, from (2.8), (2.9) and (2.10) it follows that

$$S\sigma_x^{(a)}S^{\dagger} = \sigma_x^{(a)}\bigotimes_{b\in\mathrm{nbgh}(a)}\sigma_z^{(b)}. \tag{2.11}$$

Thus, the state $|\phi\rangle_{\mathcal{C}}$ generated from $|+\rangle_{\mathcal{C}}$ via the transformation S as defined in (2.5) does indeed obey eigenvalue equations of form (2.1), with

$$\kappa_a = 0,\quad \forall\, a \in \mathcal{C}. \tag{2.12}$$

As the eigenvalues are fixed in this case, we drop them in the notation for the cluster state $|\phi\rangle_{\mathcal{C}}$. Cluster states specified by different sets $\{\kappa_a\}$ can be obtained by applying Pauli phase flip operators $\sigma_z^{(a)}$. To see this, note that

$$\sigma_z^{(a)}K^{(b)}{\sigma_z^{(a)}}^{\dagger} = (-1)^{\delta_{a,b}}\,K^{(b)}. \tag{2.13}$$

Therefore,

$$\bigotimes_{a\in\mathcal{C}}\left(\sigma_z^{(a)}\right)^{\Delta\kappa_a}\ |\phi_{\{\kappa_a\}}\rangle_{\mathcal{C}} = |\phi_{\{\kappa_a+\Delta\kappa_a\}}\rangle_{\mathcal{C}}, \tag{2.14}$$

where the addition for the κ_a is modulo 2. Cluster states with different sets $\{\kappa\}$ are equally suited for $\mathrm{QC}_{\mathcal{C}}$-computation.

Concerning a physical realization of the transformation S defined in (2.5), note that S is generated by the Hamiltonian

$$H = \hbar g \sum_{a,b\in\mathcal{C}|b-a\in\gamma_d} \frac{\mathbb{1}^{(a)} - \sigma_z^{(a)}}{2} \frac{\mathbb{1}^{(b)} - \sigma_z^{(b)}}{2}. \tag{2.15}$$

Now, to perform the required unitary transformation S the Hamiltonian H is switched on for a time span $T = \frac{\pi}{g}$. The transformation $S = \exp\left(-i\frac{\pi}{\hbar g}H\right)$ may be written in the form

$$\begin{aligned} S &= \left(\prod_{a,b\in\mathcal{C}|b-a\in\gamma_d} e^{-i\frac{\pi}{4}} \exp\left(i\frac{\pi}{4}\sigma_z^{(a)}\right) \exp\left(i\frac{\pi}{4}\sigma_z^{(b)}\right) \right) \\ &\quad \times \exp\left(-i\frac{\pi}{4} \sum_{a,b\in\mathcal{C}|b-a\in\gamma_d} \sigma_z^{(a)}\sigma_z^{(b)} \right). \end{aligned} \tag{2.16}$$

We find that the interaction part H_I of the Hamiltonian H generating S is of Ising form,

$$H_I = \hbar\frac{g}{4} \sum_{a,b\in\mathcal{C}|b-a\in\gamma_d} \sigma_z^{(a)}\sigma_z^{(b)}, \tag{2.17}$$

and, since the local part H_{local} of the Hamiltonian commutes with the Ising Hamiltonian H_I, the interaction S generated by H is local unitary equivalent to the unitary transformation generated by a Ising Hamiltonian.

For matter of presentation, the interaction S^{ab} in (2.6) and, correspondingly, the local part of the Hamiltonian H in (2.15) has been chosen in such a way that the eigenvalue equations (2.1) take the particularly simple form with $\kappa_a = 0$ for all $a \in \mathcal{C}$, irrespective of the shape of the cluster.

To create quantum states that are useful as a resource for the $\text{QC}_\mathcal{C}$, i.e. cluster- or local unitary equivalent states, all systems with a tunable Ising interaction and a local σ_z-type Hamiltonian, i.e. with a Hamiltonian

$$H' = \sum_{a\in\mathcal{C}} \Delta E_a\, \sigma_z^{(a)} + \hbar\frac{g(t)}{4} \sum_{a,b\in\mathcal{C}|b-a\in\gamma_d} \sigma_z^{(a)}\sigma_z^{(b)} \tag{2.18}$$

are suitable, provided the coupling $g(t)$ can be switched between zero and at least one nonzero value.

Even this condition can be relaxed. A permanent Ising interaction instead of a globally tunable one is sufficient, if the measurement process is much faster than the characteristic time scale for the Ising interaction, i.e. if the measurements are stroboscopic. If it takes the Ising interaction a time T_{Ising} to create a cluster state $|\phi\rangle_\mathcal{C}$ from a product state $|+\rangle_\mathcal{C}$, then the Ising interaction acting for a time $2T_{\text{Ising}}$ performs the identity operation, $S^{(\mathcal{C})}S^{(\mathcal{C})} = \mathbb{1}^{(\mathcal{C})}$. Therefore, starting with a product state $|+\rangle_\mathcal{C}$ at time $t = 0$ evolving under permanent Ising interaction, stroboscopic measurements may be performed at times $(2k+1)\,T_{\text{Ising}},\ k \in \mathbb{N}$.

One possibility to create a cluster state in practice is via cold controlled collisions in optical lattices, as described in [51]. Cold atoms representing the qubits can be arranged on a two- or three dimensional lattice and state-dependent interaction phases may be acquired via cold collisions between neighboring atoms [30] or via tunneling [31]. For a suitable choice of the collision phases φ, $\varphi = \pi \bmod 2\pi$, the state resulting from a product state $|+\rangle_{\mathcal{C}}$ after interaction is a cluster state obeying the eigenvalue equations (2.1), with the set $\{\kappa_a, a \in \mathcal{C}\}$ specified by the filling pattern of the lattice.

Let us, at the end of this section, briefly state which techniques will be used for the explanation of measurement-based quantum computation on cluster states. First, note that the operators $(-1)^{\kappa_a} K^{(a)}$ in eq. (2.1) generate the stabilizer of the state $|\phi_{\{\kappa\}}\rangle_{\mathcal{C}}$. The stabilizer formalism, as developed by Gottesman [42, 41] and by Calderbank et al. [52] (see also [20]), provides a compact characterization of the cluster state. It is also useful in understanding some of the working principles of the $\mathrm{QC}_{\mathcal{C}}$. In the subsequent sections we frequently perform stabilizer manipulations.

Further, some basic notions of graph theory will be useful later when we discuss the relation between quantum algorithms and graphs in Section 3.6. Therefore let us, at this point, establish a connection between quantum states such as the cluster state of (2.1) and graphs. The treatment here follows that of [53], adapted to our notation.

Let us recall the definition of a graph, as given e.g. in [54]. A graph $G(V, E)$ is a set V of vertices connected via edges e from the set E. The information of which vertex $a \in V$ is connected to which other vertex $b \in V$ is contained in a symmetric $|V| \times |V|$ matrix Γ, the adjacency matrix. The matrix Γ is such that $\Gamma_{ab} = 1$ if two vertices a and b are connected via an edge $e \in E$, and $\Gamma_{ab} = 0$ otherwise. We identify the cluster $\mathcal{C}$ with the vertices $V_{\mathcal{C}}$ of a graph, $\mathcal{C} = V_{\mathcal{C}}$, and in this way establish a connection to the notion introduced earlier.

To relate graphs to quantum mechanics, the vertices of a graph can be identified with local quantum systems, in this case qubits, and the edges with two-particle interactions, in the present case $\sigma_z\sigma_z$-interactions. If one initially prepares each individual qubit a in the state $\left(\sigma_z^{(a)}\right)^{\kappa_a} |+\rangle_a$ and subsequently switches on, for an appropriately chosen finite time span, the interaction

$$H_{G(V,E)} = \hbar g \sum_{(a,b)\in E} \frac{\mathbb{1}^{(a)} - \sigma_z^{(a)}}{2} \frac{\mathbb{1}^{(b)} - \sigma_z^{(b)}}{2}, \tag{2.19}$$

with $(a, b) \in E$ denoting an edge between qubits a and b, then one obtains quantum states that are graph code words as introduced in [53]. Henceforth we will refer to these graph code words as graph states and use them in a context different from coding. The graph states $|\phi\{\kappa\}\rangle_G$ are defined by a set of eigenvalue equations which read

$$\sigma_x^{(a)} \bigotimes_{b\in V} \left(\sigma_z^{(b)}\right)^{\Gamma_{ab}} |\phi\{\kappa\}\rangle_G = (-1)^{\kappa_a} |\phi\{\kappa\}\rangle_G, \tag{2.20}$$

with $\kappa_a \in \{0, 1\}\ \forall\, a \in V$. Here we use G instead of V as an index for the state $|\phi\rangle$ as the set $E \subset V \times V$ of edges is now independent and no longer implicitly specified by V as was the case in (2.1).

Note that cluster states (2.1) are a particular case of graph states (2.20). The graph $G(\mathcal{C}, E_{\mathcal{C}})$ which describes a cluster state is that of a square lattice in 2D and that of a simple cubic lattice in 3D, i.e. the set $E_{\mathcal{C}}$ of edges is given by

$$E_{\mathcal{C}} = \big\{(a,b)|\, a,b \in \mathcal{C}, b \in \mathrm{nbgh}(a)\big\}. \tag{2.21}$$

2.2.2 A universal set of quantum gates

To provide something definite to discuss right from the beginning, we now give the procedures of how to realize a CNOT gate and a general one-qubit rotation via one-qubit measurements on a cluster state. The explanation of why and how these gates work will be given in Section 2.2.7.

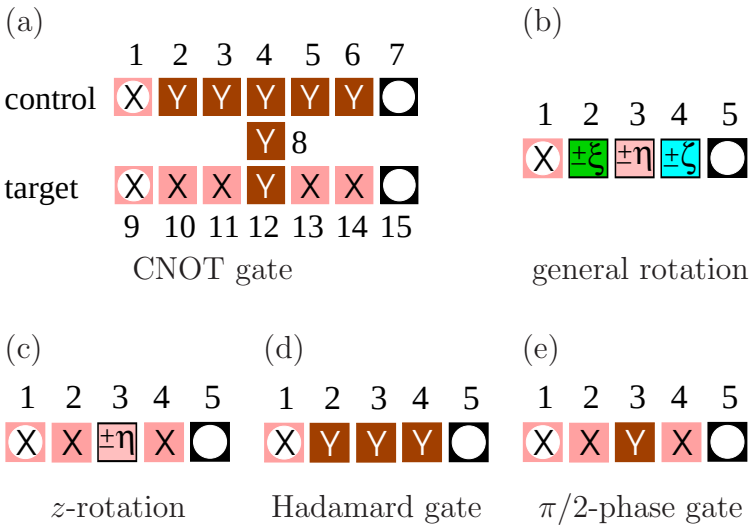


Figure 2.2: Realization of elementary quantum gates on the $\mathrm{QC}_{\mathcal{C}}$. Each square represents a lattice qubit. The squares in the extreme left column marked with white circles denote the input qubits, those in the right-most column denote the output qubits.

A CNOT gate can be realized on a cluster state of 15 qubits, as shown in Fig. 2.2. All measurements can be performed simultaneously. The procedure to realize a CNOT gate on a cluster with 15 qubits as displayed in Fig. 2.2 is

Procedure 1 Realization of a CNOT gate acting on a two-qubit state $|\psi_{\mathrm{in}}\rangle$.

1. Prepare the state
   $|\Psi_{\mathrm{in}}\rangle_{\mathcal{C}_{15}} = |\psi_{\mathrm{in}}\rangle_{1,9} \otimes \Big(\bigotimes_{i\in\mathcal{C}_{15}\setminus\{1,9\}} |+\rangle_i\Big)$.

2. Entangle the 15 qubits of the cluster $\mathcal{C}_{15}$ via the unitary operation $S^{(\mathcal{C}_{15})}$.

3. Measure all qubits of $\mathcal{C}_{15}$ except for the output qubits 7, 15 (following the labeling in Fig. 2.2). The measurements can be performed simultaneously. Qubits 1, 9, 10, 11, 13, 14 are measured in the σ_x-eigenbasis and qubits 2-6, 8, 12 in the σ_y-eigenbasis.

Dependent on the measurement results, the following gate is thereby realized:

$$U'_{CNOT} = U_{\Sigma,CNOT}\, CNOT(c,t), \tag{2.22}$$

where the byproduct operator $U_{\Sigma,CNOT}$ has the form

$$\begin{aligned} U_{\Sigma,CNOT} &= \sigma_x^{(c)^{\gamma_x^{(c)}}} \sigma_x^{(t)^{\gamma_x^{(t)}}} \sigma_z^{(c)^{\gamma_z^{(c)}}} \sigma_z^{(t)^{\gamma_z^{(t)}}}, \text{ with} \\ \gamma_x^{(c)} &= s_2 + s_3 + s_5 + s_6 \\ \gamma_x^{(t)} &= s_2 + s_3 + s_8 + s_{10} + s_{12} + s_{14} \\ \gamma_z^{(c)} &= s_1 + s_3 + s_4 + s_5 + s_8 + s_9 + s_{11} + 1 \\ \gamma_z^{(t)} &= s_9 + s_{11} + s_{13}. \end{aligned} \tag{2.23}$$

Therein, the s_i represent the measurement outcomes s_i on the qubits i. The expression (2.23) is modified if redundant cluster qubits are present and/or if the cluster state on which the CNOT gate is realized is specified by a set $\{\kappa_a\}$ different from (2.12), see Section 2.2.3. This concludes the presentation of the CNOT gate, the proof of its functioning is given in Section 2.2.7.

An arbitrary rotation $U_{Rot} \in SU(2)$ can be realized on a chain of 5 qubits. Consider a rotation in its Euler representation

$$U_{Rot}[\xi,\eta,\zeta] = U_x[\zeta]U_z[\eta]U_x[\xi], \tag{2.24}$$

where the rotations about the x- and z-axis are

$$\begin{aligned} U_x[\alpha] &= \exp\left(-i\alpha\frac{\sigma_x}{2}\right) \\ U_z[\alpha] &= \exp\left(-i\alpha\frac{\sigma_z}{2}\right). \end{aligned} \tag{2.25}$$

Initially, the first qubit is prepared in some state $|\psi_{\text{in}}\rangle$, which is to be rotated, and the other qubits are prepared in $|+\rangle$. After the 5 qubits are entangled by the unitary transformation S, the state $|\psi_{\text{in}}\rangle$ can be rotated by measuring qubits 1 to 4. At the same time, the state is also swapped to site 5. The qubits $1\,..\,4$ are measured in appropriately chosen bases

$$\mathcal{B}_j(\varphi_j) = \left\{ \frac{|0\rangle_j + e^{i\varphi_j}|1\rangle_j}{\sqrt{2}},\ \frac{|0\rangle_j - e^{i\varphi_j}|1\rangle_j}{\sqrt{2}} \right\}, \tag{2.26}$$

whereby the measurement outcomes $s_j \in \{0,1\}$ for $j = 1\,..\,4$ are obtained. Here, $s_j = 0$ means that qubit j is projected into the first state of $\mathcal{B}_j(\varphi_j)$. In (2.26) the basis states of all possible measurement bases lie on the equator of the Bloch sphere, i.e. on the intersection

of the Bloch sphere with the x-y-plane. Therefore, the measurement basis for qubit j can be specified by a single parameter, the measurement angle φ_j. The measurement direction of qubit j is the vector on the Bloch sphere which corresponds to the first state in the measurement basis $\mathcal{B}_j(\varphi_j)$. Thus, the measurement angle φ_j is the angle between the measurement direction at qubit j and the positive x-axis. In summary, the procedure to realize an arbitrary rotation $U_{Rot}[\xi, \eta, \zeta]$, specified by its Euler angles ξ, η, ζ, is this:

Procedure 2 Realization of general one-qubit rotations $U_{Rot} \in SU(2)$.

1. Prepare the state $|\Psi_{\text{in}}\rangle_{\mathcal{C}_5} = |\psi_{\text{in}}\rangle_1 \otimes \left(\bigotimes_{i=2}^{5} |+\rangle_i\right)$.

2. Entangle the five qubits of the cluster $\mathcal{C}_5$ via the unitary operation $S^{(\mathcal{C}_5)}$.

3. Measure qubits 1 - 4 in the following order and basis

$$\begin{array}{ll} 3.1 & \text{measure qubit 1 in } \mathcal{B}_1(0) \\ 3.2 & \text{measure qubit 2 in } \mathcal{B}_2\left(-\xi\,(-1)^{s_1}\right) \\ 3.3 & \text{measure qubit 3 in } \mathcal{B}_3\left(-\eta\,(-1)^{s_2}\right) \\ 3.4 & \text{measure qubit 4 in } \mathcal{B}_4\left(-\zeta\,(-1)^{s_1+s_3}\right) \end{array} \tag{2.27}$$

Via Procedure 2 the rotation U'_{Rot} is realized:

$$U'_{Rot}[\xi, \eta, \zeta] = U_{\Sigma,Rot}\, U_{Rot}[\xi, \eta, \zeta]. \tag{2.28}$$

Therein, the random byproduct operator has the form

$$U_{\Sigma,Rot} = \sigma_x^{s_2+s_4} \sigma_z^{s_1+s_3}. \tag{2.29}$$

It can be corrected for at the end of the computation, as will be explained in Section 2.2.5.

There is a subgroup of rotations for which the realization procedure is somewhat simpler than Procedure 2. These rotations form the subgroup of local operations in the Clifford group. The Clifford group is the normalizer of the Pauli group.

Among these rotations are, for example, the Hadamard gate and the $\pi/2$-phase gate. These gates can be realized on a chain of 5 qubits in the following way:

Procedure 3 Realization of a Hadamard- and $\pi/2$-phase gate.

1. Prepare the state $|\Psi_{\text{in}}\rangle_{\mathcal{C}_5} = |\psi_{\text{in}}\rangle_1 \otimes \left(\bigotimes_{i=2}^{5} |+\rangle_i\right)$.

2. Entangle the five qubits of the cluster $\mathcal{C}_5$ via the unitary operation $S^{(\mathcal{C}_5)}$.

3. Measure qubits 1 - 4. This can be done simultaneously. For the Hadamard gate, measure individually the observables $\sigma_x^{(1)}$, $\sigma_y^{(2)}$, $\sigma_y^{(3)}$, $\sigma_y^{(4)}$. For the $\pi/2$-phase gate measure $\sigma_x^{(1)}$, $\sigma_x^{(2)}$, $\sigma_y^{(3)}$, $\sigma_x^{(4)}$.

The difference with respect to Procedure 2 for general rotations is that in Procedure 3 no measurement bases need to be adjusted according to previous measurement results and therefore the measurements can all be performed at the same time.

As in the cases before, the Hadamard- and the $\pi/2$-phase gate are performed only modulo a subsequent byproduct operator which is determined by the random measurement outcomes s_k

$$\begin{aligned} U_{\Sigma,H} &= \sigma_x^{s_1+s_3+s_4}\,\sigma_z^{s_2+s_3} \\ U_{\Sigma,U_z(\pi/2)} &= \sigma_x^{s_2+s_4}\,\sigma_z^{s_1+s_2+s_3+1}. \end{aligned} \tag{2.30}$$

Before we explain the functioning of the above gates, we would like to address the following questions: First,"How does one manage to occupy only those lattice sites with cluster qubits that are required for a particular circuit but leaves the remaining ones empty?". The answer to this question is that redundant qubits will not have to be removed physically. It is sufficient to measure each of them in the σ_z-eigenbasis, as will be described in Section 2.2.3.

Second, "How can the described procedures for gate simulation be concatenated such that they represent a measurement based simulation of an entire circuit?". It seems at first sight that the described building blocks would only lead to a computational scheme consisting of repeated steps of entangling operations and measurements. This is not the case. As will be shown in Section 2.2.4, the three procedures stated are precisely of such a form that the described measurement-based scheme of quantum computation can be decomposed into them.

The third question is: "How does one deal with the randomness of the measurement results that leads to the byproduct operators (2.23), (2.29) and (2.30)?". The appearance of byproduct operators may suggest that there is a need for local correction operations to counteract these unwanted extra operators. However, there is neither a possibility for such counter rotations within the described model of quantum computation, nor is there a need. The scheme works with unit efficiency despite the randomness of the individual measurement results, as will be discussed in Section 2.2.5.

2.2.3 Removing the redundant cluster qubits

A cluster state on a two-dimensional cluster of rectangular shape, say, is a resource that allows for any computation that fits on the cluster. If one realizes a certain quantum circuit on this cluster state, there will always be qubits on the cluster which are not needed for its realization. Such cluster qubits we call redundant for this particular circuit.

In the description of the $\mathrm{QC}_{\mathcal{C}}$ as a quantum logic network, the first step of each computation will be to remove these redundant cluster qubits. Fortunately, the situation is not such that we have to remove the qubits (or, more precisely, the carriers of the qubits) physically from the lattice. To make them ineffective to the realized circuit, it suffices to measure each of them in the σ_z-eigenbasis. In this way, one is left with an entangled quantum state on the cluster $\mathcal{C}_N$ of the unmeasured qubits and a product state on $\mathcal{C}\backslash\mathcal{C}_N$,

$$|\phi_{\{\kappa\}}\rangle_{\mathcal{C}} \longrightarrow |Z\rangle_{\mathcal{C}\backslash\mathcal{C}_N} \otimes |\phi_{\{\kappa'\}}\rangle_{\mathcal{C}_N}, \tag{2.31}$$

with $|Z\rangle_{\mathcal{C}\setminus\mathcal{C}_N} = \left(\bigotimes_{i\in\mathcal{C}\setminus\mathcal{C}_N} |s_i\rangle_{i,z}\right)$ and s_i the results of the σ_z-measurements. The resulting entangled state $|\phi_{\{\kappa'\}}\rangle_{\mathcal{C}_N}$ on the sub-cluster $\mathcal{C}_N$ is again a cluster state obeying the set of equations (2.1), and the measurement outcomes determine the sign factors therein. This can be easily seen with stabilizer methods [41], [20]. Nevertheless, for completeness we give the argument here. First, by definition we have

$$|Z\rangle_{\mathcal{C}\setminus\mathcal{C}_N} \otimes |\phi_{\{\kappa'\}}\rangle_{\mathcal{C}_N} = \left(\bigotimes_{i\in\mathcal{C}\setminus\mathcal{C}_N} \frac{\mathbb{1}^{(i)} + (-1)^{s_i}\sigma_z^{(i)}}{2} \right) |\phi_{\{\kappa\}}\rangle_{\mathcal{C}}. \tag{2.32}$$

Using the eigenvalue equations (2.1), we now insert a correlation operator $K^{(a)}$ with $a \in \mathcal{C}_N$ into the r.h.s of (2.32) between the projector and the state, and obtain

$$|Z\rangle_{\mathcal{C}\setminus\mathcal{C}_N} \otimes |\phi_{\{\kappa'\}}\rangle_{\mathcal{C}_N} = (-1)^{\kappa'_a} K'^{(a)} |Z\rangle_{\mathcal{C}\setminus\mathcal{C}_N} \otimes |\phi_{\{\kappa'\}}\rangle_{\mathcal{C}_N}, \tag{2.33}$$

with the correlation operators

$$K'^{(a)} = \sigma_x^{(a)} \bigotimes_{c\in \mathrm{nbgh}(a)\cap\mathcal{C}_N} \sigma_z^{(c)}, \tag{2.34}$$

and the set $\{\kappa'_a\}$ specifying the eigenvalues

$$\kappa'_a = \left(\kappa_a + \sum_{b\in \mathrm{nbgh}(a)\cap(\mathcal{C}\setminus\mathcal{C}_N)} s_b \right) \mod 2. \tag{2.35}$$

As the new correlation operators $K'^{(a)}$ in (2.33) only act on the cluster qubits in $\mathcal{C}_N$, the states $|\phi_{\{\kappa'\}}\rangle_{\mathcal{C}_N}$ again obey eigenvalue equations of type (2.1), i.e.

$$K'^{(a)} |\phi_{\{\kappa'\}}\rangle_{\mathcal{C}_N} = (-1)^{\kappa'_a} |\phi_{\{\kappa'\}}\rangle_{\mathcal{C}_N}, \ \forall\ a \in \mathcal{C}_N. \tag{2.36}$$

There are $|\mathcal{C}_N|$ such eigenvalue equations for a state of $|\mathcal{C}_N|$ qubits. Thus, the state $|\phi_{\{\kappa'\}}\rangle_{\mathcal{C}_N}$ is specified by (2.36) up to a global phase.

From (2.35) we find that the redundant qubits have some remaining influence on the process of computation. After they have been measured, the random measurement results enter into the eigenvalues that specify the residual cluster state $|\phi_{\{\kappa'\}}\rangle_{\mathcal{C}_N}$ on the cluster $\mathcal{C}_N$. However, from (2.14) it follows that $|\phi_{\{\kappa'\}}\rangle_{\mathcal{C}_N}$ is equivalent to $|\phi\rangle_{\mathcal{C}_N}$ modulo local σ_z rotations. These can be accounted for by absorbing them into the subsequent measurements.

In this way, a $\mathrm{QC}_{\mathcal{C}}$-computation with arbitrary $\{\kappa'_a\}$ may always be traced back to the case of $\{\kappa'_a = 0 \,|\, \forall\, a \in \mathcal{C}_N\}$, and we therefore adopt the following two rules to simplify the further discussion:

1. *The redundant cluster qubits are discarded. We only consider the sub-cluster* $\mathcal{C}_N$.
2. *We assume that* $\kappa'_a = 0$ *for all* $a \in \mathcal{C}_N$.

(2.37)

2.2.4 Concatenation of gate simulations

A quantum circuit on the $\mathrm{QC}_{\mathcal{C}}$ is a spatial and temporal pattern of measurements on individual qubits which have previously been entangled to form a cluster state. To better understand its functioning we would like –as in the network model of quantum computation– to decompose the circuit into basic building blocks. These building blocks should be such that out of them any circuit can be assembled. In explaining the $\mathrm{QC}_{\mathcal{C}}$ in a network language, we can relate the building blocks of a quantum logic network –the quantum gates– to building blocks of $\mathrm{QC}_{\mathcal{C}}$-circuits. To do so, we need to prove that, in a $\mathrm{QC}_{\mathcal{C}}$-computation, measurement patterns representing the gates can be patched together like the quantum gates themselves. This proof is given next.

To realize a gate g on the $\mathrm{QC}_{\mathcal{C}}$ consider a cluster $\mathcal{C}(g)$. This cluster has an input section $\mathcal{C}_I(g)$, a body $\mathcal{C}_M(g)$ and an output section $\mathcal{C}_O(g)$, with

$$\begin{aligned} \mathcal{C}_I(g) \cup \mathcal{C}_M(g) \cup \mathcal{C}_O(g) &= \mathcal{C}(g) \\ \mathcal{C}_I(g) \cap \mathcal{C}_M(g) &= \emptyset \\ \mathcal{C}_I(g) \cap \mathcal{C}_O(g) &= \emptyset \\ \mathcal{C}_M(g) \cap \mathcal{C}_O(g) &= \emptyset. \end{aligned} \tag{2.38}$$

The measurement bases of the qubits in $\mathcal{C}_M(g)$, the body of the gate g, encode g. The general scheme for procedures to realize a gate g on a cluster $\mathcal{C}(g)$, for which examples have been given with Procedures 1-3 for the CNOT gate and the rotations, is

Scheme 1 Simulation of the gate g on $\mathcal{C}(g)$, acting on the input state $|\psi\rangle_{\mathrm{in}}$.

1. Prepare the input state $|\psi_{\mathrm{in}}\rangle$ on $\mathcal{C}_I(g)$ and the qubits in $\mathcal{C}_M(g) \cup \mathcal{C}_O(g)$ individually in the state $|+\rangle = |0\rangle_x$ such that the quantum state of all qubits in $\mathcal{C}(g)$ becomes

$$|\Psi_{\mathrm{in}}\rangle_{\mathcal{C}(g)} = |\psi_{\mathrm{in}}\rangle_{\mathcal{C}_I(g)} \otimes \bigotimes_{k \in \mathcal{C}_M(g) \cup \mathcal{C}_O(g)} |+\rangle_k. \tag{2.39}$$

2. Entangle $|\Psi_{\mathrm{in}}\rangle_{\mathcal{C}(g)}$ by the interaction

$$S^{(\mathcal{C}(g))} = \prod_{a,b \in \mathcal{C}(g) \,|\, b-a \in \gamma_d} S^{ab}, \tag{2.40}$$

   such that the resulting quantum state is $|\Psi_{\varepsilon}\rangle_{\mathcal{C}_N} = S^{(\mathcal{C}(g))} |\Psi_{\mathrm{in}}\rangle_{\mathcal{C}(g)}$.

3. Measure the cluster qubits in $\mathcal{C}_I(g) \cup \mathcal{C}_M(g)$, i.e. choose measurement bases specified by $\vec{r}_k \in S^2,\ k \in \mathcal{C}_I(g) \cup \mathcal{C}_M(g)$ and obtain the random measurement results s_k such that the projector

$$P^{(\mathcal{C}_I(g) \cup \mathcal{C}_M(g))} = \bigotimes_{k \in \mathcal{C}_I(g) \cup \mathcal{C}_M(g)} \frac{\mathbb{1}^{(k)} + (-1)^{s_k} \vec{r}_k \cdot \vec{\sigma}^{(k)}}{2} \tag{2.41}$$

   is applied. Thereby the state $|\Psi_{\mathrm{out}}\rangle_{\mathcal{C}(g)}$ is obtained.

Putting all three steps of Scheme 1 together, the relation between $|\Psi_{\text{in}}\rangle_{\mathcal{C}(g)}$ and $|\Psi_{\text{out}}\rangle_{\mathcal{C}(g)}$ is

$$|\Psi_{\text{out}}\rangle_{\mathcal{C}(g)} = P^{(\mathcal{C}_I(g)\cup\mathcal{C}_M(g))}\, S^{(\mathcal{C}(g))}\, |\Psi_{\text{in}}\rangle_{\mathcal{C}(g)}. \tag{2.42}$$

As we will show later, the state $|\Psi_{\text{out}}\rangle_{\mathcal{C}(g)}$ has the form

$$|\Psi_{\text{out}}\rangle_{\mathcal{C}(g)} = \left(\bigotimes_{k\in\mathcal{C}_I(g)\cup\mathcal{C}_M(g)} |s_k\rangle_{k,\vec{r}_k} \right) \otimes |\psi_{\text{out}}\rangle_{\mathcal{C}_O(g)}, \tag{2.43}$$

where $|s_k\rangle_{k,\vec{r}_k}$ denotes the state of the qubit k after the observable $\vec{r}_k\cdot\vec{\sigma}^{(k)}$ has been measured and the measurement outcome was s_k, and

$$|\psi_{\text{out}}\rangle = U_{\Sigma,g} U_g |\psi_{\text{in}}\rangle. \tag{2.44}$$

Therein, U_g is the desired unitary operation, and the byproduct operator $U_{\Sigma,g}$ is an extra multi-local rotation that depends on the measurement results $\{s_k \mid k \in \mathcal{C}_I(g)\cup\mathcal{C}_M(g)\}$. The byproduct operator is always in the Pauli group, i.e.

$$U_{\Sigma,g} = \bigotimes_{i=1}^{n} \left(\sigma_x^{[i]}\right)^{x_i} \left(\sigma_z^{[i]}\right)^{z_i} \tag{2.45}$$

modulo a possible global phase, and n is the number of logical qubits. In (2.45) the $\sigma^{[i]}$ denote Pauli operators acting on the *logical* qubit i, not cluster qubit. The values $x_i, z_i \in \{0,1\}$ are computed from the outcomes of the measurements by which the respective gate is realized.

As will be proved in Section 2.2.6, each gate may be realized only modulo a subsequent byproduct operator $U_{\Sigma,g}$, The byproduct operator is random, but known from the outcomes of the measurements which realize the gate. This knowledge is sufficient to drive the $\text{QC}_{\mathcal{C}}$-computation deterministically, as we will demonstrate in Section 2.2.5.

Given a quantum circuit implemented on a cluster $\mathcal{C}_N$ of qubits which is divided into two consecutive circuits. Suppose that circuit g_1 is implemented on the sub-cluster $\mathcal{C}(g_1)$ and the subsequent circuit g_2 is implemented on the sub-cluster $\mathcal{C}(g_2)$, such that $\mathcal{C}_N = \mathcal{C}(g_1)\cup\mathcal{C}(g_2)$. There is an overlap between $\mathcal{C}(g_1)$ and $\mathcal{C}(g_2)$ which consists of the output qubits of circuit 1 (identical to the input qubits of circuit 2), $\mathcal{C}_O(g_1) = \mathcal{C}_I(g_2) = \mathcal{C}(g_1)\cap\mathcal{C}(g_2)$. The location of the readout quantum register is $\mathcal{C}_O(g_2) \subset \mathcal{C}(g_2)$.

Now compare the following two strategies. Strategy i) consists of the following steps: (1) write input and entangle all qubits of $\mathcal{C}_N$; (2) measure qubits in $\mathcal{C}_N\backslash\mathcal{C}_O(g_2)$, to implement the circuit except of the readout measurements. Strategy ii) consists of the steps (1) write input and entangle the qubits on $\mathcal{C}(g_1)$; (2) measure the qubits in $\mathcal{C}(g_1)\backslash\mathcal{C}_O(g_1)$. This implements the first sub-circuit and writes the intermediate output to $\mathcal{C}_O(g_1) = \mathcal{C}_I(g_2)$; (3) entangle the qubits on $\mathcal{C}(g_2)$; (4) measure all qubits in $\mathcal{C}(g_2)\backslash\mathcal{C}_O(g_2)$. Step 3 and 4 implement the second sub-circuit, g_2, on the sub-cluster $\mathcal{C}(g_2)$. The measurements on $\mathcal{C}(g_1)\backslash\mathcal{C}_O(g_1)$, represented by the projector P_1 commute with the entanglement operation

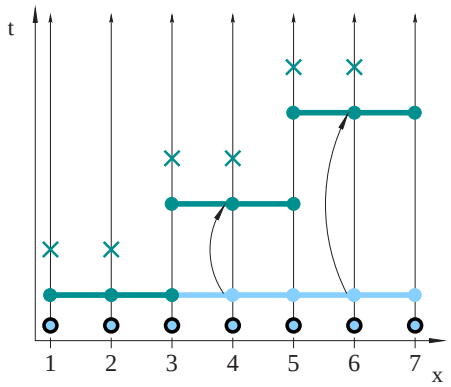


Figure 2.3: Here the exchange of the order of the measurements and the entanglement operations is shown. The crosses "×" denote the one-qubit measurements and the horizontal lines between adjacent cluster qubits denote the unitary transformations $S^{a,a+1}$.

restricted to $\mathcal{C}(g_2)$, $S^{(\mathcal{C}(g_2))} =: S_2$, $P_1 S_2 = S_2 P_1$, because these two operations act on different subsets of particles. With P_2 representing the measurements on $\mathcal{C}(g_2)\backslash\mathcal{C}_O(g_2)$ and $S_1 = S^{(\mathcal{C}(g_1))}$, it follows that $S_2 S_1 = S^{(\mathcal{C}_N)}$ and $P_2 P_1 = P^{(\mathcal{C}_N \backslash \mathcal{C}_O(g_2))}$. Therefore,

$$P_2 S_2 P_1 S_1 = P_2 P_1 S_2 S_1 = P^{(\mathcal{C}_N \backslash \mathcal{C}_O(g_2))} S^{(\mathcal{C}_N)}. \tag{2.46}$$

Thus, the two strategies are mathematically equivalent. The above argument can be iterated. It follows that entangling the whole cluster once and subsequently performing all the measurements is equivalent to simulating a quantum logic network gate by gate. The exchange of the order of operations is illustrated in Fig. 2.3.

Now, we want to specialize to the case where the quantum input is *known* and where the quantum output is measured. This is the situation which interests us most in this thesis. Examples of such a situation are Shor's factoring algorithm [1] and Grover's search algorithm [2]. In both cases, the quantum input is $|\psi_{\mathrm{in}}\rangle = \bigotimes_{i=1}^{n} |+\rangle_i$.

Let us denote the input section of the whole cluster $\mathcal{C}$, comprising the input qubits of the network simulation, as I; and the output section, comprising the qubits of the readout quantum register, as O. As long as the quantum input is known it is sufficient to consider the state $|+\rangle_I = \bigotimes_{i\in I} |+\rangle_i$. For different but known input states $|\psi_{\mathrm{in}}\rangle_I$ one can always find a transformation U_{in} such that $|\psi_{\mathrm{in}}\rangle_I = U_{\mathrm{in}}|+\rangle_I$ and instead of realizing some unitary transformation U on $|\psi_{\mathrm{in}}\rangle_I$ one realizes $U\,U_{\mathrm{in}}$ on $|+\rangle_I$.

Preparing an input state $|+\rangle_I$ and entangling it via $S^{(\mathcal{C})}$ with the rest of the cluster $\mathcal{C}\backslash I$ is the same as creating a cluster state $|\phi\rangle_{\mathcal{C}}$ on the entire cluster $\mathcal{C} = I \cup \mathcal{C}\backslash I$, $S^{(\mathcal{C})} |+\rangle_I \otimes |+\rangle_{\mathcal{C}\backslash I} = S^{(\mathcal{C})} |+\rangle_{\mathcal{C}} = |\phi\rangle_{\mathcal{C}}$. Therefore, the entire procedure of realizing a quantum computation on the $\mathrm{QC}_{\mathcal{C}}$ amounts to

Scheme 2 Performing a computation on the $\mathrm{QC}_{\mathcal{C}}$.

1. Prepare a cluster state $|\phi_{\{\kappa\}}\rangle_{\mathcal{C}}$ of sufficient size.

2. Perform a sequence of measurements on $|\phi_{\{\kappa\}}\rangle_{\mathcal{C}}$ and obtain the result of the computation from all the measurement outcomes.

For practical realization of a $\mathrm{QC}_{\mathcal{C}}$-computation, Scheme 2 is advantageous over the mathematically equivalent sequence of gate simulations according to Scheme 1. This sequence, in turn, may be used to explain the functioning of the $\mathrm{QC}_{\mathcal{C}}$ in network terminology.

2.2.5 Randomness of the measurement results

We will now show that the described scheme of quantum computation with the $\mathrm{QC}_{\mathcal{C}}$ works with unit efficiency despite the randomness of the individual measurement results.

First note that a byproduct operator U_Σ that acts after the final unitary gate $U_{g_{|\mathcal{N}|}}$ does not jeopardize the scheme. Its only effect is that the results of the readout measurements have to be reinterpreted. The byproduct operator U_Σ that acts upon the logical output qubits $1\,..\,n$ has the form

$$U_\Sigma = \prod_{i=1}^{n} \left(\sigma_x^{[i]}\right)^{x_i} \left(\sigma_z^{[i]}\right)^{z_i}, \tag{2.47}$$

where $x_i, z_i \in \{0,1\}$ for $1 \leq i \leq n$. Let the qubits on the cluster which are left unmeasured be labeled in the same way as the readout qubits of the quantum logic network.

The qubits on the cluster which take the role of the readout qubits are, at this point, in a state $U_\Sigma|\mathrm{out}\rangle$, where $|\mathrm{out}\rangle$ is the output state of the corresponding quantum logic network. The computation is completed by measuring each qubit in the σ_z-eigenbasis, thereby obtaining the measurement results $\{s_i'\}$, say. In the $\mathrm{QC}_{\mathcal{C}}$-scheme, one measures the state $U_\Sigma|\mathrm{out}\rangle$ directly, whereby outcomes $\{s_i\}$ are obtained and the readout qubits are projected into the state $|\mathcal{M}\rangle = \prod_{i=1}^{n} \frac{\mathbb{1}^{(i)}+(-1)^{s_i}\sigma_z^{(i)}}{2}\, U_\Sigma|\mathrm{out}\rangle$. Depending on the byproduct operator U_Σ, the set of measurement results $\{s\}$ in general has a different interpretation from what the network readout $\{s_i'\}$ would have. The measurement basis is the same. From (2.47) one obtains

$$\begin{aligned} |\mathcal{M}\rangle &= \prod_{i=1}^{n} \frac{\mathbb{1}^{(i)} + (-1)^{s_i}\sigma_z^{(i)}}{2}\, U_\Sigma|\mathrm{out}\rangle \\ &= U_\Sigma \left(U_\Sigma^\dagger \prod_{i=1}^{n} \frac{\mathbb{1}^{(i)} + (-1)^{s_i}\sigma_z^{(i)}}{2} U_\Sigma \right) |\mathrm{out}\rangle \\ &= U_\Sigma \prod_{i=1}^{n} \frac{\mathbb{1}^{(i)} + (-1)^{s_i+x_i}\sigma_z^{(i)}}{2} |\mathrm{out}\rangle \end{aligned} \tag{2.48}$$

From (2.48) we see that a σ_z-measurement on the state $U_\Sigma\,|\mathrm{out}\rangle$ with results $\{s\}$ represents the same algorithmic output as a σ_z-measurement of the state $|\mathrm{out}\rangle$ with the results $\{s_i'\}$, where the sets $\{s\}$ and $\{s_i'\}$ are related by

$$s_i' \equiv s_i + x_i \bmod 2. \tag{2.49}$$

The set $\{s_i'\}$ represents the result of the computation. It can be calculated from the results $\{s_i\}$ of the σ_z-measurements on the "readout" cluster qubits, and the values $\{x_i\}$ which are determined by the byproduct operator U_Σ.

Let us now discuss the sequence of the individual gate simulations. Because of (2.44) and the argument presented in Section 2.2.4, the quantum output $|\psi_{\mathrm{out}}\rangle$ of a whole sequence of unitary gates is related to the respective input via

$$|\psi_{\mathrm{out}}\rangle = \left(\prod_{i=1}^{|\mathcal{N}|} U_{\Sigma,g_i} U_{g_i} \right) |\psi_{\mathrm{in}}\rangle, \tag{2.50}$$

where the gates $g_i \in \mathcal{N}$ are labeled corresponding to the order of their action.

Thus we find that one can cope with the randomness of the measurement results provided the byproduct operators U_{Σ,g_i} in (2.50) can be propagated forward through the subsequent gates such that they act on the cluster qubits representing the output register. This can be done. To propagate the byproduct operators we use the propagation relations

$$\begin{array}{rcl} \mathrm{CNOT}(c,t)\sigma_x^{(t)} &=& \sigma_x^{(t)}\mathrm{CNOT}(c,t) \\ \mathrm{CNOT}(c,t)\sigma_x^{(c)} &=& \sigma_x^{(c)}\sigma_x^{(t)}\mathrm{CNOT}(c,t) \\ \mathrm{CNOT}(c,t)\sigma_z^{(t)} &=& \sigma_z^{(c)}\sigma_z^{(t)}\mathrm{CNOT}(c,t) \\ \mathrm{CNOT}(c,t)\sigma_z^{(c)} &=& \sigma_z^{(c)}\mathrm{CNOT}(c,t) \end{array} \tag{2.51}$$

for the CNOT gate,

$$\begin{array}{rcl} U_{Rot}[\xi,\eta,\zeta]\sigma_x &=& \sigma_x U_{Rot}[\xi,-\eta,\zeta] \\ U_{Rot}[\xi,\eta,\zeta]\sigma_z &=& \sigma_z U_{Rot}[-\xi,\eta,-\zeta] \end{array} \tag{2.52}$$

for general rotations $U_{Rot}[\xi,\eta,\zeta]$ as defined in (2.24), and

$$\begin{array}{rcl} H\sigma_x &=& \sigma_z H \\ H\sigma_z &=& \sigma_x H \\ U_z[\pi/2]\sigma_x &=& \sigma_y U_z[\pi/2] \\ U_z[\pi/2]\sigma_z &=& \sigma_z U_z[\pi/2] \end{array} \tag{2.53}$$

for the Hadamard- and $\pi/2$-phase gate. The propagation relations (2.52) apply to general rotations realized via Procedure 2 –including Hadamard- and $\pi/2$-phase gates– while the propagation relations (2.53) apply to Hadamard- and $\pi/2$-phase gates as realized via Procedure 3.

Note that the propagation relations (2.51) - (2.53) are such that Pauli operators are mapped onto Pauli operators under propagation and thus the byproduct operators remain in the Pauli group when being propagated. Further note that there is a difference between the relations for propagation through gates which are in the Clifford group and through those which are not. For CNOT-, Hadamard- and $\pi/2$-phase gates the byproduct operator changes under propagation while the gate remains unchanged. This holds for all gates in the Clifford group, because the propagation relations for Clifford gates are of the form $U_g U_\Sigma = (U_g U_\Sigma U_g^{-1}) U_g$ as (2.51) and (2.53), i.e. the byproduct operator U_Σ is conjugated

under the gate, and the Clifford group by its definition as the normalizer of the Pauli group maps Pauli operators onto Pauli operators under conjugation. The propagation relations (2.51) and (2.53) are identical to the propagation relations for Pauli errors given in [43]. For gates which are not in the Clifford group conjugation of the byproduct operator under the gate would in general not work and therefore, for rotations which are not in the Clifford group, the propagation relations are different. There, the gate is conjugated under the byproduct operator; and thus the byproduct operator remains unchanged in propagation while the gate is modified. In both cases, the forward propagation leaves the byproduct operators in the Pauli group. In particular, their tensor product structure is maintained.

Let us now discuss how byproduct operator propagation affects the scheme of computation with the $\mathrm{QC}_{\mathcal{C}}$. Using the above propagation relations, (2.50) can be rewritten in the following way

$$|\psi_{\mathrm{out}}\rangle = \left(\prod_{i=1}^{|\mathcal{N}|} U_{\Sigma,g_i}|_{\Omega}\right)\left(\prod_{i=1}^{|\mathcal{N}|} U'_{g_i}\right)|\psi_{\mathrm{in}}\rangle. \tag{2.54}$$

Therein, $U_{\Sigma,g_i}|_{\Omega}$ are forward propagated byproduct operators resulting from the byproduct operators U_{Σ,g_i} of the gates g_i. They accumulate to the total byproduct operator U_{Σ} whose effect on the result of the computation is contained in (2.49),

$$U_{\Sigma} = \prod_{i=1}^{|\mathcal{N}|} U_{\Sigma,g_i}|_{\Omega}\,. \tag{2.55}$$

Further, the U'_{g_i} are the gates modified under the propagation of the byproduct operators. As discussed above, for gates in the Clifford group we have

$$U'_g = U_g,\ \forall\ g \in \text{Clifford group}, \tag{2.56}$$

as can be seen from (2.51) and (2.53).

Gates which are not in the Clifford group are modified by byproduct operator propagation. Specifically, the general rotations (2.24) are conjugated as can be seen from (2.52). From the structure of (2.50) we see that only the byproduct operators of gates g_k earlier than g_i in the network may have an effect on U_{g_i}, i.e. those with $k < i$. To give an explicit expression, let us define $U_{\Sigma,g_k}|_{\mathcal{O}_i}$, which are byproduct operators U_{Σ,g_k} propagated forward by the propagation relations (2.51) - (2.53) to the vertical cut $\mathcal{O}_i$ through the network, see Fig. 2.4. A vertical cut through a network is a cut which intersects each qubit line exactly once and does not intersect gates. The vertical cut $\mathcal{O}_i$ has the additional property that it intersects the network just before the input of gate g_i. The relation between a rotation U'_{g_i} modified by the byproduct operators and the non-modified rotation U_{g_i} is

$$\begin{aligned} U'_{g_i} \;=\; & \left(\prod_{k|k<i} U_{\Sigma,g_k}|_{\mathcal{O}_i}\right) U_{g_i} \left(\prod_{k|k<i} U_{\Sigma,g_k}|_{\mathcal{O}_i}\right)^{\dagger}, \\ & \forall\ U_{g_i} \in SU(2). \end{aligned} \tag{2.57}$$

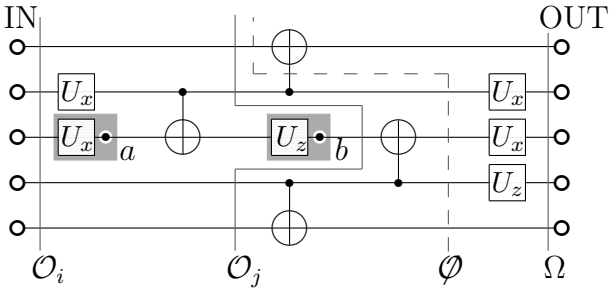


Figure 2.4: Vertical cuts. The vertical cuts intersect each qubit line exactly once but do not intersect gates. Thus, $\mathcal{O}_i$, $\mathcal{O}_j$ and Ω are vertical cuts, but $\not\mathcal{O}$ is not. The cut $\mathcal{O}_i$ intersects the rotation U_x just before its input. For two of the rotations in the displayed network, the sub-clusters on which these gates are realized are symbolically displayed in gray underlay. Via the measurement of the cluster qubits a and b (displayed as black dots with white border), the rotation angles of the respective rotations U_x and U_z are set.

Now that we have investigated the effect of byproduct operator propagation on the individual gates let us return to equation (2.54). There, we find that the operations which act on the input state $|\psi_{\text{in}}\rangle$ group into two factors. The first is composed of the modified gate operations U'_{g_i} and the second of the forward propagated byproduct operators. The second factor gives the accumulated byproduct operator U_Σ and is absorbed into the result of the computation via (2.49). It does not cause any complication.

So what remains is the first factor, and we find that the unitary evolution of the input state $|\psi_{\text{in}}\rangle$ that is realized is composed of the modified gates U'_{g_i}. The $QC_{\mathcal{C}}$gates which correspond to the gates realized in a network quantum algorithm are thus the U'_{g_i}, not U_{g_i}. The procedures 1 - 3 in Section 2.2.2, which relate a quantum gate to a measurement pattern, are for the operations U_{g_i}, though. Therefore we need to read (2.57) in reverse and deduce U_{g_i} from U'_{g_i}. This can be done only in runtime of the algorithm. Once the gates g_k for all $k < i$ have been realized, the byproduct operators $U_{\Sigma,k}$ are known for all $k < i$. With U_{g_i} determined from U'_{g_i} via (2.57), Procedure 2 then gives the measurement bases required for the realization of the gate g_i.

For proper discussion of the temporal ordering we have to step out of the network frame, which is done in Chapter 3. At this point it shall only be pointed out that in case of the $QC_{\mathcal{C}}$ the basic primitive are measurements. Thus, the temporal complexity will be determined by the temporal ordering of these measurements, unlike in quantum logic networks, where it depends on the ordering of gates. The most efficient ordering of measurements that simulates a quantum logic network is not pre-described by the temporal ordering of the gates in this network. For example, gates in the normalizer of the Pauli group, the Clifford group, do not contribute to the temporal complexity of a $QC_{\mathcal{C}}$-algorithm at all, see Section 2.2.9.

A temporal ordering among the measurements is inferred from the requirement to keep the computation on the $QC_{\mathcal{C}}$ deterministic in spite of the randomness introduced by the measurements. This causes bases for one-qubit measurements to be adapted in accordance

with outcomes obtained from the measurements of other qubits, which thus must have been performed before.

2.2.6 Using quantum correlations for quantum computation

In this section we give a criterion that allows us to demonstrate the functioning of the $\mathrm{QC}_{\mathcal{C}}$-simulations of unitary gates in a compact way. Specifically, Theorem 1 given below establishes a correspondence between general quantum gates and quantum correlations of states. Using this correspondence, the explanation of $\mathrm{QC}_{\mathcal{C}}$gates can be reduced to stabilizer manipulations.

Before we state the theorem, let us make the notion of a measurement pattern more precise. In a $\mathrm{QC}_{\mathcal{C}}$-computation one can only choose the measurement bases, while the measurement outcomes are random. This is sufficient for deterministic computation. Thus one can perform measurements specified by a spatial and temporal pattern of measurement bases but one cannot control into which of the two eigenstates the qubits are projected.

Definition 1 *A measurement pattern $\mathcal{M}^{(\mathcal{C})}$ on a cluster $\mathcal{C}$ is a set of vectors*

$$\mathcal{M}^{(\mathcal{C})} = \left\{ \vec{r}_a \in S^2 \mid a \in \mathcal{C} \right\}, \tag{2.58}$$

defining the measurement bases of the one-qubit measurements on $\mathcal{C}$.

If this pattern $\mathcal{M}^{(\mathcal{C})}$ of measurements is applied on an initial state $|\Psi_{\mathcal{E}}\rangle_{\mathcal{C}}$ and thereby the set of measurement outcomes

$$\{s\}_{\mathcal{C}} = \{s_a \in \{0,1\} \mid a \in \mathcal{C}\} \tag{2.59}$$

is obtained, then the resulting state $|\Psi_{\mathcal{M}}\rangle_{\mathcal{C}}$ is, modulo norm factor, given by $|\Psi_{\mathcal{M}}\rangle_{\mathcal{C}} = P^{(\mathcal{C})}_{\{s\}}(\mathcal{M})\,|\Psi_{\mathcal{E}}\rangle_{\mathcal{C}}$, where

$$P^{(\mathcal{C})}_{\{s\}}(\mathcal{M}) = \bigotimes_{k\in\mathcal{C}} \frac{\mathbb{1}^{(k)} + (-1)^{s_k}\vec{r}_k \cdot \vec{\sigma}^{(k)}}{2}. \tag{2.60}$$

Additionally, let us introduce some conventions for labeling. Let $\mathcal{C}_I(g)$ and $\mathcal{C}_O(g)$ be such that $|\mathcal{C}_I(g)| = |\mathcal{C}_O(g)| = n$ where n is the number of logical qubits processed by g. Operators acting on qubits $p \in \mathcal{C}_I(g)$ and $q \in \mathcal{C}_O(g)$ are labeled by upper indices $(\mathcal{C}_I(g), i)$ and $(\mathcal{C}_O(g), i')$, $1 \le i, i' \le n$, respectively. The qubits $p \in \mathcal{C}_I(g)$ and $q \in \mathcal{C}_O(g)$ are ordered from 1 to n in the same way as the logical qubits that they represent.

We make a distinction between the gate g and the unitary transformation U it realizes. The gate $g \in \mathcal{N}$ does, besides specifying the unitary transformation U, also comprise the information about the location of the gate within the network.

After these definitions and conventions we can now state the following theorem

Theorem 1 *Let $\mathcal{C}(g) = \mathcal{C}_I(g) \cup \mathcal{C}_M(g) \cup \mathcal{C}_O(g)$ with $\mathcal{C}_I(g) \cap \mathcal{C}_M(g) = \mathcal{C}_I(g) \cap \mathcal{C}_O(g) = \mathcal{C}_M(g) \cap \mathcal{C}_O(g) = \emptyset$ be a cluster for the simulation of a gate g, realizing the unitary transformation U, and $|\phi\rangle_{\mathcal{C}(g)}$ the cluster state on the cluster $\mathcal{C}(g)$.*

Suppose, the state $|\psi\rangle_{\mathcal{C}(g)} = P_{\{s\}}^{(\mathcal{C}_M(g))}(\mathcal{M})\,|\phi\rangle_{\mathcal{C}(g)}$ obeys the $2n$ eigenvalue equations

$$\begin{aligned} \sigma_x^{(\mathcal{C}_I(g),i)} \left(U\sigma_x^{(i)}U^\dagger\right)^{(\mathcal{C}_O(g))} |\psi\rangle_{\mathcal{C}(g)} &= (-1)^{\lambda_{x,i}} |\psi\rangle_{\mathcal{C}(g)} \\ \sigma_z^{(\mathcal{C}_I(g),i)} \left(U\sigma_z^{(i)}U^\dagger\right)^{(\mathcal{C}_O(g))} |\psi\rangle_{\mathcal{C}(g)} &= (-1)^{\lambda_{z,i}} |\psi\rangle_{\mathcal{C}(g)}, \end{aligned} \tag{2.61}$$

with $\lambda_{x,i}, \lambda_{z,i} \in \{0,1\}$ and $1 \leq i \leq n$.

Then, on the cluster $\mathcal{C}(g)$ the gate g acting on an arbitrary quantum input state $|\psi_{\mathrm{in}}\rangle$ can be realized according to Scheme 1 with the measurement directions in $\mathcal{C}_M(g)$ described by $\mathcal{M}^{(\mathcal{C}_M(g))}$ and the measurements of the qubits in $\mathcal{C}_I(g)$ being σ_x-measurements. Thereby, the input- and output state in the simulation of g are related via

$$|\psi_{\mathrm{out}}\rangle = U U_\Sigma\, |\psi_{\mathrm{in}}\rangle, \tag{2.62}$$

where U_Σ is a byproduct operator given by

$$U_\Sigma = \bigotimes_{(\mathcal{C}_I(g) \ni i)=1}^{n} (\sigma_z^{[i]})^{s_i + \lambda_{x,i}} (\sigma_x^{[i]})^{\lambda_{z,i}}. \tag{2.63}$$

The significance of the above theorem is that it provides a comparably simple criterion for the functioning of gate simulations on the $\mathrm{QC}_{\mathcal{C}}$. We can now base the explanation of the gates directly on the eigenvalue equations (2.1) which were also used to define the cluster states in a compact way. The quantum correlations required to explain the functioning of the gates are derived from the basic correlations (2.2) rather easily and thus the use of Theorem 1 makes the explanation of the gates more transparent and compact.

In the simulation of an individual quantum gate according to scheme 1, after reading in of the input state and the entangling operation $S^{(\mathcal{C}(g))}$, but before the measurements that realize the gate are performed, the resulting state carries the quantum input in an encoded form. This state is in general not a cluster state. It is therefore not clear a priori that cluster state correlations alone are sufficient to explain the functioning of the gate. However, this is what Theorem 1 states. To prove the functioning of a gate g on the $\mathrm{QC}_{\mathcal{C}}$ it is sufficient to demonstrate that a cluster state on $\mathcal{C}(g)$ exhibits certain quantum correlations.

Before we turn to the proof of Theorem 1 let us note that the measurements described by $P_{\{s\}}^{(\mathcal{C}_M(g))}(\mathcal{M}(g))$, as they have full rank, project the initial cluster state $|\phi\rangle_{\mathcal{C}(g)}$ into a tensor product state, $|\psi\rangle_{\mathcal{C}(g)} = |m\rangle_{\mathcal{C}_M(g)} \otimes |\psi\rangle_{\mathcal{C}_I(g) \cup \mathcal{C}_O(g)}$. Thereof only the second factor, $|\psi\rangle_{\mathcal{C}_I(g) \cup \mathcal{C}_O(g)}$, is of interest. This state alone satisfies the eigenvalue equations (2.61), and is uniquely determined by these equations. To see this, consider the state $|\psi'\rangle_{\mathcal{C}_I(g) \cup \mathcal{C}_O(g)} = U^\dagger |\psi\rangle_{\mathcal{C}_I(g) \cup \mathcal{C}_O(g)}$. It satisfies the $2n$ eigenvalue equations

$$\begin{aligned} \sigma_x^{(i,\mathcal{C}_I(g))} \sigma_x^{(i,\mathcal{C}_O(g))} |\psi'\rangle &= (-1)^{\lambda_{x,i}} |\psi'\rangle, \\ \sigma_z^{(i,\mathcal{C}_I(g))} \sigma_z^{(i,\mathcal{C}_O(g))} |\psi'\rangle &= (-1)^{\lambda_{z,i}} |\psi'\rangle, \end{aligned} \tag{2.64}$$

where we have written in short $|\psi'\rangle$ for $|\psi'\rangle_{\mathcal{C}_I(g)\cup\mathcal{C}_O(g)}$. The state $|\psi'\rangle_{\mathcal{C}_I(g)\cup\mathcal{C}_O(g)}$ is uniquely defined by the above set of commuting observables, it is a product of Bell states. Therefore, $|\psi\rangle_{\mathcal{C}_I(g)\cup\mathcal{C}_O(g)}$ is uniquely defined as well.

Proof of Theorem 1. We will discuss the functioning of the gates for two cases of inputs. First, for all input states in the computational basis. This leaves relative phases open which have to be determined. To fix them, we discuss second the input state with all qubits individually in $|+\rangle$. As we will see, from these two cases it can be concluded that the gate simulation works for all input states of the computational basis. This is sufficient because of the linearity of the applied operations; if the gate simulations work for states of the computational basis then they work for superpositions of such inputs as well.

Case 1: The input $|\psi_{\rm in}\rangle$ is one of the states of the computational basis, i.e. $|\psi_{\rm in}\rangle = |\mathbf{z}\rangle := \bigotimes_{i=1}^{n} |z_i\rangle_{z,i}$ with $z_i \in \{0,1\}$, $i = 1\,..\,n$. Then the state $|\Psi_{\rm out}(\mathbf{z})\rangle_{\mathcal{C}(g)}$ of the qubits in $\mathcal{C}$ [after performing a procedure according to Scheme 1, using a measurement pattern $\mathcal{M}^{(\mathcal{C}_M(g))}$ on the body $\mathcal{C}_M(g)$ of the gate g, and applying σ_x-measurements on $\mathcal{C}_I(g)$] is

$$\begin{aligned} &n_O(\mathbf{z})\,|\Psi_{\rm out}(\mathbf{z})\rangle_{\mathcal{C}(g)} = \\ &P_{\{s\}}^{(\mathcal{C}_I(g))}(X) P_{\{s\}}^{(\mathcal{C}_M(g))}(\mathcal{M}) S^{(\mathcal{C}(g))}\,|\mathbf{z}\rangle_{\mathcal{C}_I(g)} \otimes |+\rangle_{\mathcal{C}_M(g)\cup\mathcal{C}_O(g)}, \end{aligned} \tag{2.65}$$

with norm factors $n_O(\mathbf{z})$ that are nonzero for all $\mathbf{z}$, as we shall show later.

The input $|\mathbf{z}\rangle$ in (2.65) satisfies the equation

$$n_I(\mathbf{z})\,|\mathbf{z}\rangle = P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))} \bigotimes_{i=1}^{n} |+\rangle_i, \tag{2.66}$$

with $P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))} = \bigotimes_{i=1}^{n} \frac{\mathbf{1}^{[i]} + (-1)^{z_i} \sigma_z^{[i]}}{2}$, and $n_I(\mathbf{z}) = 1/2^{n/2}$ for all $\mathbf{z}$. Now note that $S^{(\mathcal{C}(g))}$ and $P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}$, as well as $P_{\{s\}}^{(\mathcal{C}_M(g))}(\mathcal{M})$ and $P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}$, commute. Thus, $|\Psi_{\rm out}(\mathbf{z})\rangle_{\mathcal{C}(g)}$ can be written as

$$\begin{aligned} n'_O(\mathbf{z})\,|\Psi_{\rm out}(\mathbf{z})\rangle_{\mathcal{C}(g)} = & \\ &= P_{\{s\}}^{(\mathcal{C}_I(g))}(X)\, P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}\, P_{\{s\}}^{(\mathcal{C}_M(g))}(\mathcal{M})\,|\phi\rangle_{\mathcal{C}(g)} \\ &= P_{\{s\}}^{(\mathcal{C}_I(g))}(X)\, P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}\,|\psi\rangle_{\mathcal{C}(g)}, \end{aligned} \tag{2.67}$$

where $|\psi\rangle_{\mathcal{C}(g)}$ is specified by the eigenvalue equations (2.61) in Theorem 1.

Let us, at this point, emphasize that the projections $P_{\{s\}}^{(\mathcal{C}_I(g))}(X)$ and $P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}$ in (2.67) are of very different origin. The projector $P_{\{s\}}^{(\mathcal{C}_I(g))}(X)$ describes the action of the σ_x-measurements on the qubits in $\mathcal{C}_I(g)$. These measurements are part of the procedure to realize some gate g on the cluster $\mathcal{C}(g)$. One has no control over the thereby obtained measurement outcomes $\{s\}$ specifying $P_{\{s\}}^{(\mathcal{C}_I(g))}(X)$. In contrast, the projector $P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}$ does not correspond to measurements that are performed in reality. Instead, it is introduced as an auxiliary construction that allows one to relate the processing of quantum inputs to quantum correlations in cluster states. The parameters $\mathbf{z}$ specifying the quantum input $|\mathbf{z}\rangle$ and thus the projector $P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}$ in (2.66) can be chosen freely.

The goal is to find for the state $|\Psi_{\mathrm{out}}(\mathbf{z})\rangle_{\mathcal{C}(g)}$ an expression involving the transformation U acting on the input $|\mathbf{z}\rangle$. To accomplish this, first observe that for the state on the r.h.s of (2.67) via (2.61) the following eigenvalue equations hold

$$\begin{aligned}\left(U\sigma_z^{[i]}U^\dagger\right)^{(\mathcal{C}_O)}\left[P_{\{s\}}^{(\mathcal{C}_I(g))}(X)\,P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}\,|\psi\rangle_{\mathcal{C}(g)}\right] = \\ (-1)^{\lambda_{z,i}+z_i}\left[P_{\{s\}}^{(\mathcal{C}_I(g))}(X)\,P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}\,|\psi\rangle_{\mathcal{C}(g)}\right],\end{aligned} \tag{2.68}$$

with $i = 1..n$.

To make use of the equations (2.68) we need to prove that $P_{\{s\}}^{(\mathcal{C}_I(g))}(X)\,P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}\,|\psi\rangle_{\mathcal{C}(g)} \neq 0$ for all $\mathbf{z}$ under the assumptions of Theorem 1.

For this, we consider the scalar ${}_{\mathcal{C}(g)}\langle\psi|P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}|\psi\rangle_{\mathcal{C}(g)}$ and write $P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}$ in the form

$$P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))} = \frac{1}{2^n}\left(\mathbb{1} + \sum_{I_k\in P(\mathcal{C}_I)\setminus\emptyset}\bigotimes_{i\in I_k}(-1)^{z_i}\sigma_z^{(i)}\right), \tag{2.69}$$

where $P(\mathcal{C}_I)$ is the power set of $\mathcal{C}_I$. For each I_k we choose an $i \in I_k$ and insert the respective eigenvalue equation from the upper line of (2.61) into ${}_{\mathcal{C}(g)}\langle\psi|\bigotimes_{j\in I_k}\sigma_z^{(j)}|\psi\rangle_{\mathcal{C}(g)}$. Since $\bigotimes_{j\in I_k}\sigma_z^{(j)}$ and $\sigma_x^{(i,\mathcal{C}_I(g))}\left(U\sigma_x^{(i)}U^\dagger\right)^{(\mathcal{C}_O(g))}$ anti-commute, ${}_{\mathcal{C}(g)}\langle\psi|\bigotimes_{j\in I_k}\sigma_z^{(i)}|\psi\rangle_{\mathcal{C}(g)} = 0$ for all I_k. Thus, with (2.69), one finds ${}_{\mathcal{C}(g)}\langle\psi|P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}|\psi\rangle_{\mathcal{C}(g)} = 1/2^n$, such that $P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}|\psi\rangle_{\mathcal{C}(g)} \neq 0$ and therefore also

$$P_{\{s\}}^{(\mathcal{C}_I(g))}(X)\,P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}\,|\psi\rangle_{\mathcal{C}(g)} \neq 0, \tag{2.70}$$

or, in other words, $n'_O(\mathbf{z}) \neq 0$ for all $\mathbf{z}$.

Due to the fact that the projections $P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}$ and $P_{\{s\}}^{(\mathcal{C}_M(g))}(\mathcal{M})$ are of full rank the above state has the form

$$\begin{aligned}P_{\{s\}}^{(\mathcal{C}_I(g))}(X)\,P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}\,|\psi\rangle_{\mathcal{C}(g)} = \\ n'_O(\mathbf{z})\,|\mathbf{s}\rangle_{x,\mathcal{C}_I(g)}\otimes|m\rangle_{\mathcal{C}_M(g)}\otimes|\psi_{\mathrm{out}}(\mathbf{z})\rangle_{\mathcal{C}_O(g)},\end{aligned} \tag{2.71}$$

where $|\mathbf{s}\rangle_{x,\mathcal{C}_I} = \bigotimes_{(\mathcal{C}_I\ni i)=1}^{n}|s_i\rangle_{x,i}$, and $|m\rangle_{\mathcal{C}_M(g)}$ is some product state with $\|\,|m\rangle_{\mathcal{C}_M(g)}\| = 1$. Elaborating the argument that leads to (2.70) one finds that $n'_O(\mathbf{z}) = 1/2^n$ and $n_O(\mathbf{z}) = 1/2^{n/2}$, but at this point the precise values of the normalization factors are not important as long as they are nonzero.

In (2.71) only the third factor of the state on the r.h.s. is interesting, and this factor is determined by the eigenvalue equations (2.68):

$$|\psi_{\mathrm{out}}(\mathbf{z})\rangle = e^{i\eta(\mathbf{z})}UU_\Sigma|\mathbf{z}\rangle, \tag{2.72}$$

where U_Σ is given by (2.63). Now, because of (2.67) with $n'_O(\mathbf{z}) \neq 0\ \forall\,\mathbf{z}$, a solution (2.71) with (2.72) for the state $P_{\{s\}}^{(\mathcal{C}_I(g))}(X)\,P_{Z,\mathbf{z}}^{(\mathcal{C}_I(g))}\,|\psi\rangle_{\mathcal{C}(g)}$ is also a solution for the state $|\Psi_{\mathrm{out}}(\mathbf{z})\rangle_{\mathcal{C}(g)}$, and one finally obtains

$$|\Psi_{\mathrm{out}}(\mathbf{z})\rangle_{\mathcal{C}(g)} = e^{i\eta(\mathbf{z})}|\mathbf{s}\rangle_{x,\mathcal{C}_I(g)}\otimes|m\rangle_{\mathcal{C}_M(g)}\otimes\left[UU_\Sigma|\mathbf{z}\rangle\right]_{\mathcal{C}_O(g)}. \tag{2.73}$$

There appear no additional norm factors in (2.73) because the states on the l.h.s. and the r.h.s. are both normalized to unity.

The solution (2.73) still allows for one free parameter, the phase factor $e^{i\eta(\mathbf{z})}$. Note that, a priori, the phase factors for different $\mathbf{z}$ can all be different.

This concludes the discussion of case 1. We have found in (2.73) that the realized gate acts as

$$\tilde{U} = U\, U_\Sigma\, D \tag{2.74}$$

where the gate D is diagonal in the computational basis and contains all the phases $e^{i\eta(\mathbf{z})}$. What remains is to show that $D = \mathbf{1}$ modulo a possible global phase.

Case 2. Now the same procedure is applied for the input state $|\psi_{\text{in}}\rangle = |+\rangle := \bigotimes_{i=1}^{n} |+\rangle_i$. Then, the state $|\Psi_{\text{out}}(+)\rangle_{\mathcal{C}(g)}$ that results from the gate simulation is

$$n_O(+)\, |\Psi_{\text{out}}(+)\rangle_{\mathcal{C}(g)} = P_{\{s\}}^{(\mathcal{C}_I(g))}(X)\, P_{\{s\}}^{(\mathcal{C}_M(g))}(\mathcal{M})\, |\phi\rangle_{\mathcal{C}(g)}, \tag{2.75}$$

with a nonzero norm factor $n_O(+)$. Using the upper line of eigenvalue equations (2.61), the state $|\Psi_{\text{out}}(+)\rangle_{\mathcal{C}(g)}$ is found to obey the eigenvalue equations

$$\left(U\sigma_x^{[i]}U^\dagger\right)^{(\mathcal{C}_O(g))} |\Psi_{\text{out}}(+)\rangle_{\mathcal{C}(g)} = (-1)^{\lambda_{x,i}+s_i} |\Psi_{\text{out}}(+)\rangle_{\mathcal{C}(g)}. \tag{2.76}$$

The eigenvalue equations (2.76) in combination with (2.75) imply that

$$|\Psi_{\text{out}}(+)\rangle_{\mathcal{C}(g)} = e^{i\chi} |\mathbf{s}\rangle_{x,\mathcal{C}_I(g)} \otimes |m\rangle_{\mathcal{C}_M(g)} \otimes [UU_\Sigma|+\rangle]_{\mathcal{C}_O(g)}, \tag{2.77}$$

with χ being a free parameter. Therefore, on the input state $|+\rangle$ the gate simulation acts as

$$\tilde{U} = e^{i\chi}\, U\, U_\Sigma. \tag{2.78}$$

This observation concludes the discussion of case 2.

The fact that (2.73) and (2.77) hold simultaneously imposes stringent conditions on the phases $\eta(\mathbf{z})$. To see this, let us evaluate the scalar product

$$c_\chi = {}_{\mathcal{C}(g)}\langle \Psi_{\text{out}}(+)| UU_\Sigma |\mathbf{s}\rangle_{x,\mathcal{C}_I(g)} \otimes |m\rangle_{\mathcal{C}_M(g)} \otimes |+\rangle_{\mathcal{C}_O(g)}. \tag{2.79}$$

From (2.77) it follows immediately that

$$c_\chi = e^{-i\chi}. \tag{2.80}$$

On the other hand, since $|+\rangle = 1/2^{n/2} \sum_{\mathbf{z}\in\{0,1\}^n} |\mathbf{z}\rangle$ and, by linearity,

$$|\Psi_{\text{out}}(+)\rangle = 1/2^{n/2} \sum_{\mathbf{z}\in\{0,1\}^n} |\Psi_{\text{out}}(\mathbf{z})\rangle, \tag{2.81}$$

from (2.73) it follows that

$$c_\chi = \frac{1}{2^n} \sum_{\mathbf{z}\in\{0,1\}^n} e^{-i\eta(\mathbf{z})}. \tag{2.82}$$

The sum in (2.82) runs over 2^n terms. Thus, with $|e^{-i\eta(\mathbf{z})}| = 1$ for all $\mathbf{z}$, it follows from the triangle inequality that $|c_\chi| \leq 1$. The modulus of c_χ can be unity only if all $e^{-i\eta(\mathbf{z})}$ are equal. As (2.80) shows, $|c_\chi|$ is indeed equal to unity. Therefore, the phase factors $e^{i\eta(\mathbf{z})}$ must all be the same, and with (2.80) and (2.82),

$$e^{i\eta(\mathbf{z})} = e^{i\chi}, \ \forall\, \mathbf{z}. \tag{2.83}$$

If we now insert (2.83) into (2.73) we find that the gate simulation acts upon every input state in the computational basis, and thus upon every input state, as $\tilde{U}_g = e^{i\chi} U\, U_\Sigma$. Therein, the global phase factor $e^{i\chi}$ has no effect. Thus we find that the gate simulation indeed acts as stated in (2.62) and (2.63). □

We would like to acknowledge that a similar theorem restricted to gates in the Clifford group has been obtained in [55].

Let us conclude this section with some comments on how to use this theorem. First, note that *Theorem 1 does not state anything about the temporal order of measurements within a gate simulation.* In particular it should be understood that it does not imply that first the measurements on the cluster qubits in $\mathcal{C}_M(g)$ and thereafter the measurements in $\mathcal{C}_I(g)$ are performed.

Instead, first all those cluster qubits $q \in \mathcal{C}_I(g) \cup \mathcal{C}_M(g)$ are measured whose measurement basis is the eigenbasis of either σ_x or σ_y (remember that, after the removal of the redundant cluster qubits as described in Section 2.2.3, we are dealing with clusters $\mathcal{C}_N$ such that, apart from the readout, no measurements in the σ_z-eigenbasis occur). Second, possibly in several subsequent rounds, the remaining measurements are performed in bases which are chosen according to previous measurement results.

In subsequent sections we will illustrate in a number of examples how Theorem 1 is used to demonstrate the functioning of quantum gate simulations on the $\mathrm{QC}_{\mathcal{C}}$, and how the strategies for adapting the measurement bases are found.

2.2.7 Functioning of the CNOT gate and general one-qubit rotations

In this section, we demonstrate that the measurement patterns which we have introduced do indeed realize the desired quantum logic gates.

The basis for all our considerations is the set (2.1) of eigenvalue equations fulfilled by the cluster states. Therefore let us, before we turn to the realization of the gates in the universal set, describe how the eigenvalue equations can be manipulated. Equations (2.1) are not the only eigenvalue equations satisfied by the cluster state. Instead, a vast number of other eigenvalue equations can be derived from them.

The operators $K^{(a)}$ may for example be added, multiplied by a scalar and multiplied with each other. In this way, a large number of eigenvalue equations can be generated from equations (2.1). Note, however, that not all operators generated in this way are correlation operators. Non-Hermitian operators can be generated which do not represent observables, yet will prove to be useful for the construction of new correlation operators.

Furthermore, if quantum correlation operator K for state $|\phi\rangle$ commutes with measured observable $\vec{r}_i \cdot \vec{\sigma}^{(i)}$, the correlation will still apply to the measured state. More specifically, if the state $|\phi\rangle$ satisfies the eigenvalue equation $K|\phi\rangle = \lambda|\phi\rangle$ and $[K, \vec{r}_i \cdot \vec{\sigma}] = 0$, then the state resulting from the measurement, $P_{s_i}^{(i)}|\phi\rangle$, where $P_{s_i}^{(i)} = \frac{\mathbf{1}^{(i)} + (-1)^{s_i} \vec{r}_i \cdot \vec{\sigma}^{(i)}}{2}$, satisfies the same eigenvalue equation since $\lambda(P_{s_i}^{(i)}|\phi\rangle) = (P_{s_i}^{(i)} K|\phi\rangle) = K(P_{s_i}^{(i)}|\phi\rangle)$. Thus the correlation K is inherited to the resultant state, $P_{s_i}^{(i)}|\phi\rangle$.

To demonstrate and explain the measurement patterns realizing certain quantum gates, the program is as follows. First, from the set of eigenvalue equations which define the cluster state $|\phi\rangle_{\mathcal{C}(g)}$, we derive a set of eigenvalue equations which is compatible with the measurement pattern on $\mathcal{C}_M$. Then, we use these to deduce the set of eigenvalue equations which define the state $|\psi\rangle_{\mathcal{C}(g)}$, where the qubits in $\mathcal{C}_M$ have been measured. Thus we demonstrate that the assumptions for Theorem 1, that is the set of equations (2.61), are satisfied with the appropriate unitary transformation U. Third, U_Σ is obtained from equation (2.63) as a function of the measurement results. The order of U and U_Σ is then interchanged and, in this way, the temporal ordering of the measurements becomes apparent.

Identity gate

As a simple example, let us first consider a gate which realizes the identity operation $\mathbb{1}$ on a single logical qubit.

For the identity gate $\mathcal{C}_I$, $\mathcal{C}_M$ and $\mathcal{C}_O$ each consist of a single qubit, so labeling the qubits 1, 2 and 3, $1 \in \mathcal{C}_I$, $2 \in \mathcal{C}_M$ and $3 \in \mathcal{C}_O$. The pattern $\mathcal{M}(\mathbb{1})$ corresponds to a measurement of qubit 2 in the σ_x basis.

Let $|\phi\rangle_{\mathcal{C}(\mathbf{1})}$ be the cluster state on these three qubits. The state is defined by the following set of eigenvalue equations.

$$\sigma_x^{(1)}\sigma_z^{(2)} \quad |\phi\rangle_{\mathcal{C}(\mathbf{1})} = |\phi\rangle_{\mathcal{C}(\mathbf{1})}, \tag{2.84a}$$

$$\sigma_z^{(1)}\sigma_x^{(2)}\sigma_z^{(3)} \quad |\phi\rangle_{\mathcal{C}(\mathbf{1})} = |\phi\rangle_{\mathcal{C}(\mathbf{1})}, \tag{2.84b}$$

$$\sigma_z^{(2)}\sigma_x^{(3)} \quad |\phi\rangle_{\mathcal{C}(\mathbf{1})} = |\phi\rangle_{\mathcal{C}(\mathbf{1})}. \tag{2.84c}$$

After the measurement of qubit 2, the resulting state of the cluster is

$$|\psi\rangle_{\mathcal{C}(\mathbf{1})} = P_{x,s_2}^{(2)}|\phi\rangle_{\mathcal{C}(\mathbf{1})}, \tag{2.85}$$

where $s_2 \in \{0, 1\}$, and $P_{x,s_2}^{(2)} = \frac{\mathbf{1}^{(2)} + (-1)^{s_2}\sigma_x^{(2)}}{2}$.

$P_{x,s_2}^{(2)}$ and $\sigma_x^{(2)}$ obey the following relation,

$$P_{x,s_2}^{(2)}\sigma_x^{(2)} = (-1)^{s_2} P_{x,s_2}^{(2)}. \tag{2.86}$$

Applying $P_{x,s_2}^{(2)}$ to both sides of equation (2.84b), and using equation (2.86), one obtains for $|\psi\rangle_{\mathcal{C}(\mathbf{1})}$, defined in equation (2.85),

$$\sigma_z^{(1)}\sigma_z^{(3)} \quad |\psi\rangle_{\mathcal{C}(\mathbf{1})} = (-1)^{s_2}|\psi\rangle_{\mathcal{C}(\mathbf{1})}. \tag{2.87}$$

Also from equations (2.84a) and (2.84c) we have

$$\sigma_x^{(1)}\sigma_x^{(3)} \quad |\phi\rangle_{\mathcal{C}(\mathbf{1})} = |\phi\rangle_{\mathcal{C}(\mathbf{1})}. \tag{2.88}$$

Applying $P_{x,s_2}^{(2)}$ to both sides of this equation gives

$$\sigma_x^{(1)}\sigma_x^{(3)} \quad |\psi\rangle_{\mathcal{C}(\mathbf{1})} = |\psi\rangle_{\mathcal{C}(\mathbf{1})}. \tag{2.89}$$

Now, since qubits 1 and 3 represent the input and output qubits respectively, the assumption of Theorem 1, equation (2.61), is satisfied for $U = \mathbb{1}$. The byproduct operator U_Σ is obtained from equation (2.63), and we find that the full unitary operation realized by the gate is $\tilde{U} = \mathbb{1}\ \sigma_x^{s_2}\ \sigma_z^{s_1} = \sigma_x^{s_2}\ \sigma_z^{s_1}\mathbb{1}$.

Also note that a wire with length one ($\mathcal{C}_I(H) = 1$, $\mathcal{C}_M(H) = \emptyset$, $\mathcal{C}_O(H) = 2$), i.e. half of the above elementary wire, implements a Hadamard transformation. As in this construction the input- and output qubits lie on different sub-lattices of $\mathcal{C}$, one on the even and one on the odd sub-lattice, we do not use it in the universal set of gates. Nevertheless, this realization of the Hadamard transformation can be a useful tool in gate construction. For example, we will use it in Section 2.2.7 to construct the realization of the z-rotations out of the realization of x-rotations.

Removing unnecessary measurements

In larger measurement patterns, whenever pairs of adjacent σ_x- qubits in a wire are surrounded above and below by either vacant lattice sites or σ_z-measurements, they can be removed from the pattern without changing the logical operation of the gate. This is simple to show in the case of a linear cluster. Consider six qubits, labelled a to f, which are part of a longer line of qubits, prepared in a cluster state. Four of the eigenvalue equations which define the state are

$$\begin{aligned}
\sigma_z^{(a)}\sigma_x^{(b)}\sigma_z^{(c)}|\psi\rangle_{\mathcal{C}} &= |\psi\rangle_{\mathcal{C}},\\
\sigma_z^{(b)}\sigma_x^{(c)}\sigma_z^{(d)}|\psi\rangle_{\mathcal{C}} &= |\psi\rangle_{\mathcal{C}},\\
\sigma_z^{(c)}\sigma_x^{(d)}\sigma_z^{(e)}|\psi\rangle_{\mathcal{C}} &= |\psi\rangle_{\mathcal{C}},\\
\sigma_z^{(d)}\sigma_x^{(e)}\sigma_z^{(f)}|\psi\rangle_{\mathcal{C}} &= |\psi\rangle_{\mathcal{C}}.
\end{aligned} \tag{2.90}$$

Suppose, a measurement pattern $\mathcal{M}$ on these qubits contains measurements of the observable σ_x on qubits c and d. Measurements in the σ_x basis can be made before any other measurements in $\mathcal{M}$. If these two measurements alone are carried out, the new state fulfills the following eigenvalue equations, derived from equation (2.90) in the usual way,

$$\begin{aligned}
\sigma_z^{(a)}\sigma_x^{(b)}\sigma_z^{(e)}|\psi\rangle_{\mathcal{C}} &= (-1)^{s_d}|\psi\rangle_{\mathcal{C}},\\
\sigma_z^{(b)}\sigma_x^{(e)}\sigma_z^{(f)}|\psi\rangle_{\mathcal{C}} &= (-1)^{s_c}|\psi\rangle_{\mathcal{C}}.
\end{aligned} \tag{2.91}$$

The resulting state is therefore a cluster state from which qubits c and d have been removed, and b and e play the role of adjacent qubits. Thus, the two measurements have mapped

a cluster state onto a cluster state and thus do not contribute to the logical operation realized by $\mathcal{M}$, which, in the case where both s_c and s_d equal 0, is completely equivalent to the reduced measurement pattern $\mathcal{M}'$, from which these adjacent σ_x measurements have been removed.

One-qubit rotation around x-axis

A one-qubit rotation through an angle α about the x-axis $U_x[\alpha] = \exp[-i\alpha/2\sigma_x]$ is realized on the same three qubit layout as the identity gate. Labeling the qubits 1, 2 and 3 as in the previous section, $1 = \mathcal{C}_I$, $2 = \mathcal{C}_M$ and $3 = \mathcal{C}_O$. The measurement pattern $\mathcal{M}(U_x)$ consists of a measurement, on qubit 2, of the observable represented by the vector $\vec{r}_{xy}(\eta) = (\cos(\eta), \sin(\eta), 0)$,

$$\vec{r}_{xy}(\eta) \cdot \vec{\sigma} = \cos\eta\,\sigma_x + \sin\eta\,\sigma_y = U_z[\eta]\sigma_x U_z[-\eta], \tag{2.92}$$

whose eigenstates lie in the x-y-plane of the Bloch sphere at an angle of η to the x-axis.

The cluster state $|\phi\rangle_{\mathcal{C}(U_x)}$ is defined by equations (2.84). After the measurement of $\mathcal{M}(U_x)$, the resulting state is $|\psi\rangle_{\mathcal{C}(U_x)} = P^{(2)}_{xy(\eta)}|\phi\rangle_{\mathcal{C}(U_x)}$ where $P^{(2)}_{xy(\eta)} = \frac{\mathbf{1}^{(2)}+(-1)^{s_2}\vec{r}_{xy}(\eta)\cdot\vec{\sigma}^{(2)}}{2}$. To generate an eigenvalue equation whose operator commutes with $\vec{r}_{xy}(\eta)\cdot\vec{\sigma}$ we manipulate equation (2.84c) in the following way,

$$\sigma_z^{(2)}\sigma_x^{(3)}|\phi\rangle_{\mathcal{C}(U_x)} = |\phi\rangle_{\mathcal{C}(U_x)} \tag{2.93}$$

i.e. $$\sigma_z^{(2)}|\phi\rangle_{\mathcal{C}(U_x)} = \sigma_x^{(3)}|\phi\rangle_{\mathcal{C}(U_x)}$$

i.e. $$(\sigma_z^{(2)} - \sigma_x^{(3)})|\phi\rangle_{\mathcal{C}(U_x)} = 0$$

$\therefore$ $$\exp\left(-i\eta/2\,(\sigma_z^{(2)} - \sigma_x^{(3)})\right)|\phi\rangle_{\mathcal{C}(U_x)} = |\phi\rangle_{\mathcal{C}(U_x)} \tag{2.94}$$

tpgfig where the last equation is true for all $\eta \in [0, 2\pi]$. This takes a more useful form, if we write it in terms of one-qubit rotations,

$$U_z^{(2)}[\eta]\,U_x^{(3)}[-\eta]|\phi\rangle_{\mathcal{C}(U_x)} = |\phi\rangle_{\mathcal{C}(U_x)}. \tag{2.95}$$

We use this, and the equation

$$\sigma_z^{(1)}\sigma_x^{(2)}\sigma_z^{(3)} \quad |\phi\rangle_{\mathcal{C}(U_x)} = |\phi\rangle_{\mathcal{C}(U_x)} \tag{2.96}$$

to construct the subsequent eigenvalue equation. Let us denote the operator on the l.h.s. of eigenvalue equation (2.95) as A, and the operator on the l.h.s. of (2.96) as B. With (2.95) and (2.96) it follows that $ABA^{-1}|\phi\rangle_{\mathcal{C}(U_x)} = |\phi\rangle_{\mathcal{C}(U_x)}$, i.e.

$$\begin{aligned}|\phi\rangle_{\mathcal{C}(U_x)} =& \sigma_z^{(1)}\,U_z^{(2)}[\eta]\,\sigma_x^{(2)}\,U_z^{(2)}[-\eta] \\ & U_x^{(3)}[-\eta]\,\sigma_z^{(3)}\,U_x^{(3)}[\eta]\,|\phi\rangle_{\mathcal{C}(U_x)}.\end{aligned} \tag{2.97}$$

Note that the operators A and B do not commute.

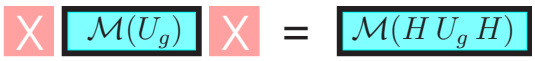


Figure 2.5: Useful identity for the realization of the rotation $U_z[\alpha]$ as the sequence $H\,U_x[\alpha]\,H$.

Applying $P_{xy(\eta),2}$ to both sides, we obtain the following eigenvalue equation for $|\psi\rangle_{\mathcal{C}(U_x)}$,

$$\sigma_z^{(1)} U_x^{(3)}[-\eta]\,\sigma_z^{(3)}\,U_x^{(3)}[\eta]\,|\psi\rangle_{\mathcal{C}(U_x)} = (-1)^{s_2}|\psi\rangle_{\mathcal{C}(U_x)}. \tag{2.98}$$

In the same way as for the identity gate we also apply the projector to an eigenvalue equation generated from equations (2.84a) and (2.84c) to obtain

$$\begin{aligned} |\psi\rangle_{\mathcal{C}(U_x)} &= \sigma_x^{(1)}\sigma_x^{(3)}|\psi\rangle_{\mathcal{C}(U_x)} \\ &= \sigma_x^{(1)}\,U_x^{(3)}[-\eta]\,\sigma_x^{(3)}\,U_x^{(3)}[\eta]\,|\psi\rangle_{\mathcal{C}(U_x)} \end{aligned} \tag{2.99}$$

and thus we see that equation (2.61) is satisfied for $U = U_x[-\eta]$ and $U_\Sigma = \sigma_z^{s_1}\sigma_x^{s_2}$. Interchanging the order of these operators is not as trivial here as for the identity gate. When σ_z is propagated through $U_x[\eta]$ the sign of the angle is reversed. We thus find that the gate operation realized by this $\mathcal{M}(U_x)$ in the $\mathrm{QC}_{\mathcal{C}}$ is

$$U_g = U_x\left[(-1)^{s_1}(-\eta)\right]. \tag{2.100}$$

The sign of the rotation realized by this gate is a function of s_1, the outcome of the measurement on qubit 1. This is an example of the temporal ordering of measurements in the $\mathrm{QC}_{\mathcal{C}}$. In order to realize $U_x[\alpha]$ deterministically, the angle of the measurement, η, on qubit 2 must be $\eta = (-1)^{s_1}(-\alpha)$. Therefore, this measurement can only be realized after the measurement of qubit 1.

Rotation around z-axis

The measurement pattern for a rotation around the z-axis $U_z[\beta] = \exp[-\mathrm{i}\beta/2\sigma_z]$ is illustrated in Fig. 2.2. It requires 5 qubits for its realization.

The measurement layout $\mathcal{M}(U_z)$ is similar to the rotation about the x-axis, except for two additional σ_x measurements on either side of the central qubit. The simplest way to understand this gate is regard it as the concatenation $U_z[\alpha] = H\,U_x[\alpha]\,H$. The Hadamard transformations may be realized as wires of length one, see Section 2.2.7. Thus, the measurement pattern of the z-rotation is that of the x-rotation plus one cluster qubit on either side measured in the eigenbasis of σ_x, as displayed in Fig 2.5.

The explanation in terms of eigenvalue equations obeyed by cluster states is as follows. Let us label the qubits 1 to 5. The cluster state $|\phi\rangle_{\mathcal{C}(U_z)}$ is defined by eigenvalue equations of the usual form. If qubits 2 and 4 are measured in the σ_x basis, the resulting state

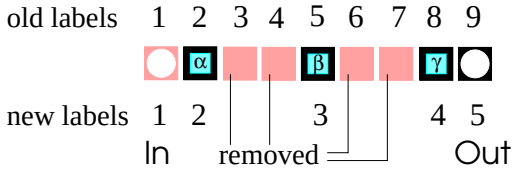


Figure 2.6: General rotation composed of two x-rotations and a z-rotation in between (Euler representation). In the $\mathrm{QC}_{\mathcal{C}}$-realization pairs of adjacent cluster qubits measured in the σ_x-eigenbasis may be removed from the measurement pattern.

$|\phi'\rangle_{\mathcal{C}(U_z)} = P^{(2)}_{x,s_2} P^{(4)}_{x,s_4} |\phi\rangle_{\mathcal{C}(U_z)}$ fulfills the following set of eigenvalue equations

$$\sigma_x^{(1)}\sigma_x^{(3)}\sigma_x^{(5)}|\phi'\rangle_{\mathcal{C}(U_z)} = |\phi'\rangle_{\mathcal{C}(U_z)}, \tag{2.101a}$$

$$\sigma_z^{(1)}\sigma_z^{(3)} \quad |\phi'\rangle_{\mathcal{C}(U_z)} = (-1)^{s_2}|\phi'\rangle_{\mathcal{C}(U_z)}, \tag{2.101b}$$

$$\sigma_z^{(3)}\sigma_z^{(5)}|\phi'\rangle_{\mathcal{C}(U_z)} = (-1)^{s_4}|\phi'\rangle_{\mathcal{C}(U_z)}. \tag{2.101c}$$

This set of equations is analogous to equations (2.84), except for the different eigenvalues and that the input and output qubits x- and z-bases have been exchanged. From here on the analysis of the measurement pattern runs parallel to the previous section.

One finds $\mathcal{M}(U_z)$ realizes the operation $U_z(\beta)$ if the basis of the measurement on qubit 3 is chosen to be the eigenbasis of $\vec{r}_{xy}((-1)^{s_2}(-\beta)) \cdot \vec{\sigma}$, where $\vec{r}_{xy}(\eta)$ is defined in equation (2.92). Qubit 2 must thus be measured prior to qubit 3. The byproduct operator for this gate is $U_{\Sigma,U_z} = \sigma_x^{s_2+s_4}\sigma_z^{s_1+s_3}$.

Arbitrary Rotation

The arbitrary Euler rotation can be realized by combining the measurement patterns of rotations around x- and z-axes by overlaying input and output qubits of adjacent patterns, as described in section 2.2.4. This creates a measurement pattern of 7 qubits plus input and output qubits, labelled as in Fig. 2.6, with measurements of σ_x on qubits 3, 4, 6 and 7, and measurements in the x-y-plane at angles α, β and γ on qubits 2, 5 and 8, respectively. The unitary operation realized by these connected measurement patterns is,

$$U_\Sigma U_{Rot}[\xi,\eta,\zeta] = \sigma_z^{s_7}\sigma_x^{s_8}U_x[(-1)^{s_7}(-\gamma)]\sigma_z^{s_3+s_5}\sigma_x^{s_4+s_6}U_z[(-1)^{s_4}(-\beta)]\sigma_z^{s_1}\sigma_x^{s_2}U_x[(-1)^{s_1}(-\alpha)] \tag{2.102}$$

As we have shown above, adjacent pairs of σ_x measurements can be removed from the pattern without changing the operation realized by the gate. The operation realized by this reduced measurement pattern is obtained by setting the measurement results from the removed qubits to zero, $s_3, s_4, s_6, s_7 = 0$. After relabelling the remaining qubits in the measurement pattern 1 to 5, we obtain

$$U_\Sigma U_{Rot}[\xi,\eta,\zeta] = \sigma_x^{s_4}U_x[-\gamma]\sigma_z^{s_3}U_z[(-\beta)]\sigma_z^{s_1}\sigma_x^{s_2}U_x[(-1)^{s_1}(-\alpha)] \tag{2.103}$$

Propagating all byproduct operators to the left hand side we find the unitary operation realized by the measurement pattern is

$$U_{Rot}[\xi,\eta,\zeta] = U_x[-(-1)^{s_1+s_3}\gamma]U_z[-(-1)^{s_2}\beta]U_x[-(-1)^{s_1}\alpha] \tag{2.104}$$

with byproduct operator $U_\Sigma = \sigma_x^{s_2+s_4}\sigma_z^{s_1+s_3}$. One finds that, to realize a specific rotation $U_{Rot}[\xi,\eta,\zeta] = U_x[\zeta]U_z[\eta]U_x[\xi]$, the angles α, β, γ specifying the measurement bases of the qubits 2,3, and 4 are again dependent on the measurement results of other qubits. We see that $\alpha = (-1)^{s_1}(-\xi)$, $\beta = (-1)^{s_2}(-\eta)$, $\gamma = (-1)^{s_1+s_3}(-\zeta)$. To realize a specific rotation deterministically, qubit 2 must thus be measured before qubits 3 and 4, and qubit 3 before qubit 4, in the bases specified in Section 2.2.2.

Hadamard- and $\pi/2$-phase gate

The Hadamard- and the $\pi/2$-phase gate have the property that under conjugation with these gates Pauli operators are mapped onto Pauli operators,

$$\begin{array}{rcl} H\sigma_x H^\dagger &=& \sigma_z, \\ H\sigma_z H^\dagger &=& \sigma_x, \end{array} \tag{2.105}$$

and

$$\begin{array}{rcl} U_z[\pi/2]\sigma_x U_z[\pi/2]^\dagger &=& \sigma_y, \\ U_z[\pi/2]\sigma_z U_z[\pi/2]^\dagger &=& \sigma_z, \end{array} \tag{2.106}$$

from which the propagation relations (2.53) follow. Related to this property is the fact that these two special rotations may be realized via σ_x- and σ_y-measurements. Such measurement bases need not be adapted to previously obtained measurement results and therefore, while these rotations might be realized in the same way as any other rotation, there is a more advantageous way to do so.

To realize either of the gates we use again a cluster state of 5 qubits in a chain $\mathcal{C}(H)$. Let the labeling of the qubits be as in Fig. 2.2d and e, i.e. qubit 1 is the input- and qubit 5 the output qubit.

A cluster state $|\phi\rangle_{\mathcal{C}(H)}$ obeys the two eigenvalue equations

$$\begin{array}{rcl} |\phi\rangle_{\mathcal{C}(H)} &=& K^{(1)}K^{(3)}K^{(4)}|\phi\rangle_{\mathcal{C}(H)} \\ &=& \sigma_x^{(1)}\sigma_y^{(3)}\sigma_y^{(4)}\sigma_z^{(5)}|\phi\rangle_{\mathcal{C}(H)}, \\ |\phi\rangle_{\mathcal{C}(H)} &=& K^{(2)}K^{(3)}K^{(5)}|\phi\rangle_{\mathcal{C}(H)} \\ &=& \sigma_z^{(1)}\sigma_y^{(2)}\sigma_y^{(3)}\sigma_x^{(5)}|\phi\rangle_{\mathcal{C}(H)}. \end{array} \tag{2.107}$$

When the qubits 2, 3 and 4 of this state are measured in the σ_y-eigenbasis and thereby the measurement outcomes s_2, s_3, $s_4 \in \{0,1\}$ are obtained, the resulting state $|\psi\rangle_{\mathcal{C}(H)}$ obeys the eigenvalue equations

$$\begin{array}{rcl} \sigma_x^{(1)}\sigma_z^{(5)}|\phi\rangle_{\mathcal{C}(H)} &=& (-1)^{s_3+s_4}|\phi\rangle_{\mathcal{C}(H)}, \\ \sigma_z^{(1)}\sigma_x^{(5)}|\phi\rangle_{\mathcal{C}(H)} &=& (-1)^{s_2+s_3}|\phi\rangle_{\mathcal{C}(H)}. \end{array} \tag{2.108}$$

From equation (2.105) we see that the correlations (2.108) are precisely those we need to explain the realization of the Hadamard gate. Using Theorem 1 we find that by procedure 3 with measurement of the operators $\sigma_x^{(1)}$, $\sigma_y^{(2)}$, $\sigma_y^{(3)}$ and $\sigma_y^{(4)}$ a Hadamard gate with a byproduct operator as given in (2.30) is realized.

A cluster state $|\phi\rangle_{\mathcal{C}(U_z[\pi/2])}$ of a chain of 5 qubits obeys the eigenvalue equations

$$\begin{aligned} |\phi\rangle_{\mathcal{C}(U_z[\pi/2])} &= K^{(1)}K^{(3)}K^{(4)}K^{(5)}|\phi\rangle_{\mathcal{C}(U_z[\pi/2])}, \\ &= -\sigma_x^{(1)}\sigma_y^{(3)}\sigma_x^{(4)}\sigma_y^{(5)}|\phi\rangle_{\mathcal{C}(U_z[\pi/2])} \\ |\phi\rangle_{\mathcal{C}(U_z[\pi/2])} &= K^{(2)}K^{(4)}|\phi\rangle_{\mathcal{C}(U_z[\pi/2])} \\ &= \sigma_z^{(1)}\sigma_x^{(2)}\sigma_x^{(4)}\sigma_z^{(5)}|\phi\rangle_{\mathcal{C}(U_z[\pi/2])}. \end{aligned} \tag{2.109}$$

When the qubits 2, and 4 of this state are measured in the σ_x- and qubit 3 is measured in the σ_y-eigenbasis, with the measurement outcomes s_2, s_3, $s_4 \in \{0,1\}$ obtained, the resulting state $|\psi\rangle_{\mathcal{C}(U_z[\pi/2])}$ obeys the eigenvalue equations

$$\begin{aligned} \sigma_x^{(1)}\sigma_y^{(5)}|\psi\rangle_{\mathcal{C}(U_z[\pi/2])} &= (-1)^{s_3+s_4+1}|\psi\rangle_{\mathcal{C}(U_z[\pi/2])}, \\ \sigma_z^{(1)}\sigma_z^{(5)}|\psi\rangle_{\mathcal{C}(U_z[\pi/2])} &= (-1)^{s_2+s_4}|\psi\rangle_{\mathcal{C}(U_z[\pi/2])}. \end{aligned} \tag{2.110}$$

Using Theorem 1 we find that by procedure 3 with measurement of the operators $\sigma_x^{(1)}$, $\sigma_x^{(2)}$, $\sigma_y^{(3)}$ and $\sigma_x^{(4)}$ a $\pi/2$-phase gate is realized, where the byproduct operator is given by (2.30).

The CNOT gate

A measurement pattern which realizes a CNOT gate is illustrated in Fig. 2.2. Labeling the qubits as in Fig. 2.2, we use the same analysis as above to show that this measurement pattern does indeed realize a CNOT gate in the $\mathrm{QC}_{\mathcal{C}}$.

Of the cluster $\mathcal{C}(CNOT)$ on which the gate is realized, qubits 1 and 9 belong to $\mathcal{C}_I$, qubits 7 and 15 belong to $\mathcal{C}_O$ and the remaining qubits belong to $\mathcal{C}_M$. Let $|\phi\rangle$ be a cluster state on $\mathcal{C}(CNOT)$, which obeys the set of eigenvalue equations (2.1).

From these basic eigenvalue equations there follow the equations

$$\begin{aligned} |\phi\rangle &= K^{(1)}K^{(3)}K^{(4)}K^{(5)}K^{(7)}K^{(8)}K^{(13)}K^{(15)}\,|\phi\rangle \\ &= -\sigma_x^{(1)}\sigma_y^{(3)}\sigma_y^{(4)}\sigma_y^{(5)}\sigma_x^{(7)}\sigma_y^{(8)}\sigma_x^{(13)}\sigma_x^{(15)}\,|\phi\rangle, \qquad (2.111a) \\ |\phi\rangle &= K^{(2)}K^{(3)}K^{(5)}K^{(6)}\,|\phi\rangle \\ &= \sigma_z^{(1)}\sigma_y^{(2)}\sigma_y^{(3)}\sigma_y^{(5)}\sigma_y^{(6)}\sigma_z^{(7)}\,|\phi\rangle, \qquad (2.111b) \\ |\phi\rangle &= K^{(9)}K^{(11)}K^{(13)}K^{(15)}\,|\phi\rangle \\ &= \sigma_x^{(9)}\sigma_x^{(11)}\sigma_x^{(13)}\sigma_x^{(15)}\,|\phi\rangle, \qquad (2.111c) \\ |\phi\rangle &= K^{(5)}K^{(6)}K^{(8)}K^{(10)}K^{(12)}K^{(14)}\,|\phi\rangle \\ &= \sigma_y^{(5)}\sigma_y^{(6)}\sigma_z^{(7)}\sigma_y^{(8)}\sigma_z^{(9)}\sigma_x^{(10)}\sigma_y^{(12)}\sigma_x^{(14)}\sigma_z^{(15)}\,|\phi\rangle. \qquad (2.111d) \end{aligned}$$

Subsequently we will often use a graphic representation of eigenvalue equations like (2.111a) - (2.111d). Each of these equations is specified by the set of correlation centers q for which

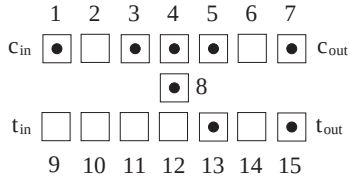


Figure 2.7: Pattern of correlation centers representing the eigenvalue equation (2.111a).

the basic correlation operators $K^{(q)}$ (2.2) enter the r.h.s. of the equation. While the information content is the same, it is often more illustrative to display the pattern of correlation centers than to write down the corresponding cluster state eigenvalue equation. As an example, the pattern of correlation centers which represents the eigenvalue equation (2.111a) is given in Fig. 2.7.

If the qubits 10, 11, 13 and 14 are measured in the σ_x- and the qubits 2, 3, 4, 5, 6, 8 and 12 are measured in the σ_y-eigenbasis, whereby the measurement results s_2 - s_6, s_8, s_{10} - s_{14} are obtained, then the cluster state eigenvalue equations (2.111a) - (2.111d) induce the following eigenvalue equations for the projected state $|\psi\rangle$

$$\sigma_x^{(1)}\,\sigma_x^{(7)}\,\sigma_x^{(15)}\,|\psi\rangle = (-1)^{1+s_3+s_4+s_5+s_8+s_{13}}|\psi\rangle, \tag{2.112a}$$

$$\sigma_z^{(1)}\,\sigma_z^{(7)}\,|\psi\rangle = (-1)^{s_2+s_3+s_5+s_6}|\psi\rangle \tag{2.112b}$$

$$\sigma_x^{(9)}\,\sigma_x^{(15)}\,|\psi\rangle = (-1)^{s_{11}+s_{13}}|\psi\rangle, \tag{2.112c}$$

$$\sigma_z^{(9)}\,\sigma_z^{(7)}\,\sigma_z^{(15)}\,|\psi\rangle = (-1)^{s_5+s_6+s_8+s_{10}+s_{12}+s_{14}}|\psi\rangle. \tag{2.112d}$$

Therein, qubits 1 and 7 represent the input and output for the control qubit and qubits 9 and 15 represent the input and output for the target qubit. Writing the CNOT unitary operation on control and target qubits $CNOT(c,t)$, we find

$$CNOT(c,t)\sigma_x^{(c)}CNOT(c,t) = \sigma_x^{(c)}\sigma_x^{(t)}, \tag{2.113a}$$

$$CNOT(c,t)\sigma_z^{(c)}CNOT(c,t) = \sigma_z^{(c)}, \tag{2.113b}$$

$$CNOT(c,t)\sigma_x^{(t)}CNOT(c,t) = \sigma_x^{(t)}, \tag{2.113c}$$

$$CNOT(c,t)\sigma_z^{(t)}CNOT(c,t) = \sigma_z^{(c)}\sigma_z^{(t)}. \tag{2.113d}$$

Comparing these equations to the eigenvalue equations (2.112a) to (2.112d), one sees that $\mathcal{M}$ does indeed realize a CNOT gate. Furthermore, after reading off the operator U_Σ using equations (2.61) and (2.63) and propagating the byproduct operators through to the output side of the CNOT gate, one finds the expressions for the byproduct operators, reported in equation (2.23).

2.2.8 Upper bounds on resource consumption

Here we discuss the spatial, temporal and operational resources required for the $\mathrm{QC}_{\mathcal{C}}$ and compare with resource requirements of a network quantum computer.

To run a specific quantum algorithm, the $\mathrm{QC}_{\mathcal{C}}$ requires a cluster of a certain size. Therefore the $\mathrm{QC}_{\mathcal{C}}$-*spatial resources* S are the number of cluster qubits in the required cluster state $|\phi\rangle_{\mathcal{C}}$, i.e. $S = |\mathcal{C}|$. The computation is driven by one-qubit measurement only. Thus, a single one-qubit measurement is one unit of operational resources, and the $\mathrm{QC}_{\mathcal{C}}$-*operational resources* O are defined as the total number of one-qubit measurements involved. The operational resources O are always smaller or equal to the spatial resources S,

$$O \leq S, \tag{2.114}$$

since each cluster qubit is measured at most once. As for the temporal resources, the $\mathrm{QC}_{\mathcal{C}}$-*logical depth* T is the minimum number of measurement rounds to which the measurements can be parallelized.

Let us briefly recall the definition of these resources in the network model. The temporal resources are specified by the network logical depth T_{qln}, which is the minimal number of steps to which quantum gates and readout measurements can be parallelized. The spatial resources S_{qln} count the number of logical qubits on which an algorithm runs. Finally, the operational resources O_{qln} are the number of elementary operations required to carry out an algorithm, i.e. the number of gates and measurements.

The construction kit for the simulation of quantum logic networks on the $\mathrm{QC}_{\mathcal{C}}$ shall contain a universal set of gates, in our case the CNOT gate between arbitrary qubits and the one qubit rotations. Already the next-neighbor CNOT with general rotations is universal since a general CNOT can be assembled of a next-neighbor CNOT and swap gates which can themselves be composed of next-neighbor CNOTs. However, in the following we would like to use for the general CNOT the less cumbersome construction described in Section 2.3.3. For this gate, the distance between logical qubits, i.e. between parallel qubit wires, is 4. The virtue of this gate is that it can always be realized on a vertical slice of width 6 on the cluster, no matter how far control and target qubit are separated. A slice of width 6 means that the distance between an input qubit of the gate and the corresponding input of the consecutive gate is 6 lattice spacings. This general CNOT gate determines the spatial dimensions of a unit cell in the measurement patterns. The size of this unit cell is 4×6. The other elementary gates, the next-neighbor CNOT and the rotations are smaller than a unit cell and therefore have to be stretched. This is easily accomplished. The next-neighbor CNOT as displayed in Fig. 2.2a has a size of 2×6 and is extended to size 4×6 by inserting two adjacent cluster qubits into the vertical bridge connecting the horizontal qubit lines. The general rotation as in Fig. 2.2b has width 4 and is stretched to width 6 by inserting two cluster qubits just before the output.

Concerning the temporal resources we first observe that we can realize the gates in the same temporal order as in the network model. To realize a general CNOT on the $\mathrm{QC}_{\mathcal{C}}$ takes one step of measurements, to realize a general rotation takes at most three. For the network model we do not assume that a general rotation has to be Euler-decomposed. Rather we assume that in the network model a rotation can be realized in a single step. Thus the temporal resources of the $\mathrm{QC}_{\mathcal{C}}$ and in the network model are related via

$$T \leq 3T_{\mathrm{qln}}. \tag{2.115}$$

As for the spatial resources, let us consider a rectangular cluster of height h and width w on which the qubit wires are oriented horizontally, with the network register state propagating from left to right. As the logical qubits have distance 4, the height of the cluster has to be $h = 4S_{\mathrm{qln}} - 3$ where S_{qln} is equal to the number n of logical qubits. Further, the number of gates in the circuit is at most $S_{\mathrm{qln}}T_{\mathrm{qln}}$ because, in the network model, in each step at most S_{qln} gates can be realized. On each vertical slice of width 6 on the cluster there fits at least one gate such that –taking into account an extra slice of width 1 for the readout cluster qubits– for the width holds $w \leq 6S_{\mathrm{qln}}T_{\mathrm{qln}} + 1$. With $S = h\,w$ one finds that

$$S \leq 24\, {S_{\mathrm{qln}}}^2 T_{\mathrm{qln}}. \tag{2.116}$$

In a similar way, a bound involving the network operational resources can be obtained. The spatial overhead S and the operational overhead O per elementary network operation is $\leq 24S_{\mathrm{qln}}$ if this operation is a unitary gate from the universal set described before, and is equal to one if this operation is a readout measurement. Thus, we also have

$$\begin{array}{rcl} S & \leq & 24\, O_{\mathrm{qln}} S_{\mathrm{qln}}, \\ O & \leq & 24\, O_{\mathrm{qln}} S_{\mathrm{qln}}. \end{array} \tag{2.117}$$

The purpose of this section was to demonstrate that the scaling of spatial and temporal resources is at worst polynomial as compared to the network model. In [10] it has been shown that the required classical processing increases the computation time only marginally (logarithmically in the number n of logical qubits) and thus there is no exponential overhead in either classical or quantum resources.

The upper bounds in (2.115), (2.116) and (2.117) should not be taken for estimates. For algorithms of practical interest the required resources usually scale much more favorably and there do not even have to be overheads at all. This is illustrated for the temporal complexity of Clifford circuits in Section 2.2.9 and in the examples of Section 2.3. A spatial overhead always exists. However, this is compensated by the fact that the operational effort to create a cluster state is independent of the cluster size.

2.2.9 Quantum circuits in the Clifford group can be realized in one step

The measurement bases to realize the Hadamard- and the $\pi/2$-phase gate need not be adapted since only operators σ_x and σ_y are measured. The same holds for the realization of the CNOT gate, see Fig. 2.2. Thus, all the Hadamard-, $\pi/2$-phase- and CNOT gates of a quantum circuit can be realized simultaneously in the first measurement round, regardless of their location in the network. In particular, quantum circuits which consist only of such gates, i.e. circuits in the Clifford group, can be realized in a single time step. As an example, many circuits for coding and decoding are in the Clifford group.

The fact that quantum circuits in the Clifford group can be realized in a single time step has previously not been known for networks. The best upper bound on the logical

depth that was known previously scales logarithmically with the number of logical qubits [56].

Note that, as stated by the Gottesman-Knill-Theorem [21], there is no need for fast Clifford circuits if the quantum output is measured in a Pauli basis because these circuits can be simulated efficiently classically. However, the purpose of this section is to point out that the whole Clifford part of *any* quantum circuit can be performed in a single time step.

Here we find a first aspect of $\mathrm{QC}_{\mathcal{C}}$-computation which is not adequately described within the network model, and with this observation we conclude the discussion of the $\mathrm{QC}_{\mathcal{C}}$ as a simulator of quantum logic networks.

2.3 Examples of practical interest

2.3.1 Multi-qubit swap gate

A multi-qubit swap gate is an n-qubit generalization of the two-qubit swap gate. It reverses the order of the n qubits, interchanging qubit i with $n+1-i$, $i = 1, 2, .., N$. This can be realized in a simple way on the $\mathrm{QC}_{\mathcal{C}}$, as shown in Fig. 2.8a. The measurement pattern $\mathcal{M}$ on $\mathcal{C}_M$ consists of a square of σ_x measurements, with sides of $2n-1$ cluster qubits. The input qubits are, simultaneously with the qubits in $\mathcal{C}_M$, also measured in the σ_x-eigenbasis.

It can be verified using the methods introduced above that realizing $\mathcal{M}$ leads to correlations between the ith input qubit and the $n+1-i$-th output qubit. Here, we discuss the four-qubit swap as a particular example.

After the σ_x-measurements of the qubits in $\mathcal{C}_M$ we obtain for the projected state $|\psi\rangle_{\mathcal{C}(\mathrm{swap})}$ the eigenvalue equations

$$\begin{aligned}
\sigma_x^{(I,1)}\sigma_x^{(O,4)}\,|\psi\rangle_{\mathcal{C}(\mathrm{swap})} &= (-1)^{\lambda_{x,1}}|\psi\rangle_{\mathcal{C}(\mathrm{swap})},\\
\sigma_x^{(I,2)}\sigma_x^{(O,3)}\,|\psi\rangle_{\mathcal{C}(\mathrm{swap})} &= (-1)^{\lambda_{x,2}}|\psi\rangle_{\mathcal{C}(\mathrm{swap})},\\
\sigma_x^{(I,3)}\sigma_x^{(O,2)}\,|\psi\rangle_{\mathcal{C}(\mathrm{swap})} &= (-1)^{\lambda_{x,3}}|\psi\rangle_{\mathcal{C}(\mathrm{swap})},\\
\sigma_x^{(I,4)}\sigma_x^{(O,1)}\,|\psi\rangle_{\mathcal{C}(\mathrm{swap})} &= (-1)^{\lambda_{x,4}}|\psi\rangle_{\mathcal{C}(\mathrm{swap})},\\
\sigma_z^{(I,1)}\sigma_z^{(O,4)}\,|\psi\rangle_{\mathcal{C}(\mathrm{swap})} &= (-1)^{\lambda_{z,1}}|\psi\rangle_{\mathcal{C}(\mathrm{swap})},\\
\sigma_z^{(I,2)}\sigma_z^{(O,3)}\,|\psi\rangle_{\mathcal{C}(\mathrm{swap})} &= (-1)^{\lambda_{z,2}}|\psi\rangle_{\mathcal{C}(\mathrm{swap})},\\
\sigma_z^{(I,3)}\sigma_z^{(O,2)}\,|\psi\rangle_{\mathcal{C}(\mathrm{swap})} &= (-1)^{\lambda_{z,3}}|\psi\rangle_{\mathcal{C}(\mathrm{swap})},\\
\sigma_z^{(I,4)}\sigma_z^{(O,1)}\,|\psi\rangle_{\mathcal{C}(\mathrm{swap})} &= (-1)^{\lambda_{z,4}}|\psi\rangle_{\mathcal{C}(\mathrm{swap})}.
\end{aligned} \tag{2.118}$$

Therein, the parameters $\lambda_{k,x}, \lambda_{k,z} \in \{0,1\}$ depend linearly on the measurement outcomes $\{s_{(i,j)}\}$, where i is the value of the x- and j the value of the y-coordinate of the respective qubit site. For example, $\lambda_{x,1} = s_{(1,2)} + s_{(2,3)} + s_{(3,4)} + s_{(4,5)} + s_{(5,6)} + s_{(6,7)} \bmod 2$.

The eigenvalue equations (2.118) can be derived from corresponding eigenvalue equations for the cluster state $|\phi\rangle_{\mathcal{C}(\mathrm{swap})}$ on the cluster $\mathcal{C}(\mathrm{swap})$. The required initial correlations are products of the basic correlation operators (2.2). The way to obtain the equations (2.118) is rather straightforward and therefore we omit the detailed derivations. In Fig. 2.8b

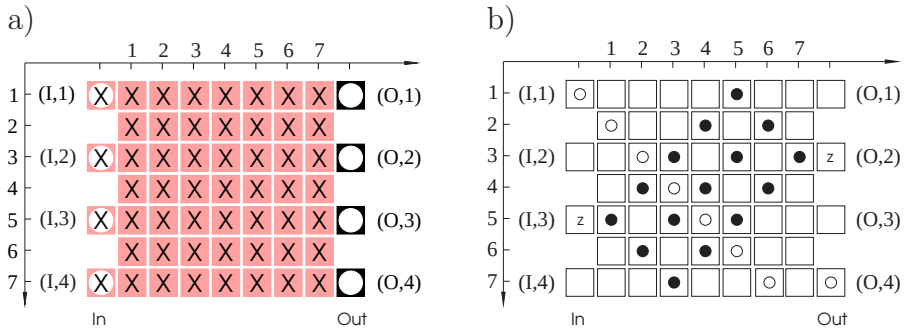


Figure 2.8: The multi-qubit swap gate. a) measurement pattern to realize the swap gate. b) Correlation centers for two correlations of the projected state $|\psi\rangle_{\mathcal{C}(\text{swap})}$ as inherited from correlations of $|\phi\rangle_{\mathcal{C}(\text{swap})}$. The correlation $\sigma_x^{(I,1)}\sigma_x^{(O,4)}$ of $|\psi\rangle_{\mathcal{C}(\text{swap})}$ stems from the product correlation for $|\phi\rangle_{\mathcal{C}(\text{swap})}$ with the centers a of basic correlation operators $K^{(a)}$ denoted by "$\circ$". The centers of the initial correlation, which after the measurements induces the correlation $\sigma_z^{(I,3)}\sigma_z^{(O,2)}$ of $|\psi\rangle_{\mathcal{C}(\text{swap})}$, are denoted by "$\bullet$".

two examples for the composition of product correlation operators from basic correlation operators $K^{(a)}$ are illustrated. The first line of (2.118),

$$\sigma_x^{(I,1)}\sigma_x^{(O,4)}\,|\psi\rangle_{\mathcal{C}(\text{swap})} = (-1)^{\lambda_{x,1}}|\psi\rangle_{\mathcal{C}(\text{swap})},$$

for example, is derived from the eigenvalue equation

$$|\phi\rangle_{\mathcal{C}(\text{swap})} = K^{(\mathcal{C}_{x,1})}|\phi\rangle_{\mathcal{C}(\text{swap})}, \tag{2.119}$$

with

$$K^{(\mathcal{C}_{x,1})} = \prod_{a\in\mathcal{C}_{x,1}} K^{(a)}, \tag{2.120}$$

and $\mathcal{C}_{x,1} = \{(I,1),(1,2),(2,3),(3,4),(4,5),(5,6),\ (6,7),\ (O,4)\}$. Evaluating the r.h.s. of (2.120) we find that all operators σ_z cancel and that

$$K^{(\mathcal{C}_{x,1})} = \prod_{a\in\mathcal{C}_{x,1}} \sigma_x^{(a)}. \tag{2.121}$$

It is now easy to see that after the σ_x measurements of the qubits in $\mathcal{C}_M$ there remains a strict $\sigma_x^{(I,1)}\sigma_x^{(O,4)}$-correlation for the state $|\psi\rangle_{\mathcal{C}(\text{swap})}$. A similar construction can be given to obtain the $\sigma_z^{(I,1)}\sigma_z^{(O,4)}$-correlation.

With the eigenvalue equations (2.118) the assumptions of Theorem 1 are fulfilled and thus via the described measurement pattern a unitary operation $U = SWAP$ is realized modulo a byproduct operator as specified in (2.63). To exchange the order of the swap gate U_{swap} and the byproduct operator U_Σ the byproduct operator is conjugated under U_{swap}, as usual for gates in the Clifford group.

2.3.2 Simulating multi-qubit Hamiltonians

Here we display a gate which simulates the unitary evolution with $U = \exp(-iH_4 t)$ of the quantum input for the multi-particle Hamiltonian

$$H_4 = g\,\sigma_z^{(1)}\sigma_z^{(2)}\sigma_z^{(3)}\sigma_z^{(4)} \tag{2.122}$$

and *arbitrary* times t. In addition, the gate performs a swap which can be undone by a subsequent swap gate as described in Section 2.3.1.

The procedure to realize the measurement pattern $\mathcal{M}$ for Hamiltonian simulation, as shown in Fig. 2.9, requires two rounds of measurements. In the first round all the σ_x-measurements are performed. In the second measurement round, of the qubit $(3,4)$ the operator

$$\vec{r}_{(3,4)} \cdot \vec{\sigma} = U_z[(-1)^{\lambda_M} 2\varphi]\,\sigma_x\,U_z^{\dagger}[(-1)^{\lambda_M} 2\varphi] \tag{2.123}$$

is measured, where $U_z[\alpha] = \exp(-i\alpha\sigma_z/2)$. Therein, the angle φ is given by

$$\varphi = gt, \tag{2.124}$$

and $\lambda_M \in \{0,1\}$, which depends linearly on outcomes of measurements in the first round, will be specified below.

To understand the functioning of the Hamiltonian simulator let us first discuss the state $|\psi'\rangle$ on the cluster $\mathcal{C}(\mathrm{sim})$ after the first round of measurements. By arguments analogous to those used in Section 2.3.1, the state $|\psi'\rangle$ obeys the following eigenvalue equations:

$$\begin{aligned}
\sigma_x^{(3,4)}\sigma_x^{(I,1)}\sigma_x^{(O,4)}\,|\psi'\rangle &= (-1)^{\lambda_{x,1}}|\psi'\rangle,\\
\sigma_x^{(3,4)}\sigma_x^{(I,2)}\sigma_x^{(O,3)}\,|\psi'\rangle &= (-1)^{\lambda_{x,2}}|\psi'\rangle,\\
\sigma_x^{(3,4)}\sigma_x^{(I,3)}\sigma_x^{(O,2)}\,|\psi'\rangle &= (-1)^{\lambda_{x,3}}|\psi'\rangle,\\
\sigma_x^{(3,4)}\sigma_x^{(I,4)}\sigma_x^{(O,1)}\,|\psi'\rangle &= (-1)^{\lambda_{x,4}}|\psi'\rangle,\\
\sigma_z^{(I,1)}\sigma_z^{(O,4)}\,|\psi'\rangle &= (-1)^{\lambda_{z,1}}|\psi'\rangle,\\
\sigma_z^{(I,2)}\sigma_z^{(O,3)}\,|\psi'\rangle &= (-1)^{\lambda_{z,2}}|\psi'\rangle,\\
\sigma_z^{(I,3)}\sigma_z^{(O,2)}\,|\psi'\rangle &= (-1)^{\lambda_{z,3}}|\psi'\rangle,\\
\sigma_z^{(I,4)}\sigma_z^{(O,1)}\,|\psi'\rangle &= (-1)^{\lambda_{z,4}}|\psi'\rangle.
\end{aligned} \tag{2.125}$$

Further, the state $|\psi'\rangle$ obeys the eigenvalue equation

$$\sigma_z^{(3,4)}\,\sigma_z^{(O,1)}\sigma_z^{(O,2)}\sigma_z^{(O,3)}\sigma_z^{(O,4)}\,|\psi'\rangle = (-1)^{\lambda}|\psi'\rangle, \tag{2.126}$$

with $\lambda \in \{0,1\}$ linear in the measurement outcomes of the first round. Equation (2.126) can be easily verified with the pattern of correlation centers displayed in Fig. 2.9b. From (2.126) it follows that

$$\exp\left(i\theta\,\sigma_z^{(3,4)}\right) U_4[(-1)^{\lambda}\theta]\,|\psi'\rangle = |\psi'\rangle \tag{2.127}$$

for arbitrary angles θ, with

$$U_4[\alpha] = \exp\left(-i\alpha\,\sigma_z^{(O,1)}\sigma_z^{(O,2)}\sigma_z^{(O,3)}\sigma_z^{(O,4)}\right). \tag{2.128}$$

Equation (2.127) is now inserted in both the l.h.s. and r.h.s. of the equations (2.125). For example, with the first equation from (2.125) one obtains

$$(-1)^{\lambda_{x,1}}\,|\psi'\rangle = \left(U_z[2\theta]\sigma_x U_z^\dagger[2\theta]\right)^{(3,4)}\sigma_x^{(I,1)}\left(U_4[-(-1)^\lambda\theta]\sigma_x^{[4]}U_4^\dagger[-(-1)^\lambda\theta]\right)^{(O)}|\psi'\rangle. \quad (2.129)$$

In the second measurement round the qubit $(3,4)$ is the only one left to be measured. As can be seen from (2.129), if of the operator $U_z[2\theta]\sigma_x U_z^\dagger[2\theta]$ of qubit (3,4) is measured then the state $|\psi\rangle$, into which the cluster qubits are projected after the second measurement round, obeys the eigenvalue equation

$$(-1)^{\lambda_{x,1}+s_{(3,4)}}\,|\psi\rangle = \sigma_x^{(I,1)}\left(U_4[-(-1)^\lambda\theta]\sigma_x^{[4]}U_4^\dagger[-(-1)^\lambda\theta]\right)^{(O)}|\psi\rangle. \quad (2.130)$$

If we carry out this procedure for all equations in (2.125) we find that the state $|\psi\rangle$ that emerges after the second measurement round obeys the eigenvalue equations

$$\begin{aligned}\sigma_x^{(I,i)}\left(U_4 U_{\rm swap}\sigma_x^{[i]}U_{\rm swap}{}^\dagger U_4^\dagger\right)^{(O)}|\psi\rangle &= (-1)^{\lambda_{x,i}+s_{(3,4)}}\,|\psi\rangle,\\ \sigma_z^{(I,i)}\left(U_4 U_{\rm swap}\sigma_z^{[i]}U_{\rm swap}{}^\dagger U_4^\dagger\right)^{(O)}|\psi\rangle &= (-1)^{\lambda_{z,i}}\,|\psi\rangle,\end{aligned} \quad (2.131)$$

for $i = 1\,..\,4$ and with U_4 written in short for $U_4[-(-1)^\lambda\theta]$.

With the set of equations (2.131) the assumptions (2.61) of Theorem 1 are fulfilled. With Theorem 1 it follows that the measurement pattern displayed in Fig. 2.9 realizes a unitary transformation

$$U_{\rm sim} = U_4[-(-1)^\lambda\theta]U_{\rm swap}U_\Sigma, \quad (2.132)$$

where the byproduct operator is given by

$$U_\Sigma = \bigotimes_{i=1}^{4}\left(\sigma_z^{[i]}\right)^{s_{(I,i)}+\lambda_{x,i}+s_{(3,4)}}\left(\sigma_x^{[i]}\right)^{\lambda_{z,i}}. \quad (2.133)$$

Finally, the order of the operators has to be exchanged. Note that $U_{\rm swap}$ and U_4 commute. From (2.132) one finds

$$U_{\rm sim} = U'_\Sigma U_{\rm swap}U_4[-(-1)^{\lambda+\sum_{i=1}^4\lambda_{z,i}}\,\theta], \quad (2.134)$$

with

$$U'_\Sigma = U_{\rm swap}\,U_\Sigma\,U_{\rm swap}{}^\dagger. \quad (2.135)$$

Thus, in order to realize $U_4[\varphi]$ with φ specified in (2.124) we must choose

$$\theta = (-1)^{1+\lambda+\sum_{i=1}^4\lambda_{z,i}}\varphi. \quad (2.136)$$

That is, in the second measurement round we measure on the qubit $(3,4)$ the operator given in (2.123), where

$$\lambda_M = \left(1+\lambda+\sum_{i=1}^{4}\lambda_{z,i}\right)\ \mathrm{mod}\,2. \quad (2.137)$$

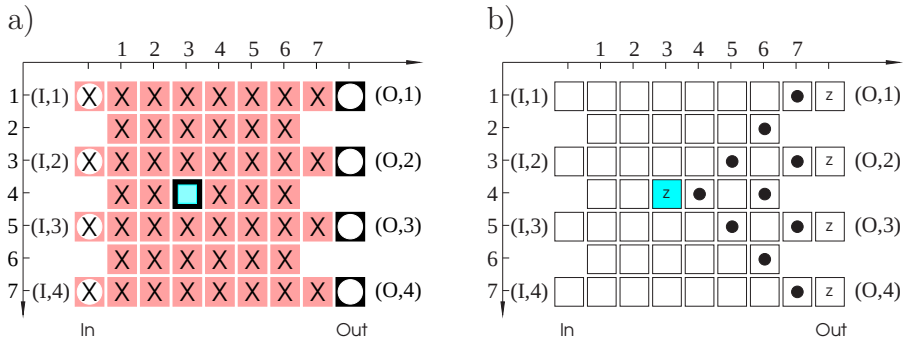


Figure 2.9: Simulation of the Hamiltonian H_4 as specified in eq. (2.122). a) measurement pattern. b) Correlation centers for additional correlation. Shaded squares (in b)) represent cluster qubits measured in adaptive bases.

The $\{\lambda_{x,i}\}$, $\{\lambda_{z,i}\}$ and λ depend linearly on the measurement outcomes $\{s_{(i.j)}\}$ obtained in the first measurement round.

The sub-circuit we have described in this section simulates the unitary evolution according to a particular four-particle Hamiltonian in a two-step process of measurements. The time for which the simulated Hamiltonian acts is encoded in the basis of the measurement in the second round. The generalization of the simulation of the 4-particle Hamiltonian H_4, shown in Fig. 2.9, to an arbitrary number n of qubits, i.e. the simulation of the Hamiltonian $H_n = \bigotimes_{i=1}^{n} \sigma_z^{[i]}$, is straightforward.

2.3.3 CNOT between distant qubits

The CNOT gate described in Section 2.2.7 operates on two logical qubits whose input qubits are adjacent to each other on the cluster. However, for universal quantum computation, one must be able to realize a CNOT gate between any two logical qubits. While this could be achieved using a combination of the CNOT gate, introduced above, and the swap gate, the width of the measurement pattern needed to realize this would grow linearly with the separation of the two logical qubits. There is, however, an alternative measurement pattern, which, at the cost of doubling the spacing between the input qubits on the cluster, has a fixed width. The measurement pattern is illustrated in Fig. 2.10 for qubits separated by an odd and even number of logical qubits, respectively.

This layout can be understood within the quantum logic network model. The "wires" for the logical qubits in between the target- and the control qubit are crossed using the measurement sub-pattern illustrated in Fig. 2.11a. However, as well as swapping the qubits, this pattern also realizes the a controlled π-phase gate, also known as a controlled σ_z gate. The quantum logic circuit realized by the whole measurement pattern, illustrated on the left-hand side of Fig. 2.11b uses these sub-patterns to swap the positions of adjacent qubits. This brings non-neighboring qubits together so that a CNOT operation may be performed on them. The networks on the left and on the right of Fig. 2.11b act identically, and thus

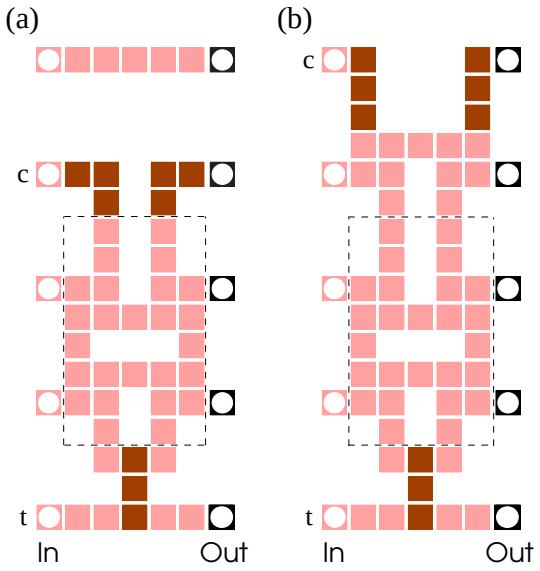


Figure 2.10: Measurement pattern for a CNOT gate between two logical qubits whose input and output qubits are not neighbors. Squares in light gray denote cluster qubits measured in the eigenbasis of σ_x, in dark gray of σ_y. Pattern (a) is for the case where the two qubits are separated by an odd number of logical qubits. Pattern (b) is for an even numbered separation. The patterns can be adapted to any separation by repeating the section enclosed by the dashed line. The width of the pattern remains the same for all separations.

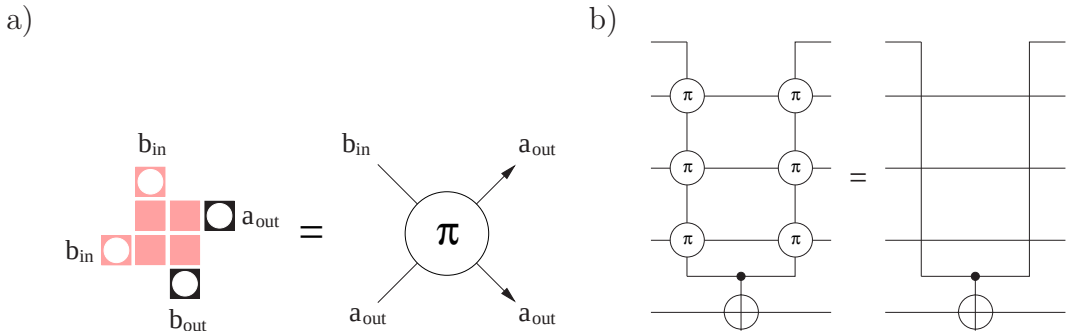


Figure 2.11: The measurement pattern in a) is one of the key components of the measurement pattern in Fig. 2.10. It performs a conditional π-phase gate and a swap gate. b) The measurement pattern in Fig. 2.10 realizes the quantum logic circuit on the left hand side. This network is equivalent to the one on the right hand side, where the only gate realized is the CNOT between the two desired non-adjacent qubits.

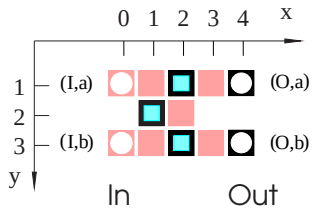


Figure 2.12: Controlled phase gate with additional swap.

the measurement pattern displayed in Fig. 2.10 realizes a distant CNOT gate.

2.3.4 Controlled phase gate

Here, we give an example of another two-qubit gate which can be realized without decomposing it into CNOTs and rotations, the controlled phase gate $U_{\rm CPG}(\theta)$. This gate realizes the unitary operation

$$U_{\rm CPG}[\theta] = \mathbb{1}^{(ab)} + (e^{i\theta} - 1)\,|11\rangle_{ab}\langle 11|, \tag{2.138}$$

applied to the two qubits a and b.

We can write this in terms of the following one- and two-qubit rotations,

$$U_{\rm CPG}[\theta] = e^{i\frac{\theta}{4}} U_{zz}^{(ab)}[-\theta/2] U_z^{(a)}[\theta/2] U_z^{(b)}[\theta/2], \tag{2.139}$$

where the two-qubit rotation is

$$U_{zz}^{(ab)}[\theta] = \exp\left(-\mathrm{i}\theta/2\sigma_z^{(a)}\sigma_z^{(b)}\right). \tag{2.140}$$

This representation is particularly convenient for finding the measurement pattern that realizes the gate, since rotations $U_z[\theta/2]$ and $U_{zz}[-\theta/2]$ are realized on the $\mathrm{QC}_{\mathcal{C}}$ in a simple natural way. The measurement pattern is illustrated in Fig. 2.12, in which the labelling of the qubits is also defined.

We follow the same method as above, beginning with the eigenvalue equations of the cluster state $|\phi\rangle_{\mathcal{C}}$ on the qubits shown. The σ_x-measurements can be considered first, using the methods already illustrated in this thesis. The resultant state of the remaining qubits $|\psi'\rangle$, after this sub-set of the measurements has been carried out, is defined by the following set of eigenvalue equations.

$$\sigma_x^{(I,a)}\sigma_x^{(1,2)}\sigma_x^{(2,3)}\sigma_x^{(O,b)}|\psi'\rangle = |\psi'\rangle, \tag{2.141a}$$
$$\sigma_x^{(I,b)}\sigma_x^{(1,2)}\sigma_x^{(2,1)}\sigma_x^{(O,a)}|\psi'\rangle = |\psi'\rangle, \tag{2.141b}$$
$$\sigma_z^{(I,a)}\sigma_z^{(O,b)}|\psi'\rangle = (-1)^{s_{(1,1)}+s_{(2,2)}+s_{(3,3)}}|\psi'\rangle, \tag{2.141c}$$
$$\sigma_z^{(I,b)}\sigma_z^{(O,a)}|\psi'\rangle = (-1)^{s_{(1,3)}+s_{(2,2)}+s_{(3,1)}}|\psi'\rangle, \tag{2.141d}$$

and

$$\sigma_z^{(2,1)}\sigma_z^{(O,a)}|\psi'\rangle = (-1)^{s_{(3,1)}}|\psi'\rangle, \tag{2.142a}$$

$$\sigma_z^{(2,3)}\sigma_z^{(O,b)}|\psi'\rangle = (-1)^{s_{(3,3)}}|\psi'\rangle, \tag{2.142b}$$

$$\sigma_z^{(1,2)}\sigma_z^{(O,a)}\sigma_z^{(O,b)}|\psi'\rangle = (-1)^{s_{(3,1)}+s_{(2,2)}+s_{(3,3)}}|\psi'\rangle. \tag{2.142c}$$

As in section 2.2.7, eigenvalue equations are now generated which commute with the remaining measurements in $\mathcal{M}$, namely the measurements of $\sigma_{xy}^{(i)}(\alpha_i)$ on qubits $i \in \{(2,1), (1,2), (2,3)\}$. First, we manipulate the equations (2.142) such that, for example, the eigenvalue equation (2.142c) attains the form

$$U_z^{(1,2)}[\xi]\, U_{zz}^{((O,a),(O,b))}\left[-(-1)^{s_{(3,1)}+s_{(2,2)}+s_{(3,3)}}\xi\right]\, |\psi'\rangle = |\psi'\rangle. \tag{2.143}$$

Similar equations containing one-qubit rotations on qubits $(2,1)$ and (O,a), and $(2,3)$ and (O,b) are derived from the other equations of (2.142) in the same way. These equations are inserted into both sides of the eigenvalue equations (2.141) so that, using the method introduced above, we obtain a set of four eigenvalue equations for $|\psi'\rangle$ which induce a set of four eigenvalue equations for the state $|\psi\rangle$ that one obtains after the remaining measurements have been carried out.

Specifically, in the second measurement round the qubits $(1,2)$, $(2,1)$ and $(2,3)$ are measured. Of these qubits one measures the observables

$$\vec{r}_a \cdot \vec{\sigma}^{(a)} = \left(U_z[\alpha_a]\sigma_x U_z[\alpha_a]^\dagger\right)^{(a)}, \tag{2.144}$$

for $a \in \{(1,2),(2,1),(2,3)\}$ and the $\{\alpha_a\}$ specified below.

The induced eigenvalue equations for the state $|\psi\rangle$ are of the form of equation (2.61), and the unitary operation realized by the gate can be read off from them using Theorem 1. The full unitary operation realized by the measurement pattern is

$$\begin{aligned} U'U'_\Sigma \;=\; & U_{zz}^{(a,b)}\left[-(-1)^{s_{(3,1)}+s_{(2,2)}+s_{(3,3)}}\alpha_{(1,2)}\right]U_z^{(a)}\left[-(-1)^{s_{(3,1)}}\alpha_{(2,1)}\right]\, U_z^{(b)}\left[-(-1)^{s_{(3,3)}}\alpha_{(2,3)}\right] \\ & \times U_{\text{swap}}^{(a,b)}\left(\sigma_x^{(a)}\right)^{s_{(1,1)}+s_{(2,2)}+s_{(3,3)}}\left(\sigma_x^{(b)}\right)^{s_{(1,3)}+s_{(2,2)}+s_{(3,1)}}\left(\sigma_z^{(a)}\right)^{s_{(I,a)}+s_{(1,2)}+s_{(2,3)}} \\ & \times \left(\sigma_z^{(b)}\right)^{s_{(I,b)}+s_{(2,1)}+s_{(1,2)}} \end{aligned} \tag{2.145}$$

such that after the order of the gate and the byproduct operator is reversed, $U'U'_\Sigma = U_\Sigma U$, one obtains

$$\begin{aligned} U_\Sigma U \;=\; & \left(\sigma_x^{(a)}\right)^{s_{(1,3)}+s_{(2,2)}+s_{(3,1)}}\left(\sigma_x^{(b)}\right)^{s_{(1,1)}+s_{(2,2)}+s_{(3,3)}}\left(\sigma_z^{(a)}\right)^{s_{(2,1)}+s_{(1,2)}+s_{(I,b)}} \\ & \times \left(\sigma_z^{(b)}\right)^{s_{(I,a)}+s_{(1,2)}+s_{(2,3)}} U_{zz}^{(a,b)}\left[-(-1)^{s_{(1,1)}+s_{(2,2)}+s_{(1,3)}}\alpha_{(1,2)}\right] \\ & \times U_z^{(a)}\left[-(-1)^{s_{(2,2)}+s_{(1,3)}}\alpha_{(2,1)}\right]U_z^{(b)}\left[-(-1)^{s_{(1,1)}+s_{(2,2)}}\alpha_{(2,3)}\right]U_{\text{swap}}^{(a,b)}. \end{aligned} \tag{2.146}$$

Using (2.146) one finds the following result: To realize the controlled phase gate (2.138) together with a swap gate, the observables (2.144) measured in the second round have to be chosen with the angles $\alpha_{(2,1)} = (-1)^{1+s_{(2,2)}+s_{(1,3)}}\,\theta/2$, $\alpha_{(1,2)} = (-1)^{s_{(1,1)}+s_{(2,2)}+s_{(1,3)}}\theta/2$ and $\alpha_{(2,3)} = (-1)^{s_{(1,1)}+s_{(2,2)}+1}\,\theta/2$. This realizes the gate $U_\Sigma U_{CPG}[\theta]$, where the byproduct operator U_Σ generated by the measurements may be read off from equation (2.146).

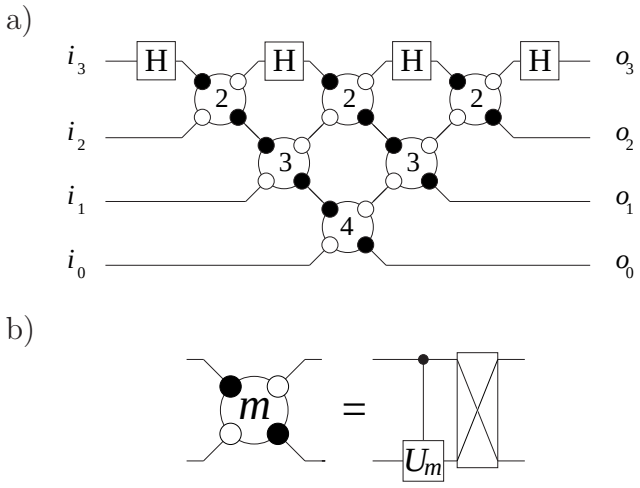


Figure 2.13: Quantum Fourier transformation. a) Network for quantum Fourier transformation on four qubits, taken from [59]. b) Component of the network shown in a) which performs a conditional phase- and a swap gate. Specifically, the gate shown is $U_{CPG}[2\pi/2^m]$, i.e. $U_m = |0\rangle\langle 0| + e^{i2\pi/2^m}|1\rangle\langle 1|$.

2.3.5 Quantum Fourier transformation

To realize the quantum Fourier transform we simulate the quantum logic network given in Fig. 2.13a. The arrangement of the gates in this network is taken from [59]. Note that in [59] it was demonstrated that the setup to perform a quantum Fourier transformation simplifies considerably in a situation where the output state is measured right after the transformation. Here, however, the quantum Fourier transformation may constitute part of a larger quantum circuit and we do not measure its output state.

As can be seen from Fig. 2.13, the quantum Fourier transform consists of Hadamard gates and combined gates which perform a conditional phase shift and a swap. These gates have been discussed in Sections 2.2.2 and 2.3.4. All that remains to do is put the measurement patterns simulating these gates together, using the network-like composition principle described in Section 2.2.4.

In this way we obtain a measurement pattern in which there are adjacent cluster qubits in "wires" that are measured in the σ_x-eigenbasis. As described in Section 2.2.7, such pairs of cluster qubits may be removed from the measurement pattern. Note, that by removing adjacent pairs of σ_x-measured cluster qubits we have moved the σ_y-measurements of the Hadamard transformations "into" the subsequent conditional phase gates, i.e. we removed a cluster qubit which was not from a wire. It can be easily verified that this is an allowed extension of the method described in Section 2.2.7. Finally, one obtains the $\mathrm{QC}_{\mathcal{C}}$-circuit displayed in Fig. 2.14.

In this circuit, as in all the others, the adaptive measurements are of observables

$$U_z[\pm\eta]\sigma_x U_z[\pm\eta]^\dagger, \tag{2.147}$$

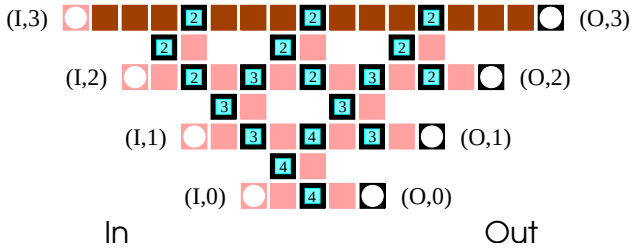


Figure 2.14: $\mathrm{QC}_{\mathcal{C}}$-realization of a quantum Fourier transformation on four qubits. The cluster qubits displayed as framed squares are measured in adapted bases. For the labels see text.

with $\eta = \pi/4$ for cluster qubits marked with "2" in Fig. 2.14, $\eta = \pi/8$ for qubits marked with "3" and $\eta = \pi/16$ for the qubits marked with "4". The sign factors of the angles in (2.147) depend on the results of previous measurements.

The $\mathrm{QC}_{\mathcal{C}}$-circuit, shown in Fig. 2.14 for the case of four qubits, is straightforwardly generalized to an arbitrary number n of logical qubits. The temporal spatial and operational resources T, S and O are, to leading order

$$T = n, \quad S, O = 2n^2. \tag{2.148}$$

The corresponding network resources are $T_{\mathrm{qln}} = 2n$, $S_{\mathrm{qln}} = n$ and $O_{\mathrm{qln}} = n^2/2$. Thus, the scaling of the $\mathrm{QC}_{\mathcal{C}}$ spatial resources is worse than in the network model, but the temporal and operational resources scale in the same way as the corresponding resources for the network. The $\mathrm{QC}_{\mathcal{C}}$-simulation of the network displayed in Fig. 2.13 requires half as many time steps and four times as many operations, albeit only one-qubit operations.

2.3.6 Multi-qubit controlled gates

In this section we describe the realization of the Toffoli phase gate and the three-qubit controlled gate $CARRY$ which we will both need for the construction of the $\mathrm{QC}_{\mathcal{C}}$-adder circuit described in Section 2.3.7.

The Toffoli phase gate is a three-qubit generalization of the two-qubit controlled phase gate. If all three qubits are in the state $|1\rangle$, the state gains a phase of $\exp(\mathrm{i}\phi)$, while all other logical basis states remain unchanged by the gate,

$$U_{\mathrm{Toffoli}}^{(c_1,c_2,t)}[\phi] = \mathbb{1}^{(c_1,c_2,t)} + (e^{i\phi} - 1)\,|111\rangle_{c_1,c_2,t}\langle 111|. \tag{2.149}$$

Like the controlled phase gate it can be represented as a product of multi-qubit rotations,

$$U_{\mathrm{Toffoli}}^{(c_1,c_2,t)}[\phi] = U_{zzz}^{(c_1,c_2,t)}\left[\frac{\phi}{4}\right] U_{zz}^{(c_1,c_2)}\left[-\frac{\phi}{4}\right] U_{zz}^{(c_1,t)}\left[-\frac{\phi}{4}\right] U_{zz}^{(c_2,t)}\left[-\frac{\phi}{4}\right] U_{z}^{(c_1)}\left[\frac{\phi}{4}\right] U_{z}^{(c_2)}\left[\frac{\phi}{4}\right] U_{z}^{(t)}\left[\frac{\phi}{4}\right]. \tag{2.150}$$

where we have dropped the global phase, and $U_{zzz}^{(c_1,c_2,t)}[\alpha] = \exp\left(-\mathrm{i}\alpha/2\sigma_z^{(c_1)}\sigma_z^{(c_2)}\sigma_z^{(t)}\right)$ is a three qubit generalized rotation. The two-qubit rotations U_{zz} are as defined in (2.140).

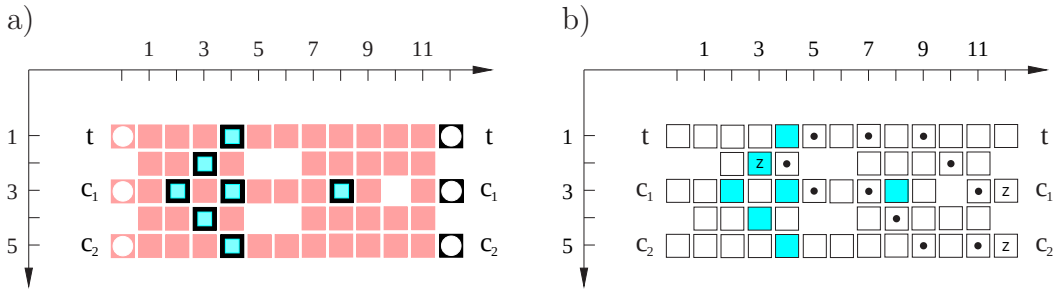


Figure 2.15: $QC_{\mathcal{C}}$-realization of the Toffoli phase gate. a) Measurement layout to realize a Toffoli phase gate with phase ϕ. The qubits marked by black boxes are simultaneously measured in adapted bases depending on previous measurement outcomes. b) Cluster state quantum correlations for the realization of $U_{zz}^{(c_1,c_2)}[\phi/4]$, used in the Toffoli phase gate.

The way to convert the sequence (2.150) of generalized rotations into a measurement pattern is as in the examples before. The measurement layout for the Toffoli phase gate is illustrated in Fig. 2.15. Each of the generalized rotations that make up the gate is directly associated with of one of the measurements made in the eigenbasis of $U_z[\pm\phi/4]\sigma_x U_z[\pm\phi/4]^\dagger$. An initial cluster-state correlations which is used for the realization of a generalized rotation is shown in Fig. 2.15: the rotation $U_{zz}^{(c_1,c_2)}[\phi/4]$ is realized via the measurement of the cluster qubit at the lattice site $(3,1)$ in the appropriate basis.

The sign factors of the angles that specify the measurement bases depend on the outcome of σ_x-measurements only. Thus, after all σ_x- measurements have been performed, the measurement bases for the remaining qubits can be deduced and the Toffoli phase gate is realized in a single further time-step. The measurement pattern realizes the generalized rotations directly and is not derived from a quantum logic network.

Now we describe the realization of a four-qubit gate $CARRY$, which has one target and three control qubits. It performs a phase-flip σ_z on the target if at least two of the control qubits are in state $|1\rangle$ and otherwise does nothing, i.e.

$$U_{CARRY} = \exp\left(-i\pi \sum_{i=000_d \,|\, w(i)\geq 2}^{111_d} |i\rangle_{c_1c_2c_3}\langle i| \otimes |1\rangle_t\langle 1|\right), \tag{2.151}$$

Expanding the projectors on the control qubits into products of Pauli operators one obtains

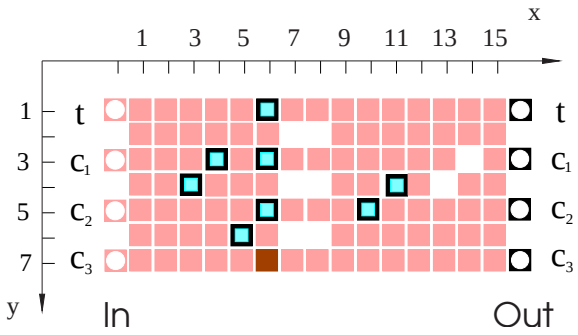


Figure 2.16: The three qubit controlled gate. Qubits displayed as squares in light gray are measured in the σ_x-eigenbasis, the qubit displayed in dark gray is measured in the σ_y-eigenbasis, and the measurement bases of the qubits displayed as framed squares are adaptive.

$$
\begin{aligned}
U_{CARRY} \;=\; & e^{-i\frac{\pi}{4}} \underbrace{\exp\left(-i\frac{\pi}{8}\sigma_z^{(t)}\sigma_z^{(c_3)}\right)}_{U_i} \underbrace{\exp\left(-i\frac{\pi}{8}\sigma_z^{(t)}\sigma_z^{(c_2)}\right)}_{U_h} \underbrace{\exp\left(i\frac{\pi}{8}\sigma_z^{(c_3)}\right)}_{U_g} \underbrace{\exp\left(i\frac{\pi}{8}\sigma_z^{(c_2)}\right)}_{U_f} \\
& \underbrace{\exp\left(i\frac{\pi}{8}\sigma_z^{(c_1)}\right)}_{U_e} \underbrace{\exp\left(i\frac{\pi}{4}\sigma_z^{(t)}\right)}_{U_d} \underbrace{\exp\left(-i\frac{\pi}{8}\sigma_z^{(t)}\sigma_z^{(c_1)}\right)}_{U_c} \underbrace{\exp\left(-i\frac{\pi}{8}\sigma_z^{(c_1)}\sigma_z^{(c_2)}\sigma_z^{(c_3)}\right)}_{U_b} \\
& \underbrace{\exp\left(i\frac{\pi}{8}\sigma_z^{(t)}\sigma_z^{(c_1)}\sigma_z^{(c_2)}\sigma_z^{(c_3)}\right)}_{U_a} .
\end{aligned}
\tag{2.152}
$$

The global phase is henceforth discarded.

The unitary transformation is now subdivided into two parts,

$$U_{CARRY} = U_{h,i}\, U_{a-g}, \tag{2.153}$$

with $U_{a-g} = U_g U_f U_e U_d U_c U_b U_a$ and $U_{h,i} = U_i U_h$. Correspondingly, the cluster on which U_{CARRY} is realized is divided into two sub-clusters. On the first sub-cluster the transformations U_a to U_g are realized, on the second sub-cluster $U_{h,i}$. The measurement pattern to realize U_{CARRY} is displayed in Fig. 2.16. The first sub-cluster stretches from $x = 0$ to $x = 8$, with the input at $x = 0$ and the intermediate output at $x = 8$. The qubits with $8 \leq x \leq 16$ belong to the second sub-cluster.

Let us now explain the sub-gate U_{a-g}. The conversion of the sequence (2.152) of generalized rotations is as in the previous examples. For each generalized rotation there is one cluster qubit in $\mathcal{C}_M(U_{a-g})$ whose measurement basis specifies the respective rotation angle. Specifically, the measurement of the cluster qubit $(3, 4)$ sets the rotation angle of U_a, the measurement of qubit $(4, 3)$ sets the angle for U_b, $(5, 6)$ sets U_c, $(6, 7)$ sets U_d, $(6, 5)$ sets U_e, $(6, 3)$ sets U_f and qubit $(6, 1)$ sets U_g. The quantum correlations of the initial cluster state which induce via the measurements of the cluster qubits in $\mathcal{C}_M(U_{a-g})$ the quantum correlations associated with the generalized rotations are displayed in Fig. 2.17.

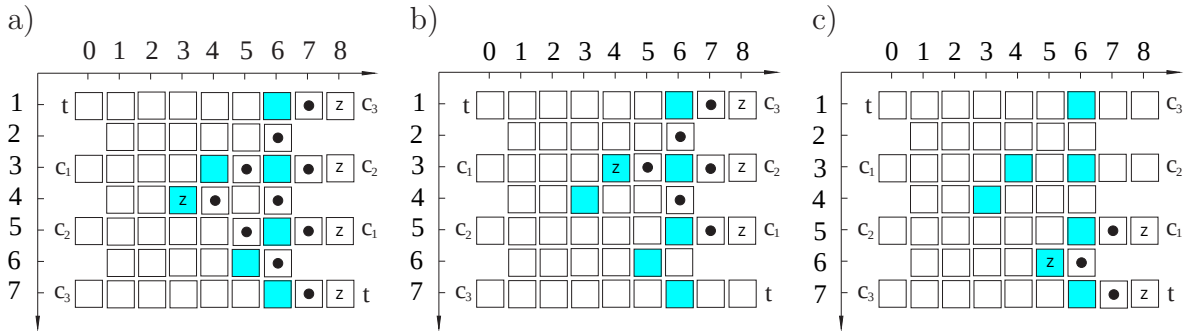


Figure 2.17: Quantum correlations of the initial cluster state $|\phi\rangle_{\mathcal{C}(U_{a-g})}$ on the cluster $\mathcal{C}(U_{a-g})$. These correlations induce via the σ_x-measurements the quantum correlations for the state $|\psi'\rangle$ which act only on the output qubits and one cluster qubit in $\mathcal{C}_M(U_{a-g})$. The pattern of correlation centers in a) displays the correlation required to realize U_a; b) and c) display the correlations for U_b, and U_c, respectively.

The realization of the gate requires two measurement rounds. In the first round the standard measurements of σ_x and σ_y are performed. Note that the rotation angle of U_d is twice as big as for the other rotations. To realize U_d of the cluster qubit $(6,7)$ the observable

$$U_z\left[\pm\frac{\pi}{2}\right]\,\sigma_x U_z\left[\mp\frac{\pi}{2}\right]=\pm\sigma_y \tag{2.154}$$

is measured. Thus, the realization of U_d belongs to the first round of measurements. Strictly speaking, this measurement round does not belong to the gate but to the circuit as a whole since all standard measurements are performed simultaneously.

In the second measurement round, of the remaining qubits in $\mathcal{C}_M(U_{a-g})$ one measures the observables

$$U_z\left[\pm\frac{\pi}{4}\right]\,\sigma_x U_z\left[\mp\frac{\pi}{4}\right]. \tag{2.155}$$

The procedure to infer the sign factors in (2.155) and (2.154) is explained in Section 2.2.6.

The reason why the measurements in the tilted bases may all be performed simultaneously in the second round can be seen as follows. Let $Q_{\nearrow}$ be the set of qubits measured in tilted bases. The contribution $U_{\Sigma,Q_{\nearrow}}$ of the cluster qubits measured in tilted bases to the byproduct operator U_Σ in (2.63) contains only a z-part but no x-part. That is, it has the form

$$U_{\Sigma,Q_{\nearrow}}=\bigotimes_{i\in I\subset\{t,c_1,c_2,c_3\}}\sigma_z^{[i]}. \tag{2.156}$$

In (2.62) the byproduct operator appears "on the wrong side" of U_{a-g} as does the contribution $U_{\Sigma,Q_{\nearrow}}$. When the order of the gate and the byproduct operator is exchanged, the byproduct operator may modify the gate. While this is, not surprisingly, indeed the case for the whole U_Σ, it is not so for the contribution $U_{\Sigma,Q_{\nearrow}}$ coming from the measurements in the tilted bases. Because $U_{\Sigma,Q_{\nearrow}}$ has only a z-part it commutes with U_{a-g}. Therefore,

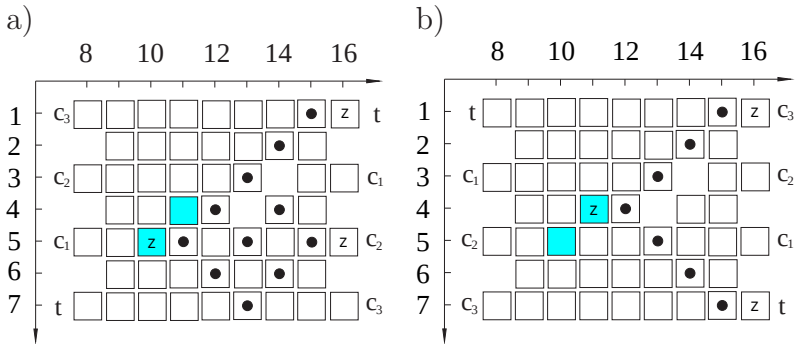


Figure 2.18: Quantum correlations of the initial cluster states on $\mathcal{C}(U_h)$ and $\mathcal{C}(U_i)$. These correlations induce, via the σ_x-measurements, the quantum correlations for the states $|\psi'\rangle_{\mathcal{C}(U_h)}$ and $|\psi'\rangle_{\mathcal{C}(U_i)}$ that involve only the respective output qubits and one qubit in the gate body. The pattern of correlation centers in a) displays the correlation required to realize U_h and b) the correlation for U_i.

the results of measurements in a tilted basis do not mutually affect the choice of their measurement bases.

The fact that hat the byproduct operator $U_{\Sigma,Q_\nearrow}$ is indeed of form (2.156) we do not show here explicitly. For the byproduct operator created in the measurement of qubit $(3,4)$ realizing the transformation U_a it may be verified from equation (2.133) in Section 2.3.2.

The explanation of the second sub-gate, $U_{h,i}$, is analogous. Fig. 2.18 displays the quantum correlations of the initial cluster state which, via the measurements in $\mathcal{C}_M(U_{h,i})$, induce the required quantum correlations associated with U_h and U_i.

Two further points we would like to address in this section. The first is to note that the whole gate U_{CARRY} can be performed on the $\mathrm{QC}_\mathcal{C}$ in two measurement rounds. The first measurement round is that of the σ_x-, σ_y- and σ_z-measurements which, strictly speaking, does not belong to the gate but to the circuit as a whole. The second measurement round is that of the simultaneous measurements in tilted measurement bases.

We have already seen that the measurements that realize the unitary transformations $U_a, .., U_g$ may be realized simultaneously, and this argument may be extended to the entire gate U_{CARRY}. All the byproduct operators created with the measurements in tilted bases have only a z- but no x-part. Therefore they all commute with U_{CARRY}. Thus, to choose the right measurement bases neither of the measurements in a tilted basis that realizes one of the rotations $U_a, .., U_i$ needs to wait for another measurement in a tilted basis.

Second, note that the for U_{CARRY} the target-input and the target-output can be interchanged, see Fig. 2.19. This holds because the (conditional) phase-flip on the target qubit is its own inverse. Thus, the target qubit may travel through the gate backwards. This property also holds for the Toffoli phase gate. We will make use of it in the construction of the quantum adder in the next section.

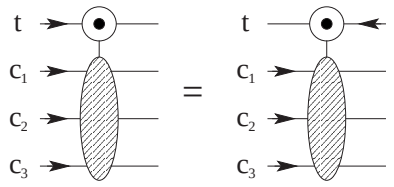


Figure 2.19: In the three-qubit controlled gate $CARRY$, the target qubit may travel either back or forth.

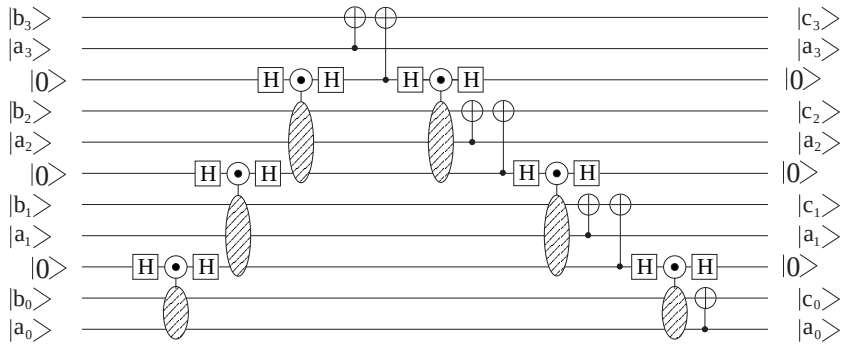


Figure 2.20: Quantum logic network for 4-qubit adder, $c = a + b \bmod 2^4$. The adder network is taken from [60]. The two-qubit controlled gate in this network is the Tofolli phase gate as discussed in Section 2.3.6. A straightforward simulation of this network on the $\mathrm{QC}_{\mathcal{C}}$ would result in a quadratic scaling of spatial resources. However, the more compact realization discussed below requires only a linear overhead.

2.3.7 Circuit for addition

The $\mathrm{QC}_{\mathcal{C}}$-version of the quantum adder corresponds to the quantum logic network as given in [60], see Fig. 2.20. In this thesis we use the three-qubit controlled phase gate $CARRY$ together with a prior and subsequent Hadamard gate on the target qubit while in [60] the equivalent tree-qubit controlled spin-flip gate is used directly.

At first sight it appears as if the horizontal dimension of the cluster to realize the adder circuit would grow linearly with the number of logical qubits n. This is, however, not the case. The $\mathrm{QC}_{\mathcal{C}}$-circuit may be formed in such a way that the horizontal size of the required cluster is constant such that the cluster size increases only linearly with the number n of logical qubits. To see what the $\mathrm{QC}_{\mathcal{C}}$-realization of the quantum adder will look like, the network displayed in Fig. 2.20 may be bent in a way displayed in Fig. 2.21.

To "bend a network" is a rather informal notion. We therefore now specify what we mean by this. If a quantum circuit is displayed as a quantum logic network, the vertical axis usually denotes some spatial dimension, i.e. the location of the qubit carriers, and the horizontal axis corresponds to the sequence of steps of a quantum computation, i.e. a logical time. As the basic blocks of quantum computation in the network model, the universal gates, are unitary transformations generated by suitably chosen Hamiltonians,

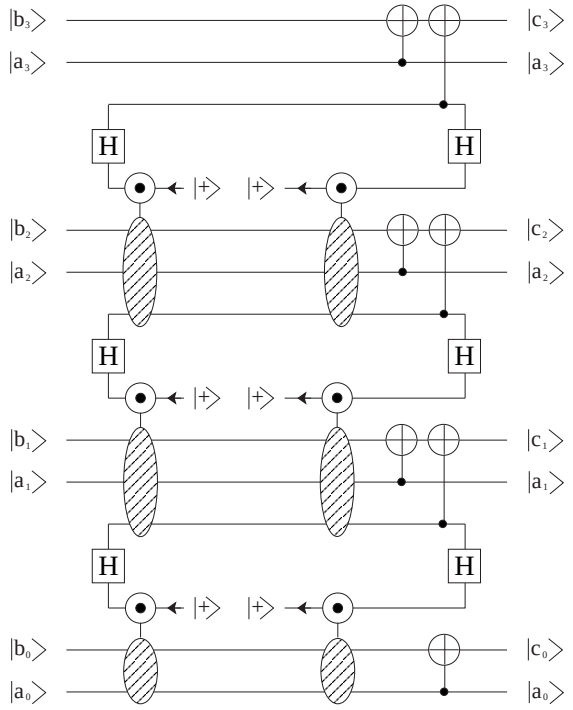


Figure 2.21: Quantum logic network for 4-qubit adder, bent.

the logical time becomes associated with physical time. This is, however, a peculiarity of the network model. If on the $\mathrm{QC}_{\mathcal{C}}$ a quantum logic network is simulated, the temporal axis is converted into an additional spatial axis. The temporal axis in a $\mathrm{QC}_{\mathcal{C}}$-computation emerges anew. It has no counterpart in the network model. If we modify a quantum logic network in such a way that qubits travel from right to left, as done in Fig. 2.21, it does not mean that we propose to use particles that travel backwards in time because we do not need to respect the temporal axis implied by the network model. If one wants a semi-network picture that accounts for this, one may imagine the logical qubits as traveling through pipes on a two-dimensional surface.

The reason why we may let the auxiliary qubits travel "backwards" is the identity displayed in Fig. 2.19. This arrangement of gates makes the circuit more compact. To complete the description of components from which the $\mathrm{QC}_{\mathcal{C}}$-version of the quantum adder is built, a compact measurement pattern for the two combined CNOT gates is displayed in Fig. 2.22. The realization of the quantum adder in the network layout of Fig. 2.21 directly leads to the $\mathrm{QC}_{\mathcal{C}}$-circuit for the quantum adder displayed in Fig. 2.23. Please note that the displayed $\mathrm{QC}_{\mathcal{C}}$-adder is for eight qubits while the networks in Figs. 2.20 and 2.21 are only for four qubits.

For the quantum adder circuit in Fig. 2.23 we have made two further minor simplifications. The first concerns the ancilla preparation. To prepare an ancilla qubit on the cluster in the state $|+\rangle$ means to measure the respective cluster qubit in the σ_x-eigenbasis (the

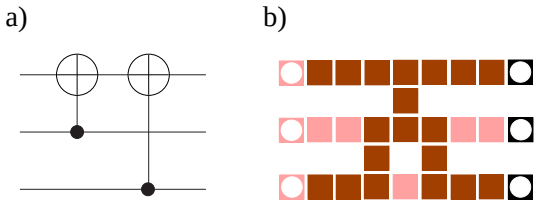


Figure 2.22: Combination of two CNOT gates (a) and its $\mathrm{QC}_{\mathcal{C}}$-realization (b).

randomness of the measurement outcome does not jeopardize the deterministic character of the circuit). As can be seen from the Toffoli gate and the three-qubit controlled gate displayed in Figs. 2.15 and 2.16, the ancilla qubits are located on cluster qubits which have only one next neighbor. As can be verified from the eigenvalue equations (2.1), to measure a qubit of a cluster state which only has one next neighbor in the eigenbasis of σ_x also has the effect of projecting this neighboring cluster qubit into an eigenstate of σ_z. Such cluster qubits may be removed from the cluster as explained in Section 2.2.3. With these neighboring qubits removed the cluster qubits on which the initial ancilla qubits were located become disconnected from the remaining cluster and may thus be removed as well. With the same argument, the cluster qubits carrying the ancillas in their output state, and their next neighbors may also be removed.

Second, between the $\mathrm{QC}_{\mathcal{C}}$-realization of the CARRY gates on the left and the subsequent blocks of CNOT gates we have removed pairs of adjacent cluster qubits that would be measured in the eigenbasis of σ_x. Why this can be done has been explained for adjacent qubits in wires in Section 2.2.7. Here the situation is a little more involved since, like in case of the circuit for Fourier transformation displayed in Section 2.3.5, one of the removed qubits in each pair has more than two neighbors. But the method still works as can be easily verified.

Let us now briefly discuss the resources required for the $\mathrm{QC}_{\mathcal{C}}$-realization of an n-qubit adder. As can be seen directly from the circuit displayed in Fig. 2.23 and the underlying network shown in Fig. 2.21 with its repeating sub-structure, the adder requires a cluster of height $8n-5$ and of constant width 38. Thus the spatial and operational resources are, to leading order,

$$S = O = 304\, n. \tag{2.157}$$

Concerning the temporal resources note that each pair of three-qubit controlled phase gates using the same control qubits and the pair of Toffoli phase gates may be completed at one time instant but that one pair of gates is completed after another. The reason why the measurements in the tilted bases that complete each pair of gates may be performed simultaneously is the same as the one given previously for the measurements in tilted bases of a single three-qubit controlled gate. The propagation of byproduct operators is most easily followed in the network of Fig. 2.20. The temporal complexity T of an n-qubit $\mathrm{QC}_{\mathcal{C}}$-adder is

$$T = n, \tag{2.158}$$

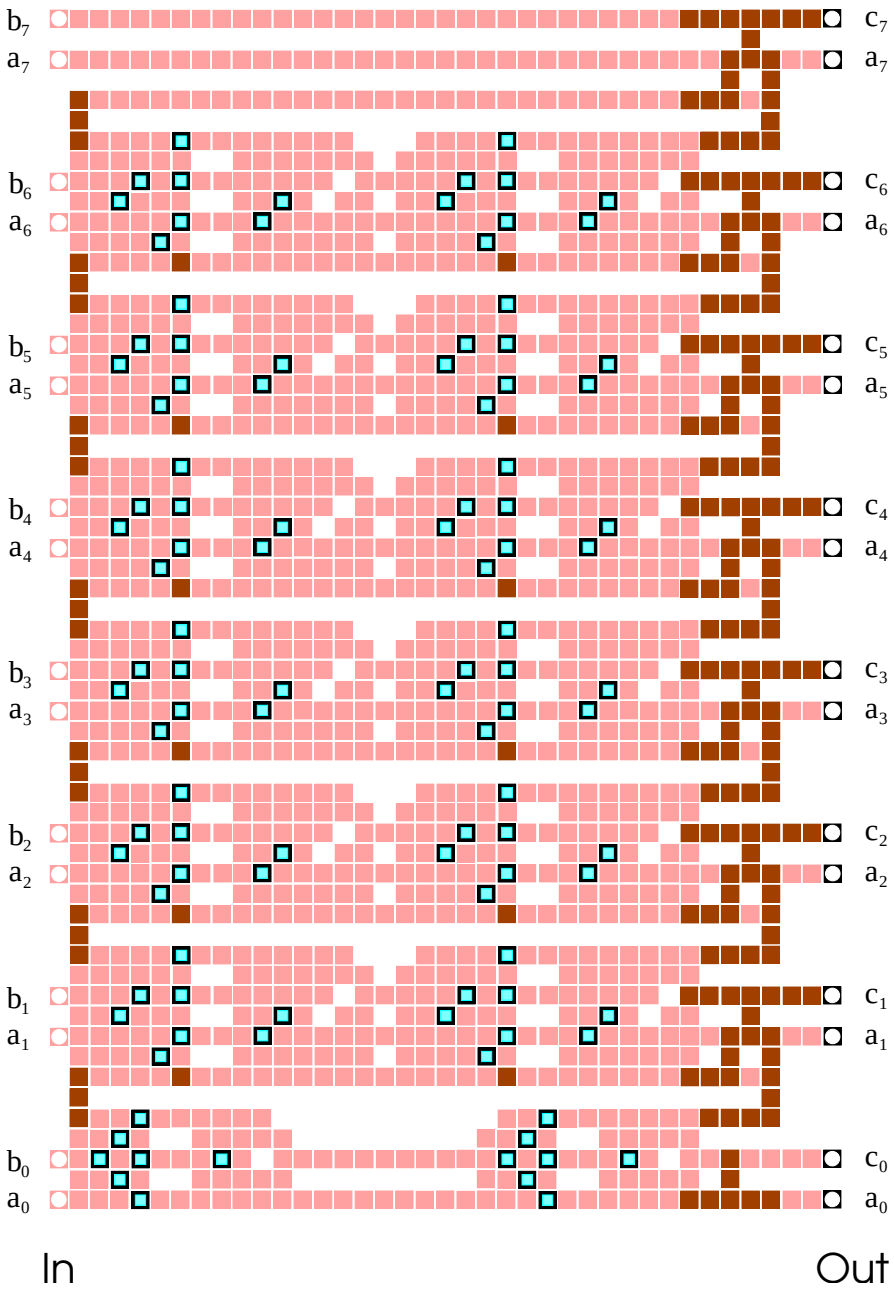


Figure 2.23: Quantum adding circuit for two 8-qubit states. As in all figures displaying $\mathrm{QC}_{\mathcal{C}}$-circuits, squares in light and dark gray represent cluster qubits measured in the σ_x- and σ_y-eigenbasis, respectively. The measurement bases of qubits displayed as framed squares are adaptive.

plus one step of σ_x-,σ_y- and σ_z-measurements for the entire circuit.

The corresponding network resources are to leading order $S_{\text{qln}} = 3n$ and $O_{\text{qln}} = T_{\text{qln}} = 8n$. For the counting of the operational and temporal network resources we have assumed that the three-qubit controlled spin-flip gate used in the addition circuit is composed of two Toffoli gates and one CNOT gate as described in [60], and that the CNOT- and the Toffoli gate are regarded as elementary.

Thus we find for both the network and the $\text{QC}_\mathcal{C}$-realization of the quantum adder that the spatial, temporal and operational resources scale linearly with n. Therefore, the resource overheads in one realization as compared to the other one are only constant. For the $\text{QC}_\mathcal{C}$ this is much better than what is indicated by the bounds (2.115), (2.116) and (2.117), in particular for the spatial and operational resources. Equation (2.116) yields an upper bound on S which is $\sim n^3$ and (2.117) gives bounds on O and S which are $\sim n^2$. Thus, the quantum adder is an example for which these bounds are very loose. In general they should not be mistaken as estimates.

If the pre-factors are compared, one finds that for the realization of a quantum adder the $\text{QC}_\mathcal{C}$ requires about 100 times more spatial and 38 times more operational resources, while it is 8 times faster. However, since we compare different objects these ratios do not mean much apart from the fact that they are constant. It may be argued that in case of the $\text{QC}_\mathcal{C}$ spatial resources are not as precious as they usually are, for to create cluster states one needs a system with non-selective uniform interaction only while for quantum logic networks one generally requires a system with selective interactions among the qubits. Concerning the operational and temporal resources, the $\text{QC}_\mathcal{C}$ only uses one-qubit measurements while the corresponding network uses two- and three-qubit gates as elementary operations.

2.4 Computation with limited spatial resources

In this section we describe how to perform $\text{QC}_\mathcal{C}$-computation on finite and possibly small clusters. If the cluster that may be provided by a specific device is too small for a certain measurement pattern it does not mean that the respective $\text{QC}_\mathcal{C}$-algorithm cannot be run on this device. Instead, the $\text{QC}_\mathcal{C}$-computation may be split into several parts such that each of them fits on the cluster.

To see this consider Scheme 1 for the realization of gates. Scheme 1 is applicable to any gate or sub-circuit. It is thus possible to divide the circuit into sub-circuits each of which fits onto the cluster. The adapted scheme is a process of repetitive re-entangling steps alternating with rounds of measurements.

Specifically, one starts with the realization of the first sub-circuit acting on the fiducial input state located on $I_1 \subset \mathcal{C}$. The fiducial input is, while being processed, teleported to some subset O_1 of the cluster $\mathcal{C}$. The set O_1 of qubits forms the intermediate output of the first sub-circuit. These qubits remain unmeasured while all the other qubits are measured to realize the first sub-circuit. Now the realization of the second sub-circuit begins. Its input state has already been prepared, $I_2 = O_1$. The cluster qubits $a \in \mathcal{C}\backslash O_1$ which have

been measured in the realization of the first sub-circuit are now prepared individually in the state $|+\rangle_a$. This completes step 1 of Scheme 1 to realize the second sub-circuit. Step 2 is to entangle the whole cluster via the Ising interaction. In the third step all cluster qubits except those of the intermediate output O_2 are measured whereby the realization of the second sub-circuit is completed. The intermediate output is now located on O_2. For the realization of the subsequent sub-circuits one proceeds accordingly.

An advantage of this modified procedure is that one gets by with smaller clusters. A disadvantage is that the Clifford part of the circuit may no longer be performed in a single time step.

2.5 Discussion

Let us, at this point, recapitulate how we have so far explained the $\mathrm{QC}_{\mathcal{C}}$. At the end of the universality proof we arrived at a picture very closely resembling that of a network. For each universal gate we found a corresponding measurement pattern on a sub-cluster, and these measurement patterns could be put together like building bricks, as the quantum gates can. In this way, a quantum logic network could be straightforwardly imprinted on a two-dimensional cluster state, with one spatial dimension of the cluster representing the time axis of the network model, and the other the spatial axis of the quantum register. Subsequently, in the search for efficient $\mathrm{QC}_{\mathcal{C}}$-circuits we found that some rules that hold for network circuits do not hold for their respective $\mathrm{QC}_{\mathcal{C}}$-simulations. In this way, we obtained a "network picture with modifications". In the construction of the quantum adder we found that the $\mathrm{QC}_{\mathcal{C}}$ can be regarded as a network quantum computer with the additional feature that the logical qubits may equally run backward and forward on the network time axis. Further, in the discussion of the Clifford part of quantum circuits we found that the $\mathrm{QC}_{\mathcal{C}}$ can be regarded as a network quantum computer with the additional features that some gates do not need to wait for their input to arrive before they can be executed and that the readout quantum register is not measured last, but first. Clearly, these extensions question the suitability of the network picture as a coherent description of the $\mathrm{QC}_{\mathcal{C}}$ altogether. Therefore, we will introduce in the next chapter a more appropriate computational model.

We would like to add two further remarks, one with regard to the elementary constituents of the $\mathrm{QC}_{\mathcal{C}}$, and one with regard to their composition principle. For the particular set of gate simulations used in the $\mathrm{QC}_{\mathcal{C}}$ universality proof in Section 2.2, the CNOT gate and arbitrary one-qubit rotations, there is only a single instance in all examples of Section 2.3 where one of these simple gates has been used as part of a more complicated gate. It is the next-neighbor CNOT gate which has been used as part of the long-distance CNOT described in Section 2.3.3. The observation that the universal gates occur almost not at all is remarkable since the usefulness of a universal set of gates derives from the fact that any circuit is composed of them.

One could say, though, that the used set of gates is not a good choice for the universal set. In fact, in realizations of network quantum computers it is often the physics of the

specific implementation that determines which gates are elementary. For the $\mathrm{QC}_{\mathcal{C}}$ this is not so. The $\mathrm{QC}_{\mathcal{C}}$ may simulate, for example, general one-qubit rotations and Toffoli gates alike. Any gate simulation may be called "elementary" with the same right as any other, but they cannot be all elementary. The elementary constituents of the $\mathrm{QC}_{\mathcal{C}}$ are not gate simulations.

As a consequence, the composition principle for these elements will be different from gate composition. At first sight, if we go through the examples of Section 2.3, we find that this is not yet reflected in the larger and more complicated constructions. For the quantum Fourier transform and the addition circuit we have, though playing with some tricks, ultimately imitated network composition.

However, in the smaller gates and sub-circuits such as the controlled phase gate, the Toffoli phase gate and the gate $CARRY$ we find something that might give rise to a new and more appropriate composition principle. First, for the $\mathrm{QC}_{\mathcal{C}}$ it is not the one-qubit and two-qubit operations that are particularly simple. In the Hamiltonian simulation circuit of Section 2.3.2 we found that it is easy to realize generalized rotations $\exp(i\varphi\,\sigma^{(J)})$ where $\sigma^{(J)}$ is a composite Pauli operator, $\sigma^{(J)} = \bigotimes_{a\in J} \sigma_{k_a}^{(a)}$, $k_a \in \{x, y, z\}$. Furthermore, in the subsequent examples of the multi-qubit gates in Sections 2.3.4 and 2.3.6 we have decomposed the gates into such generalized rotations rather than into known standard gates on fewer qubits.

Any unitary transformation may be decomposed into a single unitary transformation in the Clifford group followed by generalized rotations. So, is this a new composition principle? With our present state of knowledge, the answer must be "Not yet.". First, though any transformation may be rewritten in this form, it is presently not clear how to design quantum algorithms with these elements directly. Second, as we will see in the next chapter, the $\mathrm{QC}_{\mathcal{C}}$ has no quantum register. However, the above decomposition uses the very concept of applying unitary transformations to the state of a quantum register. Therefore, the generalized rotations and their concatenation at least have to be reformulated to fit the description of the $\mathrm{QC}_{\mathcal{C}}$. Nevertheless, it appears that they should be reflected in what may emerge as elementary constituents and the composition principle for the $\mathrm{QC}_{\mathcal{C}}$.

Chapter 3

Computational model underlying the one-way quantum computer

In Chapter 2 we have shown that universal quantum computation can be entirely built on one-qubit measurements on a certain class of highly entangled multi-qubit states, the cluster states [51]. In this scheme, the one-way quantum computer, the cluster state forms a resource for quantum computation and the set of measurements forms the program.

The main point of this chapter is to show that the $\mathrm{QC}_{\mathcal{C}}$ has an independent structure which, among other things, determines the temporal order of measurements. As we shall show, the $\mathrm{QC}_{\mathcal{C}}$ has no quantum register and does not consist of quantum gates. The quantities that are processed with the $\mathrm{QC}_{\mathcal{C}}$ are the outcomes of one-qubit measurements and thus processing of information exists only at the classical level. The $\mathrm{QC}_{\mathcal{C}}$ is nevertheless quantum mechanical as it uses a highly entangled cluster state as the central physical resource.

3.1 Motivation for a non-network model of the $\mathrm{QC}_{\mathcal{C}}$

In the previous section we have described the $\mathrm{QC}_{\mathcal{C}}$ in a network terminology, which has been useful to prove the universality of the scheme. On the other hand, the cluster qubits do not have to be measured in the order prescribed by the order of the gates in the corresponding network. This observation indicates that the network picture does not describe the $\mathrm{QC}_{\mathcal{C}}$ in every respect.

Suppose that in the simulation of a quantum logic network $\mathcal{N}$ on the $\mathrm{QC}_{\mathcal{C}}$ in a network manner –i.e. measuring the "readout" qubits at last– the processing has reached the stage where all but those cluster qubits have been measured which form the output register.

The accumulated byproduct operator U_Σ to be applied to the logical output qubits $1, .., n$ is known. It has the form (2.47)

$$U_\Sigma = \prod_{i=1}^{n} \left(\sigma_x^{[i]}\right)^{x_i} \left(\sigma_z^{[i]}\right)^{z_i},$$

where $x_i, z_i \in \{0,1\}$ for $1 \leq i \leq n$. Let us now label the unmeasured qubits on the cluster in the same way as the readout qubits on the quantum logic network are labelled.

As shown in Section 2.2.5, the output bits s_i' depend in the same way on the parameter x_i of the byproduct operator and on the respective readout measurement result (2.49),

$$s_i' \equiv s_i + x_i \bmod 2.$$

Indeed, there is no need to distinguish between these two contributions. On the level of the byproduct operators, the readout measurement result is translated into an additional contribution U_R to the accumulated byproduct operator, which in this way becomes the extended byproduct operator $U_{\Sigma R}$,

$$U_{\Sigma R} = U_\Sigma U_R. \tag{3.1}$$

From the x-part of $U_{\Sigma R}$ the result of the computation can be read off directly. In (3.1), U_R is given by

$$U_R = \prod_{(O \ni k)=1}^{n} (U_k|_\Omega)^{s_k}, \text{ with } U_k|_\Omega = \sigma_x^{(k)} \; \forall k \in O \tag{3.2}$$

Further, as in (2.55), the accumulated byproduct operator U_Σ can be written as the product of the forward propagated byproduct operators of gates g_i, $U_{\Sigma,g_i}|_\Omega$. Note that the byproduct operator of a gate, (2.23), (2.29) and (2.30), can be written as the product of byproduct operators of individual qubits times a constant byproduct operator. As the forward propagated product of byproduct operators is the same as the product of the individually forward propagated byproduct operators, U_Σ can be written in the form $U_\Sigma = \text{const} \times \prod_{k \in \mathcal{C}_N \setminus O} (U_{\Sigma,k}|_\Omega)^{s_k}$, such that with (3.1) and (3.2) one obtains

$$U_{\Sigma R} = \text{const} \times \prod_{k \in \mathcal{C}_N} (U_{\Sigma,k}|_\Omega)^{s_k}. \tag{3.3}$$

Both contributions to the such extended byproduct operator $U_{\Sigma R}$ stem from random measurement results. In the way they contribute to $U_{\Sigma R}$ there is no difference as to whether these qubits stem from the $\mathrm{QC}_\mathcal{C}$-representation of the network quantum input state, from the sub-cluster of qubits for the realization of the gates or from the sub-cluster for the $\mathrm{QC}_\mathcal{C}$-representation of the readout quantum register. The distinguished role of the readout qubits is a remnant of the interpretation of the $\mathrm{QC}_\mathcal{C}$ as a quantum logic network. For the $\mathrm{QC}_\mathcal{C}$, a distinction of the above three groups of cluster qubits, input qubits, gate qubits and output qubits, as it is suggested by the network model of quantum computation, is not adequate. *All cluster qubits contribute to the result of the $QC_\mathcal{C}$-computation in the same way.* As a consequence, the notions of a quantum register, of quantum input and quantum output are not suitable for the $\mathrm{QC}_\mathcal{C}$ and are therefore abandoned. To find the structures which fill their place is the main motivation for a non-network model of the $\mathrm{QC}_\mathcal{C}$.

3.2 Beyond the network picture

3.2.1 The sets Q_t of simultaneously measurable qubits

The cluster qubits which we have chosen to take the role of the readout register, for example, are just qubits like any other cluster qubits. It turns out that, in a more efficient way of running the $\mathrm{QC}_{\mathcal{C}}$, the "readout" qubits are not the last ones to be measured but among the first. It is advantageous to forget about the network altogether and to view the $\mathrm{QC}_{\mathcal{C}}$ as a set of one-qubit measurements on a resource quantum state, the cluster state. These measurements have to be performed in a certain order and in a certain basis. The classical information of how to measure subsequent qubits must all be contained in the results of the already performed measurements. Similarly, the final result of the computation must be contained in all the measurement outcomes together.

In the following we will adopt the strategy that every cluster qubit is measured at the earliest possible time. This means that each qubit is measured as soon as the required measurement results from other qubits which determine its measurement basis are known. Let us denote by Q_t the set of qubits which can be measured at the same time in the measurement round t. So, how can the sets Q_t be determined? Q_0 is the set of qubits which are measured in the first round. These are all the qubits whose observables σ_x, σ_y or σ_z are measured. The measurement bases for these qubits do not depend on the results of any previous measurements. To determine the subsequent set Q_1, one looks at which qubits can be measured with the knowledge of the measurement results from the qubits in Q_0. Next, one looks which qubits can be measured with the measurement results from the qubits in Q_0 and Q_1 known. These qubits form the set Q_2. In this manner one proceeds until the whole cluster $\mathcal{C}$ is divided into disjoint subsets Q_t.

As will become clear later, it is useful to introduce the sets $Q^{(t)}$ of yet-to-be measured qubits. More precisely, $Q^{(t)}$ is the set of qubits which remain to be measured after measurement round No. $t-1$,

$$Q^{(t)} = \bigcup_{i=t}^{t_{\max}} Q_i. \tag{3.4}$$

Mathematically, the sets Q_t are derived from a strict partial ordering in $\mathcal{C}$. The strict partial ordering, in turn, is generated by forward cones which are explained in the next section.

3.2.2 The forward- and backward cones

Be g a gate in the network $\mathcal{N}$ to be simulated and $k \in \mathcal{C}(g)$ a cluster qubit that belongs to the implementation of g. Further, be $\mathcal{O}$, A and Ω three vertical cuts through the network $\mathcal{N}$. A vertical cut is such that it intersects each qubit line in a network only once and that it does not intersect gates. $\mathcal{O}$ intersects $\mathcal{N}$ just after the gate g, i.e. the byproduct operator $U_{\Sigma,g}$ caused by the implementation of g, as given in (2.23) and (2.29), is located on $\mathcal{O}$. Note that the byproduct operators generated on $\mathcal{O}$ depend on the measurement

results obtained in course of the gate implementation via $(U_k)^{s_k}$. A intersects $\mathcal{N}$ just be fore the input, i.e. an operator propagated to A acts on the input register of $\mathcal{N}$, and Ω intersects $\mathcal{N}$ just before the output such that an operator propagated to Ω acts on the output register of $\mathcal{N}$. For an illustration of the vertical cuts see Fig 2.4 in Section 2.2.5. We can now define the forward- and backward cones of the cluster qubits $k \in \mathcal{C}$.

Definition 2 *The forward cone* $\mathrm{fc}(k)$ *of a cluster qubit* $k \in \mathcal{C}$ *is the set of all those cluster qubits* $j \in Q^{(1)}$ *whose measurement basis* $\mathcal{B}(\varphi_{j,meas})$ *depends on the result* s_k *of the measurement of qubit* k *after the byproduct operator* $(U_k)^{s_k}$ *is propagated from* $\mathcal{O}$ *to* Ω.

Definition 3 *The backward cone* $\mathrm{bc}(k)$ *of a cluster qubit* $k \in \mathcal{C}$ *is the set of all those cluster qubits* $j \in Q^{(1)}$ *whose measurement basis* $\mathcal{B}(\varphi_{j,meas})$ *depends on the result* s_k *of the measurement of qubit* k *after the byproduct operator* $(U_k)^{s_k}$ *is propagated from* $\mathcal{O}$ *to* A.

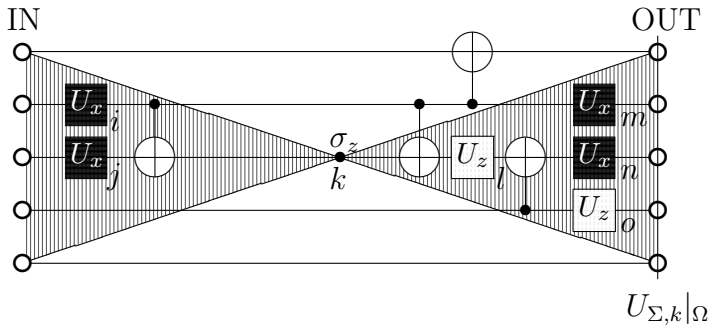


Figure 3.1: Forward and backward cones. The measurement of a cluster qubit k for the implementation of the shown quantum logic network may, depending on the measurement outcome, result in a byproduct operator σ_z (the underlying cluster and measurement pattern is not shown). This byproduct operator is propagated forward to act upon the output register as $U_{\Sigma,k}|_\Omega$. In forward propagation, it flips the measurement angles of the cluster qubits m, n by whose measurement one-qubit rotations are implemented. The cluster qubits m and n are thus in the forward cone of k, $m, n \in \mathrm{fc}(k)$, while l and o are not. Similarly, $i, j \in \mathrm{bc}(k)$.

It will turn out that only the backward cones of the qubits $k \in Q_0$ constitute part of the information specifying an algorithm on the $\mathrm{QC}_\mathcal{C}$, but nevertheless all the backward- and forward cones are important objects in the scheme. Either of the sets, the set of the forward- and that of the backward cones, separately contains the full information of the temporal structure of a computation on the $\mathrm{QC}_\mathcal{C}$.

Let us examine the definitions 2 and 3 for a particular example, the general one-qubit rotation (2.24) as implemented by the Procedure 2 modulo a byproduct operator $U_{\Sigma,Rot}$ as given in (2.29). The measurement result s_1 of qubit 1 (cf. Fig. 2.2) modifies the measurement angle of qubit 2, which is responsible for implementing an x-rotation $U_x(\xi)$, by a factor $(-1)^{s_1}$. Further, it causes a byproduct operator $(\sigma_z)^{s_1}$ at $\mathcal{O}$. If this byproduct operator is propagated forward from $\mathcal{O}$ to Ω it has no effect on qubit 2, because qubit 2 is

behind $\mathcal{O}$. The dependence on s_1 of the basis in which qubit 2 has to be measured persists and thus qubit 2 is in the forward cone of qubit 1, $2 \in \mathrm{fc}(1)$. The situation is different if the byproduct operator $(\sigma_z)^{s_1}$ is propagated backwards from $\mathcal{O}$ to A: via the propagation relation (2.52) the Euler angle ξ is modified by a factor $(-1)^{s_1}$ which has to be accounted for by multiplying the measurement angle $\varphi_{2,\mathrm{meas}}$ by a factor $(-1)^{s_1}$, too. Thus, the factor $(-1)^{s_1}$ modifies the measurement angle $\varphi_{2,\mathrm{meas}}$ twice, once via the Procedure 2 and once in backward propagation, and there is no net effect. Qubit 2 is not in the backward cone of qubit 1, $2 \notin \mathrm{bc}(1)$.

What does it mean that a cluster qubit j is in the forward cone of another cluster qubit k, $j \in \mathrm{fc}(k)$? According to the definition, a byproduct operator created via the measurement at cluster qubit k influences the measurement angle $\varphi_{j,\mathrm{meas}}$ at cluster qubit j. To determine the measurement angle at j one must thus wait for the measurement result at k to see what the byproduct operator created randomly by the measurement at k is. If $j \in \mathrm{fc}(k)$, the measurement at qubit j is performed later than that at qubit k. This we denote by $k \prec j$

$$j \in \mathrm{fc}(k) \Rightarrow k \prec j. \tag{3.5}$$

Please note that the converse of (3.5) is not true. If $k \prec j$ holds, still $j \in \mathrm{fc}(k)$ may not. This can be easily verified for the example of a general rotation (2.24). There, according to the Procedure 2 for implementing such a rotation described in Section 2.2.2, the result of the measurement of qubit 1 enters into in which basis qubit 2 has to be measured. Hence, $2 \in \mathrm{fc}(1)$. By (3.5), $1 \prec 2$ which means that the measurement at qubit 2 has to wait for the result of the measurement on qubit 1. Similarly, the measurement result on qubit 2 enters in the choice of the measurement basis for the measurement on qubit 3. $3 \in \mathrm{fc}(2)$ and thus $2 \prec 3$. Then $1 \prec 3$ also holds as shown below in (3.6) , but $3 \notin \mathrm{fc}(1)$, since the measurement result on qubit 1 does not influence the choice of the measurement basis for the measurement on qubit 3.

The relation "$\prec$" is a strict partial ordering. Suppose, that besides $k \prec j$, for another cluster qubit l one had $l \in \mathrm{fc}(j)$ and thus $j \prec l$. This would mean that the measurement at l must wait for the measurement at j, which itself had to wait for the measurement at k. Thus, the measurement at l also had to wait for the measurement at k. Therefore the relation "$\prec$" is transitive,

$$k \prec j \ \wedge \ j \prec l \longrightarrow k \prec l. \tag{3.6}$$

Further, a measurement to implement a gate cannot and does not need to wait for its own result. Therefore the relation "$\prec$" is anti-reflexive,

$$\neg \exists j \in \mathcal{C} : j \prec j. \tag{3.7}$$

Let us now cast the procedure to construct the sets of simultaneously measured qubits given above in more precise terms. Be $Q_t \subset \mathcal{C}$ the set of cluster qubits measured in measurement round t, and $Q^{(t)} \subset \mathcal{C}$ the set of qubits which are to be measured in the measurement round t and all subsequent rounds, as defined in (3.4). Then, Q_0 is the set of qubits which are measured in the first round. These are the qubits of which the observables σ_x, σ_y or

σ_z are measured, so that the measurement bases are not influenced by other measurement results. Further, $Q^{(0)} = \mathcal{C}$. Now, the sequence of sets Q_t can be constructed using the following recursion relation

$$\begin{array}{lcl} Q_t & = & \left\{q \in Q^{(t)} | \neg \exists p \in Q^{(t)} : p \prec q\right\} \\ Q^{(t+1)} & = & Q^{(t)} \backslash Q_t. \end{array} \tag{3.8}$$

All those qubits which have no precursors in some remaining set $Q^{(t)}$ and thus do not have to wait for results of measurements of qubits in $Q^{(t)}$ are taken out of this set to form Q_t. The recursion proceeds until $Q^{(t_{\max}+1)} = \emptyset$ for some maximal value $t_{\max}$ of t.

Can it happen that the recursion does not terminate? That were the case if for a number m of qubits $j_1, ..., j_m \in \mathcal{C}$ formed a cycle $j_1 \prec j_2 \prec ... \prec j_m \prec j_1$. Then, none of the qubits $j_1, .., j_n$ could ever taken out of the set. However, by transitivity (3.6) we then had $j_1 \prec j_1$ which contradicts anti-reflexivity (3.7). Hence, such a situation cannot occur.

Let us at the end of this section define the forward- and backward cones $\mathrm{fc}(g)$, $\mathrm{bc}(g)$ of the gates g. In eqs. (2.23) and (2.30) we have seen that the byproduct operator caused by the implementation of a CNOT- and the $\pi/2$-phase gate contain a constant contribution, $U_0(CNOT) = \sigma_z^{(c)}$ and $U_0(U_z[\pi/2]) = \sigma_z$. This contributions to the respective byproduct operators do not depend on any local variables such as the measurement results and are thus attributed to the gate as a whole. These byproduct operators are of the same form as those depending on the individual measurement results and can influence measurement angles when being propagated forward or backward. Thus we define the forward- and backward cones of gates, in analogy to those of the cluster qubits $k \in \mathcal{C}$, as follows:

The forward cone $\mathrm{fc}(g)$ of a gate $g \in \mathcal{N}$ is the set of all those cluster qubits $j \in Q^{(1)}$ of which the measurement basis $\mathcal{B}(\varphi_{j,\mathrm{meas}})$ is modified if the byproduct operator $U_{0,g}$ is propagated forward from $\mathcal{O}$ to Ω.

The backward cone $\mathrm{bc}(g)$ of a gate $g \in \mathcal{N}$ is the set of all those cluster qubits $j \in Q^{(1)}$ of which the measurement basis $\mathcal{B}(\varphi_{j,\mathrm{meas}})$ is modified if the byproduct operator $U_{0,g}$ is propagated backward from $\mathcal{O}$ to A.

The forward- and backward cones of gates do not form part of the information representing a quantum algorithm on the $\mathrm{QC}_{\mathcal{C}}$, they will be absorbed into the algorithm angles and the initial value of the information flow vector $\mathbf{I}_{\mathrm{init}}$ introduced below. Their role for the description of a computation on the $\mathrm{QC}_{\mathcal{C}}$ is a technical one.

3.2.3 The algorithm- and measurement angles

There are three different types of angles involved in the described scheme of quantum computation of which the most prominent are the algorithm angles and the measurement angles.

The *algorithm angles* $\left\{\varphi_{j,\mathrm{algo}}, j \in Q^{(1)}\right\}$ are part of the information that specifies an algorithm on the $\mathrm{QC}_{\mathcal{C}}$. They are derived from the network angles $\left\{\varphi_{j,\mathrm{qln}}, j \in Q^{(1)}\right\}$, i.e. the Euler angles of the one-qubit rotations in the quantum logic network. Further, the algorithm angles depend on the set $\left\{\kappa_k, k \in \mathcal{C}\right\}$ characterizing the cluster state $|\phi\rangle_{\mathcal{C}}$ in

(2.1), and on special properties of the measurement pattern. We see that the network angles are absorbed into the algorithm angles. They do not constitute part of the information specifying a $\mathrm{QC}_{\mathcal{C}}$-algorithm.

As described before, the process of computation with the $\mathrm{QC}_{\mathcal{C}}$ comprises several measurement rounds. The first round, in wich the qubits in the set Q_0 are measured, is somewhat different from the following rounds. Therein, all gates of the circuit that belong to the Clifford group are implemented at the same time, no matter where they are located in the corresponding quantum logic network and in which step they would be carried out there. This results in byproduct operators scattered all over the place. These byproduct operators are, according to the scheme described in Section 3.4.2, propagated backwards. To account for the effect that the byproduct operators have on the algorithm angles, these angles have to updated to the *modified algorithm angles* $\left\{\varphi'_{j,\mathrm{algo}}, j \in Q^{(1)}\right\}$. The modified algorithm angles $\varphi'_{j,\mathrm{algo}}$ are calculated from the respective algorithm angles $\varphi_{j,\mathrm{algo}}$ and the results obtained in the first measurement round $\left\{s_k, k \in Q_0\right\}$. In the subsequent measurement rounds no further update of the modified algorithm angles occurs. Finally, each qubit $j \in Q_t \subset Q^{(1)}$ is measured in some measurement round t in the basis $\mathcal{B}(\varphi_{j,\mathrm{meas}})$ where $\varphi_{j,\mathrm{meas}}$ denotes the *measurement angle* of qubit j. The measurement angle $\varphi_{j,\mathrm{meas}}$ of a qubit $j \in Q_t$ is calculated from the modified algorithm angle, $\varphi'_{j,\mathrm{algo}}$ and the results $\left\{s_k, k \in \bigcup_{i=0}^{t-1} Q_i\right\}$ of the so far obtained measurements.

Before a quantum algorithm is run on the $\mathrm{QC}_{\mathcal{C}}$, the algorithm angles are determined from the cluster and the properties of the algorithm. During runtime of the $\mathrm{QC}_{\mathcal{C}}$, in the first measurement round ($t = 0$), the algorithm angles are replaced by the modified algorithm angles, i.e. only the latter are kept while the former are erased. Then, in the measurement round t a qubit $j \in Q_t$ is measured in the basis determined by the measurement angle $\varphi_{j,\mathrm{meas}}$. After the measurement of qubit j both $\varphi'_{j,\mathrm{algo}}$ and $\varphi_{j,\mathrm{meas}}$ can be erased.

Now there arises the question of how the measurement angles of the actual measurements are calculated from the results of previous measurements. This question will be answered in Section 3.4.2. The question which interests us most, of course, is: "How can the final result of the computation be determined from all the measurement outcomes?" It will turn out that the answers to both questions are very much related.

3.2.4 Quantities for the processing of the measurement results

The information vector

First, we define the *information vector* $\mathbf{I}$, a $2n$-component binary vector which is a function of the quantities $\{x_i, z_i\}$ and the results $\{s_i\}$ of the measurements on the cluster output register. The information vector contains the computational result. It can be read off from the extended byproduct operator $U_{\Sigma R}$.

Definition 4 *The information vector* $\mathbf{I}$ *is given by*

$$\mathbf{I} = \begin{pmatrix} \\ \mathbf{I}_x \\ \\ \\ \mathbf{I}_z \\ \\ \end{pmatrix}, \quad \textit{with } \mathbf{I}_x = \begin{pmatrix} x_1 + s_1 \\ x_2 + s_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n + s_n \end{pmatrix}, \quad \mathbf{I}_z = \begin{pmatrix} z_1 \\ z_2 \\ \cdot \\ \cdot \\ \cdot \\ z_n \end{pmatrix}. \tag{3.9}$$

As can be seen from (2.49) and (3.9), $\mathbf{I}_x$ is a possible result of a readout measurement in a corresponding quantum logic network. $\mathbf{I}_z$ is redundant. However, in Section 3.4.1 the flow quantity $\mathbf{I}(t)$, the *information flow vector*, will be defined for which $\mathbf{I}(t_{\max}) = \mathbf{I}$, with $t_{\max}$ the index of the final computational step. For $t < t_{\max}$, in $\mathbf{I}(t)$ both the z-part $\mathbf{I}_z(t)$ and the x-part $\mathbf{I}_x(t)$ are required to determine the bases for the one-qubit measurements in Q_{t+1}. As $\mathbf{I}_z(t)$ is of equal importance as $\mathbf{I}_x(t)$ throughout the process of computation we keep $\mathbf{I}_z$ in the definition of $\mathbf{I}$ as well.

The set of possible information vectors $\mathbf{I}$ forms a $2n$ dimensional vector space over F_2, $\mathcal{V}$. Let us consider the group $\mathcal{U}^{\text{local}}$ of all possible extended byproduct operators $U_{\Sigma R}$. If we divide out the normal divisor $\{\pm 1\}$ of $\mathcal{U}^{\text{local}}$, the resulting factor group $\overline{\mathcal{U}} = \mathcal{U}^{\text{local}}/\{\pm 1\}$ is isomorphic to $\mathcal{V}$. From the viewpoint of physics, dividing out the normal divisor $\{\pm 1\}$ means that we ignore a global phase. The isomorphism $\mathcal{I}$ which maps an $\mathbf{I} \in \mathcal{V}$ to the corresponding $U_{\Sigma R} \in \overline{\mathcal{U}}$ is given by

$$\mathcal{I}: \ \mathcal{V} \ni \mathbf{I} \longrightarrow U_{\Sigma R} = \prod_{i=1}^{n} \left(\sigma_x^{(i)}\right)^{[\mathbf{I}_x]_i} \left(\sigma_z^{(i)}\right)^{[\mathbf{I}_z]_i} \in \overline{\mathcal{U}}, \tag{3.10}$$

where $[\mathbf{I}_x]_i$ and $[\mathbf{I}_z]_i$ are the respective components of $\mathbf{I}_x$ and $\mathbf{I}_z$. The component-wise addition of vectors in $\mathcal{V}$ corresponds, via the isomorphism $\mathcal{I}$, to the multiplication of byproduct operators modulo a phase factor $\{\pm 1\}$. The procedure to implement this product is to first use the operator product, then bring the factors into normal order according to (3.10) and finally drop the phase. Multiplication of vectors $\mathbf{I} \in \mathcal{V}$ with the scalars 0,1 corresponds to raising the byproduct operators $U_{\Sigma R} \in \overline{\mathcal{U}}$ to the respective powers. One may switch between the two pictures via the isomorphism (3.10). The algebraic structures involved will be more apparent in the representation using the information vector $\mathbf{I} = \mathbf{I}(\{x_i, z_i, s_i\})$ than in the formulation of the operator $U_{\Sigma R}$.

Now that we have defined the information vector $\mathbf{I}$ in (3.9) and have seen that the result of the computation can be directly read off from the x-part of $\mathbf{I}$, we would like to find out how $\mathbf{I}$ depends on the measurement outcomes $\{s_k\}$ and the set $\{\kappa_k\}$ of binary numbers that determine the cluster state $|\phi\rangle_{\mathcal{C}}$ in (2.1). This task is left until Section 3.4.1. Before we can accomplish it we need some further definitions. It will turn out that the information vector $\mathbf{I}$ can be written as a linear combination of the *byproduct images* which are explained next.

The byproduct images

Be Ω the "cut" through a network $\mathcal{N}$ which intersects the qubit lines just before its output. This is the cut at which the extended byproduct operator $U_{\Sigma R}$ is accumulated. Consider a qubit k on the cluster $\mathcal{C}$ which is measured in the course of computation. Depending on the result of the measurement on qubit k, a byproduct operator $(U_k)^{s_k}$ is introduced in $\mathcal{N}$ at the location of the logical output qubits of the gate for whose implementation the cluster qubit k was measured. This byproduct operator U_k can –by using the propagation relations (2.52), (2.51) and (2.53)– propagated from where it occurred to the cut Ω. There it appears as the forward propagated byproduct operator $U_k|_\Omega$. Now we can define the *byproduct image* $\mathbf{F}_k$ of a cluster qubit $k \in \mathcal{C}$. Each cluster qubit $k \in \mathcal{C}$ has a byproduct image.

Definition 5 *For each cluster qubit $k \in \mathcal{C}$ the byproduct image $\mathbf{F}_k$ is the vector that corresponds via the isomorphism $\mathcal{I}^{-1}$ (3.10) to the forward propagated byproduct operator $U_k|_\Omega$,*

$$\mathbf{F}_k = \mathcal{I}^{-1}\left(U_k|_\Omega\right). \tag{3.11}$$

In the definition (3.11) of the byproduct image $\mathbf{F}_k$ it is mentioned only implicitly that the image is evaluated on the cut Ω. Later in the discussion it will become apparent that we could evaluate the byproduct image on every vertical cut $\mathcal{O}$. Sometimes, if we compare to other vertical cuts, we will explicitly write $\mathbf{F}_k|_\Omega$ for $\mathbf{F}_k$.

The set of byproduct images $\{\mathbf{F}_k, k \in \mathcal{C}\}$ is an important quantity for the scheme. It represents part of the information which is needed to run a quantum algorithm with the $\mathrm{QC}_\mathcal{C}$.

In eq. (2.23) the byproduct operator for the CNOT gate as realized according to Fig. 2.2 is given. This byproduct operator contains a constant contribution $U_0(CNOT) = \sigma_z^{(c)}$. As U_0 does not depend on any local variables, neither on $\{s_k\}$ nor on $\{\kappa_k\}$, it makes no sense to attribute it to any of the cluster qubits that were measured to realize the gate. Instead, it is attributed to the part of the measurement pattern that implements the gate as a whole, or –for simplicity– to the gate itself. For any gate g, $U_{0,g}$ can be propagated forward to the cut Ω to act upon the "readout" qubits. There it appears as the forward propagated byproduct operator $U_{0,g}|_\Omega$. In analogy to the byproduct images of the cluster qubits, we can now define the byproduct images of the gates g of the quantum logic network that is simulated on the $\mathrm{QC}_\mathcal{C}$. For any such gate g the byproduct image $\mathbf{F}_g$ is the vector that corresponds to $U_{0,g}|_\Omega$ via

$$\mathbf{F}_g = \mathcal{I}^{-1}(U_{0,g}|_\Omega). \tag{3.12}$$

Please note that in contrast to the byproduct images $\mathbf{F}_k$ of cluster qubits $k \in \mathcal{C}$ the byproduct images $\mathbf{F}_g$ of gates do not form a separate part of the information specifying a quantum algorithm on the $\mathrm{QC}_\mathcal{C}$. They will be absorbed into the initialization value $\mathbf{I}_{\mathrm{init}}$ of the information flow vector defined in Section 3.4.1 and they are thus only a convenient tool in the derivation of the computational model.

Via $\mathcal{I}^{-1}$ we map the multiplication of byproduct operators, i.e. their accumulation, onto addition modulo 2 on the level of the vectors in $\mathcal{V}$. Now there arises the question

whether other operations on the byproduct operators could be expressed in terms of the corresponding vectors, too. Specifically, one may ask how the byproduct operator propagation looks like on the level of the $\mathbf{I} \in \mathcal{V}$.

The propagation matrices

The answer to this question is that on the level of the vector quantities in $\mathcal{V}$ propagation is described by multiplication with certain $2n \times 2n$-matrices C. Consider two cuts $\mathcal{O}_1$ and $\mathcal{O}_2$ through a network which intersect each qubit line only once. Further, be the two cuts such that they do not intersect each other and that $\mathcal{O}_1$ is earlier than $\mathcal{O}_2$. The part of the quantum logic network between $\mathcal{O}_1$ and $\mathcal{O}_2$ is denoted by $\mathcal{N}_{\mathcal{O}_1 \to \mathcal{O}_2}$. Be $\mathbf{I}_k|_{\mathcal{O}_1}$ and $\mathbf{I}_k|_{\mathcal{O}_2}$ the vectors describing a byproduct operator resulting from the measurement of qubit k, propagated to the cuts $\mathcal{O}_1$ and $\mathcal{O}_2$, respectively. Then we have

$$\mathbf{I}_k|_{\mathcal{O}_2} = C(\mathcal{N}_{\mathcal{O}_1 \to \mathcal{O}_2})\, \mathbf{I}_k|_{\mathcal{O}_1}. \tag{3.13}$$

To any quantum logic network $\mathcal{N}$ a matrix $C_{\mathcal{N}}$ can be assigned. For a network $\mathcal{N}_2 \circ \mathcal{N}_1$ composed of two subnetworks $\mathcal{N}_1$ and $\mathcal{N}_2$ (of which $\mathcal{N}_1$ is carried out first) the propagation matrix is equal to the product of the propagation matrices of the subnetworks

$$C(\mathcal{N}_2 \circ \mathcal{N}_1) = C(\mathcal{N}_2) C(\mathcal{N}_1). \tag{3.14}$$

Because of property (3.14) we only need to find the propagation matrices for the general one-qubit rotations, the CNOT-, the Hadamard- and the $\pi/2$-phase gate. The one-qubit rotations and the CNOT gate alone form a universal set of gates. The reason why we also include the Hadamard- and the $\pi/2$-phase gate is that here they are treated differently from the general rotations, as can be seen from the propagation relations (2.52) and (2.53). By propagation through a Hadamard- or $\pi/2$-phase gate, the gate is left unchanged while the byproduct operator changes; whereas for the propagation through a general rotation, the rotation changes and the byproduct operator stays the same. Thus, for finding the byproduct images the general rotations in $\mathcal{N}$ can be replaced by the identity. Only the CNOT-, Hadamard and $\pi/2$-phase gates have an effect. The special treatment of the Hadamard and the $\pi/2$-phase gate is advantageous with respect to the temporal complexity of a computation, because if one uses the propagation relation (2.53) the implementation of the Hadamard- and the $\pi/2$-phase gate does not need to wait for results of any previous measurements. To sum up, to each possible $\mathcal{N}$ belongs a unitary operation $U(\mathcal{N})$ in the Clifford group and a corresponding matrix $C(\mathcal{N})$, such that

$$\mathcal{I}\left(C(\mathcal{N})\mathbf{I}\right) = U(\mathcal{N})\, \mathcal{I}\left(\mathbf{I}\right)\, U(\mathcal{N})^{\dagger}, \quad \forall \mathbf{I} \in \mathcal{V}. \tag{3.15}$$

Let us now give the propagation matrices for propagation through CNOT-, Hadamard and $\pi/2$-phase gates. The propagation matrices C are conveniently written in block form

$$C = \left(\begin{array}{c|c} C_{xx} & C_{zx} \\ \hline C_{xz} & C_{zz} \end{array}\right), \tag{3.16}$$

where C_{xx}, C_{zx}, C_{xz} and C_{zz} are $n \times n$ matrices with binary-valued entries.

For the Hadamard gate $H^{(i)}$ on the logical qubit i one finds

$$\begin{array}{rcl} \left[C_{xx}(H^{(i)})\right]_{kl} = \left[C_{zz}(H^{(i)})\right]_{kl} &=& \delta_{kl} + \delta_{ki}\delta_{il}, \\ \left[C_{zx}(H^{(i)})\right]_{kl} = \left[C_{xz}(H^{(i)})\right]_{kl} &=& \delta_{ki}\delta_{il}, \end{array} \tag{3.17}$$

where e.g. $\left[C_{xx}(H^{(i)})\right]_{kl}$ denotes the entry of row k and column l in C_{xx}. Note that the qubit index i is not summed over in (3.17) and that the addition is modulo 2.

For the $\pi/2$-phase gate $U_z^{(i)}(\pi/2)$ on the logical qubit i one finds

$$\begin{array}{rcl} \left[C_{xx}(U_z^{(i)}(\pi/2))\right]_{kl} &=& \delta_{kl}, \\ \left[C_{zz}(U_z^{(i)}(\pi/2))\right]_{kl} &=& \delta_{kl}, \\ \left[C_{xz}(U_z^{(i)}(\pi/2))\right]_{kl} &=& \delta_{ki}\delta_{il}, \\ \left[C_{zx}(U_z^{(i)}(\pi/2))\right]_{kl} &=& 0. \end{array} \tag{3.18}$$

For the CNOT gate on control qubit c and target qubit t one finds the propagation matrix $C(\mathrm{CNOT}(c,t))$ with

$$\begin{array}{rcl} \left[C_{xx}(\mathrm{CNOT}(c,t))\right]_{kl} &=& \delta_{kl} + \delta_{kt}\delta_{cl}, \\ \left[C_{zz}(\mathrm{CNOT}(c,t))\right]_{kl} &=& \delta_{kl} + \delta_{kc}\delta_{tl}, \\ C_{zx}(\mathrm{CNOT}(c,t)) &=& 0, \\ C_{xz}(\mathrm{CNOT}(c,t)) &=& 0. \end{array} \tag{3.19}$$

We will make use of the propagation matrices in the discussion of temporal complexity of algorithms on the $\mathrm{QC}_{\mathcal{C}}$ in Section 3.5.1.

For the action of the propagation matrices C on the vectors $\mathbf{I} \in \mathcal{V}$ there exist conserved quantities. One of them, $\mathbf{I}_{x,1}^T\mathbf{I}_{z,2} + \mathbf{I}_{z,1}^T\mathbf{I}_{x,2}$, is discussed in the next section.

Conservation of the symplectic scalar product

The symplectic scalar product

$$(\mathbf{I}_1, \mathbf{I}_2)_S = \mathbf{I}_{x,1}^T\mathbf{I}_{z,2} + \mathbf{I}_{z,1}^T\mathbf{I}_{x,2} \bmod 2 \tag{3.20}$$

is conserved. For any $\mathbf{I}_1, \mathbf{I}_2 \in \mathcal{V}$ and C the identity

$$(\mathbf{I}_1, \mathbf{I}_2)_S = (C\mathbf{I}_1, C\mathbf{I}_2)_S \tag{3.21}$$

holds. Let us briefly explain why the symplectic scalar product (3.20) is conserved. First, note that the symplectic scalar product tells whether two operators $\mathcal{I}(\mathbf{I}_1)$, $\mathcal{I}(\mathbf{I}_2)$ in the Pauli group commute or anti-commute,

$$\mathcal{I}(\mathbf{I}_1)\mathcal{I}(\mathbf{I}_2) = (-1)^{(\mathbf{I}_1,\mathbf{I}_2)_S}\mathcal{I}(\mathbf{I}_2)\mathcal{I}(\mathbf{I}_1). \tag{3.22}$$

Relation (3.22) is the only place in this thesis where we pay attention to the sign factor of a byproduct operator. There, the product, e.g. $\mathcal{I}(\mathbf{I}_1)\mathcal{I}(\mathbf{I}_2)$, denotes the usual operator product. However, everywhere else in this thesis a product $\mathcal{I}(\mathbf{I}_1)\mathcal{I}(\mathbf{I}_2)$ denotes operator multiplication modulo a global phase factor ± 1, i.e. the product is normal ordered as in (3.10) and the phase factor is dropped.

Using relation (3.22), the invariance (3.21) of the scalar product (3.20) is easily demonstrated. Consider the quantity $\mathcal{I}(C\mathbf{I}_1)\mathcal{I}(C\mathbf{I}_2)$ with $\mathcal{I}(C\mathbf{I}_1) = U\,\mathcal{I}(\mathbf{I}_1)\,U^\dagger$ and $\mathcal{I}(C\mathbf{I}_2) = U\,\mathcal{I}(\mathbf{I}_2)\,U^\dagger$ as in (3.15). Then, we find

$$\begin{aligned}
\mathcal{I}(C\mathbf{I}_1)\mathcal{I}(C\mathbf{I}_2) &= U\,\mathcal{I}(\mathbf{I}_1)\,U^\dagger\,U\,\mathcal{I}(\mathbf{I}_2)\,U^\dagger \\
&= U\,\mathcal{I}(\mathbf{I}_1)\,\mathcal{I}(\mathbf{I}_2)\,U^\dagger \\
&= (-1)^{(\mathbf{I}_1,\mathbf{I}_2)_S} U\,\mathcal{I}(\mathbf{I}_2)\,\mathcal{I}(\mathbf{I}_1)\,U^\dagger \\
&= (-1)^{(\mathbf{I}_1,\mathbf{I}_2)_S} U\,\mathcal{I}(\mathbf{I}_2)\,U^\dagger\,U\,\mathcal{I}(\mathbf{I}_1)\,U^\dagger \\
&= (-1)^{(\mathbf{I}_1,\mathbf{I}_2)_S}\mathcal{I}(C\mathbf{I}_2)\mathcal{I}(C\mathbf{I}_1),
\end{aligned} \tag{3.23}$$

where the third line holds by (3.22). On the other hand, as we can see from (3.22) directly that

$$\mathcal{I}(C\mathbf{I}_1)\mathcal{I}(C\mathbf{I}_2) = (-1)^{(C\mathbf{I}_1,C\mathbf{I}_2)_S}\mathcal{I}(C\mathbf{I}_2)\mathcal{I}(C\mathbf{I}_1). \tag{3.24}$$

From (3.23) and (3.24) together it follows that $(\mathbf{I}_1, \mathbf{I}_2)_S = (C\mathbf{I}_1, C\mathbf{I}_2)_S$, as stated in (3.21).

The symplectic scalar product (3.20) will prove useful in determining the measurement angles from previously obtained measurement results.

The cone test

The cone test is used to find out whether two measurements, which are part of some gates of a circuit, influence each other, i.e. whether one of the measurements has to wait for the result of the other. The cone test does not reveal which of the two measurements has to be performed first.

Let j, k be some cluster qubits $k \in \mathcal{C}$ and $j \in Q^{(1)}$. Qubit j is not measured in the first measurement round and thus the observable measured at qubit j is a nontrivial linear combination of σ_x and σ_y, hence j can be in the forward and backward cones of some other cluster qubits. We would like to find out whether j is in the forward or backward cone of k. For this question the cone test provides a necessary and sufficient criterion. It reads

$$\forall\, k \in \mathcal{C}, j \in Q^{(1)} : \quad j \in \mathrm{fc}(k) \vee j \in \mathrm{bc}(k) \Longleftrightarrow (\mathbf{F}_j, \mathbf{F}_k)_S = 1. \tag{3.25}$$

To check whether a qubit lies in some other qubits backward or forward cone we only need the two byproduct images and can use the symplectic scalar product.

We further observe that

$$\forall\, j,k \in Q^{(1)}: \quad k \in \mathrm{fc}(j) \Longleftrightarrow j \in \mathrm{bc}(k). \tag{3.26}$$

If we confine k to $k \in Q^{(1)} \subset \mathcal{C}$ we can insert (3.26) into (3.25) such that

$$\forall\, j,k \in Q^{(1)}: \quad j \in \mathrm{fc}(k) \vee k \in \mathrm{fc}(j) \Longleftrightarrow (\mathbf{F}_j, \mathbf{F}_k)_S = 1. \tag{3.27}$$

The expression on the l.h.s. of (3.27) is symmetric with respect to j and k. This fits in well since the r.h.s of (3.27) is also symmetric.

Similar to (3.25) we can give a criterion for whether or not a qubit $j \in Q^{(1)}$ is in the forward- or backward cone $\mathrm{fc}(g)$, $\mathrm{bc}(g)$ of some gate g. It reads

$$\forall g \in \mathcal{N},\ j \in Q^{(1)}:\ j \in \mathrm{fc}(g) \vee j \in \mathrm{bc}(g) \Longleftrightarrow (\mathbf{F}_j, \mathbf{F}_g)_S = 1. \tag{3.28}$$

Proof of (3.25), (3.28) *and* (3.26). Considering the cone test, first note that whether a one-qubit rotation at some position in the network is about the z-axis or a about the x-axis can be identified by the potential byproduct operator produced when the rotation is implemented. This can be seen by inspecting (2.24), (2.28), (2.29) and the Procedure 2 to implement a general rotation as described in Section 2.2.2. The x-rotations $U_x(\xi)$ and $U_x(\zeta)$ of $U_R(\xi,\eta,\zeta)$ in (2.24) are implemented by measurements on the qubits 2 and 4 of a 5-qubit chain. As can be seen from (2.29), they contribute to the byproduct operator U_Σ of the rotation U_R with ${\sigma_x}^{s_2+s_4}$ where s_2 and s_4 are the results of the measurements on qubits 2 and 4. Further, the rotation about the z-axis, $U_z(\zeta)$, is implemented by measurement of qubit 3. The contribution to the byproduct operator which is thereby generated is, from (2.29), ${\sigma_z}^{s_3}$. We see that x-rotations only generate byproduct operators σ_x and z-rotations only generate byproduct operators σ_z.

A byproduct operator generated via the measurement on the cluster qubit k must be propagated either forward or backward to possibly reach the rotation on the logical qubit i implemented via the measurement on the cluster qubit j. Let be $\mathcal{O}_K$ and $\mathcal{O}_J$ two cuts through the network which intersect each logical qubit line only once. More specifically, $\mathcal{O}_K$ intersects the qubit line i just before the rotation implemented by the measurement at cluster qubit k. $\mathcal{O}_J$ intersects the qubit line i just before the rotation implemented by the measurement at cluster qubit j.

There are two cases which can occur. Either the cut $\mathcal{O}_K$ is before the cut $\mathcal{O}_J$ in the network $\mathcal{N}$ which we denote by $\mathcal{O}_K \leq \mathcal{O}_J$, or $\mathcal{O}_J$ is before the cut $\mathcal{O}_K$ which we denote by $\mathcal{O}_J \leq \mathcal{O}_K$. It can also be that both is true at the same time but it cannot be that neither of the two relations hold.

Case I: $\mathcal{O}_K \leq \mathcal{O}_J$.

The byproduct operator generated via the measurement at qubit k must be propagated forward to possibly affect the measurement at qubit j. It is not possible that the result of the measurement on qubit j has an effect on the measurement basis chosen at k.

Let us introduce a further cut $\mathcal{O}_{J'}$ which is the same as $\mathcal{O}_J$, except for that it intersects the line of the logical qubit i in the network $\mathcal{N}$ just after the rotation implemented via the measurement on the cluster qubit j. The potential byproduct operator which is generated via the measurement on cluster qubit k and then propagated forward to the cuts $\mathcal{O}_J$ and $\mathcal{O}_{J'}$, is denoted by $U_k|_{\mathcal{O}_J}$ and $U_k|_{\mathcal{O}_{J'}}$, respectively (the byproduct operators which are actually generated are $(U_k|_{\mathcal{O}_J})^{s_k}$ and $\left(U_k|_{\mathcal{O}_{J'}}\right)^{s_k}$). Further, we denote the restriction of the byproduct operators $U_k|_{\mathcal{O}_J}$ and $U_k|_{\mathcal{O}_{J'}}$ to the logical qubit i by $[U_k|_{\mathcal{O}_J}]_i$ and $\left[U_k|_{\mathcal{O}_{J'}}\right]_i$. The two cuts differ only on the logical qubit i, and there only by the side of the respective cut on which the rotation is located. Therefore, using (2.52), it follows that $U_k|_{\mathcal{O}_J} = U_k|_{\mathcal{O}_{J'}}$. Hence also

$$[U_k|_{\mathcal{O}_J}]_i = \left[U_k|_{\mathcal{O}_{J'}}\right]_i . \tag{3.29}$$

If the rotation implemented via the measurement on cluster qubit j is about the x-axis, then the measurement on qubit j has to wait for the measurement on cluster qubit k iff $[U_k|_{\mathcal{O}_J}]_i$ contains a contribution σ_z. The measurement on j itself produces a potential byproduct operator $\left[U_j|_{\mathcal{O}_{J'}}\right]_i = \sigma_x$. Similarly, if the rotation implemented via the measurement on j is about the z-axis then the measurement on j has to wait for the measurement on k iff $[U_k|_{\mathcal{O}_J}]_i$ contains a contribution σ_x. The measurement on j itself produces a potential byproduct operator $\left[U_j|_{\mathcal{O}_{J'}}\right]_i = \sigma_z$.

Because of (3.29) U_k can as well be evaluated at the cut $\mathcal{O}_{J'}$ instead of $\mathcal{O}_J$. The byproduct operator on the intersection of qubit line i and cut $\mathcal{O}_{J'}$ resulting from the measurement on qubit j can be written in the form

$$\left[U_j|_{\mathcal{O}_{J'}}\right]_i = \left(\sigma_x^{(i)}\right)^{x_{j,i}} \left(\sigma_z^{(i)}\right)^{z_{j,i}} \text{ with } \begin{pmatrix} x_{j,i} \\ z_{j,i} \end{pmatrix} = \begin{cases} \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \text{for } z\text{-rotations} \\ \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \text{for } x\text{-rotations} \end{cases} . \tag{3.30}$$

The byproduct operator on the intersection of qubit line i and cut $\mathcal{O}_{J'}$ resulting from the measurement on qubit k reads

$$\left[U_k|_{\mathcal{O}_{J'}}\right]_i = \left(\sigma_x^{(i)}\right)^{x_{k,i}} \left(\sigma_z^{(i)}\right)^{z_{k,i}} . \tag{3.31}$$

One can now easily check for both the cases of an x- and a z-rotation implemented by the measurement on qubit j that the measurement of qubit j must wait for the result of the measurement of qubit k iff

$$x_{j,i} z_{k,i} + z_{j,i} x_{k,i} = 1 \bmod 2. \tag{3.32}$$

Now note that the correspondence between $\begin{pmatrix} x_{j,i} \\ z_{j,i} \end{pmatrix}$ and $\left[U_j|_{\mathcal{O}_{J'}}\right]_i$; and between $\begin{pmatrix} x_{k,i} \\ z_{k,i} \end{pmatrix}$ and $\left[U_k|_{\mathcal{O}_{J'}}\right]_i$ is via the restriction of the isomorphism (3.10) on qubit i. Thus, $x_{j,i}, z_{j,i}$ are just the i-components of $\mathbf{I}_x|_{\mathcal{O}_{J'}}$ and $\mathbf{I}_z|_{\mathcal{O}_{J'}}$, respectively. Equivalent relations hold for $x_{k,i}, z_{k,i}$. One finds

$$\begin{array}{lll} x_{j,i} = \left[I_{x,j}|_{\mathcal{O}_{J'}}\right]_i & , & z_{j,i} = \left[I_{z,j}|_{\mathcal{O}_{J'}}\right]_i \\ x_{k,i} = \left[I_{x,k}|_{\mathcal{O}_{J'}}\right]_i & , & z_{k,i} = \left[I_{z,k}|_{\mathcal{O}_{J'}}\right]_i \end{array} . \tag{3.33}$$

Further we observe that

$$\left[I_{x,j}|_{\mathcal{O}_{J'}}\right]_l = 0, \quad \left[I_{z,j}|_{\mathcal{O}_{J'}}\right]_l = 0 \text{ for all } l \neq i, \tag{3.34}$$

since the byproduct operator introduced by the implementation of the rotation acts, at the cut $\mathcal{O}_{J'}$, non-trivially only on the logical qubit i. Thus we can write

$$\begin{aligned} x_{j,i}z_{k,i} + z_{j,i}x_{k,i} &= \sum_{l=1}^{n} x_{j,l}z_{k,l} + z_{j,l}x_{k,l} \\ &= \left(\mathbf{I}_j|_{\mathcal{O}_{J'}}, \mathbf{I}_k|_{\mathcal{O}_{J'}}\right)_S \\ &= \left(\mathbf{I}_j|_{\Omega}, \mathbf{I}_k|_{\Omega}\right)_S \\ &= \left(\mathbf{F}_j, \mathbf{F}_k\right)_S, \end{aligned} \tag{3.35}$$

where the second line holds by the definition (3.20) and the third by (3.13) and the conservation (3.21) of the symplectic scalar product. Inserting (3.35) into (3.32) yields

$$\mathcal{O}_K \leq \mathcal{O}_J : \quad j \in \mathrm{fc}(k) \Longleftrightarrow \left(\mathbf{F}_j, \mathbf{F}_k\right)_S = 1. \tag{3.36}$$

For $\mathcal{O}_K \leq \mathcal{O}_J$, $j \in \mathrm{bc}(k)$ cannot occur, hence with (3.36),

$$\mathcal{O}_K \leq \mathcal{O}_J : \quad j \in \mathrm{fc}(k) \vee j \in \mathrm{bc}(k) \Longleftrightarrow \left(\mathbf{F}_j, \mathbf{F}_k\right)_S = 1. \tag{3.37}$$

Case II: $\mathcal{O}_J \leq \mathcal{O}_K$.
First we observe that j can only be in the backward cone of k, but not in the forward cone. Thus, the byproduct operator generated via the measurement on k must be propagated backwards in the network to reach the gate for whose implementation qubit j is to be measured. The reasoning is completely analogous to case I, up to the fact that the potential byproduct operator generated via the measurement of cluster qubit k is in this case propagated backwards onto the cut $\mathcal{O}_{J'}$. Qubit j is in the backward cone of qubit k iff the quantity $\left(\mathbf{I}_j|_{\mathcal{O}_{J'}}, \mathbf{I}_k|_{\mathcal{O}_{J'}}\right)_S$ is equal to 1. Again, by conservation (3.21) of the symplectic scalar product follows

$$\mathcal{O}_J \leq \mathcal{O}_K : \quad j \in \mathrm{bc}(k) \Longleftrightarrow \left(\mathbf{F}_j, \mathbf{F}_k\right)_S = 1. \tag{3.38}$$

For $\mathcal{O}_J \leq \mathcal{O}_K$, $j \in \mathrm{fc}(k)$ cannot occur, and therefore with (3.38),

$$\mathcal{O}_J \leq \mathcal{O}_K : \quad j \in \mathrm{fc}(k) \vee j \in \mathrm{bc}(k) \Longleftrightarrow \left(\mathbf{F}_j, \mathbf{F}_k\right)_S = 1. \tag{3.39}$$

Now we combine the two cases and with (3.37) and (3.39) we obtain

$$k \in \mathcal{C}, j \in Q^{(1)} : \quad j \in \mathrm{fc}(k) \ \vee \ j \in \mathrm{bc}(k) \Longleftrightarrow \left(\mathbf{F}_j, \mathbf{F}_k\right)_S = 1,$$

which proves the cone test (3.25).

The proof of the cone test for gates (3.28) goes along the same lines, only the byproduct operator $(U_k)^{s_k}$ generated via the measurement at cluster qubit $k \in \mathcal{C}$ has to be replaced with the byproduct operator $U_{0,g}$ of the gate g.

Finally, the proof the forward-backward cone relation shall be outlined. Suppose that $j \in \mathrm{fc}(k)$. With the same methods as used in the proof of (3.25) one can derive that

$$\begin{array}{rcl} j \in \mathrm{fc}(k) & \Longleftrightarrow & (\mathbf{I}_k|_{\mathcal{O}_J}, \mathbf{I}_j|_{\mathcal{O}_J})_S = 1, \\ k \in \mathrm{bc}(j) & \Longleftrightarrow & (\mathbf{I}_k|_{\mathcal{O}_K}, \mathbf{I}_j|_{\mathcal{O}_K})_S = 1. \end{array} \tag{3.40}$$

Then, with (3.40) and the invariance (3.21) of the symplectic scalar product

$$j \in \mathrm{fc}(k) \iff k \in \mathrm{bc}(j),$$

which proves (3.26).

3.2.5 To what a quantum logic network condenses

Simulating a quantum logic network on a $\mathrm{QC}_{\mathcal{C}}$ is a two-stage process. Before the genuine computation, we feed a classical computer with the network to be simulated. It returns the quantities needed to run the respective algorithm on the $\mathrm{QC}_{\mathcal{C}}$. These quantities are the sets Q_t of simultaneously measurable qubits, the measurement bases of the qubits $k \in Q_0$, the algorithm angles $\varphi_{l,\mathrm{algo}}$ for $l \in \mathcal{C}\backslash Q_0$, the backward cones $\mathrm{bc}(k)$ of the qubits $k \in Q_0$, the byproduct images $\mathbf{F}_j$ for $j \in \mathcal{C}$ and the initialization value $\mathbf{I}_{\mathrm{init}}$ of the information flow vector $\mathbf{I}(t)$. Together these quantities represent the program for the $\mathrm{QC}_{\mathcal{C}}$.

The measurement pattern representing the $\mathrm{QC}_{\mathcal{C}}$-algorithm has both a temporal and a spatial structure. The temporal structure is given by the sets Q_t of simultaneously measured qubits. The spatial structure consists of the bases (σ_x-, σ_y- or σ_z-) of the measurements in the first round and of the measurement angles in the subsequent rounds. The measurement angles can be determined only run-time, since they involve the random outcomes of previous measurements.

3.3 Symmetry considerations

We introduce a group of symmetry transformations that have, by construction, no effect on the computation. Nevertheless, the transformations are such that they act on the measurement outcomes $\{s_k\}$, the parameters $\{\kappa_k\}$ and the measurement angles $\{\varphi_{a,\mathrm{meas}}\}$; and non-trivial conclusions can be derived form them. Among these are the characterization of functions on the random measurement outcomes which lead to computationally meaningful quantities, and constraints upon the possible temporal order of the measurements in a $\mathrm{QC}_{\mathcal{C}}$-computation. The reason why we discuss the symmetry transformations here is that they make the subsequent derivation of the information vector $\mathbf{I}$ (3.9) as a function of the measurement outcomes and the parameters $\{\kappa\}$ specifying the cluster state more compact and transparent.

Recall that, when performing a $\mathrm{QC}_{\mathcal{C}}$-computation, one is only interested in the measurement results but not in the projected quantum state $P^{(\mathcal{C})}|\phi_{\{\kappa\}}\rangle_{\mathcal{C}}$. Therefore, it is of no

relevance for the computation whether we leave the projected state as it is or apply some subsequent unitary transformation U_{post} to it,

$$P^{(\mathcal{C})}|\phi_{\{\kappa\}}\rangle_{\mathcal{C}} \cong U_{\text{post}}\, P^{(\mathcal{C})}|\phi_{\{\kappa\}}\rangle_{\mathcal{C}}. \tag{3.41}$$

Therein, the symbol "$\cong$" shall denote the equivalence of these states with respect to the result of the computation, and $P^{(\mathcal{C})}$ denotes the projection operator representing the measurement sequence of which a $\text{QC}_{\mathcal{C}}$-computation consists,

$$P^{(\mathcal{C})} = \bigotimes_{i\in\mathcal{C}} \frac{\mathbb{1}^{(i)} + (-1)^{s_i}\vec{r}_i\cdot\vec{\sigma}^{(i)}}{2}. \tag{3.42}$$

The measurement outcomes $\{s\}$ are random; only the measurement bases can be chosen. They depend upon previous measurement outcomes $\vec{r}_i = \vec{r}_i(\{s_a\})$.

The unitary transformations U_{post} have no effect solely due to the fact that they are subsequent ones. In principle, one might consider the full group of subsequent unitary transformations on the Hilbert space of the cluster qubits. However, for our argument it is necessary that the group of transformations operates on the following variables: the measurement outcomes $\{s_k|\, k\in\mathcal{C}\}$, the parameters $\{\kappa_k|\, k\in\mathcal{C}\}$ and the parameters $\{\vartheta_a \in \mathbb{F}_2|\, a\in\mathcal{C}\backslash Q_0\}$ which appear in the sign factors for the measurement angles, $\varphi_{a,\text{meas}} = (-1)^{\vartheta_a}\varphi_{a,\text{qln}}$. The largest group with this property that we could identify is the Pauli group,

$$U_{\text{post}} \in \left\{\sigma_z^{(a)}, \sigma_x^{(a)},\ \forall a\in\mathcal{C}, \text{ and products thereof}\right\}. \tag{3.43}$$

Let us now investigate how the subsequent transformations (3.43) act on the variables $\{s\}$, $\{\kappa\}$ and $\{\vartheta\}$. For the $\text{QC}_{\mathcal{C}}$-realization of all the gates and sub-circuits developed so far, not all directions of the Bloch sphere are used for measurements. There occur only σ_z-measurements and measurements in the equator of the Bloch sphere, i.e. of operators $\cos\varphi\,\sigma_x + \sin\varphi\,\sigma_y$. All measurements in $\mathcal{C}_N\backslash O$ are of the latter type and all in $\mathcal{C}\backslash\mathcal{C}_N\cup O$ of the former.

We discuss two examples of the transformations (3.43) explicitly, and subsequently state the table of transformations. First, let us consider the case where a qubit $a\in\mathcal{C}$ is measured in the σ_z-eigenbasis. For the state after the measurement $P^{(\mathcal{C})}|\phi_{\{\kappa\}}\rangle_{\mathcal{C}} = P^{(\mathcal{C}\backslash a)}P^{(a)}|\phi_{\{\kappa\}}\rangle_{\mathcal{C}}$ one can write

$$P^{(\mathcal{C}\backslash a)}\frac{\mathbb{1}^{(a)} + (-1)^{s_a}\sigma_z^{(a)}}{2}|\phi_{\{\kappa\}}\rangle_{\mathcal{C}} \cong \sigma_z^{(a)}P^{(\mathcal{C}\backslash a)}\frac{\mathbb{1}^{(a)} + (-1)^{s_a}\sigma_z^{(a)}}{2}|\phi_{\{\kappa\}}\rangle_{\mathcal{C}}. \tag{3.44}$$

In the r.h.s. of eq. (3.44) the subsequent rotation is now propagated through the projectors such that it acts directly on the state $|\phi_{\{\kappa\}}\rangle_{\mathcal{C}}$, i.e.

$$\sigma_z^{(a)}P^{(\mathcal{C}\backslash a)}\frac{\mathbb{1}^{(a)} + (-1)^{s_a}\sigma_z^{(a)}}{2}|\phi_{\{\kappa\}}\rangle_{\mathcal{C}} = P^{(\mathcal{C}\backslash a)}\frac{\mathbb{1}^{(a)} + (-1)^{s_a}\sigma_z^{(a)}}{2}|\phi_{\{\tilde{\kappa}\}}\rangle_{\mathcal{C}}, \tag{3.45}$$

with $|\phi_{\{\tilde{\kappa}\}}\rangle_{\mathcal{C}} = \sigma_z^{(a)}|\phi_{\{\kappa\}}\rangle_{\mathcal{C}}$, and thus $\tilde{\kappa}_b = \kappa_b + \delta_{ab}$. The measurement bases and outcomes of the $\text{QC}_{\mathcal{C}}$-computation (3.42) remain unchanged, that is $\tilde{\vec{r}}_i = \vec{r}_i$ and $\tilde{s}_i = s_i$ for all

$i \in \mathcal{C}$. Applying the subsequent phase flip $\sigma_z^{(a)}$ causes no change for the computation, but transform $\kappa_a \to \kappa_a + 1$, according to (2.14). Thus we find that for qubits a which are measured in the σ_z-eigenbasis the parameter κ_a cannot enter into computationally relevant quantities. However, the measurement outcome s_a may, since it is not affected by the transformation (3.44).

Second, consider a subsequent σ_x acting on a cluster qubit measured in the equator of the Bloch sphere where the measurement basis is not the eigenbasis of σ_x or σ_y. Then, the measurement direction $\vec{r}_a$ is affected. One obtains

$$\sigma_x^{(a)} P^{(\mathcal{C}_N \backslash a)} \frac{\mathbb{1}^{(a)} + (-1)^{s_a} \vec{r}_a \cdot \vec{\sigma}^{(a)}}{2} |\phi_{\{\kappa'\}}\rangle_{\mathcal{C}_N} = P^{(\mathcal{C}_N \backslash a)} \frac{\mathbb{1}^{(a)} + (-1)^{s_a} \tilde{\vec{r}}_a \cdot \vec{\sigma}^{(a)}}{2} |\phi_{\{\tilde{\kappa}\}}\rangle_{\mathcal{C}_N}, \quad (3.46)$$

with $|\phi_{\{\tilde{\kappa}\}}\rangle_{\mathcal{C}_N} = \bigotimes_{b \in \mathrm{nbgh}(a)} \sigma_z^{(b)} |\phi_{\{\kappa\}}\rangle_{\mathcal{C}_N}$ and $\tilde{\vec{r}}_a(\varphi_a) = (\cos(\varphi), -\sin(\varphi), 0) = \vec{r}_a(-\varphi_a)$. For the measurement basis of qubit a is given by $\mathcal{B}\left((-1)^{\vartheta_a}\varphi_a\right)$, the transformation thus is $\vartheta_a \longrightarrow \vartheta_a + 1 \bmod 2$, $\kappa_b \longrightarrow \kappa_b + 1 \bmod 2$, $\forall b \in \mathrm{nbgh}(a)$.

Repeating the above arguments for the remaining cases we finally obtain the following set of symmetry transformations:

- Transformations caused by $U_{\mathrm{post}} = \sigma_z^{(a)}$ in (3.41):

$$\forall a \in \mathcal{C}\backslash\mathcal{C}_N \cup O: \quad \kappa_a \longrightarrow \kappa_a + 1 \bmod 2, \quad (3.47a)$$

$$\forall a \in \mathcal{C}_N\backslash O: \quad \begin{cases} s_a \longrightarrow s_a + 1 \bmod 2, \\ \kappa_a \longrightarrow \kappa_a + 1 \bmod 2, \end{cases} \quad (3.47b)$$

- Transformations caused by $U_{\mathrm{post}} = \sigma_x^{(a)}$:

$$\forall a \in \mathcal{C}\backslash\mathcal{C}_N \cup O: \quad Z: \begin{cases} s_a \longrightarrow s_a + 1 \bmod 2, \\ \kappa_b \longrightarrow \kappa_b + 1 \bmod 2, \end{cases} \quad \forall b \in \mathrm{nbgh}(a)\ , \quad (3.48a)$$

$$\forall a \in \mathcal{C}_N\backslash O: \quad X: \quad \kappa_b \longrightarrow \kappa_b + 1 \bmod 2, \quad \forall b \in \mathrm{nbgh}(a)\ , \quad (3.48b)$$

$$Y: \begin{cases} s_a \longrightarrow s_a + 1 \bmod 2, \\ \kappa_b \longrightarrow \kappa_b + 1 \bmod 2, \end{cases} \quad \forall b \in \mathrm{nbgh}(a)\ , \quad (3.48c)$$

$$\nearrow: \begin{cases} \vartheta_a \longrightarrow \vartheta_a + 1 \bmod 2, \\ \kappa_b \longrightarrow \kappa_b + 1 \bmod 2, \end{cases} \quad \forall b \in \mathrm{nbgh}(a)\ . \quad (3.48d)$$

Therein, "X", "Y", "Z" denote σ_x-, σ_y- and σ_z-measurement of qubit a, respectively. The symbol "$\nearrow$" denotes a measurement of an operator in the equator of the Bloch sphere which is neither of σ_x nor σ_y.

We now introduce the notion of *computationally meaningful quantities*. So far, we have some idea of what the computational meaningful quantities should be from the phenomenological viewpoint. The measurements which drive a $\mathrm{QC}_{\mathcal{C}}$-computation produce a pile of random bits. From this random bit stream information can be extracted which is required to steer the $\mathrm{QC}_{\mathcal{C}}$-computation and to identify its result. Such information is

certainly meaningful in the colloquial sense of the word. Among the computational meaningful quantities there is, of course, the final result of the quantum computation. By this we mean the bits of information that are obtained from the read-out of the quantum register after, if required, classical post-processing[1]. These bits should, if the algorithm is deterministic, not be random. So, is it the non-randomness in value that should be the criterion for being computationally meaningful? No. If "computationally meaningful" is a proper notion, the measurement angles (2.26), or more precisely, the sign factor parameters ϑ_a, $\forall a \in \mathcal{C}_N \backslash O$ required to adjust the measurement angles $\varphi_{a,\mathrm{meas}} = (-1)^{\vartheta_a} \varphi_{a,\mathrm{qln}}$ should be such quantities. The parameters ϑ_a can be determined only runtime from previously obtained measurement outcomes and they are completely random in their value. Nevertheless, they are meaningful. If in the adjustment of the measurement basis for the measurement of some qubit a no attention is payed to the value of ϑ_a, the $\mathrm{QC}_{\mathcal{C}}$-computation gets immediately off track. Therefore, non-randomness in value cannot be the appropriate criterion. Below we will present a necessary condition for computational meaningfulness, requiring that the defining relations of computational meaningful quantities are consistent with the group of symmetry transformations (3.43).

Let us, at this point, emphasize that here we are concerned only with the randomness introduced by the measurements which drive the $\mathrm{QC}_{\mathcal{C}}$-computation. This is randomness that can be compensated for. It is distinct from randomness inherent in probabilistic quantum algorithms.

A computationally meaningful quantity has a defining relation. This defining relation expresses the respective quantity as a function of the measurement outcomes $\{s_k | \, k \in \mathcal{C}\}$ and the parameters $\{\kappa_k | \, k \in \mathcal{C}\}$ which characterize the cluster state in (2.1). The readout bits $[\mathbf{I}]_m$, $1 \leq m \leq n$ and the parameters ϑ_a, $a \in \mathcal{C}_N \backslash O$ have defining relations, which read

$$[\mathbf{I}_x]_m = f^m_{\mathbf{I}}(\{s_k | \, k \in \mathcal{C}\}, \{\kappa_k | \, k \in \mathcal{C}\}), \tag{3.49}$$

and

$$\vartheta_a = f^a_{\vartheta}(\{s_k | \, k \in \mathcal{C}\}, \{\kappa_k | \, k \in \mathcal{C}\}). \tag{3.50}$$

Now, we can state a necessary condition for a quantity to be computationally meaningful:

> *A quantity is computationally meaningful only if its defining relation is invariant under the group of transformations* (3.47, 3.48). (3.51)

If a transformation that has by construction no effect on the computation changes a defining relation then the belonging quantity can have no meaning.

Note that we have given a criterion for 'computational meaningfulness' before defining it in a precise way. However, the above criterion must hold for any reasonable definition. The reason why we refrain from giving a definition at this point is the following: We

[1] An example of where classical post-processing is required is Shor's factoring algorithm. Even in case of success, the bits read off from the measurement of the quantum register are random, i.e. many computationally equivalent sets of readout measurement outcomes are possible. However, the final result of the quantum computation is a smaller bit string which is obtained after classical post-processing of the readout measurement outcomes.

have not proved that the group of transformations generated by (3.47, 3.48) is the largest that is induced by the transformations (3.41). A sensible definition of the notion of a 'computational meaningful quantity' is by invariance of the respective defining relation under the largest group of transformations of type (3.47, 3.48) induced by (3.41).

To summarize the first part of this investigation, we have characterized the computationally meaningful quantities as quantities whose defining relations remain invariant under the transformations (3.47) and (3.48). This holds in particular for the defining relations (3.49) and (3.50) of the components of the information vector $\mathbf{I}$ and the parameters ϑ_a for the choice of the non-Pauli measurement bases.

Let us now investigate which conclusions can be drawn from the invariance under these transformations. We proceed in two steps. First, we consider the subgroup (3.47), (3.48a) induced by subsequent unitary transformations $\sigma_z^{(c)},\ c \in \mathcal{C}$ and $\sigma_x^{(b)},\ b \in \mathcal{C}\backslash\mathcal{C}_N \cup O$. As the $[\mathbf{I}]_k$ remain invariant under these transformations it must be possible to express them as functions of a set of variables which are invariant themselves. As can be easily verified, for each qubit $a \in \mathcal{C}_N\backslash O$ there exists a variable $\overline{s}_a$,

$$\overline{s}_a = \kappa_a + s_a + \sum_{b\in \mathrm{nbgh}(a)\cap((\mathcal{C}\backslash\mathcal{C}_N)\cup O)} s_b \mod 2, \tag{3.52}$$

which is invariant under the transformations (3.47), (3.48a). These variables are the only ones that are invariant under the considered transformations, which can be seen as follows. A priori, there are four types of variables involved, $\{\kappa_a, a \in \mathcal{C}\backslash\mathcal{C}_N \cup O\}$, $\{\kappa_a, a \in \mathcal{C}_N\backslash O\}$, $\{s_a, a \in \mathcal{C}\backslash\mathcal{C}_N \cup O\}$, $\{s_a, a \in \mathcal{C}_N\backslash O\}$. First, as follows from (3.47a), the parameters $\{\kappa_a, a \in \mathcal{C}\backslash\mathcal{C}_N \cup O\}$ do not appear in the invariant variables. Then it follows from (3.48b) that among the $\{s_a, a \in \mathcal{C}\backslash\mathcal{C}_N \cup O\}$ those s_a can be discarded for which $\forall b \in \mathrm{nbgh}(a)$: $b \in \mathcal{C}\backslash\mathcal{C}_N \cup O$. Now, suppose an invariant variable contains one measurement outcome s_a, $a \in \mathcal{C}_N\backslash O$. Then, via (3.47b) it can depend only on $s_a + \kappa_a$. But if it contains κ_a then because of invariance under the transformation (3.48a) it is of form (3.52). Next, suppose the invariant variable contains no s_a, $a \in \mathcal{C}_N\backslash O$. Then, via (3.47b) it cannot depend on either of the κ_a. Then, because of invariance under the transformation (3.48a) it cannot depend on the s_b, $b \in \mathcal{C}\backslash\mathcal{C}_N \cup O\}\cap\mathrm{nbgh}(a)$. There are no further quantities left upon which the invariant variable could depend. Thus, all invariant variables contain at least one s_a, $a \in \mathcal{C}_N\backslash O$; equation (3.52) describes a complete set of independent invariant variables.

The parameter ϑ_a for the choice of measurement bases also can only depend upon the effective variables (3.52). To see this, first note that if ϑ_a did depend upon s_a it would render the $\mathrm{QC}_{\mathcal{C}}$-computation probabilistic. We have not observed such a case in the so far given circuit constructions. As we have demonstrated universality of the $\mathrm{QC}_{\mathcal{C}}$, a situation where some ϑ_a depends upon the outcome s_a of the measurement of the same qubit a can always be avoided and we thus discard it. So, if ϑ_a does not depend on s_a, it cannot depend upon κ_a either, as implied by (3.47b). Thus ϑ_a depends only on the variables $\{s_b, \kappa_b \,|\, b \in \mathcal{C}\backslash a\}$. As both sides of the defining relation (3.50) do not change under Pauli transformations (3.41) restricted to qubits $b \in \mathcal{C}\backslash a$, by the same arguments as above for $[\mathbf{I}]_k$, ϑ_a can only depend on the invariant variables $\{\overline{s}_b \,|b \in \mathcal{C}_N\backslash(O \cup a)\}$.

To work with the effective variables $\{\overline{s}_a |\, a \in \mathcal{C}_N \backslash O\}$ instead of $\{s_b, \kappa_b |\, b \in \mathcal{C}\}$ is convenient because the computationally meaningful quantities depend only upon them. We will make use of the effective variables (3.52) in Section 3.4.1 when deriving the defining relation for the information vector $\mathbf{I}$. The symmetry transformations (3.47), (3.48a) act trivially on the variables (3.52). Thus, by choosing invariant effective variables $\{\overline{s}\}$ we have absorbed these symmetry transformations.

Second, let us consider the remaining symmetry transformations (3.48b) - (3.48d) induced by the subsequent transformations $\sigma_x^{(a)}$, $a \in \mathcal{C}_N \backslash O$ and products thereof. An equivalent and convenient choice for the remaining subsequent transformations (3.43) are the basic correlation operators $K^{(a)}$, $a \in \mathcal{C}_N \backslash O$, defined in (2.2),

$$U[J] = \prod_{a \in J \subset \mathcal{C}_N \backslash O} K^{(a)}. \tag{3.53}$$

The transformations (3.53) describe the effect of randomness introduced by the measurements in the following sense: They act only on the measurement outcomes $\{s_k \,|\, k \in \mathcal{C}\}$ and on the measurement angles φ_a, represented by the binary variables $\{\vartheta_a \,| a \in \mathcal{C} \backslash Q_0\}$, but they do not act on the set of parameters $\{\kappa_k \,|\, k \in \mathcal{C}\}$. For a set of fixed parameters $\{\kappa_k \,|\, k \in \mathcal{C}\}$, one may define equivalence classes of sets $\{s_k \,|\, k \in \mathcal{C}\}$ of measurement outcomes obtained in a $\mathrm{QC}_{\mathcal{C}}$-computation. Two such sets $\{s\}$, $\{s'\}$ are equivalent if they yield the same computational result $\mathbf{I}_x$. In doing so, one separates the randomness introduced by the measurements from the randomness that may be inherent in the quantum algorithm. While the former leads to the fact that different sets $\{s\}$ may be obtained in different runs of the $\mathrm{QC}_{\mathcal{C}}$-algorithm that yield the same computational result, the latter has the effect that there may exist numerous equivalence classes of sets $\{s\}$ with nonzero total probability. The transformations (3.53) take one around within these equivalence classes of sets of measurement outcomes, and in this way mimic the effect of measurement-induced randomness. The invariance condition (3.51) ensures that the randomness caused by the measurements does not affect the logical processing.

Besides characterizing the randomness of measurement outcomes, the symmetry transformations (3.53) may be used to derive severe constraints upon the possible temporal order of measurements in a $\mathrm{QC}_{\mathcal{C}}$-computation where all randomness of the measurement outcomes is accounted for. To see this in a particular example, we consider a five-qubit chain on a cluster $\mathcal{C}_5 \subset \mathcal{C}$ to implement a general one-qubit rotation. Qubits 2, 3 and 4 are measured in a non-standard basis to adjust the Euler angles of the rotation. Qubits 1 and 5 are the input- and output qubit; however, it is not specified which is which. The transformation $K^{(3)}$ of (3.41) has the effect

$$K^{(3)}: \quad \begin{array}{rcl} \vartheta_3 & \longrightarrow & \vartheta_3 + 1 \bmod 2, \\ \overline{s}_2 & \longrightarrow & \overline{s}_2 + 1 \bmod 2, \\ \overline{s}_4 & \longrightarrow & \overline{s}_4 + 1 \bmod 2. \end{array} \tag{3.54}$$

Consider the accumulated byproduct operator $U_{\Sigma,\mathcal{C}'}|_{\mathcal{O}}$, $\mathcal{C}' \subset \mathcal{C}$, at a vertical cut $\mathcal{O}$ after qubits 2 and 4, i.e. the byproduct operator including the contributions from both these

cluster qubits. Depending on what the forward direction is, an intuitive choice (from the network perspective) for $\mathcal{O}$ would be after qubit 1 or before 5. Anyway, the quantities we are going to consider are invariant under the displacement of $\mathcal{O}$. Under the transformation (3.54), the byproduct operator undergoes the change

$$U_{\Sigma,\mathcal{C}'}|_{\mathcal{O}} \longrightarrow U_{\Sigma,\mathcal{C}'}|_{\mathcal{O}}\, U_2|_{\mathcal{O}} U_4|_{\mathcal{O}}, \tag{3.55}$$

where $U_2|_{\mathcal{O}}$, $U_4|_{\mathcal{O}}$ are such that $U_{\Sigma,2}|_{\mathcal{O}} = (U_2|_{\mathcal{O}})^{s_2}$ and $U_{\Sigma,4}|_{\mathcal{O}} = (U_4|_{\mathcal{O}})^{s_4}$. Under the transformation (3.54) no measurement angle except $\varphi_{3,\mathrm{meas}}$ is affected, whatever the operations later than $\mathcal{O}$ in the simulated network maybe. This is only possible if

$$U_2|_{\mathcal{O}} U_4|_{\mathcal{O}} = \mathbb{1}. \tag{3.56}$$

Hence, $U_2|_{\Omega} U_4|_{\Omega} = \mathbb{1}$. Using (3.10), one obtains $\mathbf{F}_2 + \mathbf{F}_4 = \vec{0} \bmod 2$, and therefore

$$(\mathbf{F}_2, \mathbf{F}_3)_S = (\mathbf{F}_4, \mathbf{F}_3)_S. \tag{3.57}$$

From (3.54) also follows that the defining relation of ϑ_3 depends either on $\overline{s}_2$ or $\overline{s}_4$ but not on both. There exist only two choices

$$\vartheta_3 = \overline{s}_2 + f_\vartheta^3(\{\overline{s}_k \,|\, k \in \mathcal{C}\backslash\{2,3,4\}\}), \tag{3.58a}$$
$$\vartheta_3 = \overline{s}_4 + f_\vartheta^3(\{\overline{s}_k \,|\, k \in \mathcal{C}\backslash\{2,3,4\}\}). \tag{3.58b}$$

We now discuss these two choices separately.

Case 1: (3.58a) is valid. From (3.58a) follows that, by definition 2 of the forward cone, $3 \in \mathrm{fc}(2)$. From this follows, by construction of the relation "$\prec$", $2 \prec 3$. Also, via (3.25), there follows $(\mathbf{F}_2, \mathbf{F}_3)_S = 1$. Hence, with (3.57), $(\mathbf{F}_4, \mathbf{F}_3)_S = 1$. And, using (3.27) (backwards), $3 \in \mathrm{fc}(4) \ \vee\, 4 \in \mathrm{fc}(3)$. $3 \in \mathrm{fc}(4)$ is excluded by the case assumption (3.58a), thus $4 \in \mathrm{fc}(3)$ and therefore $3 \prec 4$. Putting both pieces together, we obtain $2 \prec 3 \prec 4$.

Case 2: (3.58b) is valid. The argument is the same as in case 1, only the roles of qubit 2 and 4 are interchanged. The result is $4 \prec 3 \prec 2$.

Therefore, out of the six possibilities that there exist in principle for the temporal order of the measurements on the qubits 2, 3 and 4 only two can yield to deterministic computation, namely 2 - 3 - 4 and 4 - 3 - 2. This result is in accordance with the Procedure 2 to realize an arbitrary rotation. The remaining ambiguity of the temporal order in the measurements is caused by the fact that we have not introduced the forward direction of the network logical time, i.e. have not specified what is "Input" and what "Output".

So, we have not produced a new result. Rather, we have derived a known result from basic principles. Objects like the temporal semi ordering "$\prec$" of the measurements we would usually derive from an underlying quantum logic network, claiming at the same time that this network is no part of the description of the respective $\mathrm{QC}_{\mathcal{C}}$-algorithm. This is no contradiction since the network description is completely absorbed into the quantities required for the processing of information with the $\mathrm{QC}_{\mathcal{C}}$, see Section 3.2.5. However, one may easily get the impression that, although the quantum logic network

finally disappears from the description of a $\mathrm{QC}_{\mathcal{C}}$-algorithm, no such description could be obtained without a quantum logic network. Here we have given a counterexample to this objection. We have derived the temporal order from an invariance property under the transformations (3.53) linked to the effect of randomness in the measurement outcomes. For the discussed example, we have asked and answered the following question: "Requiring that the randomness introduced by the measurements does not lead to randomness in the logical processing, what can the temporal order of measurements be?"

3.4 Computational model for the $\mathrm{QC}_{\mathcal{C}}$

In the preceding sections we have established the notions of the sets of simultaneously measurable qubits, backward cones, byproduct images, measurement angles and the information vector. In this section, the computational model underlying the $\mathrm{QC}_{\mathcal{C}}$ is described in these terms. First, we would like to give a summary of the characteristic features of the model:

- The $\mathrm{QC}_{\mathcal{C}}$ has no quantum input and no quantum output.
- For any given quantum algorithm, the cluster $\mathcal{C}$ is divided into disjoint subsets $Q_t \subset \mathcal{C}$ of qubits, $t = 0, 1, \ldots, t_{\max}$, where $Q_p \cap Q_q = \emptyset$ for $p \neq q$ and $\bigcup_{t=0}^{t_{\max}} Q_t = \mathcal{C}$. These subsets are measured one after the other in the order given by the index t. In measurement round t the set Q_t of qubits is measured.
- The classical information gained by the measurements is processed within a flow scheme. The flow quantity is a classical $2n$-component binary vector $\mathbf{I}(t)$, where n is the number of logical qubits of a corresponding quantum logic network and t the number of the measurement round.
- This vector $\mathbf{I}(t)$, the *information flow vector*, is updated after every measurement round. That is, after the one-qubit measurements of all qubits of a set Q_t have been performed simultaneously, $\mathbf{I}(t-1)$ is updated to $\mathbf{I}(t)$ through the results of these measurements. In turn, $\mathbf{I}(t)$ determines which one-qubit observables are to be measured of the qubits of the set Q_{t+1}.
- The result of the computation is given by the information flow vector $\mathbf{I}(t_{\max})$ after the last measurement round. From this quantity the readout measurement result of the quantum register in the corresponding quantum logic network can be read off directly without further processing.

We should make a comment on the first point. The $\mathrm{QC}_{\mathcal{C}}$ has no quantum output. Of course, the final result of any computation –including quantum computations– is a classical number, but for the quantum logic network the state of the output register before the readout measurements plays a distinguished role. For the $\mathrm{QC}_{\mathcal{C}}$ this is not the case, there are just cluster qubits measured in a certain order and basis. If, to perform a particular

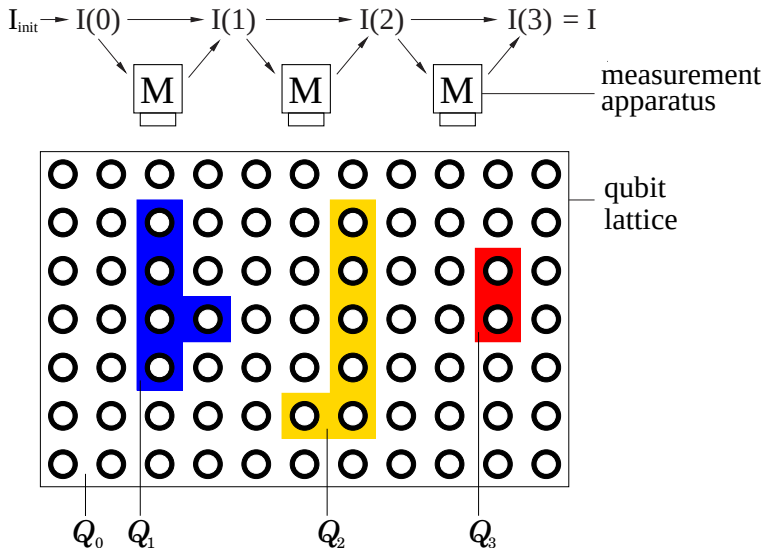


Figure 3.2: General scheme of the quantum computer via one-qubit measurements. The sets Q_t of lattice qubits are measured one after the other. The results of earlier measurements determine the measurement bases of later ones. All classical information from the measurement results needed to steer the $\mathrm{QC}_{\mathcal{C}}$ is contained in the information flow vector $\mathbf{I}(t)$. After the last measurement round $t_{\max}$, $\mathbf{I}(t_{\max})$ contains the result of the computation.

algorithm on the $\mathrm{QC}_{\mathcal{C}}$, a quantum logic network is implemented on a cluster state there is a subset of cluster qubits which play the role of the output register. These qubits are, however, not the final qubits to be measured, but among the first (!).

The $\mathrm{QC}_{\mathcal{C}}$ has no quantum input. This means that the quantum input state is *known* and can thus be created from some standard quantum state, e.g. $|00...0\rangle$, by a circuit preceding the main part of the computation. Shor's algorithm where one starts with an input state $\bigotimes_{i=1}^{n} 1/\sqrt{2}(|0\rangle_i + |1\rangle_i)$ is an example for such a situation. Other scenarios are conceivable, e.g. where an unknown quantum input is processed and the classical result of the computation is retransmitted to the sender of the input state; or the unmeasured network output register state is retransmitted. These scenarios would lead only to slight modifications in the computational model. They are, however, not in the focus of this thesis. The reader who is interested in how to read in and process an unknown quantum state with the $\mathrm{QC}_{\mathcal{C}}$ is referred to [9].

3.4.1 Obtaining the computational result from the measurement outcomes

Now that we have defined the information vector $\mathbf{I}$ in (3.9) and have seen that the result of the computation can be directly read off from the x-part of $\mathbf{I}$, we will explain how $\mathbf{I}$ depends on the measurement outcomes $\{s_k\}$ and the set $\{\kappa_k\}$ of binary numbers that determine the cluster state $|\phi\rangle_{\mathcal{C}}$ in (2.1). For this purpose, we will express $U_{\Sigma R}$ in terms of $\{s_k\}$, and

use the isomorphism (3.10) to obtain $\mathbf{I}$. We will proceed in three steps. First, we focus on the special case of a cluster state $|\phi\rangle_{\mathcal{C}_N}$ (2.1) on a cluster $\mathcal{C}_N$ where $\{\kappa_a = 0 \,|\, \forall a \in \mathcal{C}\}$. For this situation, we derive an expression for $U_{\Sigma R}$, the quantity from wich we can directly read off the result of the computation. This is a somewhat unnatural expression for we have excluded the irrelevant cluster qubits $q \in \mathcal{C}\backslash\mathcal{C}_N$ which are measured in the eigenbasis of σ_z, but have included the qubits of the output register O which are also measured in the σ_z-eigenbasis. This unequal treatment is a remnant of the network model. There, the output qubits are, of course, very important. For the $\mathrm{QC}_{\mathcal{C}}$, however, as we have already seen in Section 2.2.3, the "readout"- qubits $q \in O$ are as redundant as any other cluster qubits measured in the σ_z-eigenbasis. To account for this, in the second step we derive an expression for $U_{\Sigma R}$ for a $\mathrm{QC}_{\mathcal{C}}$-computation on a cluster $\mathcal{C}_N\backslash O$. After the influence of all the redundant qubits has thus been eliminated, we reintroduce it in the third step. Using a symmetry argument, we derive the expression for $U_{\Sigma R}$ for the general case, i.e. for the cluster state on the universal cluster $\mathcal{C}$ and for arbitrary parameters $\{\kappa_a \,|\, a \in \mathcal{C}\}$. From this expression for $U_{\Sigma R}$ we deduce the information vector $\mathbf{I}$ via the isomorphism (3.10).

To derive $\mathbf{I}$ as a function of $\{s_k\}$ and $\{\kappa_k\}$, we need to define the following sets. $\mathcal{C}$ is a universal cluster. Let $O \subset \mathcal{C}$ be the subset of the cluster which, in the simulation of a quantum logic network on the $\mathrm{QC}_{\mathcal{C}}$, consists of the readout qubits. Let $\mathcal{C}_N \subset \mathcal{C}$ denote the cluster that contains only the relevant cluster qubits, i.e. those which are measured in a direction in the equator of the Bloch sphere, and the "readout" qubits. Be $Q_{0,z} \subset \mathcal{C}$ the set of qubits of which the operator σ_z is measured. Among these sets, the following relations hold:

$$\begin{array}{rcl} \mathcal{C}_N \cup Q_{0,z} & = & \mathcal{C} \\ \mathcal{C}_N \cap Q_{0,z} & = & O. \end{array} \tag{3.59}$$

We can now start to express $\mathbf{I}$ in terms of $\{s_k, k \in \mathcal{C}\}$ and $\{\kappa_k, k \in \mathcal{C}\}$. Let us –in the first step– discuss the accumulated byproduct operator U_Σ for a computation on the special cluster $\mathcal{C}_N$. To U_Σ contribute all the byproduct operators $U_{\Sigma,g}$ that are created in the implementation of the gates g. For all necessary cases, the general rotations (2.29), the CNOT gate (2.23) and the special rotations Hadamard gate and $\pi/2$-phase gate (2.30), the byproduct operators $U_{\Sigma,g}$ can be written in the form

$$U_{\Sigma,g} = \left(\prod_{k \in \mathcal{C}_I(g) \cup \mathcal{C}_M(g)} (U_k)^{s_k} \right) U_{0,g}. \tag{3.60}$$

$U_{0,g}$ is constant in the measurement outcomes $\{s_k, k \in \mathcal{C}_N\backslash O\}$ and we therefore attribute it to the gate g as a whole rather than to a particular cluster qubit. For all rotations we have $U_{0,g} = \mathbf{1}$, but for the CNOT gate –if realized as depicted in Fig. 2.2– the contribution is nontrivial as can be read off from (2.23), $U_0(CNOT) = \sigma_z^{(c)}$.

To determine the effect of $U_{\Sigma,g}$ on U_Σ we propagate, by use of the propagation relations (2.52), (2.51) and (2.53), the byproduct operators $U_{\Sigma,g}$ forward to the cut Ω which intersects the corresponding network $\mathcal{N}$ just before the output. The forward propagated byproduct operator that results from the byproduct operator $U_{\Sigma,g}$ we denote by $U_{\Sigma,g}|_\Omega$.

In the same way, the forward propagated byproduct operator originating from U_k, the byproduct operator generated via the measurement of qubit k, is denoted by $U_k|_\Omega$ for all $k \in \mathcal{C}_N \backslash O$. Finally, the forward propagated byproduct operator originating from $U_{0,g}$, the byproduct operator attributed to the gate g as a whole, is denoted by $U_{0,g}|_\Omega$. To give an explicit expression, be $\mathcal{O}$ the vertical cut through a network $\mathcal{N}$ at the output of a gate g and $U(\mathcal{N}_{\mathcal{O}\to\Omega})$ the unitary operation in the Clifford group which corresponds to the part of the network $\mathcal{N}$ with all the one-qubit rotations except for the Hadamard- and $\pi/2$-phase gates replaced by the identity, as explained in Section 3.2.4. Then, the forward propagated byproduct operators are given by

$$\begin{array}{rcl} U_{\Sigma,g}|_\Omega & = & U(\mathcal{N}_{\mathcal{O}\to\Omega})\, U_{\Sigma,g}\, U(\mathcal{N}_{\mathcal{O}\to\Omega})^\dagger \\ U_k|_\Omega & = & U(\mathcal{N}_{\mathcal{O}\to\Omega})\, U_k\, U(\mathcal{N}_{\mathcal{O}\to\Omega})^\dagger \\ U_{0,g}|_\Omega & = & U(\mathcal{N}_{\mathcal{O}\to\Omega})\, U_{0,g}\, U(\mathcal{N}_{\mathcal{O}\to\Omega})^\dagger \end{array} \tag{3.61}$$

The contribution $U_{\Sigma,g}|_\Omega$ from the gate g to U_Σ is

$$U_{\Sigma,g}|_\Omega = \left(\prod_{k\in\mathcal{C}_I(g)\cup\mathcal{C}_M(g)} U_k|_\Omega{}^{s_k} \right) U_{0,g}|_\Omega. \tag{3.62}$$

The total byproduct operator U_Σ is the product of all forward propagated byproduct operators $U_{\Sigma,g}|_\Omega$, $U_\Sigma = \prod_{g\in\mathcal{N}} U_{\Sigma,g}|_\Omega$, and thus given by

$$\begin{array}{rcl} U_\Sigma & = & \displaystyle\prod_{g\in\mathcal{N}} \left(U_{0,g}|_\Omega \prod_{k\in\mathcal{C}_I(g)\cup\mathcal{C}_M(g)} U_k|_\Omega{}^{s_k} \right) \\ & = & \displaystyle\left(\prod_{g\in\mathcal{N}} U_{0,g}|_\Omega \right) \left(\prod_{k\in\mathcal{C}_N\backslash O} U_k|_\Omega{}^{s_k} \right). \end{array} \tag{3.63}$$

In the second line of (3.63) we have used the facts that $\bigcup_{g\in\mathcal{N}} \mathcal{C}_I(g) \cup \mathcal{C}_M(g) = \mathcal{C}_N \backslash O$ and $\mathcal{C}_I(g) \cap \mathcal{C}_I(g') = \mathcal{C}_M(g) \cap \mathcal{C}_M(g') = \emptyset\ \forall g, g' \neq g \in \mathcal{N}$, $\mathcal{C}_I(g) \cap \mathcal{C}_M(g') = \emptyset\ \forall g, g' \in \mathcal{N}$.

We now include U_R as given in (3.2) to find for $U_{\Sigma R}$, using (3.1) and (3.63),

$$U_{\Sigma R} = \left(\prod_{g\in\mathcal{N}} U_{0,g}|_\Omega \right) \left(\prod_{k\in\mathcal{C}_N} U_k|_\Omega{}^{s_k} \right). \tag{3.64}$$

This is the result of the first step.

In the second step, we remove the influence of the readout qubits $q \in O$ from the expression (3.64). As initially stated, there is no reason to treat the redundant qubits which form the network output register O differently from the other redundant qubits. The qubits in the readout register O are measured in the σ_z-eigenbasis. Such measurements are non-adaptive and can therefore be performed in the first measurement round. Let us now artificially split the first measurement round into two sub-rounds, such that first the

σ_z-measurements of the qubits $k \in O$ and second the σ_x- and σ_y-measurements of the qubits $k \in \mathcal{C}_N \backslash O$ are performed.

The outcomes of the measurements of the qubits $k \in O$, obtained in the first sub-round of the first measurement round are individually random and uncorrelated. It is therefore possible that all these measurement outcomes are zero,

$$s_k = 0, \quad \forall k \in O. \tag{3.65}$$

For the special case of the measurement results (3.65) the extended byproduct operator (3.64) reduces to

$$U_{\Sigma R} = \left(\prod_{g \in \mathcal{N}} U_{0,g} |_{\Omega} \right) \left(\prod_{k \in \mathcal{C}_N \backslash O} U_k |_{\Omega}{}^{s_k} \right). \tag{3.66}$$

Further, the measurements on the qubits of the set O project the unmeasured qubits $k \in \mathcal{C}_N \backslash O$ into a cluster state $|\phi_{\{\kappa\}}\rangle_{\mathcal{C}_N \backslash O}$ (2.1). With the measurement results (3.65) obtained, the parameters κ are all zero, $\kappa_k = 0$, $\forall k \in \mathcal{C}_N \backslash O$. Now, instead of creating the state $|\phi_{\{\kappa\}}\rangle_{\mathcal{C}_N \backslash O}$ with $\kappa_k = 0$, $\forall k \in \mathcal{C}_N \backslash O$ via measurements on O, one may equivalently start the $\mathrm{QC}_{\mathcal{C}}$-computation with this state right away. How the state $|\phi\rangle_{\mathcal{C}_N \backslash O}$ is created is of no concern for the computation, such that the expression (3.66) for $U_{\Sigma R}$ still applies. In this way, we have derived an expression for the extended byproduct operator for the case where no redundant qubits are present, i.e. where we use the cluster $\mathcal{C}_N \backslash O$ for computation,

$$U_{\Sigma R, \mathcal{C}_N \backslash O} = \left(\prod_{g \in \mathcal{N}} U_{0,g} |_{\Omega} \right) \left(\prod_{k \in \mathcal{C}_N \backslash O} U_k |_{\Omega}{}^{s_k} \right). \tag{3.67}$$

We have added the label $\mathcal{C}_N \backslash O$ in $U_{\Sigma R, \mathcal{C}_N \backslash O}$ to stress that this expression holds only for a computation on $\mathcal{C}_N \backslash O$. With (3.67) step two is completed.

After we have removed the influence of the subset O of the redundant qubits, in the third step we bring the influence of all the redundant qubits in $\mathcal{C}$ back in. Also, we include the effect of non-vanishing parameters κ which have so far been set equal to zero.

To do so, let us now make use of the symmetry transformations (3.47) for finding the general expression for the information vector $\mathbf{I}$. Via the isomorphism (3.10) we find for the information vector $\mathbf{I}_{\mathcal{C}_N \backslash O}$ corresponding to $U_{\Sigma R, \mathcal{C}_N \backslash O}$ (3.67),

$$\mathbf{I}_{\mathcal{C}_N \backslash O} = \sum_{k \in \mathcal{C}_N \backslash O} s_k \mathbf{F}_k + \sum_{g \in \mathcal{N}} \mathbf{F}_g. \tag{3.68}$$

Suppose, as be fore, that in a $\mathrm{QC}_{\mathcal{C}}$-computation on the whole cluster $\mathcal{C}$ the first measurement round is split. The σ_z-measurements are performed first, and the σ_x- and σ_y-measurements second. Further suppose that one started with a cluster state $|\phi_{\{\kappa\}}\rangle_{\mathcal{C}}$ for which all parameters κ are zero, $\{\kappa_k = 0 \,|\forall k \in \mathcal{C}\}$. It is possible that all σ_z-measurements yield the outcome zero, $\{s_a = 0 \,|\, \forall a \in Q_{0,z}\}$. Then, the resulting state of the unmeasured qubits $q \in \mathcal{C}_N \backslash O$ is a cluster state $|\phi\rangle_{\mathcal{C}_N \backslash O}$ with $\{\kappa_q = 0 \,|\, \forall q \in \mathcal{C}_N \backslash O\}$. Therefore, the

formula (3.68) also applies for $\mathbf{I}$, provided that one started from a cluster state $|\phi_{\{\kappa\}}\rangle_{\mathcal{C}}$ with property $\{\kappa_k = 0\,|\forall k \in \mathcal{C}\}$ and obtained only measurement outcomes $s_a = 0$ in the σ_z-measurements on the qubits in $Q_{0,z}$. As follows from (3.52), under these assumptions we have

$$s_q = \overline{s}_q, \quad \forall\ q \in \mathcal{C}_N \backslash O. \tag{3.69}$$

Therefore, we can replace –for the so far considered special case– $\{s_a\}$ by $\{\overline{s}_a\}$, the set of quantities which are invariant under the symmetry transformations (3.47), and obtain by use of (3.52),

$$\mathbf{I} = \sum_{k \in \mathcal{C}_N \backslash O} s_k \mathbf{F}_k + \sum_{k \in \mathcal{C}_N \backslash O} \sum_{j\,|\, \substack{j \in \mathrm{nbgh}(k) \wedge \\ j \in Q_{0,z}}} s_j \mathbf{F}_k + \sum_{k \in \mathcal{C}_N \backslash O} \kappa_k \mathbf{F}_k + \sum_{g \in \mathcal{N}} \mathbf{F}_g. \tag{3.70}$$

The equation (3.70) is valid for all sets of parameters $\{\kappa_k\,|\,k \in \mathcal{C}\}$ and measurement results $\{s_a\,|\,\forall a \in Q_{0,z}\}$ for $\mathbf{I}$ is a computationally meaningful quantity and thus has to be invariant under the symmetry transformations (3.47). Invariance of (3.70) is guaranteed by the construction of the $\overline{s}_q$ in (3.52).

In (3.70), the second factor can be rewritten in the following way

$$\begin{aligned} \sum_{k \in \mathcal{C}_N \backslash O} \sum_{j\,|\, \substack{j \in \mathrm{nbgh}(k) \wedge \\ j \in Q_{0,z}}} s_j \mathbf{F}_k &= \sum_{j \in \mathcal{C}_N \backslash O} \sum_{k\,|\, \substack{j \in \mathrm{nbgh}(k) \wedge \\ k \in Q_{0,z}}} s_k \mathbf{F}_j \\ &= \sum_{\substack{(j,k)\,|\, j \in \mathcal{C}_N \backslash O \wedge \\ j \in \mathrm{nbgh}(k) \wedge k \in Q_{0,z}}} s_k \mathbf{F}_j \\ &= \sum_{k \in Q_{0,z}} \sum_{j\,|\, \substack{j \in \mathrm{nbgh}(k) \wedge \\ j \in \mathcal{C}_N \backslash O}} s_k \mathbf{F}_j. \end{aligned} \tag{3.71}$$

In the first line of (3.71) the labels j and k were interchanged and the relation $j \in \mathrm{nbgh}(k) \iff k \in \mathrm{nbgh}(j)$ was used. In the second and third line the order of the products over k and j was interchanged.

We now define the forward propagated byproduct operators $U_k|_\Omega$ for qubits k in the set $Q_{0,z}\backslash O = \mathcal{C}\backslash\mathcal{C}_N$ as

$$\mathbf{F}_k = \sum_{j\,|\, \substack{j \in \mathrm{nbgh}(k) \wedge \\ j \in \mathcal{C}_N \backslash O}} \mathbf{F}_j\,, \qquad \forall k \in Q_{0,z}. \tag{3.72}$$

In this way, we have traced back the forward propagated byproduct operators for qubits $k \in Q_{0,z}$ to those for qubits $j \in \mathcal{C}_N\backslash O$ which are already known. Note that this includes the qubits $k' \in O$ for which the byproduct operator $U_{k'}|_\Omega$ has been defined in (3.2). Using the isomorphism (3.10) also yields a definition for $\mathbf{F}_{k'}$, $k' \in O$. Both definitions are equivalent.

We insert (3.71) into (3.70) and, with the definition (3.72), we finally obtain

$$\mathbf{I} = \sum_{k \in \mathcal{C}} s_k \mathbf{F}_k + \sum_{k \in \mathcal{C}\backslash Q_{0,z}} \kappa_k \mathbf{F}_k + \sum_{g \in \mathcal{N}} \mathbf{F}_g \tag{3.73}$$

To derive the expression (3.73) for the information vector has been the primary purpose of this section.

With the expression (3.73) at hand we are finally able to define the quantity which carries the algorithmic information during the computational process and which has already been mentioned on earlier occasions in this thesis, the *information flow vector* $\mathbf{I}(t)$.

Definition 6 *The information flow vector* $\mathbf{I}(t)$ *is given by*

$$\mathbf{I}(t) = \sum_{k \in \bigcup_{i=0}^{t} Q_i} s_k \, \mathbf{F}_k + \sum_{k \in \mathcal{C} \setminus Q_{0,z}} \kappa_k \mathbf{F}_k + \sum_{g \in \mathcal{N}} \mathbf{F}_g \tag{3.74}$$

The quantity $\mathbf{I}(t)$ is similar to $\mathbf{I}$ as given in (3.73), but to $\mathbf{I}(t)$ only contribute the byproduct images of qubits from a subset $\bigcup_{i=1}^{t} Q_i$ of $\mathcal{C}$. The information flow vector $\mathbf{I}(t_{\max})$ after the final measurement round $t_{\max}$ equals the information vector $\mathbf{I}$,

$$\mathbf{I} = \mathbf{I}(t_{\max}). \tag{3.75}$$

As will be shown later, during all steps of the computation, except for after the final one, the information flow vector determines the measurement bases for the cluster qubits that are to be measured in the next round. After the final round it contains the result of the computation. Thus, it has a meaning in every step of the computation. No further information obtained from the measurements is needed. In this sense, the information flow vector can be regarded as the carrier of the *algorithmic information* on the $\mathrm{QC}_{\mathcal{C}}$.[2]

The information flow vector has a constant part which does not depend on the measurement results $\{s_k\}$. This part alone forms its initialization value $\mathbf{I}_{\mathrm{init}}$,

$$\mathbf{I}_{\mathrm{init}} = \sum_{k \in \mathcal{C} \setminus Q_{0,z}} \kappa_k \mathbf{F}_k + \sum_{g \in \mathcal{N}} \mathbf{F}_g, \tag{3.76}$$

such that $\mathbf{I}(t)$ becomes

$$\mathbf{I}(t) = \mathbf{I}_{\mathrm{init}} + \sum_{k \in \bigcup_{i=0}^{t} Q_i} s_k \, \mathbf{F}_k. \tag{3.77}$$

From eq. (3.76) we see that the byproduct images of the gates $\mathbf{F}_g$ do not form an independent part of the information specifying a quantum algorithm on the $\mathrm{QC}_{\mathcal{C}}$. Instead, they are absorbed into the initialization value $\mathbf{I}_{\mathrm{init}}$ of $\mathbf{I}(t)$.

The measurement bases in which the results s_k are obtained –referred to implicitly in (3.73) and (3.74)– are not fixed a priori, but must be determined during the computation. They will be calculated using the byproduct images $\{\mathbf{F}_k, k \in \mathcal{C}\}$ and $\mathbf{I}(t)$, as explained in

[2]The way we use the term "algorithmic information" has nothing to do with the –in general non-computable– algorithmic information content of an object as it is defined in Kolmogorov complexity theory [63].

Sections 3.4.2 and 3.4.3. Besides the byproduct images, the algorithm angles $\varphi_{j,\mathrm{algo}},\ j \in Q^{(1)}$ are also needed to determine the appropriate measurement bases. They are related to the network angles $\varphi_{j,\mathrm{qln}},\ j \in Q^{(1)}$ that specify the one-qubit rotations in the corresponding quantum logic network via

$$\varphi_{j,\mathrm{algo}} = (-1)^{\eta_j}\, \varphi_{j,\mathrm{qln}}\,, \quad j \in Q^{(1)}, \tag{3.78}$$

where η_j is given by

$$\eta_j = \sum_{\substack{k|\ k\in\mathcal{C}\setminus Q_{0,z},\\ j\in \mathrm{bc}(k)}} \kappa_k + \sum_{\substack{g|\ g\in\mathcal{N},\\ j\in\mathrm{bc}(g)}} 1. \tag{3.79}$$

The pair of equations (3.78), (3.79) is, for the moment, just a definition of the algorithm angles. It will become apparent in Sections 3.4.2 and 3.4.3 that this definition is indeed useful.

3.4.2 Description of the model

As already listed in Section 3.2.5, a quantum algorithm on the $\mathrm{QC}_{\mathcal{C}}$ is specified by the sets Q_t of simultaneously measured qubits, the backward cones $\mathrm{bc}(k)$ of the qubits $k \in Q_0$, the measurement bases of the qubits $k \in Q_0$, the byproduct images $\mathbf{F}_j$ for $j \in \mathcal{C}$, the algorithm angles $\varphi_{l,\mathrm{algo}}$ for $l \in Q^{(1)}$ and the initialization value $\mathbf{I}_{\mathrm{init}}$ of the information flow vector $\mathbf{I}(t)$. If an algorithm is not given in this form but rather as a quantum logic network composed of CNOT gates and one-qubit rotations, the above quantities can be derived from the network as explained in the previous sections.

Let us summarize this step of classical pre-processing. First, the measurement pattern is obtained –if one has no better idea– by patching together the measurement patterns for the individual gates displayed in Fig. 2.2. This gives the measurement directions for the qubits $k \in Q_0$. The network angles $\varphi_{j,\mathrm{qln}}$ for the qubits $j \in Q^{(1)}$ are taken from the quantum logic network to be simulated. To determine the sets $\{Q_t, t = 0..t_{\max}\}$, we need the forward cones. The forward cones $\mathrm{fc}(k)$ for all qubits $k \in \mathcal{C}$ can be obtained using the expressions (2.29), (2.23) for the byproduct operators and the propagation relations (2.52), (2.51) and (2.53). From the forward cones we derive a strict partial ordering "$\prec$" (3.5) among the cluster qubits, and from the strict partial ordering we derive the sets $Q_t \subset \mathcal{C}$ via (3.8). The byproduct images $\mathbf{F}_k$ for the qubits $k \in \mathcal{C}\setminus Q_{0,z}$ are obtained from their definition (3.11) once the corresponding forward propagated byproduct operators are obtained from (3.61). The byproduct images of the qubits $k \in Q_{0,z}$ are traced back to those in the set $\mathcal{C}\setminus Q_{0,z}$ via eq. (3.72). To determine the algorithm angles we need the backward cones $\mathrm{bc}(k)$ for the qubits $k \in Q_0$ and the backward cones of gates $\mathrm{bc}(g)$. Then, the algorithm angles are given by (3.78), (3.79). Finally, for the initialization value $\mathbf{I}_{\mathrm{init}}$ of the information flow vector we need the byproduct images $\mathbf{F}_g$ of the gates g which we obtain from eq. (3.12). $\mathbf{I}_{\mathrm{init}}$ is set via (3.76). All the pre-processing required to extract the listed quantities from a quantum logic network can be performed efficiently on a classical computer, see Section 3.5.3.

The scheme of quantum computation on the $\mathrm{QC}_{\mathcal{C}}$ comprises several measurement rounds in which the following steps have to be performed:

1. First measurement round.

   (a) Measure all qubits $k \in Q_0$. Obtain measurement results $\{s_k | k \in Q_0\}$.

   (b) Modify the angles $\varphi_{j,\mathrm{algo}}$ for the continuous gates

   $$\varphi_{j,\mathrm{algo}} \longrightarrow \varphi'_{j,\mathrm{algo}} = \varphi_{j,\mathrm{algo}} \, (-1)^{\eta'_j}, \tag{3.80}$$

   with

   $$\eta'_j = \sum_{k \in Q_0 | j \in \mathrm{bc}(k)} s_k \tag{3.81}$$

   for all $j \in Q^{(1)}$.

   (c) Update the information flow vector from $\mathbf{I}_{\mathrm{init}}$ to $\mathbf{I}(0)$

   $$\mathbf{I}(0) = \mathbf{I}_{\mathrm{init}} + \sum_{k \in Q_0} s_k \mathbf{F}_k. \tag{3.82}$$

2. Subsequent measurement rounds.

   Perform the following three steps (2a) - (2c) for all qubit sets $Q_t \subset \mathcal{C} \backslash Q_0$ in ascending order, beginning with Q_1. In the measurement round t,

   (a) Determine the measurement bases for $j \in Q_t$ according to

   $$\varphi_{j,\mathrm{meas}} = \varphi'_{j,\mathrm{algo}} \, (-1)^{(\mathbf{I}(t-1), \mathbf{F}_j)_S} \tag{3.83}$$

   (b) Perform the measurements on the qubits $j \in Q_t$. Thereby obtain the measurement results $\{s_j \in \{0,1\} \mid j \in Q_t\}$.

   (c) Update the information flow vector $\mathbf{I}$

   $$\mathbf{I}(t) = \mathbf{I}(t-1) + \sum_{j \in Q_t} s_j \, \mathbf{F}_j. \tag{3.84}$$

The information flow vector $\mathbf{I}(t_{\max})$ after the final measurement round $t_{\max}$ equals the information vector $\mathbf{I}$, as can be seen from (3.75). At the end of the computation, from $\mathbf{I}$ we can directly read off the result $\mathbf{I}_x$ of the computation. $\mathbf{I}_x$ is identical to the readout of the corresponding quantum logic network.

Remark 1. Note that in the first measurement round the byproduct operators created by the measurements on qubits in Q_0 have been propagated *backwards* to set the angles $\{\varphi'_{j,\mathrm{algo}}\}$. There is also a scheme in which the byproduct operators caused by the measurements in the initialization round are propagated forward to set the modified algorithm

angles $\{\varphi'_{j\text{algo}}\}$. In that scheme, the update of the information flow vector $\mathbf{I}(t)$ and the rule to determine the measurement angles $\varphi_{j,\text{meas}}$ are the same as in the described scheme, given by (3.83) and (3.84). What is different is the initialization and the appearance of a step of post-processing. In the modified scheme, in eqs. (3.79) and (3.81) the backward cones $\text{bc}(k)$ are replaced by the respective forward cones $\text{fc}(k)$ and $\mathbf{I}_{\text{init}}$ is set to zero. The quantity which was $\mathbf{I}_{\text{init}}$ in (3.76) is computed as well but now stored as an auxiliary quantity $\Delta\mathbf{I}$ until the end of the computation. After the last measurement round $t_{\max}$, the information vector $\mathbf{I}$ then is obtained by the relation $\mathbf{I} = \mathbf{I}(t_{\max}) + \Delta\mathbf{I}$, which requires the extra post-processing step and extra memory during the computation. We have chosen to present the scheme with backward propagation of byproduct operators in order to avoid this superfluous post-processing. This way, the quantity $\mathbf{I}(t)$ which steers the computational process directly displays the result of the computation after the final update to $\mathbf{I}(t_{\max})$.

Remark 2. This comment concerns the choice $\mathcal{O} = \Omega$ of the cut on which the byproduct images $\mathbf{F}_k$ and $\mathbf{F}_g$ are evaluated. In the visualization of the $\text{QC}_\mathcal{C}$ as an implementation of a quantum logic network the cut Ω plays a distinguished role. The byproduct operators accumulated at Ω determine how the "readout" measurements have to be interpreted. In the computational model underlying the $\text{QC}_\mathcal{C}$, however, the former readout qubits are just qubits to be measured like any other cluster qubits. Here, the cut Ω is not distinguished. Due to the invariance (3.21) of the symplectic scalar product (3.20) the byproduct images $\mathbf{F}_k$, which enter the expression (3.83) for the $\varphi_{k,\text{meas}}$ directly and via (3.76) and (3.84), can be evaluated with respect to *any* vertical cut $\mathcal{O}$ through the corresponding quantum logic network. The information vector $\mathbf{I}$ which displays the result of the computation in its x-part $\mathbf{I}_x$ would then be related to the information flow vector after the final measurement round $\mathbf{I}(t_{\max})$ via $\mathbf{I} = C(\mathcal{N}_{\mathcal{O}\to\Omega})\,\mathbf{I}(t_{\max})$. Thus, the particular vertical cut $\mathcal{O} = \Omega$ was chosen just to avoid an additional step of post-processing. The dependence on the cut $\mathcal{O}$ would vanish altogether if one would write the n output bits of the quantum computation in the form $[I_x]_i = (\mathbf{I}|_\mathcal{O}, f_i|_\mathcal{O})_S$ for suitably chosen $\{f_i \in \mathcal{V}, i = 1, .., n\}$, e.g. for the case $\mathcal{O} = \Omega$, $f_1 = (0, .., 0; 1, 0, .., 0)^T$, $f_2 = (0, .., 0; 0, 1, 0, .., 0)^T$, and the other f_i, $i \leq n$ accordingly.

3.4.3 Proof of the model

In this section it is shown that if we run the $\text{QC}_\mathcal{C}$ according to the scheme described in Section 3.4.2, we obtain the same result as in the corresponding quantum logic network. This requires to prove that (a) one does indeed choose all the measurement angles correctly and (b) obtains at the end of the computation the result $\mathbf{I}_x$, the x-part of the information vector $\mathbf{I}$ as given in (3.73).

To show point (b), we use (3.76), (3.82) and (3.84) and obtain for the information vector

$$\mathbf{I} = \sum_{k\in\mathcal{C}\setminus Q_{0,z}} \kappa_k \mathbf{F}_k + \sum_{g\in\mathcal{N}} \mathbf{F}_g + \sum_{k\in Q_0} s_k \mathbf{F}_k + \sum_{k\in\bigcup_{i=1}^{t_{\max}} Q_i} s_k \mathbf{F}_k$$

which coincides with (3.73). This ensures that we obtain the right vector $\mathbf{I}$ at the end of

the computation, provided the measurement bases were chosen appropriately, as required for (a). This is checked below.

First we observe that the measurement angle $\varphi_{j,\mathrm{meas}}$ and the network angle $\varphi_{j,\mathrm{qln}}$ are for all $j \in Q^{(1)}$ related in the following way

$$\varphi_{j,\mathrm{meas}} = (-1)^{\vartheta_j}\, \varphi_{j,\mathrm{qln}}\,, \tag{3.85}$$

with

$$\vartheta_j = \sum_{k\in\mathcal{C}|j\in\mathrm{fc}(k)} s_k + \sum_{k\in\mathcal{C}\setminus Q_{0,z}|j\in\mathrm{fc}(k)} \kappa_k + \sum_{g\in\mathcal{N}|j\in\mathrm{fc}(g)} 1 \mod 2. \tag{3.86}$$

Why does the pair of equations (3.85), (3.86) hold? As can be seen from the propagation relation for rotations (2.52), the network and the measurement angle of a qubit $j \in Q^{(1)}$ can differ only by a sign factor ± 1 and can therefore always be related as in (3.85). The first and the third sum in (3.86) follow from the definition of the forward cones of the cluster qubits and of gates in Section 3.2.2. The measurement angle at j acquires a factor $(-1)^{s_k}$ if $j \in \mathrm{fc}(k)$ and a factor of (-1) for each gate g with $j \in \mathrm{fc}(g)$.

Now, we rewrite the quantity ϑ_j in the following way

$$\begin{aligned}
\vartheta_j &= \sum_{k\in\mathcal{C}|j\in\mathrm{fc}(k)} s_k + \sum_{k\in\mathcal{C}\setminus Q_{0,z}|j\in\mathrm{fc}(k)} \kappa_k + \sum_{g\in\mathcal{N}|j\in\mathrm{fc}(g)} 1 \mod 2 \\
&= \sum_{k\in\mathcal{C}|j\in\mathrm{fc}(k)} s_k + \sum_{k\in\mathcal{C}\setminus Q_{0,z}|j\in\mathrm{fc}(k)} \kappa_k + \sum_{g\in\mathcal{N}|j\in\mathrm{fc}(g)} 1 + \\
&\quad +2\left(\sum_{k\in Q_0|j\in\mathrm{bc}(k)} s_k + \sum_{k\in\mathcal{C}\setminus Q_{0,z}|j\in\mathrm{bc}(k)} \kappa_k + \sum_{g\in\mathcal{N}|j\in\mathrm{bc}(g)} 1\right) \mod 2 \\
&= \underbrace{\sum_{k\in Q_0|j\in\mathrm{fc}(k)\vee j\in\mathrm{bc}(k)} s_k}_{S_1} + \underbrace{\sum_{k\in Q^{(1)}|j\in\mathrm{fc}(k)} s_k}_{S_2} + \underbrace{\sum_{k\in Q_0|j\in\mathrm{bc}(k)} s_k}_{S_3} + \\
&\quad + \underbrace{\sum_{k\in\mathcal{C}\setminus Q_{0,z}|j\in\mathrm{fc}(k)\vee j\in\mathrm{bc}(k)} \kappa_k}_{S_4} + \underbrace{\sum_{k\in\mathcal{C}\setminus Q_{0,z}|j\in\mathrm{bc}(k)} \kappa_k}_{S_5} + \\
&\quad + \underbrace{\sum_{g\in\mathcal{N}|j\in\mathrm{fc}(g)\vee j\in\mathrm{bc}(g)} 1}_{S_6} + \underbrace{\sum_{g\in\mathcal{N}|j\in\mathrm{bc}(g)} 1}_{S_7} \mod 2.
\end{aligned} \tag{3.87}$$

We now discuss the seven terms $S_1, .., S_7$. All sums are evaluated modulo 2.

Term S_1 of (3.87):

$$S_1 = \sum_{k\in Q_0|j\in\mathrm{fc}(k)\vee j\in\mathrm{bc}(k)} s_k = \sum_{k\in Q_0} s_k\, (\mathbf{F}_k, \mathbf{F}_j)_S, \tag{3.88}$$

where the last identity holds by the cone test (3.25).

Term S_2 of (3.87):
Let be $j \in Q_t$ and $k \in Q_i$. Qubit j can only then be in the forward cone of k, $j \in \mathrm{fc}(k)$, if $i < t$. Hence

$$\begin{aligned} S_2 &= \sum_{k \in Q^{(1)} \big| j \in \mathrm{fc}(k)} s_k \\ &= \sum_{k \in \bigcup_{i=1}^{t-1} Q_i | j \in \mathrm{fc}(k)} s_k \\ &= \sum_{k \in \bigcup_{i=1}^{t-1} Q_i} s_k \left(\mathbf{F}_k, \mathbf{F}_j\right)_S. \end{aligned} \tag{3.89}$$

In (3.89) the last line again follows by using the cone test (3.25).

Term S_3 of (3.87):

$$S_3 = \sum_{k \in Q_0 | j \in \mathrm{bc}(k)} s_k = \eta_j'. \tag{3.90}$$

This equity follows by the definition of η_j' in (3.81). Thus, the term S_3 is the contribution to ϑ_j coming from the first measurement round where the algorithm angles $\{\varphi_{j,\mathrm{algo}}\}$ are changed to the modified algorithm angles $\{\varphi'_{j,\mathrm{algo}}\}$.

Term S_4 of (3.87):

$$S_4 = \sum_{k \in \mathcal{C} \setminus Q_{0,z} | j \in \mathrm{fc}(k) \vee j \in \mathrm{bc}(k)} \kappa_k = \sum_{k \in \mathcal{C} \setminus Q_{0,z}} \kappa_k \left(\mathbf{F}_k, \mathbf{F}_j\right)_S, \tag{3.91}$$

which follows by the cone test (3.25).

Terms $S_5 + S_7$ of (3.87):

$$S_5 + S_7 = \sum_{k \in \mathcal{C} \setminus Q_{0,z} | j \in \mathrm{bc}(k)} \kappa_k \; + \sum_{g \in \mathcal{N} | j \in \mathrm{bc}(g)} 1 = \eta_j, \tag{3.92}$$

via the definition (3.79) of the η_j.

Finally, term S_6 of (3.87):

$$S_6 = \sum_{g \in \mathcal{N} | j \in \mathrm{fc}(g) \vee j \in \mathrm{bc}(g)} 1 = \sum_{g \in \mathcal{N}} \left(\mathbf{F}_g, \mathbf{F}_j\right)_S, \tag{3.93}$$

which follows by the cone criterion (3.28) for gates.

Now we combine these seven terms $S_1, .., S_7$. By (3.87) - (3.93) we obtain

$$\begin{aligned} \vartheta_j &= \eta_j + \eta_j' + \sum_{g \in \mathcal{N}} \left(\mathbf{F}_g, \mathbf{F}_j\right)_S \; + \sum_{k \in \mathcal{C} \setminus Q_0} \kappa_k \left(\mathbf{F}_k, \mathbf{F}_j\right)_S \; + \sum_{k \in \bigcup_{i=0}^{t-1} Q_i} s_k \left(\mathbf{F}_k, \mathbf{F}_j\right)_S \\ &= \eta_j + \eta_j' + \left(\mathbf{I}(t-1), \mathbf{F}_j\right)_S. \end{aligned} \tag{3.94}$$

The last line follows from the definition (3.74) of the information flow vector. If we consider the relations (3.78), (3.80) and (3.85) between the angles $\varphi_{j,\text{algo}}$, $\varphi'_{j,\text{algo}}$ and $\varphi_{j,\text{meas}}$, we find

$$\varphi_{j,\text{meas}} = (-1)^{\vartheta_j - \eta_j - \eta'_j} \varphi'_{j,\text{algo}}. \tag{3.95}$$

Now we insert (3.94) into (3.95) and obtain

$$\varphi_{j,\text{meas}} \quad = \quad \varphi'_{j,\text{algo}} \, (-1)^{(\mathbf{I}(t-1),\mathbf{F}_j)_S},$$

which proofs that the assignment of the measurement angles (3.83) is correct, and thereby concludes the proof of the computational model described in Section 3.4.2.

3.5 Logical depth and temporal complexity

The logical depth has, to our knowledge, only been defined in the context of quantum logic networks, but it can straightforwardly be generalized to the $\text{QC}_\mathcal{C}$. In networks one groups gates which can be performed in parallel to layers. The logical depth of a quantum logic network then is the minimum number of its layers. Similarly in case of the $\text{QC}_\mathcal{C}$, one can group the cluster qubits which can be measured simultaneously to sets Q_t. There, the logical depth of the $\text{QC}_\mathcal{C}$-realization of an algorithm is the minimal number of such sets.

Since the one-qubit measurements on the cluster state mutually commute, one may be led to think that they can always be performed all in parallel. They could, but then the measurements would in general not drive a deterministic computation.

In the following, we will denote the logical depth in the context of the $\text{QC}_\mathcal{C}$ by D and the logical depth of a quantum logic network by $D_\mathcal{N}$.

3.5.1 $D = 2$ for circuits of CNOT gates and $U(1)$-rotations

In Section 2.2.9 we have already seen that the whole Clifford part of a circuit can be performed in a single step of measurements, which is not obvious from the network perspective. In this section we give a further example for a $\text{QC}_\mathcal{C}$-circuit with constant logical depth. Specifically, we prove that the logical depth D of a circuit composed of either CNOT gates and rotations about the x-axis or of CNOT gates and rotations about the z-axis is $D = 2$. This set of circuits contains all circuits of diagonal 2-qubit gates as a special case. For circuits of diagonal 2-qubit gates we can compare our result $D = 2$ to the best known result [56] for quantum logic networks where the logical depth scales logarithmically in the number of gates.

Here we give the proof for circuits of CNOT gates and rotations about the z-axis $U_z(\alpha) = e^{-i\alpha\frac{\sigma_z}{2}}$. The elementary gates used are (a) the rotations about the z-axis $U_z(\alpha) = e^{-i\alpha\frac{\sigma_z}{2}}$, and (b) the CNOT gate between neighbouring logical qubits. The realization of the rotation U_z is depicted in Fig. 2.2. Of the CNOT gate between neighbouring qubits we construct the swap gate between neighbouring qubits and by that the general CNOT gate, as in Section 2.2.9. The strategy to implement the circuit is then: (1) to measure all those

qubits on $\mathcal{C}$ which are to be measured in the eigenbases of σ_x, σ_z or σ_y; and (2) to measure the remaining qubits, i.e. the ones which are measured in a direction in the $x-y$-plane.

The result that the measurements in step (1) can be performed in one step has already been shown in Section 2.2.9. It remains to be shown that the measurements in the tilted measurement directions of step (2) can also be performed in parallel. Let j and l be two cluster qubits which are measured in a tilted basis in step 2 in order to implement the rotations. Using (3.8) one finds

$$D > 2 \Longrightarrow \exists\, j, l \in Q^{(1)} :\ l \prec j \ \ (\text{that is, } Q^{(2)} \neq \emptyset). \tag{3.96}$$

Further holds

$$l \prec j \Longrightarrow \exists\, k \in Q^{(1)} :\ j \in \mathrm{fc}(k), \tag{3.97}$$

because the strict partial ordering "$\prec$" is generated by the forward cones, i.e. $l \prec j \Longleftrightarrow$ either $j \in \mathrm{fc}(l)$, or $\exists (k_1, \ldots k_r) :\ k_1 \in \mathrm{fc}(l) \ \wedge\ \left\{ k_s \in \mathrm{fc}(k_{s-1}) |\, 2 \leq s \leq r \right\} \ \wedge\ j \in \mathrm{fc}(k_r)$.

Moreover, from the criterion (3.25) one derives

$$j \in \mathrm{fc}(k) \Longrightarrow (\mathbf{F}_j, \mathbf{F}_k)_S = 1. \tag{3.98}$$

Now, by putting the implications (3.96), (3.97) and (3.98) together we obtain

$$D > 2 \Longrightarrow \exists\, j, k \in Q^{(1)} :\ (\mathbf{F}_j, \mathbf{F}_k)_S = 1, \tag{3.99}$$

which we negate to obtain

$$\forall j, k \in Q^{(1)} : (\mathbf{F}_j, \mathbf{F}_k)_S = 0 \Longrightarrow D \leq 2. \tag{3.100}$$

Next it is proved that $(\mathbf{F}_j, \mathbf{F}_k)_S = 0$ does indeed hold for all $j, k \in Q^{(1)}$.

A measurement of a qubit at site k , which is part of the implementation of a rotation about the z-axis (central qubit 3 in Fig. 2.2c), generates a byproduct operator $(U_k)^{s_k} = (\sigma_z)^{s_k}$. This can be seen from equations (2.24), (2.28) and (2.29). Note that in Fig. 2.2c, qubits 1,2,4 are measured in the σ_x-eigenbasis, they belong to the set Q_0. Now let be i the number of the *logical* qubit on which the rotation $U_z(\varphi_k)$ is performed by the measurement of cluster qubit k. Further, let $\mathcal{O}$ be a vertical cut through the network simulated by the $\mathrm{QC}_{\mathcal{C}}$. $\mathcal{O}$ intersects each qubit line only once. In particular, it shall intersect the qubit line i just at the output side of the rotation $U_z(\varphi_k)$. Thus, the image $\mathbf{F}_k|_{\mathcal{O}}$ of U_k on the cut $\mathcal{O}$ is

$$\mathbf{F}_k|_{\mathcal{O}} = \begin{pmatrix} 0 \\ \\ \mathbf{F}_{kz}|_{\mathcal{O}} \end{pmatrix}, \quad \text{with } F_{kz,l} = \delta_{il}. \tag{3.101}$$

What we see from (3.101) is that $\mathbf{F}_{kx}|_{\mathcal{O}} = 0$. Be $\mathcal{N}_{\mathcal{O}\to\Omega}$ the part of the network $\mathcal{N}$ which is located between the two cuts $\mathcal{O}$ and Ω. The byproduct image $\mathbf{F}_k$ corresponding to U_k is then given by

$$\mathbf{F}_k \equiv \mathbf{F}_k|_{\Omega} = C(\mathcal{N}_{\mathcal{O}\to\Omega})\, \mathbf{F}_k|_{\mathcal{O}}. \tag{3.102}$$

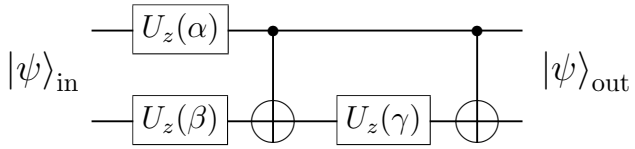


Figure 3.3: Network for a diagonal gate composed of rotations U_z and CNOT gates.

The only gates that contribute to $C(\mathcal{N}_{\mathcal{O}\to\Omega})$ are the CNOT gates, as described in section 3.2.4. The propagation matrices for CNOT gates (3.19) have block-diagonal form. Hence, using (3.14) the propagation matrix for the network $\mathcal{N}_{\mathcal{O}\to\Omega}$ has block-diagonal form

$$C(\mathcal{N}_{\mathcal{O}\to\Omega}) = \left(\begin{array}{c|c} C_{xx}(\mathcal{N}_{\mathcal{O}\to\Omega}) & 0 \\ \hline 0 & C_{zz}(\mathcal{N}_{\mathcal{O}\to\Omega}) \end{array}\right). \tag{3.103}$$

From (3.101), (3.102) and (3.103) it follows that the x-part of the byproduct image vector $\mathbf{F}_k$ vanishes for all k

$$[\mathbf{F}_x]_k = 0 \;\; \forall k \in Q^{(1)}. \tag{3.104}$$

Hence by the definition of the symplectic scalar product (3.20), we obtain $(\mathbf{F}_j, \mathbf{F}_k)_S = 0$ for all $j, k \in Q^{(1)}$. This proves via (3.100) $D \leq 2$. The measurements to implement the one-qubit rotations can thus all be performed at the same time. In (3.100) the case $D = 1$ can be easily be excluded for all interesting cases such that only $D = 2$ remains. This concludes the proof of $D = 2$ for circuits of CNOT gates and rotations of the form $e^{-i\varphi\frac{\sigma_z}{2}}$. The proof for circuits of CNOT gates and rotations $e^{-i\varphi\frac{\sigma_x}{2}}$ runs analogously. Now let us discuss the special case of circuits composed of diagonal two-qubit gates. A diagonal gate G_d in the computational basis is of the form

$$G_d = \begin{pmatrix} e^{i\varphi_1} & & & \\ & e^{i\varphi_2} & & \\ & & e^{i\varphi_3} & \\ & & & 1 \end{pmatrix}, \tag{3.105}$$

modulo a possible global phase which is not relevant.

The network of rotations about the z-axis and of a CNOT gate shown in Fig. 3.3 realizes a general diagonal two-qubit gate. In order to obtain the angles φ_1, φ_2 and φ_3 specifying the diagonal gate G_d in (3.105), one chooses the following angles for the three z-rotations

in this network

$$\begin{aligned} \alpha &= \frac{1}{2}(-\varphi_1 - \varphi_2 + \varphi_3), \\ \beta &= \frac{1}{2}(-\varphi_1 + \varphi_2 - \varphi_3), \\ \gamma &= \frac{1}{2}(-\varphi_1 + \varphi_2 + \varphi_3). \end{aligned} \tag{3.106}$$

Thus, a circuit of diagonal two-qubit gates can also be regarded as a circuit of z-rotations and CNOT gates. Therefore we find $D = 2$ for circuits of diagonal two-qubit gates on the $\mathrm{QC}_{\mathcal{C}}$. This result can be compared to the best known upper bound [56] for quantum logic networks where the logical depth is of the order $\mathcal{O}(\log n_G)$ with n_G the number of two-qubit gates.

3.5.2 The logical depth D is a good measure for temporal complexity

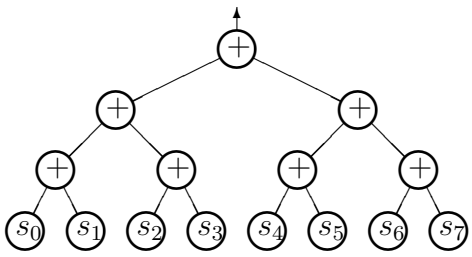
In this section, we discuss the temporal resources consumed by a $\mathrm{QC}_{\mathcal{C}}$-computation at runtime. Specifically, we will express the computation time as a function of the logical depth. Note that additional temporal resources are required for the design of the $\mathrm{QC}_{\mathcal{C}}$-circuit which will be discussed in the next section.

The computational model described in section 3.4.2 consists of an alternating series of measurement rounds and classical processing of the thereby obtained measurement results. The classical processing contributes to the duration of the computation and will therefore enter into the relation between the computation time and the logical depth. For the computation time, this results in a correction logarithmic in the number n of logical qubits involved, and thus the computation time is no longer the logical depth times a constant. For all practical purposes, however, this logarithmic correction is small compared to the time required for the genuine quantum part of the computation, consisting of the measurements.

Let Δ_Q be the time required to perform the simultaneous measurements in one measurement round and Δ_{cl} the time required for the elementary steps of classical processing: say, addition modulo 2 or multiplication of two bits. The time $T_{cl}(t)$ required for classical processing after each measurement round has two contributions. First, the time $T_{cl,\mathbf{I}}(t)$ to update the information flow vector $\mathbf{I}(t)$ and second, the time $T_{cl,\pm}(t)$ to determine the signs of the measurement angles of all measurements in the next round. The total computation time T_{comp} is given by

$$T_{\mathrm{comp}} = D\Delta_Q + \sum_{t=0}^{D-1} T_{cl,\mathbf{I}}(t) + T_{cl,\pm}(t) \tag{3.107}$$

The update of the information vector $\mathbf{I}(t)$ according to (3.84) can be done for all $2n$ components in parallel. The update $\mathbf{I}(t-1) \longrightarrow \mathbf{I}(t)$ following measurement round t requires the time that it takes to add up $\|Q_t\|$ bits modulo 2. As the drawing below illustrates, $T_{cl,\mathbf{I}}(t)$ is logarithmic in $\|Q_t\|$.



The number of qubits in the set Q_t is bounded from above by $\|\mathcal{C}\|$ since $Q_t \subset \mathcal{C}$. Here, $\mathcal{C}$ is any cluster sufficiently large to carry the network to be simulated. Thus

$$T_{cl,\mathbf{I}}(t) \leq \Delta_{cl} \log \|\mathcal{C}\|. \tag{3.108}$$

To determine the proper measurement angle $\varphi_{k,\text{meas}}$ for the measurement on qubit $k \in Q_{t+1}$ in the next measurement round requires, according to (3.83), the evaluation of the symplectic scalar product $(\mathbf{I}(t), F_k)_S$. This requires 1 step for multiplication and $\log 2n$ steps for addition modulo 2. Thus,

$$T_{cl,\pm} = \Delta_{cl} \left(\log n + 2\right). \tag{3.109}$$

Combining (3.107), (3.108) and (3.109), the total computation time T_{comp} is bounded from above by

$$T_{\text{comp}} \leq D\, \Delta_Q \left(1 + \frac{\Delta_{cl}}{\Delta_Q} \left[\log \|\mathcal{C}\| + \log n + 2\right]\right). \tag{3.110}$$

We see that, although the computation time T_{comp} is linear in the logical depth D, it contains contributions logarithmic in the number n of logical qubits and in the cluster size $\|\mathcal{C}\|$. These logarithmic contributions are, however, suppressed by the ratio between the characteristic time for classical processing and the characteristic time for the von-Neumann measurements, Δ_{cl}/Δ_Q. This ratio can, in practice, be very small. Therefore, the logarithmic corrections become important only in the limit of large clusters and large n. As will be argued below, even in the regimes where a quantum computer is believed to become useful, say $n \approx 10^5$, the logarithmic corrections have only a minor influence on the total computation time.

We now eliminate the dependence of the total computation time on the cluster size $\|\mathcal{C}\|$. For this we assume that on the $\text{QC}_{\mathcal{C}}$ we simulate a quantum logic network with the network logical depth $D_{\mathcal{N}}$. Now, we give an upper bound on $\|\mathcal{C}\|$ as a function of n and $D_{\mathcal{N}}$. As displayed in Fig. 2.2, a single CNOT gate has height 3 and width 6 on the cluster $\mathcal{C}$. Here we do not count the output qubits of the gates since they also form the input qubits of the gates in the next slice. As in Fig. 2.2, the rotation has height 1 and width 4, if the output qubit is again not counted for the width. The wires for the logical qubits on the cluster can be arranged with distance 2. Each set of parallelized gates will at most require a slice of width 6 on the cluster. The circuit as a whole requires an additional slice of width 1 for the output. A swap gate that is composed of three CNOT gates, requires an array of 3×18 qubits on the cluster. If a general CNOT gate on the cluster were composed of a

next-neighbour CNOT gate and swap gates (in practice it is not), then it would require at most an array of [height] $\times$ [width] =$[2n-1] \times [18(n-2)+6]$ qubits. Hence, a CNOT gate would, to leading order, consume at most $36n^2$ cluster qubits. Each rotation would require at most –in the worst case where on the network it could not be performed in parallel with other gates– a slice of width 4 on the cluster, so it consumes, to leading order, at most $8n$ cluster qubits. The total number of gates in the network is bounded from above by $n\, D_{\mathcal{N}}$. The simulation of each gate costs at most $\max(36n^2, 8n) = 36n^2$ cluster qubits. Hence, the size of the required cluster is bounded by

$$\|\mathcal{C}\| \leq 36n^3 D_{\mathcal{N}}. \tag{3.111}$$

If we now use the assumption about a good quantum algorithm that the logical depth scales polynomially in the number of qubits n,

$$D_{\mathcal{N}} = c\, n^p, \tag{3.112}$$

and insert (3.111) and (3.112) into (3.110), we obtain

$$T_{\text{comp}} \leq D\, \Delta_Q \left(\left[1 + \frac{\Delta_{cl}}{\Delta_Q} (4 + \log 9c) \right] + \frac{\Delta_{cl}}{\Delta_Q} (p+4) \log n \right). \tag{3.113}$$

From a practical point of view, we find that the logarithmic corrections –even for numbers n of logical qubits in the range of 10^5– play a minor role since they are suppressed by the ratio Δ_{cl}/Δ_Q. We could plug in some typical numbers, say $\Delta_Q = 1\ \mu$s, $\Delta_{cl} = 1$ ns, $p = 3$ and $n = 10^5$, to obtain $\Delta_{cl}/\Delta_Q\, (p+4) \log n \approx 0.12$ (or ≈ 0.24 for $n = 10^{10}$).

The spatial overhead $\|\mathcal{C}\|$ is polynomial in the number n of logical qubits. But, if one adopts this more practical viewpoint one may not be satisfied by the mere result that the spatial overhead scales polynomially, but might want to know what the scaling power actually is. Above we found that $\|\mathcal{C}\|$ scales with the $(p+3)$th power of n. However, in the above argument, we focused on the computation time where the precise value of exponent for the spatial scaling did not play an important role, and thus have been extremely wasteful with spatial resources. A more careful discussion yields a more favorable scaling of the spatial overhead.

From a strict scaling point of view, we find in (3.113) that the computation time is no longer equal to the logical depth D times a constant, but there are $\log n$ -corrections due to the classical processing. This is, as the above numbers illustrate, of little relevance for practical purposes. The classical processing can be parallelized to such a degree that it increases the total computation time only marginally.

3.5.3 Temporal complexity of computing the circuit layout

The time that it takes for a classical computer (i.e. a compiler) to translate an algorithm into a machine-specific set of operations (i.e. the machine code) is usually not regarded as to count for the temporal complexity of that algorithm. For quantum logic networks this

viewpoint is certainly justified because there the complexity to compute the circuit layout is well understood and known not to exceed the complexity of the quantum logic network itself.

A similar result needs to be established for the $\mathrm{QC}_{\mathcal{C}}$. We must exclude the possibility that for the $\mathrm{QC}_{\mathcal{C}}$ the algorithmic complexity of a quantum computation is shuffled from the genuine quantum part of the computation to the classical pre-processing, and that this classical pre-processing may be exponentially hard. As will be shown below, such a case does not occur. All the classical pre-processing can be done in polynomial time.

To see this, we assume that the quantum algorithm on n logical qubits is given as a sequence of $\|\mathcal{N}\|$ elementary gates. For good quantum algorithms, $\|\mathcal{N}\|$ is polynomial in n, as is $\|\mathcal{C}\|$, the number of physical qubits in the cluster $\mathcal{C}$ required to run the algorithm (see Section 3.5.2).

The layout of the measurement pattern requires to assign $\|Q_0\|$ measurement bases and $\|\mathcal{C}\backslash Q_0\|$ angles. Creating the pattern is for itself not a problem since it can be obtained by patching together the measurement patterns of the elementary gates which are available in block form. The temporal complexity for this step is thus $O(\|\mathcal{C}\|)$.

To obtain the byproduct images we introduce $\|\mathcal{N}\|$ vertical cuts $\mathcal{O}_i,\ i = 1, .., \|\mathcal{N}\|$ to the network, one after each gate (such that $\mathcal{O}_{\|\mathcal{N}\|} = \Omega$) and compute the $2n \times 2n$-matrices $C(\mathcal{N}_{\mathcal{O}_i \to \Omega})$ for $i = 1, .., \|\mathcal{N}\| - 1$, starting with $i = \|\mathcal{N}\| - 1$. The operational effort for this is of the order $O(n^3\|\mathcal{N}\|)$. By use of these matrices the byproduct images for cluster qubits $k \in \mathcal{C}\backslash Q_{0,z}$ can now be obtained via (3.13), which requires $O(n^2)$ elementary operations per byproduct image. The way to obtain the byproduct images $\mathbf{F}_g$ of the gates is the same. For $k \in Q_{0,z}\backslash O$ at most four byproduct images have to be added in (3.72), which requires $O(n)$ operations. The computation of $\mathbf{F}_k$ for $k \in O$ is trivial. Thus, to compute a byproduct image requires at most $O(n^2)$ operations per cluster qubit or gate such that the complexity to compute all of them is at most $O\left(n^2(\|\mathcal{C}\| + \|\mathcal{N}\|)\right)$.

The backward- and forward cones of the cluster qubits $k \in \mathcal{C}$ are computed using the temporal ordering of gates in a sequence representing the quantum logic network and the cone test (3.25). The number of cone tests that have to be performed in each case is $\|\mathcal{N}\|(\|\mathcal{N}\| - 1)/2$ where the computational effort for each test scales like $O(n)$. Thus, the complexity of this step is $O(n\|\mathcal{N}\|^2)$.

The forward cones generate the anti-reflexive semi ordering "$\prec$". The semi ordering can be computed from them in $O(\|\mathcal{C}\|^5)$ steps.

For each set Q_t there have to be $\|Q^{(t)}\| \leq \|\mathcal{C}\|$ test of the relation $j \prec k,\ j \in Q^{(t)}$ performed to check whether some qubit $k \in Q^{(t)}$ is in Q_t. Also, $\|Q^{(t)}\|$ qubits have to be checked for each Q_t. At most $\|\mathcal{C}\|$ sets Q_t exist such that the operational effort to obtain the these sets is $O(\|\mathcal{C}\|^3)$.

As far as the stated upper bounds are conclusive, it looks as if the computation of the anti-reflexive semi ordering is the toughest part. However, as elementary a relation between the cluster qubits "$\prec$" is, for the conversion of a quantum logic network into a $\mathrm{QC}_{\mathcal{C}}$-algorithm it needs not be computed. Please note that the semi ordering is finally only needed to compute the sets Q_t via (3.8). But instead of computing "$\prec$" from the forward cones and the sets Q_t from "$\prec$", the sets Q_t can also be computed from the forward cones

directly. For this, please note that $\exists j \in Q_t |\ j \prec k \in Q^{(t)} \Longleftrightarrow \exists j' \in Q_t |\ k \in \mathrm{fc}(j')$. The direction "$\Longleftarrow$" is obvious with $j = j'$. The opposite direction, "$\Longrightarrow$", holds by an argument analogous to the one justifying (3.97). In fact, statement "$\Longrightarrow$" is the same as (3.97) with $Q^{(1)}$ replaced by $Q^{(t)}$. Thus, eq. (3.8) can be replaced by

$$Q_t = \left\{k \in Q^{(t)} |\ \neg\exists j' \in Q^{(t)} :\ k \in \mathrm{fc}(j')\right\}. \tag{3.114}$$

To set the algorithm angles via (3.78), (3.79) requires at most $\|\mathcal{C}\| + \|\mathcal{N}\|$ additions per angle and there are at most $\|\mathcal{C}\|$ such angles. Hence, in total it takes $O\left(\|\mathcal{C}\|(\|\mathcal{C}\| + \|\mathcal{N}\|)\right)$ operations to set them. Finally, to initialize the information flow vector via (3.76) requires $O\left(n(\|\mathcal{C}\| + \|\mathcal{N}\|)\right)$ operations. Thus we see that all classical processing requires only a polynomial overhead of elementary operations and can therefore be done in polynomial time.

3.6 Quantum algorithms and graphs

In this section we relate $\mathrm{QC}_{\mathcal{C}}$-algorithms to graphs. We do this by considering non-universal graph states suited for the specific algorithm in question. For the $\mathrm{QC}_{\mathcal{C}}$, the Clifford part of each algorithm can be removed. We show that a mathematical graph comprises all the information that needs to be kept of the Clifford part.

While the network formulation of a quantum algorithm is given as a sequence of quantum gates applied to a fiducial input state, the $\mathrm{QC}_{\mathcal{C}}$-version of a quantum algorithm is specified by a measurement pattern on the universal cluster state plus the structure [10] for the processing of the measurement outcomes.

To motivate the considerations of this section, note that the measurement pattern is, in the simplest case, just a copy of the network layout to the substrate cluster state, imprinted by the measurements. As such it contains information about the precise location of the gate simulations and about the way the "wires" connecting the gates are bent around. These are all details of the realization of an algorithm but do not belong to the description of the algorithm itself. Thus, the measurement pattern introduces a large amount of redundancy into the description of a $\mathrm{QC}_{\mathcal{C}}$-algorithm. This redundancy may be reduced to a large extent by allowing for non-universal, algorithm-specific quantum resources.

Clearly, at this point one has to specify how special the algorithm-specific resource is allowed to be. Obviously it would make no sense to take the quantum output of the entire network as the required quantum resource and to regard the subsequent readout measurements as the algorithm. Here, we allow for any graph state [53], (2.20) as the quantum resource. Graph states are easy to create, e.g. via unitary networks or from cluster states via measurements.

To allow for an algorithm-specific graph state as the quantum resource of a $\mathrm{QC}_{\mathcal{C}}$-computation reduces the redundancy of both the description and the realization of a quantum algorithm. This can easily be seen from the material presented in Section 2.2.3. All the cluster qubits $q \in \mathcal{C}\backslash\mathcal{C}_N$ can be get rid of either by measuring them in the σ_z-eigenbasis or equivalently by not placing them initially into their positions at all. The remaining

state on the sub-cluster $\mathcal{C}_N$ is again a cluster state. Hence it is also a graph state. It is less redundant and no longer universal.

But we can go further. Not only the qubits measured in the σ_z-eigenbasis may be removed from the cluster but instead all those qubits of which one of the Pauli operators σ_x, σ_y or σ_z is measured, i.e. all the qubits which form the set Q_0. The state of the unmeasured qubits that emerges after the measurement of the cluster qubits in Q_0 is again (local equivalent to) a graph state.

This may be seen as follows. First note that the operators $\sigma_x^{(a)} \bigotimes_{b\in V} \left(\sigma_z^{(b)}\right)^{\Gamma_{ab}}$ which appear in (2.20) form a stabilizer of the state $|\phi\{\kappa\}\rangle_G$. The generator of the stabilizer contains $|\mathcal{C}|$ elements for a state of $|\mathcal{C}|$ qubits. After all the qubits $q \in Q_0$ have been measured, the resulting state $|\Psi\rangle_{\mathcal{C}\backslash Q_0}$ of the $|\mathcal{C}\backslash Q_0|$ unmeasured qubits is again described by a stabilizer of the form

$$\bigotimes_{i=1}^{|\mathcal{C}\backslash Q_0|} \left(\sigma_x^{(i)}\right)^{X_{a,i}} \left(\sigma_z^{(i)}\right)^{Z_{a,i}} |\Psi\rangle_{\mathcal{C}\backslash Q_0} = \pm|\Psi\rangle_{\mathcal{C}\backslash Q_0} \qquad \forall a = 1\,..\,|\mathcal{C}\backslash Q_0|, \tag{3.115}$$

with two $|\mathcal{C}\backslash Q_0| \times |\mathcal{C}\backslash Q_0|$-matrixes X and Z, for which $X_{a,i}, Z_{a,i} \in \{0,1\}$. The $|\mathcal{C}\backslash Q_0| \times 2|\mathcal{C}\backslash Q_0|$-compound matrix $(X|Z)$ [52] is called the generator matrix of the stabilizer for $|\Psi\rangle_{\mathcal{C}\backslash Q_0}$. The state $|\Psi\rangle_{\mathcal{C}\backslash Q_0}$ is uniquely determined by the generator of its stabilizer.

The state $|\Psi\rangle_{\mathcal{C}\backslash Q_0}$ can thus be regarded as a $[|\mathcal{C}\backslash Q_0|, 0, d]$-stabilizer code, with the distance d not specified. This state fulfills the assumptions of Theorem 1 in [57]. The cited theorem states that any stabilizer code over the alphabet $A = \mathbb{F}_{p^m}$ is [local unitary] equivalent to a graph code.

We now specialize to the case of our interest, $A = \mathbb{F}_{2^2}$. It follows from the above quoted theorem that the state $|\Psi\rangle_{\mathcal{C}\backslash Q_0}$ specified in (3.115) is local unitary equivalent to a graph state $|\phi\{\kappa\}\rangle_{G(\mathcal{C}\backslash Q_0, E_{\mathcal{C}\backslash Q_0})}$ (2.20). That is, the state $|\Psi\rangle_{\mathcal{C}\backslash Q_0}$ obtained in a $\mathrm{QC}_{\mathcal{C}}$-computation after the first round of measurements may as well be obtained from a graph state $|\phi\{\kappa\}\rangle_{G(\mathcal{C}\backslash Q_0, E_{\mathcal{C}\backslash Q_0})}$ via local unitary transformations; and the subsequent measurements may be performed as usual. Alternatively, one may use the graph state $|\phi\{\kappa\}\rangle_{G(\mathcal{C}\backslash Q_0, E_{\mathcal{C}\backslash Q_0})}$ directly, only modifying the measurement bases instead of performing the local rotations prior to the measurements. Thus, in a $\mathrm{QC}_{\mathcal{C}}$-computation with a special graph state as the quantum resource and the first measurement round omitted, the way of processing the classical information is the same as in a $\mathrm{QC}_{\mathcal{C}}$-computation with a universal resource and the first measurement round performed.

The graphs associated with states (3.115) are in general not unique [57]. A constructive way to obtain graphs on $\mathcal{C}\backslash Q_0$ from $G(\mathcal{C}, E_{\mathcal{C}})$ and the measurement bases of the qubits in Q_0 has been described in [58].

Now note that the measurement of the qubits in Q_0 realize the Clifford part of a quantum circuit. The fact that we can reduce the quantum resource by these qubits means that *we can remove from each quantum algorithm its Clifford part.* This represents, in a way, an extension to the Knill-Gottesman-Theorem [21], stating that a quantum

computation that consist only of quantum input state preparation in the computational basis, unitary gates in the Clifford group, measurement of observables in the Pauli group, and gates in the Clifford group conditioned on the outcomes of such measurements, may be simulated efficiently classically and thus requires no quantum resources at all.

With only a single non-Clifford operation in the circuit, such as a one-qubit rotation about most axes and angles, the efficient classical formalism upon which the Gottesman-Knill theorem rests can no longer be applied. The $\mathrm{QC}_{\mathcal{C}}$-construction, on the other hand, is not affected by this. Each quantum network algorithm in question may be reduced by its Clifford part. Only the non-Clifford gates require quantum resources. The price is that the universal quantum resource, the cluster state, is changed into a non-universal, algorithm-specific resource –a graph state (2.20)– on fewer qubits. The Clifford part of the network algorithm specifies the corresponding graph.

In conclusion, instead of describing a quantum algorithm as a network of gates applied to some fiducial input state, a quantum algorithm may (arguably more effectively) be characterized by a graph specifying the quantum resource and the structure [10] for the processing of the measurement outcomes.

3.7 Discussion

The discussion about the logical depth of certain algorithms with the $\mathrm{QC}_{\mathcal{C}}$ in Section 3.5 showed that there exist ways of information processing with the $\mathrm{QC}_{\mathcal{C}}$ which cannot be explained adequately in network model terms. This made a computational model appropriately describing the $\mathrm{QC}_{\mathcal{C}}$ desirable. The computational model underlying the $\mathrm{QC}_{\mathcal{C}}$ that we found does not seem to have much in common with the network model. It is based on objects of a different sort which require an interpretation. In this section, we attempt to clarify the role of the binary valued information flow vector $\mathbf{I}(t)$ and that of the stepwise measured quantum state.

What is the role of the information flow vector $\mathbf{I}(t)$? In every computational step except the final one the information flow vector $\mathbf{I}(t)$ is completely random. So one might ask whether it contains information at all. It does, since in every step except the last one it tells what has to be done next. After the final computational step at time $t_{\max}$ the quantity $\mathbf{I}(t_{\max})$ contains the result of the computation. Thus, the quantity $\mathbf{I}(t)$ has a meaning in every computational step. For a "computational meaningful" quantity we have given a precise criterion, the invariance of the respective defining relation under the set of symmetry transformations (3.47, 3.48). In this way, we have also clarified the possibly puzzling observation that a quantity might be computationally meaningful despite its random value. The information to steer a $\mathrm{QC}_{\mathcal{C}}$-computation and to read off the computational result is all comprised by the information flow vector $\mathbf{I}(t)$. In this sense, it represents the algorithmic information in the described scheme of quantum computation.

What is the role of the stepwise measured quantum state? To see that explicitly, let us consider the scenario where a quantum computation is halted in the middle and continued at a later time by another person who only knows which steps of the computation are left

to perform but does not know what has been done so far. In analogy to a teleportation protocol where both the result from the Bell measurement and the quantum state at the receivers side are required to reconstruct the initial state, the halted computation can be successfully completed only if both pieces –the intermediate information flow vector $\mathbf{I}(t_i)$ and the half-measured quantum state– are stored until the computation proceeds. Thus, the quantum state cannot be neglected just because it does not appear in the formal description of the computational model. The quantum correlations in the stepwise measured state are what basically enables the described way of quantum information processing. However, the role of this state is a passive one. It serves as a resource that is used up during the course of computation.

Let us at the end of this discussion come back to the role which the randomness of the individual measurement results plays for the $\mathrm{QC}_{\mathcal{C}}$. It may surprise that a set of classical binary numbers represents the algorithmic information in a scheme of quantum computation. In the network model the quantum state (of the quantum register) is usually considered to represent the processed information. For the $\mathrm{QC}_{\mathcal{C}}$, the situation is different. There, the randomness of the individual measurement results makes it necessary to store classical steering information. The need to process this information has called for a novel information carrying quantity. What, in a network-like description of the $\mathrm{QC}_{\mathcal{C}}$, has been regarded as a mere byproduct turns out to be the central quantity of information processing with the $\mathrm{QC}_{\mathcal{C}}$.

Chapter 4

Fault-tolerant quantum computation with the $\mathrm{QC}_{\mathcal{C}}$

In this chapter we prove that the $\mathrm{QC}_{\mathcal{C}}$ can be operated fault-tolerantly. Further, we describe the concept of checksums for the $\mathrm{QC}_{\mathcal{C}}$ which may become an element in future efficient techniques for fault-tolerant $\mathrm{QC}_{\mathcal{C}}$-computation.

We start with a brief review on fault-tolerant quantum computation in the network model in Section 4.1. In Section 4.2 we derive the error model for the physical errors on which we will base the subsequent investigation. We also expect this error model to be more generally useful. In Section 4.3 we give the proof that fault-tolerant $\mathrm{QC}_{\mathcal{C}}$-computation is possible. The technique used is to trace back the fault-tolerance of the $\mathrm{QC}_{\mathcal{C}}$ to that of a network quantum computer with next-neighbor and local gates [44, 64]. The purpose of the fault-tolerance proof presented here is to demonstrate that there *exist* nonzero error thresholds for the $\mathrm{QC}_{\mathcal{C}}$. In Section 4.4 we describe how checksums arise in the context of the $\mathrm{QC}_{\mathcal{C}}$ and how they reduce the effect of noise. We illustrate the concept of checksums in the example of an encoded CNOT-gate for the seven qubit Steane code.

4.1 Fault-tolerant quantum computation in the network model

In the realization of a quantum computer as a physical system one is, among other things, faced with the following problem: One wants the quantum system used for computation to be well isolated from the environment, since the interaction with environmental degrees of freedom introduces decoherence. This would drive the quantum computer towards the classical regime, thereby degrading its power. On the other hand, the system cannot be so well isolated that it could not be accessed anymore by control fields to steer the computation. In practice, any quantum computation will therefore be erroneous where both the interaction with the environment and imperfect operation contribute to the errors.

A priori, the regimes of perfect and imperfect operation are different, however small the error per gate cycle is. For any fixed values of the gate and memory error rate,

the quantum register will end up close to a totally depolarized state if the computation only is long enough. Therefore it was a great discovery from the perspective of both the practical construction of a quantum computer and the theory of quantum computation that quantum errors can be corrected, and further that any quantum computation can be stabilized against the degradation of quantum coherence, provided the error rate is below a certain threshold. Fault-tolerant quantum computation is thus possible.

Specifically, it was shown by Calderbank and Shor [35], [36], and by Steane [37, 38] that logical qubits can be protected from errors if they are encoded in a larger number of physical qubits. Upon these encoded qubits one repeatedly performs a recovery operation. Therein, one measures a number of observables which reveal no information about the encoded state but identify the possible "error" that the encoded qubit has been affected by. It has to be taken into account that the recovery procedure may itself introduce errors. The error is corrected in a subsequent multi-local unitary operation, conditioned on the measured error syndrome. Fault-tolerant quantum computation is more complicated. If the computation is performed in a sequence of steps of encoding, decoding and gate operation, then the logical qubits are unprotected during the action of the gate, which will lead to unrecoverable errors. Instead, the gates need to be performed on encoded qubits. These encoded gates will also introduce errors. Therefore, the encoded gates and the error recovery procedures have to be constructed in such a way that the errors introduced by the gate operations and the memory errors can be identified and accounted for. Encoded gates and recovery procedures allowing for fault-tolerant quantum computation have been given by Shor [40] and by Gottesman [41, 42].

In these papers it was established that fault-tolerant quantum computation is possible in principle. The requirements for the realization of a quantum computer are, nevertheless, extremely demanding. The error thresholds obtained in proofs for fault-tolerant quantum computation are at the level of 10^{-6} for both the gate- and memory error [43, 44]. However, in simulations it was found that error thresholds may be as high as 10^{-3} [50, 65]. Recently it was shown by Dür and Briegel [66], using entanglement purification, that the thresholds for fault-tolerant quantum computation can be made to depend solely on the quality of local operations on 2^5-dimensional quantum systems. High fidelity interaction between these systems can be generated from extremely noisy interaction via purification of Bell states followed by teleportation.

As has been shown by Kitaev [67], a two-dimensional quantum system with anyonic excitations can be considered as a quantum computer. Unitary transformations are performed by moving the excitations around another, and measurements by fusion of pairs of anyons. This form of quantum computation is fault-tolerant by its physical nature: the ground state of the system coincides with the protected code space and is separated from the lowest excited states by a finite energy gap. The quantum codes naturally exhibited by these systems are called surface codes and have been studied further e.g. in [68] and [69]. In [68] procedures for encoding, measurement and fault-tolerant universal quantum computation with surface codes are discussed, and an estimate for the error threshold of about 10^{-4} is given. Further, it is shown that the capabilities of the system to store quantum information can be related to a three-dimensional $\mathbb{Z}_2$-gauge theory on a lattice that

exhibits an ordered to disordered phase transition at a non-zero critical value of the error rate. This phase transition has been analyzed in detail in [69].

Let us now give a brief account on techniques for quantum error correction and fault-tolerant quantum computation within the network model.

Coding and quantum error correction. The main idea is to protect quantum information from decoherence by coding, like in the classical case. That is, via an encoding operation $U_{\rm enc}$ one encodes a number k of qubits in a larger number n of qubits,

$$|\psi\rangle_{1..k} \otimes |0...0\rangle_{k+1..n} \longrightarrow U_{\rm enc}\,|\psi\rangle_{1..k} \otimes |0...0\rangle_{k+1..n} = |\Psi\rangle_{1..n}. \tag{4.1}$$

Therein, $|\psi\rangle_{1..k}$ is the bare and $|\Psi\rangle_{1..n}$ the respective encoded state. The encoded multi-qubit state may be decoded at the end of the storage process via the inverse of the coding transformation, $U_{\rm dec} = {U_{\rm enc}}^{-1}$. While being encoded, the protected quantum state may be monitored in such a way that information can be obtained about which errors have occurred without affecting the encoded information. The effect of noise accumulating over a fixed time interval T, say, may be described by a superoperator $\mathcal{E}$ acting on the density operator representing the encoded quantum state. The error channel $\mathcal{E}$ in its Kraus representation [70] reads

$$\rho_{1..n} \longrightarrow \mathcal{E}(\rho_{1..n}) = \sum_i E_i \rho_{1..n} E_i^\dagger, \tag{4.2}$$

with the trace-preserving condition

$$\sum_i E_i^\dagger E_i = \mathbb{1}. \tag{4.3}$$

Therein, the Kraus operators E_i form the set of errors $E = \{E_i\}$.

To undo the action of the error channel on the encoded data, a recovery operation $\mathcal{R}$ is subsequently applied,

$$\rho'_{1..n} \longrightarrow \mathcal{R}(\rho'_{1..n}) = \sum_s R_s \rho'_{1..n} R_s^\dagger, \tag{4.4}$$

where the operators R_s are again constrained by a trace-preserving condition

$$\sum_s R_s^\dagger R_s = \mathbb{1}. \tag{4.5}$$

For a proper recovery operation $\mathcal{R}$ for the set E of errors we require that

$$\mathcal{R} \circ \mathcal{E}(|\Psi\rangle\langle\Psi|) = |\Psi\rangle\langle\Psi|, \tag{4.6}$$

where $|\Psi\rangle$ is an arbitrary state in the code space. That is, the recovery superoperator $\mathcal{R}$ undoes $\mathcal{E}$ if $\mathcal{E}$ is acting on the code space.

Clearly, the choice of the recovery procedure depends on the set E of expected errors, and not for every error superoperator $\mathcal{E}$ there will exist a procedure $\mathcal{R}$ to invert it. It

may now occur that an error superoperator $\mathcal{E}'$ realized in nature is very close such a noise superoperator $\mathcal{E}$ which is invertible on the code space. If we apply the recovery procedure $\mathcal{R}$ after the action of $\mathcal{E}'$, we may still get a high success probability for the recovery. Higher than if we had done nothing.

For an error channel $\mathcal{E}(E)$ to be correctable there exist the following conditions [39]:

$$\langle\Psi_i|E_a^\dagger E_b|\Psi_j\rangle = 0 \qquad \forall i \neq j, \forall\, E_a, E_b \in E,\ |\Psi_i\rangle, |\Psi_j\rangle \in \mathcal{H}_C, \tag{4.7a}$$

$$\langle\Psi_i|E_a^\dagger E_b|\Psi_i\rangle = \langle\Psi_j|E_a^\dagger E_b|\Psi_j\rangle \qquad \forall\, E_a, E_b \in E,\ |\Psi_i\rangle, |\Psi_j\rangle \in \mathcal{H}_C. \tag{4.7b}$$

The condition (4.7a) says that after the action of the error channel different codewords must remain distinguishable. Condition (4.7b) ensures that during the recovery procedure we do not learn anything about the encoded state.

The conditions (4.7) are necessary and sufficient. The proof for their necessity is as follows: Taking the expectation value with $|\Psi\rangle$ on both sides of equation (4.6) and inserting the definitions (4.2) and (4.4) of $\mathcal{E}$ and $\mathcal{R}$, we obtain $\sum_{i,s} |\langle\Psi|R_sE_i|\Psi\rangle|^2 = 1$. In this equation, we may now use the Schwartz inequality and the constraints (4.3), (4.5) to estimate the l.h.s., and obtain $\sum_{i,s} |\langle\Psi|R_sE_i|\Psi\rangle|^2 \leq \sum_{i,s}\langle\Psi|\Psi\rangle\langle\Psi|E_i^\dagger R_s^\dagger R_sE_i|\Psi\rangle = 1$. Thus, the Schwartz inequalities must all hold as equalities. This is possible only if $R_sE_i|\Psi\rangle = \lambda_{i,s}(|\Psi\rangle)|\Psi\rangle$ for all $E_i \in E$, for all s and for all $|\Psi\rangle \in \mathcal{H}_C$. Further, because of linearity of R_s and E_i, the eigenvalues $\lambda_{i,s}$ cannot depend on $|\Psi\rangle$, such that

$$R_sE_i|\Psi\rangle = \lambda_{i,s}|\Psi\rangle, \quad \forall\, E_i \in E, \forall\, s, \forall\, |\Psi\rangle \in \mathcal{H}_C. \tag{4.8}$$

Now, following [39], we evaluate for two arbitrary codewords $|\Psi_i\rangle$, $|\Psi_j\rangle$ the scalar product $\langle\Psi_i|E_a^\dagger E_b|\Psi_j\rangle$. We find that $\langle\Psi_i|E_a^\dagger E_b|\Psi_j\rangle = \langle\Psi_i|E_a^\dagger \mathbb{1} E_b|\Psi_j\rangle = \sum_s\langle\Psi_i|E_a^\dagger R_s^\dagger R_sE_b|\Psi_j\rangle$, with (4.5). Therefore, using (4.8), it follows that $\langle\Psi_i|E_a^\dagger E_b|\Psi_j\rangle = \langle\Psi_i|\Psi_j\rangle\sum_s \lambda_{a,s}^\dagger\lambda_{b,s}$, and with $\alpha_{ab} := \sum_s \lambda_{a,s}^\dagger\lambda_{b,s}$ one obtains

$$\langle\Psi_i|E_a^\dagger E_b|\Psi_j\rangle = \alpha_{ab}\delta_{ij} \tag{4.9}$$

The conditions (4.7) for reversibility of a set E of errors can be read off directly from (4.9). For the opposite direction of the proof, i.e. that conditions (4.7) are sufficient, see [39]. Note that in order to derive the conditions (4.7) we have required the existence of a recovery operation, i.e. that $\mathcal{R}$ is a physical operation (4.4), (4.5) and acts as it is supposed to, (4.6), but have not assumed a particular form of it.

Sometimes it is sufficient to detect errors instead of identifying them. An example for such a situation is ancilla preparation. If an error in the ancilla is detected, then the ancilla is discarded and prepared anew. Error detection is successful if each error E_a from a set E is distinguishable from the identity or acts like the identity on the prepared ancilla state. Thus the condition for error detection is as (4.7), with the error E_b replaced by the identity.

Stabilizer Codes. Let us now describe a particular class of quantum codes, the stabilizer codes, and illustrate the functioning of the corresponding error recovery procedures. In order to simplify the discussion, we assume that the error superoperator $\mathcal{E}$ is diagonal in

the Pauli basis, i.e. that all E_i are Pauli operators $E_i = \bigotimes_{a=1}^{n} \left(\sigma_x^{(a)}\right)^{x_a} \left(\sigma_z^{(a)}\right)^{z_a}$, $x_a, z_a \in \{0,1\}$ $\forall a$. The error channel $\mathcal{E}$ then takes the form

$$\mathcal{E}(|\Psi\rangle\langle\Psi|) = \sum_{E_i \in E} p(E_i)\, E_i\, |\Psi\rangle\langle\Psi|\, E_i^\dagger, \tag{4.10}$$

where $p(E_i)$ is the probability for the error E_i to occur. In (4.2), the description of the general error channel, the error probabilities $p(E_i)$ have been absorbed in the error operators E_i. For the special case of Pauli operators, we have the normalization condition $E_i^\dagger E_i = \mathbb{1}$, $\forall E_i$ and therefore write the error probability separately. The trace-preserving constraint (4.3) translates into $\sum_{E_i \in E} p(E_i) = 1$.

An $[n, k, d]_2$ stabilizer code is a quantum code encoding k qubits into n where the code space is the common eigenspace of a set of $(n-k)$ independent commuting operators G_l, the generators of the code stabilizer S. Each codeword $|\Psi_m\rangle$ obeys the eigenvalue equations

$$|\Psi_m\rangle = G_l|\Psi_m\rangle, \;\; \forall\, m = 0, .., 2^k - 1, \; \forall\, l = 1, .., n-k \tag{4.11}$$

The parameter d is the distance of the code. A code with distance d can correct $\lfloor d/2 \rfloor$ simultaneous errors from the set E. For $d = 3$ one usually assumes that the set E of errors consists of all one-qubit spin-flip, phase-flip and the combined spin+phase-flip errors, $E = \{\sigma_x^{(i)}, \sigma_z^{(i)}, \sigma_y^{(i)} |\, i = 1, .., n\}$. If each qubit in the encoded state decoheres due to interaction with its own environment, which seems a reasonable assumption for data storage, then the one-qubit errors are the most likely ones.

The generators $\{G_l |\, l = 1, .., n-k\}$ of the code stabilizer S are also chosen to be Pauli operators,

$$G_l = \bigotimes_{i=1}^{n} \left(\sigma_x^{(i)}\right)^{x_i(G_l)} \left(\sigma_z^{(i)}\right)^{z_i(G_l)}, \quad x_i(G_l), z_i(G_l) \in \{0,1\}\; \forall i = 1, .., n. \tag{4.12}$$

The stabilizer generators mutually commute, $[G_l, G_m] = 0$, $\forall l, m = 1, .., n-k$. Further, each of the generators does either commute or anti-commute with each error operator from the set E. Therefore, to each error E_i there belongs a set of $n-k$ binary numbers $Sy_l(E_i)$,

$$E_i G_l = (-1)^{Sy_l(E_i)} G_l E_i. \tag{4.13}$$

They form the syndrome $\mathbf{Sy}(E_i)$ of the error E_i, $[\mathbf{Sy}(E_i)]_l = Sy_l(E_i)$.

In the error recovery procedure $\mathcal{R}$ for stabilizer codes first the operators G_l generating the code stabilizer are measured. This provides one with an error syndrome as defined in (4.13) for the individual errors. In the second step, conditioned on this syndrome a unitary correction operation is applied to the encoded quantum state. Formally, the recovery procedure reads

$$\mathcal{R} = \sum_{\mathbf{Sy} \in \{0,1\}^{n-k}} \left[U(\mathbf{Sy}) \prod_{l=1}^{n-k} \frac{\mathbb{1} + (-1)^{Sy_l} G_l}{2} \right]. \tag{4.14}$$

Therein, the brackets "[..]" indicate a superoperator, i.e. the sum is over superoperators, not operators.

In each individual run of the recovery procedure we obtain a set of measurement outcomes $\{s_l \in \{0,1\} | \, l = 1, .., n-k\}$ where s_l describes the eigenvalue $(-1)^{s_l}$ obtained in the measurement of G_l. A posteriori, that is knowing the measurement results, the recovery procedure may be described as a series of projections. Let us now investigate how these projections act on each term $E_i \, |\Psi\rangle\langle\Psi| \, E_i^\dagger$ in the l.h.s. of (4.10):

$$\begin{aligned} \left(\prod_{l=1}^{n-k} \frac{\mathbb{1} + (-1)^{s_l} G_l}{2} \right) E_i |\Psi\rangle &= E_i \left(\prod_{l=1}^{n-k} \frac{\mathbb{1} + (-1)^{s_l + Sy_l(E_i)} G_l}{2} \right) |\Psi\rangle \\ &= \left(\prod_{l=1}^{n-k} \frac{1 + (-1)^{s_l + Sy_l(E_i)}}{2} \right) E_i |\Psi\rangle. \end{aligned} \tag{4.15}$$

Therefore, under the action of the projections only those error operators E_i persist for which $s_l = Sy_l(E_i), \ \forall l = 1, .., n-k$. All other error operators are annihilated. We see that the syndrome defined as an algebraic property of the errors in (4.13) can be measured. This measurement, as an irreversible quantum process, changes the encoded state in such a way that errors E_i of a subset of $E(\mathbf{Sy}) \subset E$ *are made to happen* while the other errors are eliminated. This process is known as *error discretization*. For successful error correction all errors in the remaining set $E(\mathbf{Sy})$, which are indistinguishable by the syndrome, are required to act identically on the code space. The condition (4.7) in stabilizer form thus reads [41],

$$\forall E_i, E_j \in E : \ (\exists G_l \in S \, \text{s.th.} \, [G_l, E_i E_j] \neq 0) \ \vee \ E_i E_j \in S \tag{4.16}$$

The notion of "error discretization" is particular illustrative if we describe decoherence as a unitary process acting on the pure state of the encoded quantum information and that of an environment. This unitary operation on a larger Hilbert space may deviate slightly from the identity operation on the "data register" times whatever operation on the environment, and entangle the quantum "data" state with the environmental degrees of freedom. In this picture, Pauli errors appear formally if we expand this unitary evolution in a basis of Pauli operators. This way, we obtain errors in linear combination where the coefficients associated with the individual errors may vary continuously. Now, under the action of the stabilizer measurement this continuous set of errors in superposition is mapped onto a much smaller one. After the projection there may still appear errors in linear combination, but only such ones which act identically on the code. Thus, from the initially continuous set there remains effectively only a single error. This single discrete error is chosen by the measurements at random, but it can be identified from the syndrome represented by the measurement outcomes.

In this thesis, we have chosen to describe the noise by a superoperator (4.2) in the Kraus representation, acting on quantum system without the environment. Such a description may always be obtained from the former if one traces over the environmental degrees of freedom. In the Kraus representation (4.2) of the error channel the set E of errors is discrete right from the beginning, as in the sum (4.2) there may appear at most 4^n errors

E_i. In the description chosen here the projective measurements select errors from a discrete set rather than inducing a discretization.

Vector space formulation for stabilizer codes. Let us state a further formulation of the error correction condition (4.7) using vectors in $\mathcal{V}(\mathbb{F}_2)$ [52]. In this language, the error correction condition for stabilizer codes, and the conditions for fault-tolerant error recovery and fault-tolerant quantum gates discussed subsequently can all be stated in a compact form.

Note that phase factors in front of error operators E_i have no physical effect, for if we transform $E_k \longrightarrow e^{i\varphi_k} E_k$, the error superoperator $\mathcal{E}$ (4.2) remains unchanged. Further, the bits of the error syndrome $\mathbf{Sy}(E_k)$ of an error E_k are specified by whether the operators E_k and G_l commute or anti-commute. This property is not changed by additional phases of either the errors or the stabilizer generators. Therefore, Pauli error operators which are equal up to a phase are grouped into equivalence classes. These equivalence classes themselves form a group, which is isomorphic to a vector space $\mathcal{V}(\mathbb{F}_2)$. The isomorphism is the same as the one (3.10) introduced in Section 3.2.4 to relate the byproduct images to the forward propagated byproduct operators,

$$\mathcal{I}: \ \mathcal{V}(\mathbb{F}_2) \ni \mathbf{F}_i^E \longrightarrow E_i = \prod_{a=1}^{n} \left(\sigma_x^{(a)}\right)^{[\mathbf{F}_{i,x}^E]_a} \left(\sigma_z^{(a)}\right)^{[\mathbf{F}_{i,z}^E]_a}. \tag{4.17}$$

The vectors in $\mathcal{V}(\mathbb{F}_2)$ onto which we map the errors E_i under $\mathcal{I}^{-1}$ we call the *error images* $\mathbf{F}_i^E$. We distinguish them from the byproduct images $\mathbf{F}_i$ corresponding to the forward propagated byproduct operators by the upper index "E". The fact that the error images $\mathbf{F}_i^E$ and the byproduct images $\mathbf{F}_i$ employ the same mathematical structures and are given similar names is no coincidence, as we shall see below.

We may also apply the isomorphism $\mathcal{I}^{-1}$ to the code stabilizer S, in particular to its generators G_l. In this way, we obtain the generator matrix G [52] of the code. The generator matrix G has $2n$ columns and $n-k$ rows, and consists of an x- and a z-block,

$$G = (X_G|Z_G). \tag{4.18}$$

Each row in G encodes a generator G_l of the stabilizer, $(\mathcal{I}^{-1}G_l)^T$. The x-part of G is for Pauli operators σ_x and the z-part is for Pauli operators σ_z. In both blocks, X_G and Z_G, the column index labels the qubits. Specifically, with the definition (4.12) of G_l, we identify

$$[X_G]_{l,i} = x_i(G_l), \ \ [Z_G]_{l,i} = z_i(G_l), \ \ \forall l = 1, .., n-k, \ \forall i = 1, .., n. \tag{4.19}$$

With these specifications, the Pauli operator formulation (4.16) of the condition (4.7) can now be translated straightforwardly into a vector space formulation,

$$\forall E_i, E_j \in E: \ \mathbf{Sy}(\mathbf{F}_i^E) + \mathbf{Sy}(\mathbf{F}_j^E) \neq 0 \ \vee \ \mathbf{F}_i^E + \mathbf{F}_j^E \in \mathcal{I}^{-1}(S). \tag{4.20}$$

Therein, all addition is modulo 2 and the syndrome $\mathbf{Sy}(\mathbf{F}_i^E)$ of the error E_i is, in accordance with (4.13), given by

$$\mathbf{Sy}(\mathbf{F}_i^E) = (Z_G|X_G) \ \mathbf{F}_i^E. \tag{4.21}$$

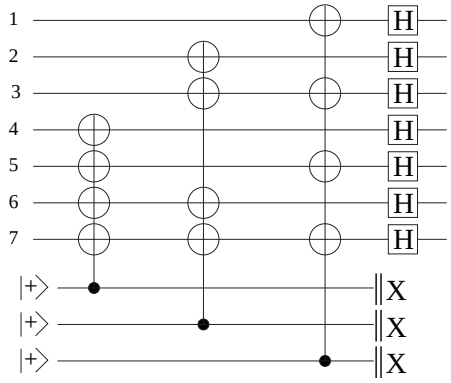


Figure 4.1: Stabilizer measurement circuit for the Steane code. Only the sub-circuit for the measurement of the first three generators of (4.22) are shown. To measure all six generators, the displayed circuit has to be repeated twice. Performing the circuit just once also results in an encoded Hadamard transformation. The stabilizer measurement requires ancilla qubits, which are each prepared in an eigenstate of σ_x and measured in the σ_x-eigenbasis. The circuit shown is not fault-tolerant.

The condition (4.20) is of quite general form. We will obtain the same type of condition for quantum error correction with imperfect operations, and very similar conditions for fault-tolerant gates on encoded qubits and for their simulation on the $\mathrm{QC}_{\mathcal{C}}$. What will change is the set E of considered errors.

For illustration, let us give the generator matrix of the 7-qubit Steane code [37], which encodes one qubit in seven and corrects all the one-qubit errors. The generator matrix is

$$G_{[7,1,3]} = \left(\begin{array}{ccccccc|ccccccc} 0&0&0&1&1&1&1&0&0&0&0&0&0&0\\ 0&1&1&0&0&1&1&0&0&0&0&0&0&0\\ 1&0&1&0&1&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&1&1&1\\ 0&0&0&0&0&0&0&0&1&1&0&0&1&1\\ 0&0&0&0&0&0&0&1&0&1&0&1&0&1 \end{array}\right). \tag{4.22}$$

For example, the generator G_1 corresponding to the first row of the generator matrix in (4.22) is

$$G_1 = \sigma_x^{(4)}\sigma_x^{(5)}\sigma_x^{(6)}\sigma_x^{(7)}. \tag{4.23}$$

The elementary cell of the network for error identification in the Steane code is displayed in Fig. 4.1. The shown circuit is for the illustration of stabilizer measurement only. It is not yet fault-tolerant, because it introduces more severe errors than it corrects for. The circuit needs to be applied twice in a row to measure the entire syndrome. In the first three measurements the generators involving Pauli spin-flip operators σ_x are measured, checking for σ_z-errors. The subsequent Hadamard transformations convert the so far undetected σ_x-errors into σ_z-errors which are subsequently detected in the second application of the

circuit. The Hadamard transformations also rotate the code space, and therefore have to be counteracted by a second step of local Hadamard transformations after the second half of syndrome measurements. The conditional spin- and phase-flips which restore the encoded state to the code space are not shown. They can be omitted anyway. It is sufficient to keep track of the error by forward propagation in an efficient classical computation running in parallel with the quantum computation [43] and adapt the subsequent non-Clifford gates accordingly. This has the advantage that it reduces the number of operations, and thereby also the errors.

Error thresholds and code concatenation. A reasonably realistic model for an error channel is the multi-local $SU(2)$-invariant channel

$$\mathcal{E}_{\mathrm{local},n}[\epsilon_1] = \bigotimes_{a=1}^{n} \mathcal{T}_1^{(a)}[\epsilon_1], \tag{4.24}$$

where

$$\mathcal{T}_1^{(a)}[\epsilon_1](\rho) = (1-\epsilon_1)\rho + \frac{\epsilon_1}{3} \sum_{i=x,y,z} \sigma_i^{(a)} \rho\, \sigma_i^{(a)}. \tag{4.25}$$

If the error superoperator $\mathcal{E}_{\mathrm{local},n}[\epsilon_1]$ is expanded into a basis of Pauli operators one finds that all 4^n possible Pauli errors occur. No code could correct for that. However, some errors are more likely than others. The probability for no error to occur is $O(1)$, the probability for a single one-qubit error is $O(\epsilon_1)$, for two independent one-qubit errors $O({\epsilon_1}^2)$, and so on. Thus, if the error probability ϵ_1 is small, $\epsilon_1 \ll 1$, then it is almost certain that at most one error occurs. These errors can be corrected for if a suitable code with distance 3 is used. If a two- or more-qubit error occurs then the error recovery procedure will most likely fail because the syndrome is misinterpreted. The probability for damage of the encoded qubit is thus $O({\epsilon_1}^2)$ whereas for the unprotected qubit it is $O(\epsilon_1)$. Therefore, if ϵ_1 is below a certain threshold, the error probability with coding will be smaller than without coding.

In such a situation, the code may be concatenated [41, 71]. That is, each qubit in the code is encoded again, and this procedure may be iterated several times. Let the probability for failure, i.e. for two or more errors, be $c\,{\epsilon_1}^2$ to leading order. If a second level of coding is used, the code will fail if two blocks of the first-level encoding are corrupted, and the probability for that is $c\,(c\,{\epsilon_1}^2)^2$. If N levels of concatenation are used, the probability for failure is $p_N = (c\,\epsilon_1)^{2^N}/c$ and thus gets small very rapidly if $c\,\epsilon_1 < 1$. At the same time the code size increases to n^N.

Error recovery with imperfect means. The error recovery procedure $\mathcal{R}$ needs to satisfy two requirements. First, it needs to act as the identity on the code space, i.e. the encoded Pauli operators $\overline{X}^{(a)}, \overline{Z}^{(a)}$, for all $a = 1, .., k$ need to be mapped onto themselves, and the stabilizer operators need to be mapped onto stabilizer operators. Second, the procedure $\mathcal{R}$ needs to correct for errors. Among them are the errors which occur during data storage. But these are not the only errors. As we can see from the error identification circuit

displayed in Fig. 4.1, quantum error correction is itself a small quantum computation, and errors may be introduced by imprecise gate operation.

So, what happens if the operations intended to correct errors are erroneous themselves? Do they not introduce more errors than they correct for? For the circuit displayed in Fig. 4.1 this is indeed the case. However, with a careful –and more complex– design it can be achieved that the recovery procedure identifies all the errors, those which occur during data storage and those it introduces itself, and thereby becomes fault-tolerant.

We shall give a detailed model for errors, due to both decoherence and imperfect operation, further below, but let us –as we deal here with errors introduced by imperfect operations– briefly sketch such a model now. Each imperfect gate operation may be modeled by the perfect gate preceded or succeeded by an error channel [50]. If the gate acts on more than one qubit, the error channel should not be a product of local channels as e.g. the local $SU(2)$-invariant channel (4.24) but instead contain genuine multi-qubit errors to lowest order in the error rate. For simplicity, we assume again that the corresponding error superoperator is diagonal in the Pauli basis. In this way, we can speak of Pauli-errors, some of them being local and some multi-local, and discuss each of these errors separately.

As in any quantum computation, in an error recovery procedure errors propagate, and for the propagation of errors in the Pauli group hold the same propagation relations as for the byproduct operators, (2.51), (2.52), (2.53) and (2.48). Specifically, if such an error propagates through a gate in the normalizer of the Pauli group, it is conjugated under that gate, and thereby remains in the Pauli group. If a Pauli error operator propagates through the measurement of an observable in the Pauli group, it flips the measured eigenvalue $\lambda_l = \pm 1$ if it anti-commutes with the measured observable or otherwise leaves it unchanged.

To track the errors through the circuit and to identify the signature they leave in the error syndrome, we consider the circuit of all the participating qubits. That is, the n data qubits and further n_a ancilla qubits for the ancilla state preparation and -verification and for the syndrome measurement. The forward propagation of each Pauli operator is followed in a classical computation from the location where it occurs to a vertical cut Ω after the recovery procedure. That is, for errors on the data qubits the errors are propagated into the domain of the subsequent storage errors, and errors which affect the ancilla qubits are propagated ante the ancilla measurements.

Due to the non-trivial error propagation, we distinguish between *physical* and *logical* errors. The physical errors E_i are caused by a physical process at a certain location in the circuit, and the logical errors are the equivalent forward propagated errors $E_i|_\Omega$ restricted to the data qubits.

We now collect the errors which need to be identified by the syndrome measurement. These are the storage errors in between the current and the previous syndrome measurement $E_a \in E_{\mathcal{T}}$, the errors in the current recovery procedure which affect the current syndrome, $E_b \in E_{\mathcal{R},1}$, and the errors of the previous recovery procedure which did not affect the previous syndrome, $E_c \in E_{\mathcal{R},0}$. The recovery procedure thus needs to be designed to correct for the set of errors

$$E_{\mathcal{R}} = \{E_a \,|E_a \in E_{\mathcal{T}}\} \cup \{E_c \,|E_c \in E_{\mathcal{R},0}\} \cup \{E_b \,|E_b \in E_{\mathcal{R},1}\}. \tag{4.26}$$

To each error E_i in $E_{\mathcal{R}}$ there belongs an error image

$$\mathbf{F}_i^E = \mathcal{I}^{-1}(E_i|_\Omega), \ \ \forall\, E_i \in E_{\mathcal{R}}. \tag{4.27}$$

Also, to each error E_i in $E_{\mathcal{R}}$ there belongs a syndrome $\mathbf{Sy}(E_i)$, which is nontrivial by definition and can, for a given circuit, easily worked out in a classical computation.

The condition for fault-tolerance of the recovery procedure is, with some simplification, that all errors in the set $E_{\mathcal{R}}$ need to be identifiable by the error syndrome up to equivalence. The simplification is that some errors only need to be detected, not identified. Specifically, recovery procedures usually invoke certain ancilla states. These ancillas can be verified before they interact with the data. If they are found to be erroneous they may just be discarded and prepared anew. Therefore, to check the ancilla states error detection with a non-trivial syndrome is sufficient. To take account of this fact we split the syndrome space into two subspaces,

$$\mathbf{Sy} = \mathbf{Sy}^1 \oplus \mathbf{Sy}^2. \tag{4.28}$$

$\mathbf{Sy}^1(E_i)$ is the vector of syndrome bits of an error E_i obtainable by measurement before an operation invoking the data qubits is performed. Practically, this is ancilla verification. $\mathbf{Sy}^2(E_i)$ is the vector of syndrome bits of an error E_i obtainable only after the encoded quantum data has been operated on.

Let us define the set $E_{\mathcal{R}}^1$ of errors which can be detected by the ancilla verification syndrome,

$$E_{\mathcal{R}}^1 = \{E_i \in E_{\mathcal{R}} |\, \mathbf{Sy}^1(E_i) \neq 0\}. \tag{4.29}$$

The condition for fault-tolerance of the recovery procedure $\mathcal{R}$ is

$$\forall E_i, E_j \in E_{\mathcal{R}} \backslash E_{\mathcal{R}}^1 :\ \mathbf{Sy}^2(E_i) + \mathbf{Sy}^2(E_j) \neq 0 \ \vee\ \mathbf{F}_i^E + \mathbf{F}_j^E \in \mathcal{I}^{-1}(S). \tag{4.30}$$

The error recovery procedure is fault-tolerant if all those errors which cannot be detected by a non-trivial syndrome in the ancilla preparation can be identified up to equivalence by the remaining part of the syndrome.

Note that the condition (4.30) for fault-tolerance of an error recovery procedure based on a stabilizer code takes the same form as the condition (4.20) for a stabilizer code to correct errors from a set E. Only the set of errors is a different one, and the definitions of the error image and the syndrome are extensions of their former counterparts.

To practically construct a proper error recovery procedure it is suitable to divide it into smaller parts such as ancilla preparation, ancilla verification (measurement of $\mathbf{Sy}^1$), syndrome extraction, i.e. interaction between data and ancilla, syndrome measurement (part of $\mathbf{Sy}^2$), and verification of the syndrome measurement [72] (remaining part of $\mathbf{Sy}^2$). Examples for fault-tolerant error recovery procedures are presented e.g. in [20].

Fault-tolerant quantum computation. To prevent a quantum computation from errors requires coding, and one needs to find a universal set of gates which operates on encoded qubits. Further, these encoded gates must be fault-tolerant. That is, they need

to be able to correct for or to identify just the errors they introduce (a precise form of this statement is given below).

In such a situation, if the probability of failure of each bare gate from a universal set, including wire, is bound by η, then the probability of failure for the corresponding fault-tolerant encoded gates will be $O(\eta^2)$. Then, for small enough η the encoded gates will perform better than the bare ones, and again code concatenation may be used to make the gate error probability arbitrarily small.

For any fixed value of the gate error probability, however small, the quantum computation will end up close to the totally mixed state if the computation is long enough. However, concatenated codes may be used with the number of coding levels adapted to the length of the computation. Fortunately, the increase in the code block size is moderate with increasing duration of the computational process, and error thresholds can be obtained for fault-tolerant quantum computation.

Proven error thresholds are of the order of $\eta = 10^{-6}$ [44], but computer simulations suggest that the actual values may be significantly higher [65, 50]. As an example, in [50] a constraint on the common value of the one- and two-qubit gate error and the value of the memory error rate has been obtained. For the case that syndrome measurements are as fast as the unitary gates, and for suitable codes, values of 10^{-3} for both the gate- and the memory error probability per gate cycle are consistent with this constraint.

Let us return to the conditions which need to be obeyed by a fault-tolerant gate on encoded qubits. First, as an encoded gate, it needs to preserve the code space. Second, it needs to be fault tolerant. An encoded circuit will consist of encoded gates, and in between each gate and its successor gate for each encoded qubit there is an error recovery procedure. This means in turn that, before and after each encoded gate, for every in- and outgoing encoded qubit there is an error recovery step. The errors belonging to a gate need to be identified up to equivalence by the error recovery procedure(s) after the gate. So let us summarize these errors. They are the errors introduced by the encoded gate, the errors introduced by the recovery procedure(s) just before the gate which did have a trivial syndrome there, and the errors introduced in the recovery procedure(s) just after the gate which have a nontrivial syndrome. These errors form, in analogy to (4.26), the error set E_G of the encoded gate G. As in the case of error recovery, the syndrome $\mathbf{Sy}$ may have two parts, one for error detection, $\mathbf{Sy}^1$, and one for error identification, $\mathbf{Sy}^2$, $\mathbf{Sy} = \mathbf{Sy}^1 \oplus \mathbf{Sy}^2$. In analogy to the case of error recovery, we may identify for each error $E_i \in E_G$ a syndrome $\mathbf{Sy}(E_i)$ and an error image $\mathbf{F}_i^E$, and define the set $E_G^1 = \{E_i \in E_G | \, \mathbf{Sy}^1(E_i) \neq 0\}$. The fault-tolerance condition for the encoded gate G then is

$$\forall E_i, E_j \in E_G \backslash E_G^1 : \; \mathbf{Sy}^2(E_i) + \mathbf{Sy}^2(E_j) \neq 0 \; \vee \; \mathbf{F}_i^E + \mathbf{F}_j^E \in \mathcal{I}^{-1}(S), \tag{4.31}$$

which is of the same form as the fault-tolerance condition (4.30) for an error recovery procedure.

As quantum codes were first introduced for coding, they are designed to correct errors with low weight which are the most likely errors in data storage. For example, codes with distance three, such as the Steane code, correct all one-qubit errors. As for encoded gates

the same codes are used, a sufficient condition for fault-tolerance of these gates is that each physical error may lead to at most one logical error (or $\lfloor d/2 \rfloor$ errors) [41]. In this way, the condition (4.31) on a set of errors is converted into a condition on individual errors, which is a lot more manageable and helpful in the design of fault-tolerant gates. Circuits for fault-tolerant quantum computation can be found e.g. in [20, 41, 42, 43, 44].

Fault-tolerance is an important question for any type of a quantum computer. Therefore, in Section 4.3 we will give a proof for the fault-tolerance of the $\mathrm{QC}_{\mathcal{C}}$. The proof is based on a theorem by Aharonov and Ben-Or [44], stating that fault-tolerant quantum computation is possible in quantum logic networks of qubits with local and next-neighbor gates only, even if the qubits are arranged in one dimension. We will trace back the $\mathrm{QC}_{\mathcal{C}}$ on a two-dimensional cluster state to such a device.

A characteristic feature of the scheme for fault-tolerant quantum computation envisioned by Aharonov and Ben-Or is that it completely avoids measurement except for the final readout. To simulate such a device on a quantum computer which entirely consist of measurements is probably as counter-intuitive as can be. We have nevertheless chosen this approach because it simplifies the fault-tolerance proof for the $\mathrm{QC}_{\mathcal{C}}$. Adequate and efficient methods for $\mathrm{QC}_{\mathcal{C}}$-fault-tolerant computation cannot be expected to arise from this procedure. For such, see Section 4.4.

In preparation for the fault-tolerance proof in Section 4.3, let us now restate some definitions and conventions for objects introduced in [44] and quote a theorem stated therein. In [44], the quantum computation is regarded as a succession of layers g_{t_i} of perfect quantum gates alternating with error superoperators $\mathcal{E}_{t_i}$, acting on the density operator representing the state of the quantum register, $\rho_{\mathrm{final}} = \mathcal{E}_t \circ g_t \circ \mathcal{E}_{t-1} \circ g_{t-1} \circ ... \circ \mathcal{E}_1 \circ g_1 \, \rho_{\mathrm{init}}$. Each $\mathcal{E}_{t_i}$ comprises a number of individual errors, each of which occurs at a *location* A. A set $(q_1, q_2, \ldots, q_l, t)$ is a location $A_{l_t,t}$ in the quantum circuit Q if the qubits $q_1, q_2, \ldots, q_l$ participate in the same gate at time t, and no other qubit participates. The list of times and places where faults have occurred (in a specific run of the computation), is called a *fault path*. The error model with which the discussion of fault-tolerance in [44] starts is

Error model 1 *The errors in the network are probabilistic local errors acting in between the gates. The noise superoperator at time step t, that is after the layer t of gates has been applied, takes the form*

$$\mathcal{E}(t) = \mathcal{E}_{A_{1,t}}(t) \otimes \mathcal{E}_{A_{2,t}}(t) \otimes ... \otimes \mathcal{E}_{A_{l_t,t}}(t). \tag{4.32}$$

There, $A_{i,t}(t)$ runs over all possible locations at time t, and for each of the associated superoperators

$$\|\mathcal{E}_{A_i,t}(t) - \mathbb{1}\| \leq \eta. \tag{4.33}$$

A suitable norm $\|\cdot\|$ for superoperators $\mathcal{T}$ in (4.33) should satisfy the properties

$$\|\mathcal{T}\rho\| \leq \|\mathcal{T}\|\|\rho\|, \tag{4.34a}$$

$$\|\mathcal{T}\mathcal{T}'\| \leq \|\mathcal{T}\|\|\mathcal{T}'\|, \tag{4.34b}$$

$$\|\mathcal{T} \otimes \mathcal{T}'\| = \|\mathcal{T}\|\|\mathcal{T}'\|, \tag{4.34c}$$

$$\|\mathcal{T}_{\mathrm{Phy}}\| = 1, \tag{4.34d}$$

where $\mathcal{T}_{\mathrm{Phy}}$ is any physically allowed superoperator. Such norms exist. An example is the diamond norm [73].

The initial error model 1 is further generalized in [44] to allow for exponentially decaying correlations in both space and time. Specifically, it is required that the probability p for a fault path which contains k locations is bounded by some constant times the probability for this fault path in error model 1,

$$p(\text{fault path with } k \text{ errors}) \leq c\eta^k (1-\eta)^{v-k}, \tag{4.35}$$

where v is the number of locations in the circuit.

The theorem stated below is formulated for this more general type of noise, which contains model 1 as a special case. The noise that we obtain in the simulation of quantum logic networks on the $\mathrm{QC}_{\mathcal{C}}$ will, under assumptions specified in the next section, be exactly as in model 1.

A quantum computer is said to have no geometry, if any subset of qubits in it may interact in quantum gates; and it is said to have a geometry if the qubits are arranged on a d-dimensional lattice and only neighboring qubits are allowed to perform gates on. With the above notions introduced, we now quote from [44] the following theorem:

Threshold theorem for d-dimensional circuits
Let $\epsilon > 0$. Let $d \geq 1$. Let $\mathcal{G}'$, $\mathcal{G}''$ be two universal sets of quantum gates. There exists a threshold $\eta'' > 0$, and constants c_1, c_2, c_3, c_4 such that the following holds. Let Q' be a d-dimensional quantum circuit, with n qubits, which operates t time steps, uses s gates from $\mathcal{G}'$, and has v locations. There exists a d-dimensional quantum circuit Q'' which operates on $c_1 n \log^{c_2}\left(\frac{v}{\epsilon}\right)$ qubits, for time $c_3 t \log^{c_2}\left(\frac{v}{\epsilon}\right)$, and uses $c_4 s \log^{c_2}\left(\frac{v}{\epsilon}\right)$ gates from $\mathcal{G}'' \cup \{SWAP\}$ such that in the presence of general noise with error rate $\eta < \eta''_c$, Q'' computes a function which is ϵ-close to that computed by Q'. (4.36)

In Section 4.3 we will prove the fault-tolerance of the $\mathrm{QC}_{\mathcal{C}}$ by showing that, with the error model described in Section 4.2, the $\mathrm{QC}_{\mathcal{C}}$ on a two-dimensional cluster state can simulate one-dimensional quantum circuits with errors specified by the error model 1.

4.2 The error model for the $\mathrm{QC}_{\mathcal{C}}$

In this section we will introduce the model of physical errors in a $\mathrm{QC}_{\mathcal{C}}$-computation upon which we base the following discussions of fault-tolerant computation with the $\mathrm{QC}_{\mathcal{C}}$. The error modell 2 of the physical errors –which is the result of this section– takes the form of an error channel sandwiched between a perfect cluster state and perfect measurements. The subsequent assumptions are to keep the form of the error channel simple.

Among the ways to introduce an error model the following two seem extremal. Either, one might postulate it right away. Or, to the contrary, one may base the discussion of errors on a specific physical system, say, ultra-cold atoms in an optical lattice. To provide

a starting point, in this thesis we have chosen an intermediate approach. We show that the error model 2 is valid if the required cluster state is created via the Ising interaction in a model system with properties defined below.

Perhaps the most important advantage of the construction in Section 2.4, i.e. to divide a quantum computation into sub-circuits and realize each sub-circuit on a smaller cluster, is that in this way decoherence can be controlled. If a single large cluster is used, the computation might reach certain cluster qubits only after a long time such that the cluster would have already decohered significantly and it is not clear how error-correction could help in such a situation. This might, for any error rate, limit the duration of a computation. On the other hand, if the computation is split, then the size of the sub-circuits may be adjusted such that each of them can be performed within a time T and in this way, each cluster qubit is, before being measured, exposed to a bounded amount of decoherence specified by T.

We base our model on the following assumptions for the physical errors:

A1 The cluster state is created from a product state $|+\rangle_{\mathcal{C}} = \otimes_{a\in\mathcal{C}}|+\rangle_a$ via Ising interaction

$$S_I^{(\mathcal{C})} = \exp\left(-i\frac{\pi}{4}\sum_{(a,b)\in E_{\mathcal{C}}}\sigma_z^{(a)}\sigma_z^{(b)}\right), \tag{4.37}$$

followed by multi-local rotations which adjust the local bases

$$U_C^{(\mathcal{C})} = \bigotimes_{a\in\mathcal{C}}\exp\left(i\frac{\pi}{4}\deg(a)\,\sigma_z^{(a)}\right). \tag{4.38}$$

Therein, $\deg(a)$ denotes the degree of the vertex a. $U_C^{(\mathcal{C})}S_I^{(\mathcal{C})}$ is identical to $S^{(\mathcal{C})}$.

In the process of computation errors arise due to erroneous preparation of the state $|+\rangle_{\mathcal{C}}$, erroneous Ising interaction and subsequent basis correction, decoherence due to interaction with environmental degrees of freedom, and erroneous measurements. The individual error processes are modeled as follows:

A2 a) Decoherence is modeled by a multi-local depolarizing channel (4.25),

$$\rho_{\mathcal{C}} \longrightarrow \mathcal{T}_1^{(\mathcal{C})}[p_D]\,\rho_{\mathcal{C}} = \bigotimes_{a\in\mathcal{C}}\mathcal{T}_1^{(a)}[p_D]\,\rho_{\mathcal{C}}. \tag{4.39}$$

Therein, p_D is the characteristic error probability associated with decoherence.

b) The one-qubit measurements are modeled by perfect one-qubit measurements preceded by a local depolarizing channel,

$$P_{\mathrm{Phy}}^{(\mathcal{C})} = P^{(\mathcal{C})}\circ\mathcal{T}_1^{(\mathcal{C})}[p_M]. \tag{4.40}$$

Therein, p_M is the characteristic error probability associated with the measurement. Similarly, the operations for basis adjustment (4.38) are modeled by the perfect operation followed by a depolarizing channel with error probability p_C,

$$U_{C,\mathrm{Phy}}^{(\mathcal{C})} = \mathcal{T}_1^{(\mathcal{C})}[p_C]\circ\left[U_C^{(\mathcal{C})}\right]. \tag{4.41}$$

c) The erroneous Ising interaction and the erroneous preparation of the state $|+\rangle_{\mathcal{C}}$, i.e. the globally tunable operations, are modeled as controlled by classical fields, whose parameters vary according to given distributions. The superoperator describing the erroneous operation is obtained by averaging over a sample of transformations with varying control parameters.

Below we show that, under the assumptions A1 and A2, we obtain the following set of elementary errors:

$$E_{\mathrm{phy}} = \left\{\sigma_x^{(a)}, \sigma_y^{(a)}, \sigma_z^{(a)}, \sigma_z^{(b)}\sigma_z^{(c)} \mid a, b, c \in \mathcal{C}, c \in \mathrm{nbgh}(b)\right\}. \tag{4.42}$$

A3 The elementary errors of E_{phy} are stochastically independent.

Assumption A3 may seem the most restrictive one; in particular as we have an eye on cold controlled collisions in optical lattices, which are, at present, one of the most promising candidates for the realization of the $\mathrm{QC}_{\mathcal{C}}$. There, e.g. the strength and duration of the interaction is controlled by a single global parameter. Its control can, of course, not be perfect. Therefore, in individual runs of the process, the conditional interaction phases acquired by pairs of neighboring qubits will vary. But, as the control parameter is global, the phases will be correlated. They are either all precise, all too small or all too large. In this way, we are faced with classically correlated errors. The same problem arises in the preparation of the initial state $|+\rangle_{\mathcal{C}}$, if it is created from the ground state of the system via a simultaneous multi-local (global) Hadamard transformation. Therefore, it has to be investigated how the correlations in the noise introduced by the imperfect operations affect the fault-tolerance of the $\mathrm{QC}_{\mathcal{C}}$. At present one statement can be made: Classical correlations among inequivalent Pauli errors E_i, E_j, i.e. $E_i|\phi\rangle_{\mathcal{C}} \neq E_j|\phi\rangle_{\mathcal{C}}$, have no effect at all to a $\mathrm{QC}_{\mathcal{C}}$-computation. See Appendix A.

Let us further comment on the assumptions A2/c). There we start with a more detailed and complicated model for the errors of the Ising interaction and the preparation of the initial $|+\rangle_{\mathcal{C}}$ state than the one used to describe decoherence and imperfect measurement. Eventually, we will present a similar superoperator describing the respective error processes, but it seems saver not to write down one ad hoc.

The over-rotation applied to the state $|+\rangle_{\mathcal{C}}$ in the initial step of preparation in an individual run may be described by

$$U_P^{(\mathcal{C})}[\{\vec{\alpha}_j\}] = \bigotimes_{i\in\mathcal{C}} U_P^{(i)}[\vec{\alpha}_i], \tag{4.43}$$

with

$$U_P^{(i)}[\vec{\alpha}_i] = \cos\alpha_i \mathbb{1}^{(i)} + i \sin\alpha_i \frac{\vec{\alpha}_i}{\alpha_i} \cdot \vec{\sigma}^{(i)}. \tag{4.44}$$

The over-rotation is now averaged over with an underlying probability distribution $p(\{\vec{\alpha}_i\})$ and the noise introduced in the preparation of $|+\rangle_{\mathcal{C}}$ is described by the superoperator

$$\mathcal{T}_P^{(\mathcal{C})} = \int d^3\vec{\alpha}_1 ... d^3\vec{\alpha}_{|\mathcal{C}|}\; p(\{\vec{\alpha}_j\}) \left[U_P^{(\mathcal{C})}[\{\vec{\alpha}_i\}]\right]. \tag{4.45}$$

The brackets $[\cdot]$, like the outer ones in $\left[U_P^{(\mathcal{C})}[\{\vec{\alpha}_i\}]\right]$ denote that the object in between is to be understood as a superoperator. The trace-preserving property of $\mathcal{T}_P^{(\mathcal{C})}$ is guaranteed by the constraint $\int d^3\vec{\alpha}_1 ... d^3\vec{\alpha}_{|\mathcal{C}|}\ p(\{\vec{\alpha}_i\}) = 1$.

The transformation generated by the Ising Hamiltonian $H_I = \sum_{(a,b)\in E_{\mathcal{C}}} \hbar g_{ab}(t) \frac{\sigma_z^{(a)}}{2} \frac{\sigma_z^{(b)}}{2}$ takes, for the specific set $\{\varphi_{ab} := \int_0^T g_{ab}(t)\, dt,\ \forall (a,b) \in E_{\mathcal{C}}\}$ of interaction phases, the form

$$U_I^{(\mathcal{C})}[\{\Delta\varphi_{ab}\}] = \exp\left(-i \sum_{(a,b)\in E_{\mathcal{C}}} \Delta\varphi_{ab} \frac{\sigma_z^{(a)}}{2} \frac{\sigma_z^{(b)}}{2}\right) S_I^{(\mathcal{C})}, \tag{4.46}$$

where $S_I^{(\mathcal{C})}$ is the perfect transformation (4.37), and the deviating interaction phases $\Delta\varphi_{ab}$ are given by

$$\Delta\varphi_{ab} = \int_0^T g_{ab}(t)dt\ - \pi. \tag{4.47}$$

These phases are distributed according to a probability density $p'(\{\Delta\varphi_{ab}\})$. The superoperator $\mathcal{T}_{I,2}^{(\mathcal{C})}$ describing the deviation from the ideal Ising interaction then is

$$\mathcal{T}_{I,2}^{(\mathcal{C})} = \int \left(\prod_{(a,b)\in E_{\mathcal{C}}} d\Delta\varphi_{ab}\right) p'(\{\Delta\varphi_{cd}\}) \left[\exp\left(-i \sum_{(e,f)\in E_{\mathcal{C}}} \Delta\varphi_{ef} \frac{\sigma_z^{(e)}}{2} \frac{\sigma_z^{(f)}}{2}\right)\right]. \tag{4.48}$$

Again, the brackets $[\cdot]$ indicate that the object in between is understood as a superoperator, not as an operator.

The process of computation in the presence of noise is now described by the sequence

$$\rho_{\text{final}} = \left[P^{(\mathcal{C})}\right] \circ \mathcal{T}_1^{(\mathcal{C})}[p_M] \circ \mathcal{T}_1^{(\mathcal{C})}[p_D] \circ \mathcal{T}_1^{(\mathcal{C})}[p_C] \circ \left[U_C^{(\mathcal{C})}\right] \circ \mathcal{T}_{I,2}^{(\mathcal{C})} \circ \left[S_I^{(\mathcal{C})}\right] \circ \mathcal{T}_P^{(\mathcal{C})} \left(|+\rangle_{\mathcal{C}}\langle +|\right). \tag{4.49}$$

As can be seen in (4.49), the operations which act first on the product state $|+\rangle_{\mathcal{C}}\langle +|$ are the local rotations $\bigotimes_{i\in\mathcal{C}} U_P^{(i)}[\vec{\alpha}_i]$ (4.43). We now note that $\sigma_x^{(i)}$ operators are absorbed by the state $|+\rangle_{\mathcal{C}}$, $\sigma_x^{(i)}|+\rangle_{\mathcal{C}} = |+\rangle_{\mathcal{C}}$, for all $i \in \mathcal{C}$. Therefore, the rotations $U_P^{(i)}[\vec{\alpha}_i]$ (4.44) may be replaced by the rotations

$$\tilde{U}_P^{(j)}[\vec{\alpha}_j] = \left(\cos\alpha_j + i \sin\alpha_j \frac{\alpha_{j,x}}{\alpha_j}\right) \mathbb{1}^{(j)} + i \sin\alpha_j \left(\frac{\alpha_{j,z} + i\alpha_{j,y}}{\alpha_j}\right) \sigma_z^{(j)}, \tag{4.50}$$

which act equivalently on the state $|+\rangle_{\mathcal{C}}$, $\bigotimes_{i\in\mathcal{C}} \tilde{U}_P^{(i)}[\vec{\alpha}_i]\,|+\rangle_{\mathcal{C}} = \bigotimes_{i\in\mathcal{C}} U_P^{(i)}[\vec{\alpha}_i]\,|+\rangle_{\mathcal{C}}$. Consequently, $\mathcal{T}_P^{(\mathcal{C})}$ may be replaced by

$$\tilde{\mathcal{T}}_P^{(\mathcal{C})} = \int d^3\vec{\alpha}_1 ... d^3\vec{\alpha}_{|\mathcal{C}|}\ p(\{\vec{\alpha}_j\}) \left[\bigotimes_{i\in\mathcal{C}} \tilde{U}_P^{(i)}[\vec{\alpha}_i]\right]. \tag{4.51}$$

The rotations $\tilde{U}_P^{(j)}[\vec{\alpha}_j]$ contain only operators $\mathbb{1}^{(j)}$ and $\sigma_z^{(j)}$. Therefore, $\tilde{\mathcal{T}}_P^{(\mathcal{C})}$, commutes with $\mathcal{T}_{I,2}^{(\mathcal{C})}$, $S_I^{(\mathcal{C})}$ and $U_C^{(\mathcal{C})}$, which also commute mutually. We can thus exchange the order

of these operations, $\left[U_C^{(\mathcal{C})}\right] \circ \mathcal{T}_{I,2}^{(\mathcal{C})} \circ \left[S_I^{(\mathcal{C})}\right] \circ \tilde{\mathcal{T}}_P^{(\mathcal{C})} = \tilde{\mathcal{T}}_P^{(\mathcal{C})} \circ \mathcal{T}_{I,2}^{(\mathcal{C})} \circ \left[U_C^{(\mathcal{C})}\right] \circ \left[S_I^{(\mathcal{C})}\right]$, and, noting that $U_C^{(\mathcal{C})} S_I^{(\mathcal{C})} = S^{(\mathcal{C})}$, write the computational process as

$$\rho_{\text{final}} = \left[P^{(\mathcal{C})}\right] \circ \mathcal{T}_1^{(\mathcal{C})}[p_M] \circ \mathcal{T}_1^{(\mathcal{C})}[p_D] \circ \mathcal{T}_1^{(\mathcal{C})}[p_C] \circ \tilde{\mathcal{T}}_P^{(\mathcal{C})} \circ \mathcal{T}_{I,2}^{(\mathcal{C})}(|\phi\rangle_{\mathcal{C}}\langle\phi|). \tag{4.52}$$

So far, we see that we have independent one-qubit errors caused by decoherence, local rotation and imperfect measurement, as expressed by $\mathcal{T}_1^{(\mathcal{C})}[p_D]$, $\mathcal{T}_1^{(\mathcal{C})}[p_C]$, and $\mathcal{T}_1^{(\mathcal{C})}[p_M]$, respectively. With $\tilde{\mathcal{T}}_P^{(\mathcal{C})}$, we encounter further one-qubit errors which are, depending on the probability distribution $p(\{\vec{\alpha}_i\})$, potentially classically correlated. And we have one type of two-qubit errors, namely $\sigma_z^{(a)}\sigma_z^{(b)}$-errors on next neighboring qubits. Also these errors may, depending on the form of the probability distribution $p'(\{\Delta\varphi_{ab}\})$, be classically correlated.

The next step is to invoke assumption A3 requiring that the potentially classically correlated errors are, in fact, uncorrelated. For relaxation see Appendix A. We require

$$p(\{\vec{\alpha}_i\}) = \prod_{i\in\mathcal{C}} p_i(\vec{\alpha}_i), \tag{4.53a}$$

$$p'(\{\Delta\varphi_{ab}\}) = \prod_{(a,b)\in E_{\mathcal{C}}} p'_i(\Delta\varphi_{ab}). \tag{4.53b}$$

Using (4.53a) in (4.51), we obtain for the local errors become $\tilde{\mathcal{T}}_P^{(\mathcal{C})} = \bigotimes_{i\in\mathcal{C}} \tilde{\mathcal{T}}_P^{(i)}$, with

$$\tilde{\mathcal{T}}_P^{(i)} = \int d^3\vec{\alpha}_i \, p(\vec{\alpha}_i) \left[\tilde{U}_P^{(i)}[\vec{\alpha}_i]\right]. \tag{4.54}$$

For any distribution $p(\vec{\alpha}_i)$, there exists a probability p_P such that

$$\mathcal{T}_1^{(i)}[p_P] = \mathcal{E}_1^{(i)} \circ \tilde{\mathcal{T}}_P^{(i)}, \tag{4.55}$$

where $\mathcal{E}_1^{(i)}$ is a physical one-qubit operation and $\mathcal{T}_1^{(i)}[p_P]$ is the $SU(2)$-invariant depolarizing one-qubit channel (4.25) with error probability p_P. We majorize the error channel $\tilde{\mathcal{T}}_P^{(i)}$ by $\mathcal{T}_1^{(i)}[p_P]$ with the smallest value for p_P. That is, in the sequence (4.52) we replace the $\tilde{\mathcal{T}}_P^{(i)}$ by $\mathcal{T}_1^{(i)}[p_P]$, thereby introducing more noise. If the additional local noise $\bigotimes_{i\in\mathcal{C}} \mathcal{E}^{(i)}$ turned out to improve the computation it might just be applied actively before the measurement sequence.

With this replacement, we have three local depolarizing channels $\mathcal{T}_1^{(\mathcal{C})}[p_M] \circ \mathcal{T}_1^{(\mathcal{C})}[p_D] \circ \mathcal{T}_1^{(\mathcal{C})}[p_C] \circ \mathcal{T}_1^{(\mathcal{C})}[p_P]$ acting in a row, and we may replace them by a single local depolarizing channel

$$\mathcal{T}_1^{(\mathcal{C})}[p_1] := \mathcal{T}_1^{(\mathcal{C})}[p_M] \circ \mathcal{T}_1^{(\mathcal{C})}[p_D] \circ \mathcal{T}_1^{(\mathcal{C})}[p_C] \circ \mathcal{T}_1^{(\mathcal{C})}[p_P]. \tag{4.56}$$

Therein, p_1 is the characteristic error probability for one qubit errors. To leading order, p_1 is the sum of the individual one-qubit error probabilities.

Besides the $SU(2)$-invariant one-qubit errors described by (4.56) there is one other type of errors, namely the $\sigma_z^{(a)}\sigma_z^{(b)}$-error on next-neighboring qubits. The corresponding error channel is

$$\mathcal{T}_{I,2}^{(\mathcal{C})}[p_2] = \bigotimes_{(a,b)\in E_{\mathcal{C}}} \mathcal{T}_{I,2}^{(a,b)}[p_2], \tag{4.57}$$

with

$$\mathcal{T}_{I,2}^{(a,b)}[p_2](\rho) = (1-p_2)\rho + p_2\, \sigma_z^{(a)}\sigma_z^{(b)}\rho\, \sigma_z^{(a)}\sigma_z^{(b)}. \tag{4.58}$$

In (4.58) we have assumed that all error probabilities are equal, $q_{ab} = p_2$, for all $(a,b) \in E_{\mathcal{C}}$, $b \in \mathrm{nbgh}(a) \subset \mathcal{C}$. Thus, we finally obtain

Error model 2 *$QC_{\mathcal{C}}$-computation in the presence of errors specified by the assumptions* A1, A2 *and* A3 *may be described by a noisy channel of independent one- and two-qubit errors applied to a perfect cluster state, followed by perfect measurements.*

$$\rho_{\mathcal{C},\mathrm{out}} = \left[\bigotimes_{a\in\mathcal{C}} P^{(a)}\right] \circ \mathcal{T}_1^{(\mathcal{C})}[p_1] \circ \mathcal{T}_{I,2}^{(\mathcal{C})}[p_2]\Big(|\phi\rangle_{\mathcal{C}}\langle\phi|\Big). \tag{4.59}$$

Therein, the error channel $\mathcal{T}_1^{(\mathcal{C})}[p_1]$ is a tensor product of local depolarizing channels (4.25) *with probability p_1 for an individual error, and $\mathcal{T}_{I,2}^{(\mathcal{C})}[p_2]$, defined in* (4.57)*, is a channel of independent $\sigma_z\sigma_z$-errors on next neighboring qubits, with error probability p_2.*

4.3 Fault-tolerance of the $\mathrm{QC}_{\mathcal{C}}$

Fault-tolerant quantum computation has been discussed in detail within the network model [40, 41, 43, 44, 64, 71]. Therefore it appears that the shortest way to obtain a result about fault-tolerance of the $\mathrm{QC}_{\mathcal{C}}$ is to show that an imperfect $\mathrm{QC}_{\mathcal{C}}$ can simulate a quantum logic network with a certain gate- and memory error efficiently. It is important that uncorrelated physical errors in a $\mathrm{QC}_{\mathcal{C}}$-computation lead to uncorrelated logical errors in the simulated network because such errors can be handled.

For networks without geometry, i.e. with no restriction of the interaction to neighboring qubits, one would need to provide $\mathrm{QC}_{\mathcal{C}}$ gate simulations for nonlocal gates whose error does not increase with distance. This does not seem to be the most straightforward thing to do. We chose instead to simulate one-dimensional networks (i.e. with local and next-neighbor gates only) on the $\mathrm{QC}_{\mathcal{C}}$, which is a simpler task. Fault-tolerance of a one-dimensional network quantum computer has been proven in [64] and [44]. We base our proof of $\mathrm{QC}_{\mathcal{C}}$-fault-tolerance on the respective Theorem [44] for one-dimensional networks stated in (4.36). All that needs to be shown is that the $\mathrm{QC}_{\mathcal{C}}$ can simulate one-dimensional networks below their error thresholds for fault-tolerant quantum computation. One does not need to worry about constructions for fault-tolerant gates on encoded qubits; all this is taken care of by the network proof [44]. The theorem which guarantees fault-tolerance of the $\mathrm{QC}_{\mathcal{C}}$ is

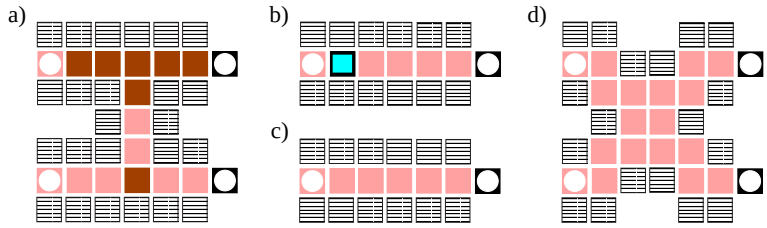


Figure 4.2: Set of $QC_{\mathcal{C}}$ gate simulations for a one-dimensional network. a) CNOT, b) x-rotation, c) wire, d) SWAP gate. The z-rotation is not shown, see text.

Theorem 2 *There exist finite error thresholds* $p_1^{\text{crit}}, p_2^{\text{crit}} > 0$ *such that the* $QC_{\mathcal{C}}$ *with error model 2 is fault-tolerant if* $p_1 < p_1^{\text{crit}}$ *and* $p_2 < p_2^{\text{crit}}$.

Proof. We proceed along similar lines as in Section 2.2.4 where we have shown that the $QC_{\mathcal{C}}$ may be viewed as a succession of gate simulations. Here, with the presence of noise, we demonstrate that an imperfect $QC_{\mathcal{C}}$-simulation of a network quantum computer may be viewed as the simulation of an imperfect network.

The proof consists of two steps. First, we show that all the locations of the individual physical errors can be grouped into error location sets I_k such that the errors in one location set affect only the functioning of the $QC_{\mathcal{C}}$-simulation of a single gate. In this way, the sets of error locations can be assigned to the gates, $I_k \longrightarrow I(g_k)$. Second, we show that the errors located in $I(g)$ lead to an error η_g of the gate g such that any bound on the gate error, $\eta_g \leq \eta_c''$ may be matched if the error probabilities p_1, p_2 of the physical error are sufficiently small but positive. Then, the $QC_{\mathcal{C}}$ can simulate one-dimensional network quantum computers in the fault-tolerant regime.

Step 1. A physical error has the potential to affect a gate only if a qubit on which this error has support is measured for the realization of this gate. That is, an error E_i may affect the gate g if it has support on $\mathcal{C}(g)\backslash\mathcal{C}_O(g)$. As the clusters overlap, errors which are located on $\mathcal{C}_O(g)$ are at the same time located on the input zone $\mathcal{C}_I(g')$ of the sub-cluster for a subsequent gate, and are counted as affecting the latter. This criterion allows one to assign the site errors unambiguously to the gates. But we have also bond errors, i.e. errors which have support on two cluster qubits. Among these errors there are some which potentially affect two consecutive gates. However, such an error may be relocated in a way that it affects only a single gate, as is explained below.

For illustration, let us first display in Fig. 4.2 the $QC_{\mathcal{C}}$-realizations for a number of gates from a possible universal set $\mathcal{G}''$, consisting of x- and z-rotations and the CNOT gate, plus the SWAP gate. The realization of the rotation about the z-axis, which is not shown, is analogous to that of the x-rotation. The only difference is that the cluster qubit by whose measurement the rotation angle is set is one site further to the right.

We are not restricted in the choice of the universal gate set $\mathcal{G}''$, and the specific choice plays no role for the proof. However, there is a constraint that needs to be respected in the construction of the respective $QC_{\mathcal{C}}$-gate simulations. Specifically, the vertical distance

between the qubit wires on the cluster needs to be grater than two. For convenience of the gate set displayed in Fig. 4.2, we have chosen distance of four. The reason for this requirement is the following. First note that the cluster qubits measured in the σ_z-eigenbasis which are neighbors of cluster qubits measured in the equator of the Bloch sphere, even though they are not needed for the computation, have an impact once they are present. This may be easily verified from equation (3.52). The measurement outcome obtained on such a qubit enters into the byproduct operator of a gate simulation in the same way as the measurement outcome from the neighboring cluster qubit measured in the equator of the Bloch sphere. We may therefore call the cluster qubits of which σ_z is measured and which are neighbors of cluster qubits measured in the equator of the Bloch sphere the qubits of the "insulating layer". The cluster qubits outside the insulating layer have no impact on the computation, and they do not even have to be measured. The qubits of the insulating layer are shown in the measurement patterns of the gate simulations in Fig. 4.2. For the block shaped gates displayed in Fig. 4.2 with their simple composition rule the assignment of an insulating layer qubit to a gate simulation is unambiguous. Each qubit in the insulating layer belongs to exactly one gate simulation. As the measurement outcomes obtained from insulating layer qubits enter into the byproduct operator of the simulated gates these qubits are also sensitive to errors. Leading order physical errors, i.e. one qubit- and next-neighbor two-qubit errors must not cause logical errors in two parallel gates. This is guaranteed if we chose distance four, see Fig. 4.3a.

It is, however, impossible to avoid that elementary physical errors affect two consecutive gates. Namely there is one case, displayed Fig. 4.3b, where a bond error affects an input qubit of a gate g_2 and its left neighbor which belongs to the realization of the preceding gate g_1. Let us denote the qubit in $\mathcal{C}(g_1)\backslash\mathcal{C}_O(g_1)$ as qubit a, and the qubit in $\mathcal{C}(g_2)\backslash\mathcal{C}_O(g_2)$ as qubit b. The discussed bond error E then is $E = \sigma_z^{(a)}\sigma_z^{(b)}$. As the correlation operator $K^{(a)}$ is absorbed by the cluster state, $K^{(a)}|\phi\rangle_{\mathcal{C}} = |\phi\rangle_{\mathcal{C}}$, (2.1), the error $E' = EK^{(a)}$ is equivalent to E, $E' \cong E$. The error E' has only support on $\mathcal{C}(g_1)\backslash\mathcal{C}_O(g_1)$ and hence only affects the gate g_1. In this way, each bond error can be assigned unambiguously to exactly one gate. *Remark:* Despite the fact that the error E' is formally a four-qubit error, we continue to count it as a bond error, labeled by the bond (a, b). The reason for this is that it has an error probability of p_2 characteristic for bond errors. The fact that it acts on four qubits instead of on two qubits is of no concern to the proof. The requirement which needs to be obeyed is that the error E' is located only on $\mathcal{C}(g_1)\backslash\mathcal{C}_O(g_1)$, which it is. We will come back to this point at the relevant passage of the proof below eq. (4.73). The need to call site- and bond-errors by distinct names is caused only by the fact that they occur with different probabilities p_1 and p_2, respectively.

The set of errors locations $I(g)$ belonging to the simulation of a gate g thus consists of the set of locations $I_{\mathrm{site}}(g)$ for site errors and of the set of locations $I_{\mathrm{bond}}(g)$ for bond errors. The σ_z-errors on qubits which are subsequently measured in the eigenbasis of σ_z do not affect the computation, and the bond errors, except for the relocated ones, are all $\sigma_z\sigma_z$. Therefore, only those edges in $E_{\mathcal{C}(g)}$ belong to $I_{\mathrm{bond}}(g)$ for which at least one end vertex is not in the insulating layer. This is true for the relocated errors, too. The set

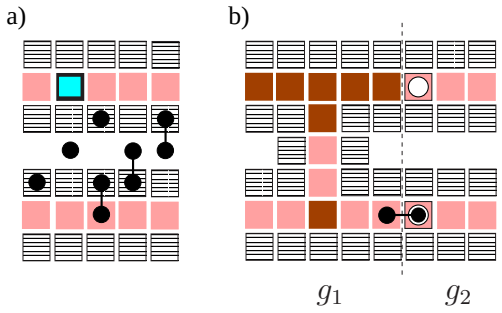


Figure 4.3: a) Distance four is chosen to avoid a two-qubit error in parallel gates caused by an elementary physical one- or two qubit error. b) A physical error which affects two consecutive gates. Such errors cannot be avoided, but they can be handled.

$I_{\text{site}}(g)$ is given by $\mathcal{C}(g)\backslash\mathcal{C}_O(g)$ for all gates g.

Step 2. We now consider a gate simulation on a cluster $\mathcal{C}(g)$ with imperfect means. In such a gate simulation first an input state

$$\rho_{\text{in},\mathcal{C}(g)} = \left[S^{(\mathcal{C}(g))}\right]\left(\rho^{\log}_{\text{in},\mathcal{C}_I(g)} \otimes |+\rangle_{\mathcal{C}(g)\backslash\mathcal{C}_I(g)}\langle +|\right) \tag{4.60}$$

is prepared, where $\rho^{\log}_{\text{in}}$ is the state of the quantum register. Second, this state is acted upon by the noise super operator $\mathcal{T}_g$, and third the perfect measurements

$$P_g(\{\vec{r}_a, s_a\}) = \bigotimes_{a\in\mathcal{C}(g)\backslash\mathcal{C}_O(g)} P^{(a)}(\vec{r}_a, s_a) \tag{4.61}$$

are performed, where $\{\vec{r}_a, s_a\}$ was used in short for $\{\vec{r}_a, s_a \,|\, a \in \mathcal{C}(g)\backslash\mathcal{C}_O(g)\}$. The quantum state resulting from this procedure is

$$\rho_{\text{out},\mathcal{C}(g)} = [P_g(\{\vec{r}_a, s_a\})] \circ \mathcal{T}_g\left(\rho_{\text{in},\mathcal{C}(g)}\right). \tag{4.62}$$

The noise superoperator $\mathcal{T}_g$ is of the form

$$\mathcal{T}_g = \left(\prod_{i\in I(g)} \mathcal{T}_i\right). \tag{4.63}$$

In the error model 2 the physical errors are associated with the vertices and with the bonds in $\mathcal{C}(g)$. The set $I(g)$ is labeling the physical error locations. For each $i \in I(g)$ there is an error superoperator $\mathcal{T}_i$ associated with it. $\mathcal{T}_i = \mathcal{T}_1^{(a_i)}[p_1]$, if i labels a one-qubit error at site $a_i \in \mathcal{C}(g)\backslash\mathcal{C}_O(g)$, and $\mathcal{T}_i = \mathcal{T}_{I,2}^{(a,b)_i}[p_2]$, if i labels an ordinary bond error at bond $(a,b)_i \in E_{\mathcal{C}(g)}$. We now expand the errors into a Pauli basis, and obtain

$$\mathcal{T}_g\left(\rho_{\text{in},\mathcal{C}(g)}\right) = \sum_{\{e_i|i\in I(g)\}} p(\{e_i\})E(\{e_i\})\,\rho_{\text{in},\mathcal{C}(g)}\,E(\{e_i\})^\dagger, \tag{4.64}$$

with

$$p(\{e_i\}) = \prod_{i\in I(g)} p_i(e_i), \;\; E(\{e_i\}) = \prod_{j\in I(g)} E_j(e_j). \tag{4.65}$$

To label each individual error operator we require a further index, e_i. For example, if i is a site error, then we have four individual Pauli errors $E_i(e_i)$: $E_i(0) = \mathbb{1}^{(a_i)}$, $E_i(1) = \sigma_x^{(i)}$, $E_i(2) = \sigma_y^{(i)}$ and $E_i(3) = \sigma_z^{(i)}$. The corresponding error probabilities are $p_i(0) = 1 - p_1$ and $p_i(1) = p_i(2) = p_i(3) = p_1/3$. Equivalently, if $\mathcal{T}_i$ is a bond error on $(a,b)_i$, then $E_i(0) = \mathbb{1}^{(a,b)_i}$ and $E_i(1) = \sigma_z^{(a_i)}\sigma_z^{(b_i)}$ (for an ordinary bond error only). The respective error probabilities are $p_i(0) = 1 - p_2$, $p_i(1) = p_2$.

Now, in eq. (4.62) we exchange the order of the noise superoperator and the projections,

$$\rho_{\mathrm{out},\mathcal{C}(g)} = \sum_{\{e_i|i\in I\}} p(\{e_i\})E(\{e_i\})\tilde{P}_g(\{e_i\})\rho_{\mathrm{in},\mathcal{C}(g)}\tilde{P}_g(\{e_i\})E(\{e_i\})^\dagger. \tag{4.66}$$

where

$$\tilde{P}_g(\{\vec{r}_a, s_a\}, \{e_i\}) = E(\{e_i\})^\dagger \left(\bigotimes_{a\in\mathcal{C}(g)\setminus\mathcal{C}_O(g)} P^{(a)}(\vec{r}_a, s_a) \right) E(\{e_i\}). \tag{4.67}$$

$\tilde{P}_g(\{\vec{r}_a, s_a\}, \{e_i\})$ is a projector onto qubit a like $P_g(\{\vec{r}_a, s_a\})$ is. Just the sets $\{\vec{r}_a\}$ and $\{s_a\}$ may differ. Thus, in the same way as $P_g(\{\vec{r}_a, s_a\})$ projects $|\Psi_{\mathrm{in}}\rangle_{\mathcal{C}(g)}$ into

$$P_g(\{\vec{r}_a, s_a\})|\Psi_{\mathrm{out}}\rangle_{\mathcal{C}(g)} = |\{\vec{r}_a, s_a\}\rangle_{\mathcal{C}(g)\setminus\mathcal{C}_O(g)} \otimes (U(\{\vec{r}_a, s_a\})|\psi_{\mathrm{in}}\rangle)_{\mathcal{C}_O(g)}\,, \tag{4.68}$$

the multi-local projector $\tilde{P}_g(\{\vec{r}_a, s_a\}, \{e_i\})$ projects the state $|\Psi_{\mathrm{in}}\rangle_{\mathcal{C}(g)}$ into

$$\tilde{P}_g(\{\vec{r}_a, s_a\}, \{e_i\})|\Psi_{\mathrm{out}}\rangle_{\mathcal{C}(g)} = |\{\vec{r}_a, s_a\}, \{e_i\}\rangle_{\mathcal{C}(g)\setminus\mathcal{C}_O(g)} \otimes (U(\{\vec{r}_a, s_a\}, \{e_i\})\,|\psi_{\mathrm{in}}\rangle)_{\mathcal{C}_O(g)}\,. \tag{4.69}$$

Thus, we eventually obtain for the output state ρ_{out}

$$\begin{aligned}
\rho_{out} &= \sum_{\{e_i|\, i\in I(g)\}} p(\{e_i\})E(\{e_i\})\tilde{P}_g E(\{e_i\})^\dagger \otimes \Big(U(\{\vec{r}_a, s_a\}, \{e_i\})\rho_{\mathrm{in}}^{\log} U(\{\vec{r}_a, s_a\}, \{e_i\})^\dagger\Big)_{\mathcal{C}_O(g)} \\
&= P_g(\{\vec{r}_a, s_a\}) \otimes \sum_{\{e_i|\, i\in I(g)\}} p(\{e_i\}) \Big(U(\{\vec{r}_a, s_a\}, \{e_i\})\rho_{\mathrm{in}}^{\log} U(\{\vec{r}_a, s_a\}, \{e_i\})^\dagger.\Big)_{\mathcal{C}_O(g)}
\end{aligned} \tag{4.70}$$

Therein, $P_g(\{\vec{r}_a, s_a\})$ denotes the density operator of the qubits in $\mathcal{C}(g)\setminus\mathcal{C}_O(g)$ after applying the operator $P_g(\{\vec{r}_a, s_a\})$ describing the measurements on these qubits.

With (4.70), we find that the noisy channel $\mathcal{T}_g = \prod_{i\in I(g)} \mathcal{T}_i$ on the physical qubits in the location of the gate g is converted into noise on the logical qubits in the state $\rho_{\mathrm{in}}^{\log}$. The imperfect gate simulation amounts to the simulation of an imperfect gate.

Let us now rewrite the unitary transformation $U(\{\vec{r}_a, s_a\}, \{e_i\})$ in (4.70) in a way that is suitable for the remaining part of the proof. For brevity, we omit the set of variables

$\{\vec{r}_a, s_a\}$ labeling the unitary transformations, and henceforth denote $U(\{\vec{r}_a, s_a\}, \{e_i\})$ as $U(\{e_1, e_2, ..., e_{|I(g)|}\})$.

$$\begin{aligned} U(\{e_1, e_2, ..., e_{|I(g)|}\}) &= U(\{e_1, e_2, ..., e_{|I(g)|}\})U(\{0, e_2, ..., e_{|I(g)|}\})^{-1} \\ &\quad U(\{0, e_2, ..., e_{|I(g)|}\})U(\{0, 0, e_3, ..., e_{|I(g)|}\})^{-1}... \\ &\quad U(\{0, ..., 0, e_{|I(g)|}\})U(\{0, ..., 0\})^{-1}\, U(\{0, ..., 0\}). \end{aligned} \tag{4.71}$$

Therein, we define

$$U_{\Sigma,i}(\{e_i, ..., e_{|I(g)|}\}) := U(\{0, ..., 0, e_i, e_{i+1}, ..., e_{|I(g)|}\})U(\{0, ..., 0, e_{i+1}, ..., e_{|I(g)|}\})^{-1}, \tag{4.72}$$

such that

$$U(\{e_1, e_2, ..., e_{|I(g)|}\}) = \left(\prod_{i\in I(g)} U_{\Sigma,i}(\{e_i, ..., e_{|I(g)|}\})\right) U(\{0, ..., 0\}). \tag{4.73}$$

The temporal ordering in the product in (4.73) is as in (4.71). Let us, at this point, briefly discuss whether the four-qubit relocated bond errors cause any trouble. As shown in (4.70), the errors e_i, $i \in I(g)$, affect the logical processing because the transformation $U(\{\vec{r}_a, s_a\}, \{e_i\})$ applied to the register state $\rho_{\text{in}}^{\log}$ depends on them. This is the case regardless of whether e_i is a site- or bond error, ordinary or relocated. As we shall see in eq. (4.91), for the fault-tolerance proof it is only relevant that the transformations $U_{\Sigma,i}(\{e_i, ..., e_{|I(g)|}\})$, defined in (4.72) on the basis of the transformations $U(\{\vec{r}_a, s_a\}, \{e_i\})$, are unitary for all sets of errors $\{e_1, ..., e_{|I(g)|}\}$, and are thus physical operations.

To quantify the gate error we need an operator norm. In reference to theorem (4.36), in this proof we will use a norm $\|\cdot\|$ which obeys the relations (4.34), such as the diamond norm [73]. Throughout the proof we will make use of these special norm properties, and two further properties which are derived thereof and the basic properties that a norm satisfies by definition [74],

$$\|\mathcal{T}\| \geq 0, \text{ where } \|\mathcal{T}\| = 0 \Longleftrightarrow \mathcal{T} = 0, \tag{4.74a}$$

$$\|\lambda\mathcal{T}\| = |\lambda|\,\|\mathcal{T}\|, \tag{4.74b}$$

$$\|\mathcal{T}_r + \mathcal{T}_s\| \leq \|\mathcal{T}_r\| + \|\mathcal{T}_s\|. \tag{4.74c}$$

First, if $\mathcal{T}_U$ corresponds to a unitary transformation, then

$$\|\mathcal{T} - \mathcal{T}_U\| = \left\|\mathcal{T} \circ {\mathcal{T}_U}^{-1} - \mathbb{1}\right\|. \tag{4.75}$$

This holds because, with (4.34b) and (4.34d), $\|\mathcal{T} - \mathcal{T}_U\| = \|(\mathcal{T} \circ {\mathcal{T}_U}^{-1} - \mathbb{1}) \circ \mathcal{T}_U\| \leq \|\mathcal{T} \circ {\mathcal{T}_U}^{-1} - \mathbb{1}\|\,\|\mathcal{T}_U\| = \|\mathcal{T} \circ {\mathcal{T}_U}^{-1} - \mathbb{1}\|$. In the same way but opposite direction, $\|\mathcal{T} \circ {\mathcal{T}_U}^{-1} - \mathbb{1}\| \leq \|\mathcal{T} - \mathcal{T}_U\|$. From both inequalities there follows (4.75).

Second, for sets I with a finite number of elements, the following inequality holds.

$$\left\|\prod_{i\in I} \mathcal{T}_i - \mathbb{1}\right\| \leq \sum_{i\in I} \|\mathcal{T}_i - \mathbb{1}\|. \tag{4.76}$$

For $|I| = 2$, note that $\|\mathcal{T}_2\mathcal{T}_1 - \mathbb{1}\| = \|\mathcal{T}_2\mathcal{T}_1 - \mathcal{T}_1 + \mathcal{T}_1 - \mathbb{1}\| \leq \|(\mathcal{T}_2 - \mathbb{1})\mathcal{T}_1\| + \|\mathcal{T}_1 - \mathbb{1}\| \leq \|\mathcal{T}_2 - \mathbb{1}\| + \|\mathcal{T}_1 - \mathbb{1}\|$. Now, (4.76) follows by induction.

Having these notions introduced, we now prove that any bound η_c'' on the gate errors can be matched with constant nonzero bounds on the physical errors in a $\mathrm{QC}_{\mathcal{C}}$ gate simulation. The error of the simulated gate g is

$$\eta_g = \left\| \left(\sum_{\{e_i|\, i\in I(g)\}} p(\{e_i\}) \left[U(\{e_1, e_2, ..., e_{|I(g)|}\})\right] \right) - \left[U(\{0, ..., 0\})\right] \right\|. \tag{4.77}$$

Substituting (4.73) into (4.77), and using (4.75), we obtain

$$\eta_g = \left\| \left(\sum_{\{e_i\}|\, i\in I(g)} p(\{e_i\}) \left[\prod_{i\in I(g)} U_{\Sigma,i}(\{e_i, ..., e_{|I(g)|}\}) \right] \right) - \left[\mathbb{1}\right] \right\|. \tag{4.78}$$

For better illustration, we do the proof first for the Clifford gates where it is technically easier. However, the essence of the proof is covered by the case of Clifford gates already. There, a simplification arises because for simulations of Clifford gates the unitary transformations $U_{\Sigma,i}(\{e_i, ..., e_{|I(g)|}\}, \{\vec{r}_a, s_a\})$ are, in fact, the byproduct operators introduced earlier. They are elements of the Pauli group, and depend solely on the error parameter e_i but neither on e_j for $j \neq i$, nor on $\{\vec{r}_a, s_a\}$),

$$U_{\Sigma,i}(\{e_i, ..., e_{|I(g)|}\}, \{\vec{r}_a, s_a\}) = U_{\Sigma,i}(e_i), \qquad \text{(for Clifford gates)}. \tag{4.79}$$

The logical errors $U_{\Sigma,i}(e_i)$ induced by the physical errors $E_i(e_i)$ are all independent, and, with (4.65),

$$\sum_{\{e_i\}|\, i\in I(g)} p(\{e_i\}) \left[\prod_{j\in I(g)} U_{\Sigma,j}(e_j) \right] = \prod_{i\in I(g)} \sum_{e_i} p_i(e_i) \left[U_{\Sigma,i}(e_i)\right]. \tag{4.80}$$

We now insert (4.80) into (4.78) and obtain, using (4.76),

$$\eta_g \leq \sum_{i\in I(g)} \left\| \left(\sum_{e_i} p_i(e_i) \left[U_{\Sigma,i}(e_i)\right] \right) - \left[\mathbb{1}\right] \right\|. \tag{4.81}$$

To discuss this expression further let us split the sum in two parts $I(g) = I_{\mathrm{site}}(g) \cup I_{\mathrm{bond}}(g)$, one for the errors on sites, $i \in I_{\mathrm{site}}(g)$, and one for the errors on bonds, $i \in I_{\mathrm{bond}}(g)$. For the site errors we have, with (4.25) and (4.74b), and with $U_{\Sigma,i}(0) = \mathbb{1}$ for all $i \in I(g)$,

$$\begin{aligned} \sum_{i\in I_{\mathrm{site}}(g)} \left\| \left(\sum_{e_i=0}^{3} p_i(e_i) \left[U_{\Sigma,i}(e_i)\right] \right) - \left[\mathbb{1}\right] \right\| &= p_1 \sum_{i\in I_{\mathrm{site}}(g)} \left\| \left(\frac{1}{3} \sum_{e_i=1}^{3} \left[U_{\Sigma,i}(e_i)\right] \right) - \left[\mathbb{1}\right] \right\| \\ &\leq 2p_1 |I_{\mathrm{site}}(g)|. \end{aligned} \tag{4.82}$$

By an analogous argument, the contribution from to the gate error bound coming from the bond errors is

$$\sum_{i\in I_{\mathrm{bond}}(g)} \left\| \left(\sum_{e_i=0}^{1} p_i(e_i) \left[U_{\Sigma,i}(e_i) \right] \right) - \left[\mathbb{1} \right] \right\| \leq 2p_2 |I_{\mathrm{bond}}(g)|. \tag{4.83}$$

Combining (4.82) and (4.83) with (4.81) we obtain

$$\eta_g \leq 2p_1 \, |I_{\mathrm{site}}(g)| + 2p_2 \, |I_{\mathrm{bond}}(g)|. \tag{4.84}$$

Note that the obtained upper bound (4.84) on the gate error η_g is linear in the probabilities p_1, p_2 of the individual one- and two-qubit errors. Thus, for any threshold value η_c'' of the gate error there exist values for the physical error probabilities p_1, p_2 such that $\eta_g(p_1, p_2) \leq \eta_c''$. Further, the obtained upper bound on the gate error is linear in the size $|I_{\mathrm{site}}(g)|$, $|I_{\mathrm{bond}}(g)|$ of the error location corresponding to the sub-cluster $\mathcal{C}(g)$ on which the gate g is realized.

Let us now restate the proof for the general case. We define

$$\eta_g[k] = \left\| \left(\sum_{\{e_i \,|\, i\geq k\}} \prod_{i=k}^{|I(g)|} p(e_i) \left[\prod_{j=k}^{|I(g)|} U_{\Sigma,j}(\{e_j, \ldots, e_{|I(g)|}\}) \right] \right) - \left[\mathbb{1} \right] \right\|. \tag{4.85}$$

Note that $\eta_g[1] = \eta_g$. Further holds the inequality

$$\eta_g[k] - \eta_g[k+1] \leq 2p_k, \tag{4.86}$$

where p_k is the error probability for the kth error channel,

$$p_k = \sum_{e_k>0} p_k(e_k) = 1 - p_k(0). \tag{4.87}$$

Proof of (4.86). The difference $\eta_g[k] - \eta_g[k+1]$ may be bounded from above in the following way

$$\eta_g[k] - \eta_g[k+1] \;=\; \left\| \left(\sum_{\{e_i \,|\, i\geq k\}} \prod_{i=k}^{|I(g)|} p(e_i) \left[\prod_{j=k}^{|I(g)|} U_{\Sigma,j}(\{e_j, \ldots, e_{|I(g)|}\}) \right] \right) - \left[\mathbb{1} \right] \right\| - \eta_g[k+1],$$

which is just the definition. Into the r.h.s of the above equation we insert a zero,

$$\begin{aligned} \eta_g[k] - \eta_g[k+1] \;=\; & \left\| \left(\sum_{\{e_i \,|\, i\geq k\}} \prod_{i=k}^{|I(g)|} p(e_i) \left[\prod_{j=k}^{|I(g)|} U_{\Sigma,j}(\{e_j, \ldots, e_{|I(g)|}\}) \right] \right) - \right. \\ & - \left(\sum_{\{e_i \,|\, i\geq k+1\}} \prod_{i=k+1}^{|I(g)|} p(e_i) \left[\prod_{j=k+1}^{|I(g)|} U_{\Sigma,j}(\{e_j, \ldots, e_{|I(g)|}\}) \right] \right) + \\ & + \left(\sum_{\{e_i \,|\, i\geq k+1\}} \prod_{i=k+1}^{|I(g)|} p(e_i) \left[\prod_{j=k+1}^{|I(g)|} U_{\Sigma,j}(\{e_j, \ldots, e_{|I(g)|}\}) \right] \right) - \left[\mathbb{1} \right] \Bigg\| - \\ & - \eta_g[k+1]. \end{aligned} \tag{4.88}$$

In (4.88), we now pair the first and the second two terms in the norm together and use the triangle inequality (4.74c). The summand arising from the latter two terms is $\eta_g[k+1]$, and thus cancels. The remaining term we rewrite as

$$\begin{aligned}\eta_g[k]-\eta_g[k+1] \;\leq\; & \Bigg\| \sum_{\{e_i|\, i\geq k+1\}} \left(\sum_{e_k} p_k(e_k) \left[U_{\Sigma,k}(\{e_k,\dots,e_{|I(g)|}\})\right] - \left[\mathbb{1}\right] \right) \\ & \prod_{i=k+1}^{|I(g)|} p(e_i) \left[\prod_{j=k+1}^{|I(g)|} U_{\Sigma,j}(\{e_j,\dots,e_{|I(g)|}\}) \right] \Bigg\| \end{aligned} \tag{4.89}$$

In (4.89) we may us again (4.74c), and (4.74b), to obtain

$$\eta_g[k]-\eta_g[k+1] \;\leq\; \sum_{\{e_i|i\geq k+1\}} \prod_{i=k+1}^{|I(g)|} p(e_i) \left\| \sum_{e_k} p_k(e_k) \left[U_{\Sigma,k}(\{e_k,\dots,e_{|I(g)|}\})\right] - \left[\mathbb{1}\right] \right\| \tag{4.90}$$

Let us now discuss the expression within the norm in (4.90), subsequently denoted in short as $\|\Delta\mathcal{T}_k\|$. The total error probability p_k for the physical error at the location k is given by (4.87). Since $U_{\Sigma,k}(0)=\mathbb{1}$, the identity appears twice in the expression for $\|\Delta\mathcal{T}_k\|$ and the two terms largely cancel,

$$\begin{aligned} \|\Delta\mathcal{T}_k\| &= \left\| \sum_{e_k} p_k(e_k) \left[U_{\Sigma,k}(\{e_k,\dots,e_{|I(g)|}\})\right] - \left[\mathbb{1}\right] \right\| \\ &= \left\| \left(\sum_{e_k>0} p_k(e_k) \left[U_{\Sigma,k}(\{e_k,\dots,e_{|I(g)|}\})\right] \right) - p_k\left[\mathbb{1}\right] \right\| \\ &= p_k \left\| \left(\sum_{e_k>0} \frac{p_k(e_k)}{p_k} \left[U_{\Sigma,k}(\{e_k,..,e_{|I(g)|}\})\right] \right) - \left[\mathbb{1}\right] \right\| \\ &\leq p_k \left(\sum_{e_k>0} \frac{p_k(e_k)}{p_k} \left\| \left[U_{\Sigma,k}(\{e_k,\dots,e_{|I(g)|}\})\right] \right\| + \left\| \left[\mathbb{1}\right] \right\| \right) \qquad (4.91) \\ &= 2p_k. \qquad (4.92) \end{aligned}$$

To obtain (4.91) we have used the triangle inequality (4.74c). For (4.92), we have inserted (4.87) in (4.91), and further have used (4.34d). If we insert (4.92) into (4.90), we obtain

$$\eta_g[k]-\eta_g[k+1] \leq 2p_k \sum_{\{e_i|i\geq k+1\}} \prod_{i=k+1}^{|I(g)|} p(e_i), \tag{4.93}$$

and thus $\eta_g[k]-\eta_g[k+1]\leq 2p_k$, which proves (4.86).

From the definition of $\eta_g[k]$ (4.85) one obtains for $\eta_g[|I(g)|]$ the bound $\eta_g[|I(g)|]\leq 2p_{|I(g)|}$, and hence by induction for the gate error $\eta_g=\eta_g[1]$,

$$\eta_g \leq 2 \sum_{i\in I(g)} p_i. \tag{4.94}$$

In the model 2 for the physical errors we have two types of errors, site errors $\mathcal{T}^{(a_i)}[p_1]$, $i \in I_{\mathrm{site}}(g)$, and bond errors $\mathcal{T}_{I,2}^{(a,b)_j}[p_2]$, $j \in I_{\mathrm{bond}}(g)$. The respective error probabilities are $p_i = p_1$ for all $i \in I_{\mathrm{site}}(g)$, and $p_j = p_2$ for all $i \in I_{\mathrm{bond}}(g)$, such that we finally obtain for the gate error

$$\eta_g \leq 2p_1|I_{\mathrm{site}}(g)| + 2p_2|I_{\mathrm{bond}}(g)|, \tag{4.95}$$

in accordance with (4.84).

To simulate all gates $g \in \mathcal{G}'' \cup \{SWAP\}$ with a gate error $\eta_g \leq \eta_c''$ it is sufficient to require that

$$2p_1|I_{\mathrm{site}}(g)| + 2p_2|I_{\mathrm{bond}}(g)| \leq \eta_c'', \quad \forall\, g \in \mathcal{G}'' \cup \{SWAP\}. \tag{4.96}$$

The set of gates $\mathcal{G}'' \cup \{SWAP\}$ is finite, and therefore the condition (4.96) can be obeyed with positive p_1, p_2 for all nonzero η_c''. Thus, the $\mathrm{QC}_{\mathcal{C}}$ can simulate a one-dimensional quantum computer [44] with gate error threshold η_c''. As the one-dimensional quantum computer is fault-tolerant by the theorem quoted in (4.36), the $\mathrm{QC}_{\mathcal{C}}$ is fault-tolerant, too. There exist positive error thresholds for fault-tolerant quantum computation with the one-way quantum computer. □

4.4 Checksums

Let us for illustration state the bounds (4.95) on the gate errors for the set or gates displayed in Fig. 4.2. They read $\eta_{CNOT} \leq 82p_1 + 130p_2$, $\eta_{\mathrm{wire}} = \eta_{Rot} \leq 38p_1 + 56p_2$, and $\eta_{SWAP} \leq 76p_1 + 114p_2$. Thus, the thresholds for $\mathrm{QC}_{\mathcal{C}}$-fault-tolerance are more stringent than the respective network thresholds by a factor of approximately 10^2. Already the error thresholds for one-dimensional networks appear to be rather small [64, 44]. Therefore, to simulate network quantum computers with next-neighbor interactions seems impractical for fault-tolerant quantum computation with the $\mathrm{QC}_{\mathcal{C}}$. Instead, it appears rewarding to look for methods of error identification that the $\mathrm{QC}_{\mathcal{C}}$ provides naturally. A concept for a fault-tolerant $\mathrm{QC}_{\mathcal{C}}$ which may be capable of improving the error thresholds rather substantially is the use of quantum correlations to obtain checksums of measurement outcomes.

4.4.1 A first example

For certain gates and sub-circuits it occurs that there exist multiple ways to infer the byproduct operator from the outcomes of the measurements which implement this gate or sub-circuit. An example for such a situation is the multi-qubit swap gate as redisplayed in Fig. 4.4. There, the byproduct operator $U_{\Sigma}, \mathrm{swap}$ is of the form (2.63) and has a z-part

$$U_{\Sigma,\mathrm{swap}}^{[1,z]} = \left(\sigma_z^{[1]}\right)^{\gamma_{1,z}}, \tag{4.97}$$

with

$$\gamma_{1,z} = \sum_{a \in J_{\mathrm{upper}}} s_a \bmod 2. \tag{4.98}$$

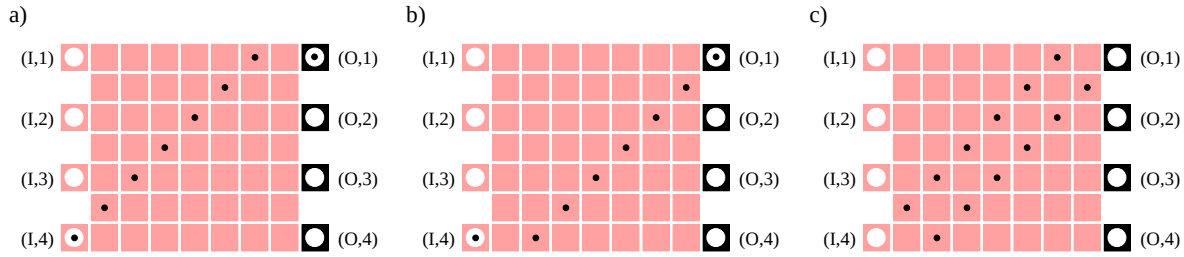


Figure 4.4: Checksums for error identification. In a), the cluster qubits whose measurement outcomes contribute to the $\sigma_z^{[1]}$-part of the byproduct operator $U_{\Sigma,\mathrm{swap}}$ are marked by dots "•". In b) an alternative choice of such cluster qubits is shown. As there are two ways to infer the $\sigma_z^{[1]}$-contribution to the byproduct operator, the consistency condition (4.101) holds. The cluster qubits whose measurement outcome contributes to this checksum are displayed in c).

The set J_{upper} over which the summation index in (4.98) runs consists of all those cluster qubits which are marked by a dot "•" in Fig. 4.4 a.

Alternatively, one may infer the $\sigma_z^{[1]}$-contribution to the byproduct operator $U_{\Sigma,\mathrm{swap}}$ from the measurement outcomes of cluster qubits $a \in J_{\mathrm{lower}}$, which are marked by dots in Fig. 4.4 b, i.e.

$$\gamma_{1,z} = \sum_{a \in J_{\mathrm{lower}}} s_a \bmod 2. \tag{4.99}$$

Both expressions for $\gamma_{1,z}$, (4.98) and (4.99) must agree for all possible sets of measurement outcomes, $\sum_{a\in J_{\mathrm{upper}}} s_a \bmod 2 = \sum_{a\in J_{\mathrm{lower}}} s_a \bmod 2$, such that, with the definition

$$J_{\mathrm{cs}} = J_{\mathrm{upper}} \cup J_{\mathrm{lower}} \setminus (J_{\mathrm{upper}} \cap J_{\mathrm{lower}}) \tag{4.100}$$

the consistency condition

$$\sum_{a \in J_{\mathrm{cs}}} s_a = 0 \bmod 2 \tag{4.101}$$

must hold. If it does not hold in an experimental realization of the gate, then an error must have occurred in the implementation procedure. The expression on the l.h.s. of eq. (4.101) is called a *checksum*. If the measurement pattern to realize a gate is designed carefully, then there exists a set of independent checksums for this gate, providing an error syndrome. If the set of checksums is sufficiently large, a possible error can be identified from the thereby provided error syndrome.

The consistency condition (4.101) can be inferred directly from the quantum correlations (2.1) defining a cluster state. Let us consider a gate operation in a procedure according to Scheme 1 described in Section 2.2.4, i.e. preparation of the state $|\psi_{\mathrm{in}}\rangle_{\mathcal{C}_I(\mathrm{swap})} \otimes |+\rangle_{\mathcal{C}_M(\mathrm{swap})\cup\mathcal{C}_O(\mathrm{swap})}$, entangling this state via the Ising interaction, thereby obtaining the state $|\Psi_{\mathcal{E}}\rangle_{\mathcal{C}(\mathrm{swap})}$, and performing the σ_x-measurements $P_X^{(\mathcal{C}_I(\mathrm{swap})\cup\mathcal{C}_M(\mathrm{swap}))}$ on the cluster $\mathcal{C}_I(\mathrm{swap}) \cup \mathcal{C}_M(\mathrm{swap})$, obtaining $|\Psi_{\mathrm{out}}\rangle_{\mathcal{C}(\mathrm{swap})} = |m\rangle_{\mathcal{C}_I(\mathrm{swap})\cup\mathcal{C}_M(\mathrm{swap})} \otimes |\psi_{\mathrm{out}}\rangle_{\mathcal{C}_O(\mathrm{swap})}$. The state $|\Psi_{\mathcal{E}}\rangle_{\mathcal{C}(\mathrm{swap})}$ is not a cluster state as it depends on the quantum input $|\psi_{\mathrm{in}}\rangle$. It does,

however, still fulfill the cluster state eigenvalue equations (2.1) for all those sites a which do not belong to the input set $\mathcal{C}_I(\mathrm{swap})$,

$$K^{(a)}|\Psi_{\mathcal{E}}\rangle_{\mathcal{C}(\mathrm{swap})} = |\Psi_{\mathcal{E}}\rangle_{\mathcal{C}(\mathrm{swap})}, \quad \forall a \in \mathcal{C}_M(\mathrm{swap}) \cup \mathcal{C}_O(\mathrm{swap}). \tag{4.102}$$

Now note that the set J_{cs} (4.100) does not contain qubits in $\mathcal{C}_I(\mathrm{swap})$, such that all cluster state correlations $K^{(a)}$ with $a \in J_{\mathrm{cs}}$ remain valid for $|\Psi_{\mathcal{E}}\rangle_{\mathcal{C}(\mathrm{swap})}$. Further, we observe that

$$\prod_{a\in J_{\mathrm{cs}}} K^{(a)} = \bigotimes_{a\in J_{\mathrm{cs}}} \sigma_x^{(a)}. \tag{4.103}$$

All the σ_z-contributions to the r.h.s. of (4.103) cancel. With (2.1) and (4.103) one finds

$$\bigotimes_{a\in J_{\mathrm{cs}}} \sigma_x^{(a)} \, |\Psi_{\mathcal{E}}\rangle_{\mathcal{C}(\mathrm{swap})} = |\Psi_{\mathcal{E}}\rangle_{\mathcal{C}(\mathrm{swap})}. \tag{4.104}$$

This implies for the state $|\Psi_{\mathrm{out}}\rangle_{\mathcal{C}(\mathrm{swap})} = n_P^{-1} \, P_X^{(\mathcal{C}_I(\mathrm{swap})\cup\mathcal{C}_M(\mathrm{swap}))}|\Psi_{\mathcal{E}}\rangle_{\mathcal{C}(\mathrm{swap})}$, n_P being a non-zero norm factor, that

$$\begin{aligned} |\Psi_{\mathrm{out}}\rangle_{\mathcal{C}(\mathrm{swap})} &= n_P^{-1} \, P_X^{(\mathcal{C}_I(\mathrm{swap})\cup\mathcal{C}_M(\mathrm{swap}))} \left(\bigotimes_{a\in J_{\mathrm{cs}}} \sigma_x^{(a)} \right) |\Psi_{\mathcal{E}}\rangle_{\mathcal{C}(\mathrm{swap})} \\ &= (-1)^{\sum_{a\in J_{\mathrm{cs}}} s_a} n_P^{-1} \, P_X^{(\mathcal{C}_I(\mathrm{swap})\cup\mathcal{C}_M(\mathrm{swap}))}|\Psi_{\mathcal{E}}\rangle_{\mathcal{C}(\mathrm{swap})} \\ &= (-1)^{\sum_{a\in J_{\mathrm{cs}}} s_a}|\Psi_{\mathrm{out}}\rangle_{\mathcal{C}(\mathrm{swap})}. \end{aligned} \tag{4.105}$$

As $|\Psi_{\mathrm{out}}\rangle_{\mathcal{C}(\mathrm{swap})} \neq 0$, the constraint $\sum_{a\in J_{\mathrm{cs}}} s_a = 0 \bmod 2$, (4.101), must hold. This is a first example of a checksum.

4.4.2 The encoded CNOT gate on the Steane code

In Fig. 4.5 CNOT gate on encoded qubits is shown, where the seven-qubit Steane code is used to encode the logical qubits. The code has twelve stabilizer generators for the two encoded qubits, and in addition there exist 24 checksums. One of the checksums are shown in Fig. 4.5. As has been shown in [75], under the idealized assumption that the subsequent measurements of the stabilizer are perfect, the displayed encoded CNOT gate operates fault-tolerantly. That is, using both the checksums obtained from the outcomes of the measurements which implement the gate and the outcomes of the code stabilizer measurements, all possible errors can be identified up to equivalence on the code.

To investigate the realistic case where errors occur in the measurement of the code stabilizer as well, it requires a measurement pattern which performs the stabilizer measurement on the $\mathrm{QC}_{\mathcal{C}}$. To date, no such measurement pattern has been tested in combination with an encoded gate. To construct a universal set of fault-tolerant encoded quantum gates for the $\mathrm{QC}_{\mathcal{C}}$ is a subject of further study.

Let us briefly explain the functioning of the encoded CNOT gate. Consider a cluster state $|\phi\rangle_{\mathcal{C}(+)}$ on a cluster $\mathcal{C}(+)$ as shown in Fig. 4.5. The state $|\Psi_{\mathrm{out}}\rangle_{\mathcal{C}(+)}$ obtained after the

measurement of the qubits in $\mathcal{C}_M(+)$ in the bases indicated in Fig. 4.5 obeys the eigenvalue equations

$$\sigma_z^{(I,c_i)}\sigma_z^{(O,c_i)}|\Psi_{\text{out}}\rangle_{\mathcal{C}(+)} = \pm|\Psi_{\text{out}}\rangle_{\mathcal{C}(+)}, \tag{4.106a}$$

$$\sigma_x^{(I,c_i)}\sigma_x^{(O,c_i)}\sigma_x^{(O,t_i)}|\Psi_{\text{out}}\rangle_{\mathcal{C}(+)} = \pm|\Psi_{\text{out}}\rangle_{\mathcal{C}(+)}, \tag{4.106b}$$

$$\sigma_z^{(I,t_i)}\sigma_z^{(O,c_i)}\sigma_z^{(O,t_i)}|\Psi_{\text{out}}\rangle_{\mathcal{C}(+)} = \pm|\Psi_{\text{out}}\rangle_{\mathcal{C}(+)}, \tag{4.106c}$$

$$\sigma_x^{(I,t_i)}\sigma_x^{(O,t_i)}|\Psi_{\text{out}}\rangle_{\mathcal{C}(+)} = \pm|\Psi_{\text{out}}\rangle_{\mathcal{C}(+)}. \tag{4.106d}$$

Therein, the labels c_i, t_i denote the ith control- and target qubit, respectively, and the labels I, O indicate whether the respective cluster qubit is from $\mathcal{C}_I(+)$ or $\mathcal{C}_O(+)$.

From the eigenvalue equations (4.106) it follows via Theorem 1 that the measurement pattern implements, modulo a byproduct operator, a bitwise CNOT from control to target. For the seven-qubit Steane code the encoded CNOT operation is transversal, the bitwise CNOT-operations on the bare qubits result in an encoded CNOT operation on the encoded qubits.

Let us conclude with a general remark. We found in this section that the checksums supplement the code stabilizer measurements in error identification for fault-tolerant quantum computation. For the $\text{QC}_{\mathcal{C}}$, the sub-circuits for fault-tolerant code stabilizer measurement will be realized as specific measurement patterns on the cluster, and therefore the outcomes of the stabilizer measurements will again appear as checksums. Further, the defining relations of the readout bits of a quantum computation, given by the components of the x-part of $\mathbf{I}$ in (3.73), have the structure of checksums. Thus, we find that the checksums –derived from the quantum correlations of the cluster state– are at work everywhere in the $\text{QC}_{\mathcal{C}}$. Some are used to infer the computational result and others to stabilize the computation.

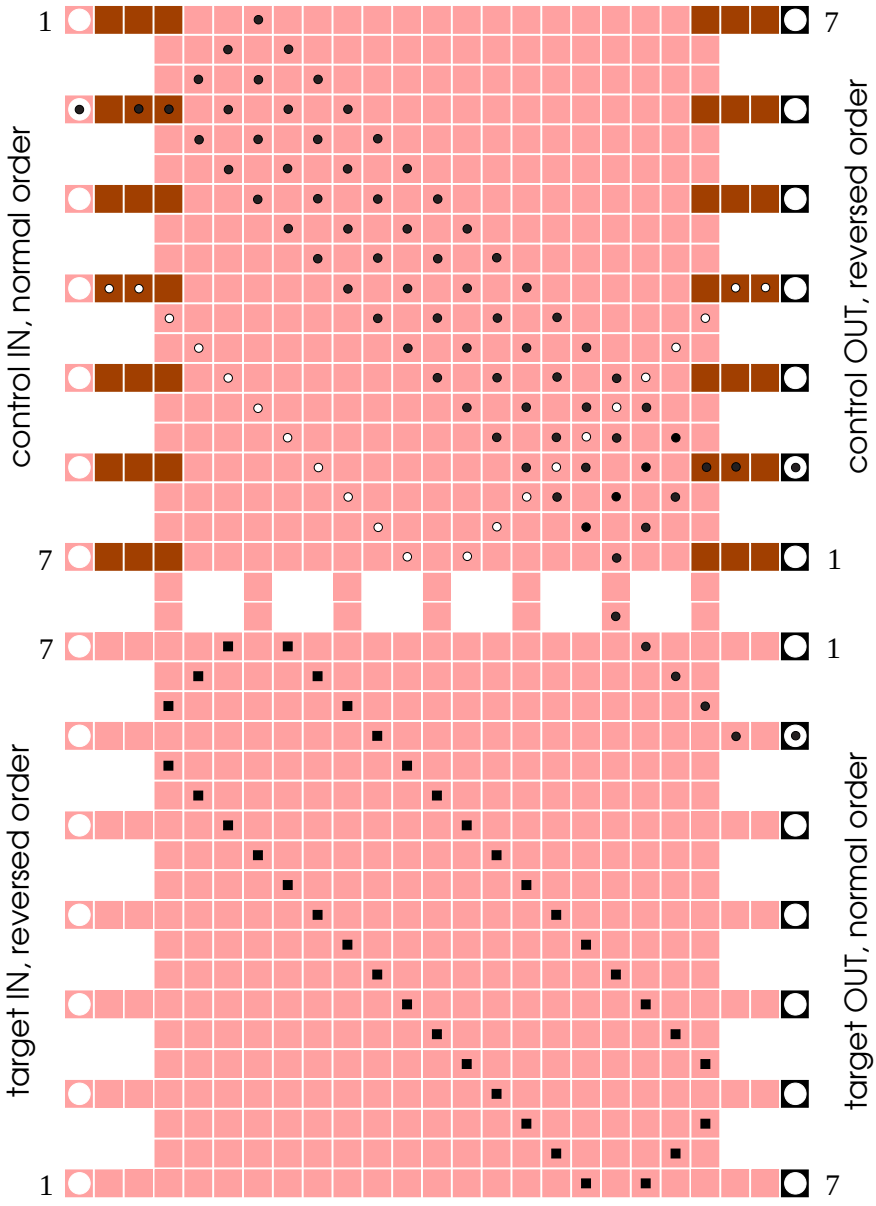


Figure 4.5: CNOT gate on qubits encoded with the seven-qubit Steane code. Squares in light and dark gray denote cluster qubits measured in the eigenbasis of σ_x and σ_y, respectively. The full circles "•" denote the correlation centers of the initial cluster state quantum correlation which leads, after measurement of the qubits in $\mathcal{C}_M(+)$, to eq. (4.106b) for qubit c_2. The circles "○" denote the correlation centers for the cluster state correlation which yields to eq. (4.106a) for the qubit c_4. The filled boxes denote the correlation centers for one of the checksums.

Chapter 5

Conclusion and outlook

In this thesis we have described the one-way quantum computer, a scheme of quantum computation that consists entirely of one-qubit measurements on a highly entangled multi-particle state. This multi-particle state, the cluster state, is a universal resource for quantum computation and provides in advance all the entanglement needed in the computational process. Each quantum circuit can be imprinted on the cluster by the one-qubit measurements.

A proper quantum computer needs to be universal, scalable and fault-tolerant. For the $\mathrm{QC}_{\mathcal{C}}$, we have proven universality in Chapter 2 and fault-tolerance in Chapter 4. Scaling has been discussed in Section 2.2.8. A quantum algorithm which is polynomial in its temporal, spatial and operational resources within the network picture is polynomial on the $\mathrm{QC}_{\mathcal{C}}$ as well. The scaling capabilities of the class of systems suitable for the implementation of the $\mathrm{QC}_{\mathcal{C}}$ are very promising. Specifically, the resource cluster state can be created by the homogeneous Ising interaction, and in this way the operational effort to create the quantum resource is independent of the system size. Furthermore, the one-qubit measurements that are used to imprint the quantum circuit on the cluster remain one-qubit measurements no matter what size the system is scaled up to. The $\mathrm{QC}_{\mathcal{C}}$ avoids by construction the difficult task of performing long-range particle-selective interactions. Nevertheless, one should not underestimate the experimental difficulties that may arise in an experimental realization of the $\mathrm{QC}_{\mathcal{C}}$.

The $\mathrm{QC}_{\mathcal{C}}$ does not only fulfill the essential requirements. We have also found that the $\mathrm{QC}_{\mathcal{C}}$ is based on a computational model different from the network model of quantum computation. This model is described in Chapter 3. The formal description of the $\mathrm{QC}_{\mathcal{C}}$ is based on primitive quantities of which the most important are the sets $Q_t \subset \mathcal{C}$ of cluster qubits defining the temporal ordering of measurements on the cluster state, and the binary valued information flow vector $\mathbf{I}(t)$ which is the carrier of the algorithmic information.

The $\mathrm{QC}_{\mathcal{C}}$ has no quantum input, no quantum output and no quantum register, and the unitary gates from some universal set are not the elementary building blocks of $\mathrm{QC}_{\mathcal{C}}$-quantum algorithms. All that is processed with the $\mathrm{QC}_{\mathcal{C}}$ are the outcomes of one-qubit measurements and thus processing of information exists only at the classical level. The byproduct operators, which were regarded as a mere byproduct in the network description

of the $QC_{\mathcal{C}}$, caused by the randomness of the measurement outcomes and the need to account for it, turn out to be the central objects for information processing with the $QC_{\mathcal{C}}$.

Despite the classical nature of information processing, the $QC_{\mathcal{C}}$ is genuinely quantum mechanical as it uses a highly entangled cluster state as the central physical resource. What enables the $QC_{\mathcal{C}}$ are the quantum correlations exhibited by the cluster state. A $QC_{\mathcal{C}}$-computation proceeds by measuring a subset of these correlations in local measurements.

In Chapter 4 we have shown that there exist nonzero error thresholds for fault-tolerant computation with the $QC_{\mathcal{C}}$. The technique used was to trace back $QC_{\mathcal{C}}$-fault-tolerance to the fault-tolerance of a network quantum computer with next-neighbor and local gates only. The fault-tolerance of such a device has been established previously [44, 64]. However, what makes the proof conceptually simple, on the other hand makes the according realization hard: the obtained error thresholds are extremely small. To simulate one-dimensional quantum computers (which use next-neighbor and local gates only) on the $QC_{\mathcal{C}}$ is no practicable way of making the $QC_{\mathcal{C}}$ fault-tolerant. As a first step towards better bounds, the concept of checksums has been introduced to the $QC_{\mathcal{C}}$ [75]. The checksums use cluster-state quantum correlations for error detection and identification, and may become an element in more efficient techniques for fault-tolerant quantum computation with the $QC_{\mathcal{C}}$.

For the future, a self-evident task is to figure out feasible and adequate methods for fault-tolerant $QC_{\mathcal{C}}$-computation and finally to come up with better error thresholds. An advantage of the $QC_{\mathcal{C}}$ over mainly unitary quantum logic networks is that operations can be parallelized to a larger degree, such that i.e. circuits for code stabilizer measurements are executed in a single time step. Fast error correction or -identification is critical for fault-tolerant quantum computation. Additionally, in appropriately designed circuits the number of checksums is enlarged beyond those provided by the code stabilizer (if a stabilizer code is used) and in this way the means for identification of errors are improved. A disadvantage of the $QC_{\mathcal{C}}$ as compared to a network quantum computer is that it has an overhead in the number of required quantum systems and is therefore more prone to error. It is presently an open question which of the effects –faster feedback and more capability for error identification, or higher sensitivity to physical errors– dominates. For the universal $QC_{\mathcal{C}}$, which was introduced as a simulator of quantum logic networks in Chapter 2, a network-independent formulation has been given in Chapter 3. The same task stands out for the fault-tolerant version of the $QC_{\mathcal{C}}$.

Returning to universal computation with perfect means one may ask whether the description of the $QC_{\mathcal{C}}$, with the computational model as derived in Chapter 3, is complete. Despite all the facts that we have collected about cluster state quantum correlations, the quantum part of the $QC_{\mathcal{C}}$, which is hidden behind the formal model of information processing, is what requires a better understanding. We may approach this part of the $QC_{\mathcal{C}}$ from both the perspectives of physics and computer science, and find that there are many questions in this context which we cannot answer presently. A physicist may ask, for instance, what condition the quantum correlations of a resource quantum state need to obey such that universal quantum computation by one-particle measurements is possible. The cluster state may be created from a product state via the Ising Interaction. What about states that can be created, say, with the Heisenberg interaction?

From the perspective of computer science one may ask how to design quantum algorithms in the $\mathrm{QC}_{\mathcal{C}}$-scheme directly, instead of translating existing network algorithms. Presently, we do not know this for the general case. Nevertheless, we identified some elements for circuit construction that may be used advantageously in future algorithmic methods. In Section 2.3, for example, we decomposed circuits into generalized rotations instead of into gates on a small number of qubits. For the $\mathrm{QC}_{\mathcal{C}}$, the generalized rotations are simple, no matter on how many qubits they act. They are, however, still unitary gates applied to a quantum register - which in Section 3.1 was found not to be a suitable construct for the $\mathrm{QC}_{\mathcal{C}}$. Further, we showed in Section 3.6 that from every quantum algorithm its Clifford part can be removed and only the remainder requires quantum resources, provided by an algorithm-specific graph state. These observations may, among others, provide starting points for investigations aiming at new tools for quantum algorithms. A well developed model for computation, such as the quantum logic network model, provides efficient algorithmic techniques and an intuition for what makes computation fast. For the one-way quantum computer we do not have this tool box and intuition yet.

The computational model of the $\mathrm{QC}_{\mathcal{C}}$ as derived in Chapter 3, the search for new construction techniques for quantum algorithms, and the investigation of the cluster state quantum correlations are all paths that lead one to address the question: "If the quantum gates are removed from the description of the $\mathrm{QC}_{\mathcal{C}}$ altogether, what replaces them as its elementary building blocks?". From the viewpoint of resources, one may regard the one-qubit measurements as the constituents. This can, however, not be the whole answer, since it helps little in understanding the structure of $\mathrm{QC}_{\mathcal{C}}$-algorithms. The elements of the $\mathrm{QC}_{\mathcal{C}}$ –as those of any model of computation– have to come with a composition principle.

We have stated earlier that the $\mathrm{QC}_{\mathcal{C}}$ separates the logic and the physics of quantum computation. On the other hand, the above raised question about the elementary constituents of the $\mathrm{QC}_{\mathcal{C}}$ is motivated from both the computer science and the physics perspective, and –as we have argued– also its answer will involve elements from both constituting sciences of the field of quantum information. In this way, the one-way quantum computer gives physics and computer science an object of joint study.

Appendix A

$\mathrm{QC}_{\mathcal{C}}$-computation in the presence of classically correlated noise

In Section 4.2 we based the error model 2 on uncorrelated physical errors, which is a reasonable assumption to start with. However, the fact that the resource cluster state can be created in a constant number of steps independent of the cluster size –which is a great advantage– brings about the side effect that errors in the elementary interactions by which the cluster state is created are, for many physical settings, classically correlated. Therefore it needs to be investigated how classical correlations in the noise affect $\mathrm{QC}_{\mathcal{C}}$-computation. In this appendix we demonstrate that, under the only assumption that the given measurement pattern realizes a $\mathrm{QC}_{\mathcal{C}}$-computation, classical correlations of inequivalent Pauli product errors E_i, E_j, $E_i|\phi\rangle_{\mathcal{C}} \neq E_j|\phi\rangle_{\mathcal{C}}$, have *no effect* on the result of the quantum computation.

We start our discussion with a more detailed version of (3.42). Taking into account the adaption of measurement bases in the process of computation explicitly, the sequence of projections is

$$P^{(\mathcal{C})}(\{s_i\}) = \bigotimes_{a\in\mathcal{C}\setminus Q_0} \left(\sigma_x^{(a)}\right)^{\vartheta_a(\{s_i,\kappa_i\})} \frac{\mathbb{1}^{(a)} + (-1)^{s_a}\vec{r}_a\cdot\vec{\sigma}^{(a)}}{2} \left(\sigma_x^{(a)}\right)^{\vartheta_a(\{s_i,\kappa_i\})} \bigotimes_{b\in Q_0} \frac{\mathbb{1} + (-1)^{s_b}\vec{r}_b\cdot\vec{\sigma}^{(b)}}{2}. \tag{A.1}$$

Herein, the vectors $\vec{r}_a$, $a\in\mathcal{C}$, are assumed fixed, and the adaption of measurement bases is taken care of by conditional conjugation under $\sigma_x^{(a)}$. As can be easily verified, the projectors $P^{(\mathcal{C})}(\{s_i\})$ have the properties

$$P^{(\mathcal{C})}(\{s_i\})P^{(\mathcal{C})}(\{s_i'\}) = \delta((\{s_i\},(\{s_i'\})P^{(\mathcal{C})}(\{s_i\}). \tag{A.2a}$$

$$\sum_{\{s_i\}} P^{(\mathcal{C})}(\{s_i\}) = \mathbb{1}^{(\mathcal{C})}. \tag{A.2b}$$

The probability to find the set of measurement outcomes $\{s_i\}$ in a $\mathrm{QC}_{\mathcal{C}}$-computation is

$$p(\{s_i\}) = {}_{\mathcal{C}}\langle\phi|P^{(\mathcal{C})}(\{s_i\})|\phi\rangle_{\mathcal{C}}. \tag{A.3}$$

More interesting than the probability for a specific set of measurement outcomes are, of course, the probability $p(R)$ for finding a certain computational result R and the success probability of an algorithm p_{succ}. $p(R)$ is the sum of all probabilities $p(\{s_i\})$ with $\mathbf{I}_x(\{s_i\}) = R$, i.e.

$$p(R) = {}_{\mathcal{C}}\langle\phi|P^{(\mathcal{C})}(R)|\phi\rangle_{\mathcal{C}}, \tag{A.4}$$

with

$$P^{(\mathcal{C})}(R) = \sum_{\{s_i\}\in\{0,1\}^{|\mathcal{C}|}\,\big|\,\mathbf{I}_x(\{s_i\})=R} P^{(\mathcal{C})}(\{s_i\}). \tag{A.5}$$

The success probability of an algorithm is given by the sum of all the probabilities $p(R)$ where R represents a valid outcome of the algorithm, $R \in \text{SOL}$,

$$p_{\text{succ}} = \sum_{R\in\text{SOL}} p(R). \tag{A.6}$$

The set SOL may contain more than one solution R. Shor's algorithm [1] is an example.

If the computation proceeds in the presence of noise, we describe this noise –as in error model 2– by an error channel $\mathcal{E}^{(\mathcal{C})}$ acting on a perfect cluster state, followed by perfect measurements. The probability to find the result R in the quantum computation then is, analogous to (A.3) , $p(R) = \text{Tr}\left(P^{(\mathcal{C})}(R)\,\mathcal{E}^{(\mathcal{C})}(|\phi\rangle_{\mathcal{C}}\langle\phi|)\right)$. We expand the noise superoperator $\mathcal{E}^{(\mathcal{C})}$ sandwiched between the perfect cluster state and the perfect measurements in error model 2 in a basis spanned by (products of) Pauli operators, E_i, E_j,

$$\mathcal{E}^{(\mathcal{C})}(\rho) = \sum_{i,j} c_{ij} E_i \rho E_j, \tag{A.7}$$

where $\sum_i c_{ii} = 1$. Therein, two Pauli errors E_i, E_j are called equivalent, $E_i \cong E_j$, iff they act identically on the cluster state, $E_i|\phi\rangle_{\mathcal{C}} = E_j|\phi\rangle_{\mathcal{C}}$. With (A.7), the probability to obtain the computational result R in the presence of noise becomes

$$p(R) = \sum_{i,j} c_{ij}\, {}_{\mathcal{C}}\langle\phi|E_j\, P^{(\mathcal{C})}(R)\, E_i|\phi\rangle_{\mathcal{C}}. \tag{A.8}$$

If the errors e_i of the generating set E_{phy} (4.42) are stochastically independent Pauli errors, then $c_{ij} = 0$ for all $i \neq j$. This special form of (A.8), which corresponds to the case that has been discussed in Section 4.2, is a consequence of the fact that the stochastically independent error channels (4.56), (4.58) have this property, and inherit it to the product errors composed of them. The form of the c_{ii} is further simplified due to the fact that the errors from the generating set are assumed to be stochastically independent in Section 4.2.

Here we discuss classical correlations of Pauli errors. Their immediate effect is that c_{ij} may be nonzero for all possible values of the error labels i, j. To illustrate this fact, let us briefly consider a situation where the error arises due to imperfect control over a single global tuning parameter for local or short-range interactions. The global tuning parameter leads to classical correlations between the errors. A specific example is a cluster state

where the errors occur solely due to erroneous Ising interaction. There, the global tuning parameter is the interaction time.

To discuss examples like this more explicitly, assume that the error superoperator $\mathcal{E}^{(\mathcal{C})}$ in error model 2 sandwiched between the perfect cluster state and the perfect measurements takes the form

$$\mathcal{E}^{(\mathcal{C})}(\rho_{\mathcal{C}}) = \int d\tau\, p(\tau) \left(\prod_{e_i \in E_{\mathrm{Phy}}} e^{-i\tau e_i} \right) \rho_{\mathcal{C}} \left(\prod_{e_i \in E_{\mathrm{Phy}}} e^{i\tau e_i} \right), \tag{A.9}$$

where the errors $e_i \in E_{\mathrm{Phy}}$ all commute. We now expand the products in (A.9) and insert the result into $p(R) = \mathrm{Tr}\left(P^{(\mathcal{C})}(R)\, \mathcal{E}^{(\mathcal{C})}(|\phi\rangle_{\mathcal{C}}\langle\phi|)\right)$. In this way, we find the probability $p(R)$ of obtaining the result R in the quantum computation

$$p(R) = \sum_{\substack{\{E\}_L \in P(E_{\mathrm{Phy}}) \\ \{E\}_R \in P(E_{\mathrm{Phy}})}} c(\{E\}_L, \{E\}_R)\ {}_{\mathcal{C}}\langle\phi| \left(\prod_{e_i \in \{E\}_L} e_i \right) P^{(\mathcal{C})}(R) \left(\prod_{e_k \in \{E\}_R} e_k \right) |\phi\rangle_{\mathcal{C}}, \tag{A.10}$$

where $P(E_{\mathrm{Phy}})$ denotes the power set of the generating set of errors E_{Phy}, and

$$c(\{E\}_L, \{E\}_R) = \int d\tau\, p(\tau)\ i^{|\{E\}_R| - |\{E\}_L|}\, (\sin\tau)^{|\{E\}_L| + |\{E\}_R|}\, (\cos\tau)^{2|E_{\mathrm{Phy}}| - |\{E\}_L| - |\{E\}_R|}. \tag{A.11}$$

As a consequence of the classical correlation between the errors $e_i \in E_{\mathrm{Phy}}$, in (A.10) we find the double-sum structure of (A.8). However, many of these classical correlations have no effect to the $\mathrm{QC}_{\mathcal{C}}$-computation, as stated by the following lemma:

Lemma 1 *Assume that a $QC_{\mathcal{C}}$-algorithm is run on a cluster state with proper adjustment of the measurement bases and method to extract the computational result. Then, for all computational results R and all inequivalent multi-local Pauli errors, E_i, $E_j \ncong E_i$,*

$${}_{\mathcal{C}}\langle\phi| E_j\, P^{(\mathcal{C})}(R)\, E_i |\phi\rangle_{\mathcal{C}} = 0. \tag{A.12}$$

Therein, proper adjustment of the measurement bases requires that the defining relations (3.50) for the sign factors of the measurement angles obey the invariance property (3.51) and a proper method to extract the computational result is one where the defining relations (3.49) for the readout bits $[\mathbf{I}_x]_m$ obey the same invariance property (3.51). Practically, this means that for a quantum algorithm of which there exists a network version, the perfect implementation on the $\mathrm{QC}_{\mathcal{C}}$ as compared to a perfect network realization does not introduce additional randomness into the quantum algorithm. The probability distribution of the computational results is exactly the same for the network- and the $\mathrm{QC}_{\mathcal{C}}$-version.

A consequence of lemma 1 and the fact that in (A.8) each E_j may be substituted by an equivalent error E'_j, $E'_j \cong E_j$ is that the probability $p(R)$ may always be written as

$$p(R) = \sum_i \tilde{p}_i\, {}_{\mathcal{C}}\langle\phi| E_i\, P^{(\mathcal{C})}(R)\, E_i |\phi\rangle_{\mathcal{C}}. \tag{A.13}$$

A hallmark of classical correlations in the noise are "off-diagonal" terms $E_i\rho E_j$ in the expression (A.7) for $\mathcal{E}^{(\mathcal{C})}(\rho)$. However, for the purpose of QC$_\mathcal{C}$-computation these terms are irrelevant. For any error channel $\mathcal{E}^{(\mathcal{C})}$ we may find a computationally equivalent error channel $\tilde{\mathcal{E}}^{(\mathcal{C})}$ which is diagonal in a Pauli basis, $\tilde{\mathcal{E}}^{(\mathcal{C})}(|\phi\rangle_\mathcal{C}\langle\phi|) = \sum_i \tilde{p}_i\, E_i|\phi\rangle_\mathcal{C}\langle\phi|E_i$. In other words, if we consider the density operator describing the state on the cluster $\mathcal{C}$ before the perfect measurements and expand it into a basis spanned by the cluster states, then, for QC$_\mathcal{C}$-computation, we may discard the off-diagonal part of this density operator.

Proof of Lemma 1. First we prove the identity

$$K^{(a)}P^{(\mathcal{C})}(R) = P^{(\mathcal{C})}(R)\,K^{(a)},\ \ \forall a \in \mathcal{C},\, \forall R. \tag{A.14}$$

To see why $K^{(a)}$ and $P^{(\mathcal{C})}$ commute, consider a single term $P^{(\mathcal{C})}(\{s_i\})$ in the sum (A.5) for $P^{(\mathcal{C})}(R)$. One finds

$$\begin{aligned} &K^{(a)}\left(\bigotimes_{b\in\mathcal{C}\setminus Q_0}\left(\sigma_x^{(b)}\right)^{\vartheta_b(\{s_i,\kappa_i\})}\frac{\mathbb{1}^{(b)}+(-1)^{s_b}\vec{r}_b\cdot\vec{\sigma}^{(b)}}{2}\left(\sigma_x^{(b)}\right)^{\vartheta_b(\{s_i,\kappa_i\})}\bigotimes_{c\in Q_0}\frac{\mathbb{1}^{(c)}+(-1)^{s_c}\vec{r}_c\cdot\vec{\sigma}^{(c)}}{2}\right)\\ &=\left(\bigotimes_{b\in\mathcal{C}\setminus Q_0}\left(\sigma_x^{(b)}\right)^{\vartheta'_b(\{s_i,\kappa_i\})}\frac{\mathbb{1}^{(b)}+(-1)^{s'_b}\vec{r}_b\cdot\vec{\sigma}^{(b)}}{2}\left(\sigma_x^{(b)}\right)^{\vartheta'_b(\{s_i,\kappa_i\})}\bigotimes_{c\in Q_0}\frac{\mathbb{1}^{(c)}+(-1)^{s'_c}\vec{r}_c\cdot\vec{\sigma}^{(c)}}{2}\right)K^{(a)}. \end{aligned} \tag{A.15}$$

Let us discuss in some detail what happens when we propagate the cluster state correlation operator through the projector $P^{(\mathcal{C})}(\{s_i\})$. First, the conditional spin flips $\left(\sigma_x^{(b)}\right)^{\vartheta_b(\{s_i,\kappa_i\})}$ remain unaffected. They may commute or anti-commute with $K^{(a)}$, but as they come in pairs there is no net sign factor. In propagation through the projections $P^{(b)}$ for $b \in \mathcal{C}\setminus Q_0$, the vector $\vec{r}_a$ is reflected about the xz-plane ($r_{a,y} \longrightarrow r'_{a,y} = -r_{a,y}$). This is accounted for by conjugating $\vec{r}_a\cdot\vec{\sigma}^{(a)}$ with $\sigma_x^{(a)}$ once more, $\vartheta_a \longrightarrow \vartheta'_a = \vartheta_a + 1$. For all $b \in \mathrm{nbgh}(a)$ the measurement result is flipped, $s_b \longrightarrow s'_b = s_b + 1 \mod 2$. For the qubits in Q_0, the measurement directions remain unaffected, and the measurement results flip if the respective measured observable anti-commutes with $K^{(a)}$. The transformations $\vartheta_a \longrightarrow \vartheta'_a = \vartheta_a + 1$, $a \in \mathcal{C}\setminus Q_0$ and $s_b \longrightarrow s'_b = s_b + 1$, $b \in \mathrm{nbgh}(a) \subset \mathcal{C}$ caused by propagation of a correlation operator we have encountered before as the invariance transformations of type (3.41) induced by the subsequent unitary transformations (3.53).

Now, via the condition (3.51) we require that the defining relation (3.50) for the ϑ_b, $b \in \mathcal{C}\setminus Q_0$, is invariant under the these transformations, i.e.

$$\vartheta'_b(\{s_i\}) = \vartheta_b(\{s'_i\}). \tag{A.16}$$

We may thus replace the r.h.s. in (A.15) by

$$\begin{aligned} & K^{(a)} P^{(\mathcal{C})}(\{s_i\}) \\ &= \left(\bigotimes_{b \in \mathcal{C} \setminus Q_0} \left(\sigma_x^{(b)}\right)^{\vartheta_b(\{s'_i,\kappa_i\})} \frac{\mathbb{1}^{(b)} + (-1)^{s'_b} \vec{r}_b \cdot \vec{\sigma}^{(b)}}{2} \left(\sigma_x^{(b)}\right)^{\vartheta_b(\{s'_i,\kappa_i\})} \bigotimes_{c \in Q_0} \frac{\mathbb{1}^{(c)} + (-1)^{s'_c} \vec{r}_c \cdot \vec{\sigma}^{(c)}}{2} \right) K^{(a)} \\ &= P^{(\mathcal{C})}(\{s'_i\}) K^{(a)}. \end{aligned} \tag{A.17}$$

Via condition (3.51) we also require that the defining relations (3.49) for the bits of the computational result are invariant under the transformations $s_b \longrightarrow s'_b$, which yields

$$[\mathbf{I}_x]'_m(\{s_b\}) = [\mathbf{I}_x]_m(\{s'_b\}) = [\mathbf{I}_x]_m(\{s_b\}), \;\; \forall m = 1..n. \tag{A.18}$$

The individual bits $[\mathbf{I}_x]_m = R_m$ of the computational result are scalars under the above transformation.

Thus, under conjugation with $K^{(a)}$, each term of the sum $P^{(\mathcal{C})}(R)$ is mapped onto another or the same term in $P^{(\mathcal{C})}(R)$. The mapping is reversible, and thus the action of $K^{(c)}$ onto $P^{(\mathcal{C})}(R)$ merely amounts to a permutation of summands,

$$\begin{aligned} K^{(a)} \left(\sum_{\{s_i\} \in \{0,1\}^{|\mathcal{C}|}} P^{(\mathcal{C})}(\{s_i\}) \right) &= \left(\sum_{\{s_i\} \in \{0,1\}^{|\mathcal{C}|}} P^{(\mathcal{C})}(\{s'_i\}) \right) K^{(a)}, \\ &= \left(\sum_{\{s'_i\} \in \{0,1\}^{|\mathcal{C}|}} P^{(\mathcal{C})}(\{s'_i\}) \right) K^{(a)}, \end{aligned} \tag{A.19}$$

which proofs (A.14).

Now consider a single term in the expansion (A.8), ${}_{\mathcal{C}}\langle\phi|E_j\, P^{(\mathcal{C})}(R)\, E_i|\phi\rangle_{\mathcal{C}}$,

$${}_{\mathcal{C}}\langle\phi|E_j\, P^{(\mathcal{C})}\, E_i|\phi\rangle_{\mathcal{C}} = {}_{\mathcal{C}}\langle\phi|E'_j\, P^{(\mathcal{C})}(R)\, E'_i|\phi\rangle_{\mathcal{C}}, \tag{A.20}$$

where $E'_i \cong E_i$ and $E'_j \cong E_j$. In particular, E'_i, E'_j are chosen to be products of Pauli phase flip operators $\sigma_z^{(a)}$ only. The Pauli spin flip operators $\sigma_x^{(a)}$ have been removed by multiplication of E_i, E_j with correlation operators $K^{(a)}$. The error operators E'_i, E'_j equivalent to E_i, E_j are in this way uniquely defined, and $E_i \cong E_j \Longleftrightarrow E'_i = E'_j$.

If $E'_i \neq E'_j$, then there exists a qubit, now called qubit a, for which there is a $\sigma_z^{(a)}$-factor only in one of the errors E'_i or E'_j. Then it follows that

$$\begin{aligned} {}_{\mathcal{C}}\langle\phi|E'_j\, P^{(\mathcal{C})}(R)\, E'_i|\phi\rangle_{\mathcal{C}} &= {}_{\mathcal{C}}\langle\phi|K^{(a)}\, E'_j\, P^{(\mathcal{C})}(R)\, E'_i\, K^{(a)}|\phi\rangle_{\mathcal{C}} \\ &= -{}_{\mathcal{C}}\langle\phi|E'_j\, K^{(a)}\, P^{(\mathcal{C})}(R)\, K^{(a)}\, E'_i|\phi\rangle_{\mathcal{C}} \\ &= -{}_{\mathcal{C}}\langle\phi|E'_j\, P^{(\mathcal{C})}(R)\, E'_i|\phi\rangle_{\mathcal{C}} \\ &= 0. \end{aligned} \tag{A.21}$$

Therein, the second line is true because $K^{(a)}$ anti-commutes with $\sigma_z^{(a)}$, and the third line because of (A.14). Inserting (A.21) into (A.20) one obtains (A.12), proving Lemma 1. □

Bibliography

[1] P.W. Shor, *Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Sci. Statist. Comput. **26**, 1484 (1997).

[2] L.K. Grover, *A fast quantum mechanical Algorithm for database search*, Proc. 28 Annual ACM Symp. on the Theory of Computing, 212 (1996).

[3] R. P. Feynman, *Simulating physics with computers*, Int. J. Phys. **21**, 467 (1982).

[4] S. Lloyd, *Universal Quantum Simulators*, Science **273**, 1073 (1996).

[5] D. Deutsch, *Quantum theory, the Church-Turing principle and the universal quantum computer*, Proc. Roy. Soc. A **400**, 97 (1985).

[6] G. Brassard, P. Høyer, M. Mosca and A. Tapp, *Quantum Amplitude Amplification and Estimation*, quant-ph/0005055 (2000).

[7] Y. Aharonov, L. Davidovich, and N. Zagury, *Quantum random walks*, Phys. Rev. **A** 48, 1687 (1993).

[8] A.M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutman, D.A. Spielman, *Exponential algorithmic speedup by quantum walk*, quant-ph/0209131 (2002).

[9] R. Raussendorf and H.J. Briegel, *A one-way quantum computer*, Phys. Rev. Lett. **86**, 5188 (2001).

[10] R. Raussendorf and H.J. Briegel, *Computational model underlying the one-way quantum computer*, Quant. Inf. Comp. **6**, 443 (2002), quant-ph/0108063 (2001).

[11] R. Raussendorf and H.J. Briegel, *Computational model for the one-way quantum computer: Concepts and Summary*, quant-ph/0207183 (2002), and in: Th. Beth and G. Leuchs (Eds.), *Quantum Information Processing*, Wiley-VCH (2003).

[12] C.H. Bennett, *Notes on the history of reversible computation*, IBM J. Res. Develop. **32**, No. 1 (1988).

[13] R. Landauer, *Irreversibility and Heat Generation in the Computing Process*, IBM J. Res. Develop. **5**, No. 3 (1961).

[14] C.H. Bennett, *Logical Reversibility of Computation*, IBM J. Res. Develop. **17**, 525 (1973).

[15] P. Benioff, *Quantum Mechanical Models of Turing Machines That Dissipate No Energy*, Phys. Rev. Lett. **48**, 1581 (1982).

[16] D. Deutsch, *Quantum computational networks*, Proc. Roy. Soc. **425**, 73 (1989).

[17] A. Barenco et al., *Elementary gates for quantum computation*, Phys. Rev. A **52**, 3457 (1995).

[18] Y. Shi, *Both Toffoli- and controlled-NOT need little help to do universal quantum computation*, quant-ph/0205115 (2002).

[19] P.O. Boykin et al. *A universal and fault-tolerant quantum basis*, Inf. Proc. Lett. **75**, 101 (2000).

[20] M.A. Nielsen and I.L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press (2000).

[21] Theorem 10.7 in [20].

[22] D. Gottesman and I.L. Chuang, *Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations*, Nature (London) **402**, 390 (1999).

[23] E. Knill, R. Laflamme and G.J. Milburn, *A scheme for efficient quantum computing with linear optics*, Nature (London) **409**, 46 (2001).

[24] R. Cleve, A. Ekert, C. Macchiavello and M. Mosca, *Quantum Algorithms Revisited*, quant-ph/9708016 (1997).

[25] R. Jozsa and N. Linden, *On the role of entanglement in quantum computational speed-up*, quant-ph/0201143 (2002).

[26] G. Vidal, *Efficient classical simulation of slightly entangled quantum systems*, quant-ph/0301063 (2003).

[27] M.A. Nielsen, *Universal quantum computation using only projective measurement, quantum memory, and preparation of the* $|0\rangle$ *state*, quant-ph/0108020 (2001).

[28] A. Fenner and Y. Zhang, *Universal quantum computation with two- and three-qubit projective measurements*, quant-ph/0111077 (2001).

[29] D.W. Leung, *Two-qubit Projective Measurements are Universal for Quantum Computation*, quant-ph/0111122 (2001).

[30] D. Jaksch *et al.*, *Entanglement of Atoms via Cold Controlled Collisions*, Phys. Rev. Lett. **82**, 1975 (1999).

[31] L.-M. Duan, E. Demler, and M.D. Lukin, *Controlling Spin Exchange Interactions of Ultracold Atoms in Optical Lattices*, cond-mat/0210564 (2002).

[32] M. Greiner, O. Mandel, T. Esslinger, Th.W. Hänsch, I. Bloch, *Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms*, Nature **415**, 39 (2002).

[33] O. Mandel et al. *Coherent transport of neutral atoms in spin-dependent optical lattice potentials*, cond-mat/0301169 (2003).

[34] M. Greiner, O. Mandel, Th.W. Hänsch, I. Bloch, *Collapse and revival of the matter wave field of a Bose-Einstein condensate*, Nature **419**, 51 (2002).

[35] P.W. Shor, *Scheme for reducing decoherence in quantum computer memory*, Phys. Rev. A **52**, R2493 (1995).

[36] A.R. Calderbank and P.W. Shor, *Good quantum error correcting codes exist*, Phys. Rev. A **54**, 1098 (1996).

[37] A.M. Steane, *Error correcting codes in quantum theory*, Phys. Rev. Lett. **77**, 793 (1996).

[38] A.M. Steane, *Multi-particle interference and quantum error correction*, Proc. R. Soc. A **452**, 2551 (1996).

[39] E. Knill and R. Laflamme, *Theory of error-correcting codes*, Phys. Rev. A **55**, 900 (1997).

[40] P.W. Shor, *Fault-tolerant quantum computation*, Proceedings of the Symposium on the Foundations of Computer Science, Los Alamitos, IEEE Press, and quant-ph/9605011 (1996).

[41] D. Gottesman, *Stabilizer Codes and Quantum Error Correction*, PhD thesis, California Institute of Technology, Pasadena, California, USA, quant-ph/9705052 (1997).

[42] D. Gottesman, *Theory of fault-tolerant quantum computation*, Phys. Rev. **A** 57, 127 (1998).

[43] E. Knill, R. Laflamme, and W. H. Zurek, *Resilient Quantum Computation: Error Models and Thresholds*, quant-ph/9702058 (1997).

[44] D. Aharonov and M. Ben-Or, *Fault-Tolerant Quantum Computation With Constant Error Rate*, quant-ph/9906129 (1999).

[45] Y. Nakamura, Yu. A. Pashkin, and J.S. Tsai, *Coherent controll of macroscopic quantum states in a single-Cooper-pair box*, Nature **398**, 786 (1999).

[46] D. Vion et al., *Manipulating the Quantum State of an Electrical Circuit*, Science **296**, 886 (2002).

[47] C.A. Sackett et al., *Experimental entanglement of four particles*, Nature **404**, 256 (2000).

[48] F. Schmidt-Kahler et al., *Realization of the Cirac-Zoller controlled-NOT quantum gate*, Nature **422**, 408 (2003).

[49] D. Leibfried et al., *Experimental demonstration of a robust, high-fidelity geometric two ion-qubit phase gate*, Nature **422**, 412 (2003).

[50] A.M. Steane, *Overhead and noise threshold for fault-tolerant quantum computation*, quant-ph/0207119 (2002).

[51] H.J. Briegel and R. Raussendorf, *Persistent Entanglement in Arrays of Interacting Qubits*, Phys. Rev. Lett. **86**, 910 (2001).

[52] A.R. Calderbank, E.M. Rains, P.W. Shor, and N.J.A. Sloane, *Quantum Errror Correction and Orthogonal Geometry*, Phys. Rev. Lett. **78**, 405 (1997).

[53] D. Schlingemann and R.F. Werner, *Quantum error-correcting codes associated with graphs*, Phys. Rev. **A** 65, 012308 (2001).

[54] R. Diestel, *Graphentheorie*, Springer-Verlag (2000).

[55] S. Perdrix, IQUING workshop on Quantum Information, Imperial College London, Sept. 2002.

[56] C. Moore and M. Nilsson, *Parallel Quantum Computation and Quantum Codes*, quant-ph/9808027 (1998).

[57] M. Grassl, A. Klappenecker, and M. Rötteler, *Graphs, Quadratic Forms, and Quantum Codes.* IEEE international symposium on information theory, Lausanne (2001).

[58] D. Schlingemann, *Cluster states, algorithms and graphs*, quant-ph/0305170 (2003).

[59] R.B. Griffiths and C.-S. Niu, *Semiclassical Fourier Transform for Quantum Computation*, Phys. Rev. Lett. **76**, 3228 (1996).

[60] V. Vedral, A. Barenco and A. Ekert, *Quantum Networks for Elementary Arithmetic Operations*, quant-ph/9511018 (1995).

[61] E. Bernstein and U. Vazirani, *Quantum complexity theory*, Proc. of the 25th Annual ACM Symposium on Theory of Computing, 11 (1997).

[62] A. Yao, *Quantum circuit complexity*, Proc. of the 34th Annual IEEE Symposium of Foundations of Computer Science, 352 (1993).

[63] M. Li and P. Vitányi, *An introduction to Kolmogorov complexity and its applications.* Springer (1997).

[64] D. Gottesman, *Fault-Tolerant Quantum Computation with Local Gates*, quant-ph/9903099 (1999).

[65] C. Zalka, *Threshold Estimate for Fault Tolerant Quantum Computing*, quant-ph/9612028 (1996).

[66] W. Dür and H. Briegel, *Entanglement Purification for Quantum Computation*, Phys. Rev. Lett. **90**, 067901 (2003).

[67] A. Yu. Kitaev, *Fault-tolerant quantum computation by anyons*, quant-ph/09707021 (1997).

[68] E. Denis, A. Kitaev, A. Landahl, and J. Preskill, *Topological quantum memory*, quant-ph/0110143 (2001).

[69] Ch. Wang and J. Preskill, *Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory*, quant-ph/0207088 (2002).

[70] K. Kraus, *States, effects and operators: Fundamental notions of quantum theory*, Springer, Heidelberg (1983).

[71] E. Knill and R. Laflamme, *Concatenated Quantum Codes*, quant-ph/9608012 (1996).

[72] J. Preskill, *Fault-tolerant quantum computation*, quant-ph/9712048 (1997).

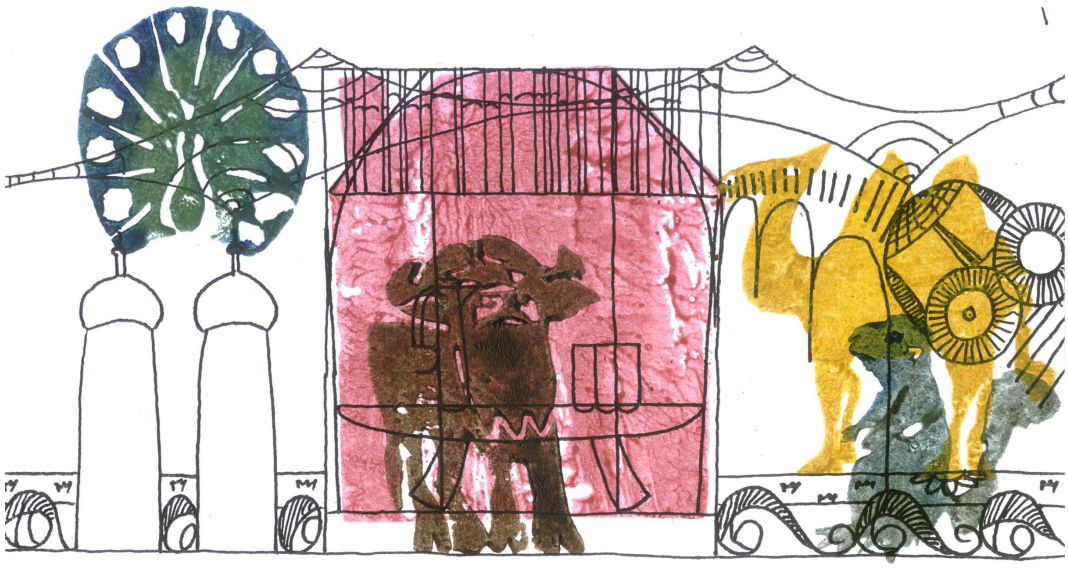
[73] D. Aharonov, A. Kitaev, N. Nisan, *Quantum circuits with mixed states*, Proc. of the 30th Annual ACM Symposium on Theory of Computing, and quant-ph/9806029 (1998).

[74] H. Heuser, *Funktionalanalysis*, B. G. Teubner Stuttgart (1975).

[75] B. Neuburger, *Untersuchung der Fehlertoleranz von messungsbasierten Quantengattern auf verschränkten Clusterzuständen*, Diplomarbeit (master thesis), Ludwig-Maximilians-Universität München (2002).

Danksagung/ Acknowledgements

Mein herzlicher Dank gilt allen, die zum Entstehen und Gelingen dieser Arbeit beigetragen haben. Ganz besonders danke ich meinem Betreuer Hans Briegel für seine Führung, seine Offenheit und Übersicht, und für unsere vielen intensiven und fruchtbaren Diskussionen. Die Gewohnheit, ein Stück Kreide in die Hand zu nehmen und an der Tafel gemeinsam Ideen anzugehen, habe ich hier kennengelernt. Es ist einfach ein Ort, an dem man gern arbeitet. Herrn Prof. Axel Schenzle danke ich dafür, dass er mich großzügig und unkompliziert hier im Institut "einziehen" ließ, bevor es überhaupt eine Stelle gab. Es hat mir große Freude bereitet, mit Dan Browne, Hans Aschauer, Benedikt Neuburger, Mark Hein und Wolfgang Dür zusammenzuarbeiten. Prof. Axel Schenzle, Prof. Herbert Wagner, Prof. Otto Forster, Prof. J. Ignacio Cirac, Dirk Schlingemann, Markus Grassl, Pawel Wocjan, Christian Kurtsiefer, Immanuel Bloch, Markus Greiner, Olaf Mandel, Artur Widera, Jesse Hersch, Christoph Gohle, Simon Anders und Jens Eisert danke ich für anregende Diskussionen. Jens Schmalzing, Robert Dahlke und Sigmund Stintzing bändigten unsere klassischen Computer. Ein besonderer Dank gilt der Deutschen Forschungsgemeinschaft, die das Patent am $QC_{\mathcal{C}}$ finanzierte, der Patentstelle der Ludwig-Maximilians-Universität für ihre Unterstützung, und den Patentanwälten Dr. Oliver Hertz und Dr. Christian Ginzel für ihren enormen Einsatz und ihren ordnenden Einfluss.



Ich bedanke mich bei Michael und Ulrike Bräuer, und bei Dominique Gobert, Christian Dekant, Hans und Susanne Aschauer, Myungseon 0, Wolfgang Becken, Ulrich Martini, Mary Kissel, Harald Hofmeier, Veronika Schreieder, Joachim Walter, Corinna Kollath, Patrik Werner, Jens Schmalzing, Robert Dahlke und Klaus Beisbart, mit denen mich gemeinsam verbrachte Zeit und schöne Erlebnisse verbinden. Mit der Studienstiftung des Deutschen Volkes und dem Zentralen Hochschulsport hatte ich an vielen interessanten Unternehmungen teil. Ganz besonders bedanke ich mich bei meiner Freundin Andrea, bei meinen Eltern und bei meiner Oma Elise Scharioth.

Curriculum Vitae

Robert Raußendorf

Date of birth:	19th March, 1973
Place of birth:	Dresden

Education

1979 - 1991	Primary school in Bautzen (1979-1987); Werner- Heisenberg-School Riesa, Special School for Natural Sciences (1987-1991).
06/1991	A-levels/ Abitur.
1991-1997	Study of Physics at the Technical University of Dresden, the Ruprecht-Karls- Universität Heidelberg, and the University of Cambridge, UK.
10/1994 - 06/1995	Mathematical Tripos Part III at the University of Cambridge.
07/1996 - 07/1997	Diploma thesis at the Institute of Theoretical Physics of the Heidelberg University under supervision of Prof. Dr. Ch. Wetterich. Title of the thesis: "A linear σ-model for vector- and axial-vector mesons".
07/1997	Diploma degree in Physics from the University of Heidelberg.
12/1997 - 12/1998	Community service at the German Cancer Research Center (DKFZ).
02/1999 - 04/1999	Researcher at the DKFZ Heidelberg.

Dissertation

07/1999 - 06/2003	PhD thesis at the University of Munich in the Theoretical Quantum Optics Group of Prof. Dr. A. Schenzle.