HYBRID QUANTUM COMPUTATION

ARUN

NATIONAL UNIVERSITY OF SINGAPORE

2011

HYBRID QUANTUM COMPUTATION

ARUN

M.Sc. (*Physics*), *IIT ROORKEE, INDIA*

A THESIS SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

CENTRE FOR QUANTUM TECHNOLOGIES

NATIONAL UNIVERSITY OF SINGAPORE

2011

Dedicated to my family,

and my teachers...

Acknowledgments

There are numerous people to whom my gratitude goes for supporting me during my PhD studies. Above all, I would like to thank my supervisor Prof. Berthold-Georg (Berge) Englert. I am deeply grateful to him for his invaluable support, guidance, and mentoring of my rewarding doctoral studies. I could not have asked for a better supervisor. He has been very positive throughout my PhD, providing encouragement, freedom and flexibility. I am deeply indebted to him for devoting so much time and patience to every aspect of my doctoral research.

I take this opportunity to convey my sincere thanks to Daniel Zemann and Le Huy Nguyen for coauthoring with me two published papers.

I would like to thank Prof. Hans J. Briegel for generously hosting me at the Institute for Quantum Optics and Quantum Information, Innsbruck, Austria and Prof. Kae Nemoto for kindly hosting me at the National Institute of Informatics, Tokyo, Japan. I am also grateful to Assoc. Prof. Wolfgang Dür, Prof. Bill Munro, Asst. Prof. Jun Suzuki and Asst. Prof. Simon Devitt for interesting discussions.

I offer special thanks to Dr. Philippe Raynal and Dr. Ng Hui Khoon for enlightening comments on the manuscript of our paper and this thesis, and for explaining fault-tolerant quantum computation to me. For helping me in the editing of this thesis, I am also eternally obliged to Naresh Susarla, Dr. Paul Constantine Condylis and Abhinav Jain.

My great appreciation goes to every member of our research group (Berge's group) for educating me through presentations and discussions in group meetings,

and my heartfelt appreciation goes to the Centre for Quantum Technologies for all the support I have received.

Finally, I thank my family and friends (especially Dr. Rohit Malik, Abhinav Jain and Dr. Philippe Raynal) for helping me in countless ways. I am truly and deeply indebted to so many people that there is no way to acknowledge them all or even any of them properly. Those who are not mentioned here, you are not forgotten.

Summary

This thesis comprises the study of two research projects: the hybrid quantum computation model presented in Part I and the test-state approach to the quantum search presented in Part II.

Part I: The hybrid model, which joins the advantages of the unitary-evolution-based quantum computation model described in Chapter 2 and the measurement-based quantum computation model given in Chapter 3, is introduced in Chapter 4. The hybrid model is a universal model, where part of a quantum circuit (of an algorithm) is simulated by unitary evolution and the rest by measurements on small (non-universal) graph states to optimize the resource consumption and to get easier experimental implementation.

The classical information processing in this model turns out to be rather simple as compared to the measurement-based model. It only requires the information flow vector and the propagation matrices. To make the picture complete, the basic ideas for a fault-tolerant version of the hybrid model are introduced in Chapter 5 in which the classical information processing accommodates nicely.

Part II: Both classical and quantum search problems with their algorithms are presented in Chapter 6. In the quantum search problem, one has to find one of a permissible set of unitary mappings, which is implemented by a given black box, without opening it. Grover's algorithm accomplished this search with a quadratic speedup as compared to its classical counterpart. Since the outcome of Grover's algorithm is probabilistic—it gives the correct answer with a high probability, not with

certainty—the answer requires verification. For this purpose, we introduce specific test states in Chapter 7, one for each unitary mapping. The test-state verification is a three-step process, named as "single iteration of the test-state approach."

The test-state approach, in itself, can complete the search deterministically, it always gives a definite answer after a finite number of such iterations. Furthermore, it is 3.41 times as fast as the purely classical search.

Contents

List of Tables

List of Figures

List of Symbols and Abbreviations

$\oplus$	Modulo-2 addition
$\otimes$	Tensor product
$\|\cdot\rangle$	Ket
$\langle\cdot\|$	Bra
D	$\pi/4$-phase gate
F	$\pi/2$-phase gate
H	Hadamard gate
I	Identity operator in the two-dimensional Hilbert space
X	Pauli σ_x operator
Y	Pauli σ_y operator
Z	Pauli σ_z operator
CH	Controlled-H gate
CNOT	Controlled-NOT gate
CPHASE	Controlled-PHASE gate
CZ	Controlled-Z gate

CCNOT	Controlled-controlled-NOT (Toffoli) gate
CCZ	Controlled-controlled-Z gate
CSWAP	Controlled-SWAP (Fredkin) gate
Δ	Control is set to $\|0\rangle$
Λ	Control is set to $\|1\rangle$
$\mathcal{D}$	Diffusion operator
$\mathcal{I}$	Information flow vector
$\mathcal{O}$	Oracle
$\mathbf{C}$	Propagation matrix
$\mathbf{I}$	Identity operator in the N-dimensional Hilbert space
CC	Classical computer
GA	Grover's search algorithm
HQCM	Hybrid quantum computation model
MQCM	Measurement-based quantum computation model
MUD	Measurement for unambiguous discrimination
POM	Probability-operator measurement
QC	Quantum computer
QIS	Quantum information science
QKD	Quantum key distribution
SRM	Square-root measurement
UQCM	Unitary-evolution-based quantum computation model

Chapter 1

Introduction

By the end of the nineteenth century, physics consisted mainly of Newtonian mechanics and Maxwell's theory of electromagnetism. Newtonian mechanics was used to study the dynamics of material bodies, while Maxwell's electromagnetism provided the proper framework to investigate radiation. Matter and radiation were described in terms of particles and waves, respectively. The interaction between matter and radiation were given by the Lorentz force or explained by thermodynamics. At the turn of the twentieth century, classical physics (classical mechanics, classical theory of electromagnetism, and thermodynamics) was challenged on two major fronts.

First, classical mechanics failed to explain the results of the Michelson-Morley experiment such as the constancy of the speed of light. In 1905, Einstein gave the special theory of relativity, which favors the Michelson-Morley experiment. Also, the theory shows that Newton's laws of motion do not hold good for objects which are moving with a velocity close to the speed of light.

Second, classical physics failed to explain a number of microscopic phenomena such as blackbody radiation, the photoelectric effect, atomic stability and discreteness of atomic spectroscopy. In 1900, Max Planck introduced the concept of quantum of energy to explain the phenomenon of blackbody radiation. Later, Einstein

gave an accurate explanation to the photoelectric effect in 1905 by taking quanta of light (photons) into consideration. In 1913, Niels Bohr introduced a model of the hydrogen atom by combining Rutherford's atomic model, Planck's quantum concept, and Einstein's photons. Bohr's atomic model explained both atomic stability and discreteness of atomic spectroscopy. These ideas are now collectively known as the old quantum theory.

In 1923, de Broglie introduced the concept of wave-particle duality, which was experimentally verified by Davisson and Germer in 1927. In 1926, Schrödinger established wave mechanics. This is a generalization of the de Broglie hypothesis, where the dynamics of microscopic matter is given by Schrödinger's wave equation. In 1927, Max Born proposed the probabilistic interpretation of Schrödinger's wave function.

Inspired by Planck's quantization of waves and Bohr's atomic model, Heisenberg developed matrix mechanics in 1925. Later, both wave mechanics and matrix mechanics were shown to be equivalent. In 1939, Dirac suggested a more general formulation of quantum mechanics dealing with abstract objects: kets (state vectors), bras, and operators. In continuous bases, Dirac's formalism gives Schrödinger's wave mechanics, and in discrete bases, it reduces to Heisenberg's matrix mechanics. Ever since, quantum mechanics has been an essential part of science and has been applied with enormous success in various fields including chemistry, biology, and computer science.

From the beginning of human socialization, *communication* and *calculation* have been indispensable of daily life. Initially, like any other task, both communication and calculation were done manually. However, the Second World War (1939–1945) created not only the need for stronger weapons but also the need for *secure* communication and *faster* computation. The aid of machines was therefore required. These necessities became the reasons for *classical information theory* and *classical computation*. As a result, a series of inventions in the field of telecommunication

such as electrical telegraphy, telephone, radio, television and Internet have been made available for public use. Likewise, personal computers have been made accessible to perform calculation at high speed. Clearly, information science is made up of these two fundamental branches, where every information processing task—communication in the field of classical information theory and calculation in the field of classical computation—has a set of basic elements such as source, encoding, processing, decoding, and detection. At first information science was based on classical physics and was therefore concerned with classical computer (CC). However, quantum mechanics has brought information science into a new age, and one now speaks of *quantum information science*[1] (QIS).

After the Second World War, the decisive events, which established the discipline of classical information theory, were the publications of Claude Shannon's seminal papers [4] in 1948. He addressed two fundamental issues of the information theory by giving two landmark theorems: The first—*Shannon's noiseless channel coding theorem*—quantifies the *minimum* amount of physical resources required to store the information being produced by a source, in such a way that at a later time it can be recovered reliably. The second—*Shannon's noisy channel coding theorem*—quantifies the *maximum* amount of information that can be reliably transmitted through a noisy communication channel. The first coding theorem established the basis for *data compression* in which the information is encoded using fewer bits than its original representation in order to reduce the consumption of expensive resources (e.g., hard disk space, transmission bandwidth). The second coding theorem triggered the development of *error-correcting codes* (e.g., repetition code, the Hamming code) since 1950, whereby the transmitted information is protected against noise by adding redundancy to it.

Information needs to be protected during transmission not just from the errors

[1]QIS is an extension of the classical information science like complex numbers are an extension of real numbers and quantum mechanics is an extension of classical mechanics. The quantum analogs of a *bit* and a *reversible logic gate* are a *qubit* and a *unitary operation*, respectively.

caused by noise, but also from the potential eavesdroppers. The task of cryptography is to secure the information from eavesdropping. Since 1917, cryptographers have been using a private key[2] with the one-time pad algorithm to secure strings of bits (classical information). The Morse code, the Enigma machine, and the RSA algorithm[3] are other milestones in the vast history of cryptography. The RSA algorithm was publicly announced in 1978, where the security relies on the assumption that the eavesdropper has a limited computational power. The first *Quantum key distribution* (QKD) protocol was introduced in 1984 by Charles Bennett and Gilles Brassard, now referred as BB84 [11]. Through a QKD protocol, private key bits can be generated over a public channel. The key bits can then be used for a classical private key cryptosystem with the one-time pad algorithm. Here, the laws of quantum mechanic insure the secure communication.

The superiority of quantum mechanics over classical mechanics is two folded in cryptography. On one hand, purely classical cryptography (the RSA cryptosystem) is vulnerable to the quantum attacks (using the Shor's factoring algorithm [33]). On the other hand, the BB84 QKD protocol is *provably secure*. Later, in 1991, Artur Ekert introduced the entanglement-based protocol for QKD [12]. Both the QKD protocol are different sides of the same coin (equivalent).

In 1992, Charles Bennett and Stephen Wiesner demonstrated the transfer of two bits of classical information using only one qubit, with the aid of quantum entanglement in *superdense coding* [13]. An unknown quantum state can be disassembled and perfectly reconstructed in another location, with the aid of quantum entanglement, by sending two bits of classical information. This, in 1993, is explained as *quantum teleportation* [14]. As quantum information has found many powerful applications, it was necessary to generalize the basic ideas, like Shannon's theorem, of classical information theory to the quantum regime.

[2]The key distribution lies at the heart of cryptography.

[3]The RSA (Rivest, Shamir and Adleman) algorithm is used for public-key cryptography, which relies on the difficulty to factorize large numbers.

In 1995, Benjamin Schumacher developed a quantum version of Shannon's noiseless channel coding theorem [5]. However, a quantum version of Shannon's noisy channel coding theorem is not yet known. Nevertheless, quantum error-correction theory—based on classical linear coding theory—has been developed [6, 7, 8, 9, 10], which allows the protection of information during computation as well as communication in the presence of noise. Thus, clearly, quantum information has the upper hand over classical information for security, and the entanglement-assisted communication is impossible in the classical regime [15].

Let us now turn our attention to another strand of the information science, computation, on which this thesis is focused. A building block for computation (to perform calculations or to execute algorithms) is the *computation model*. An algorithm is a procedure to perform a certain task on a computer. Algorithms are independent to the computational model, and vice versa.

Before the World War II, researchers like Alan Turing were studying cryptography and felt the need for fast computation to decode encrypted messages. In 1937, Alan Turing introduced the first abstract (mathematical) notion of a programmable CC—known as *Turing machine*[4] [16]. He and Alonzo Church showed that there is a *universal* Turing machine that can be used to simulate any other Turing machine. The strong form of this statement—called the strong *Church-Turing thesis*[5]—can be rewritten as follows:

Any algorithmic process (or computational model) can be simulated "efficiently[6]*" by using a Turing machine* [1].

Around 1945, John von Neumann established a basic theoretical model of a computer—known as the *von Neumann architecture*—in which the necessary com-

[4]This idea came to Alan Turing from the question "Is there a mechanical process which can be applied to a mathematical statement?" posed by M. H. A. Newman's lectures.

[5]The Church-Turing thesis is a conjecture.

[6]Basically, an algorithm is called efficient if it takes a time to solve a problem that is polynomial in the size of problem. However, if the required time is super-polynomial or exponential then the algorithm is called inefficient.

ponents of a computer such as input devices (keyboard, mouse, scanner), processor (CPU), main memory (RAM), auxiliary storages (disk drives), and output devices (monitor, printer) are assembled in such a *practical* fashion that it becomes as capable as a universal Turing machine. Since then, the development of computer hardware made of electronic components has been following an amazing pace, and every modern day computer uses the von Neumann architecture.

The *strong* Church-Turing thesis emphasizes *efficiency* and thus, the Turing machine has become a very useful model for investigating *computational complexity*. During the 1970s, the discovery of *randomized algorithms*[7] posed a challenge on the strong Church-Turing thesis. There are problems efficiently solvable by randomized algorithms, which, nevertheless, cannot be efficiently solved on a *deterministic* Turing machine. This challenge led to a small modification in the strong Church-Turing thesis:

Any algorithmic process can be simulated efficiently using a probabilistic Turing machine [1].

After this, it was completely natural to ask whether it is possible to find a computational model that can efficiently solve a computational problem that has no efficient solution on a CC or even a probabilistic Turing machine. In 1982, Richard Feynman [17], followed by David Deutsch [19], presented their response to this question. Feynman conjectured that it is advantageous to use a computer based on the principles of quantum mechanics, a *quantum computer* (QC), over a CC for simulating quantum mechanical systems. In 1982, Paul Benioff gave a classical model that could be efficiently simulated on a Turing machine, but to make it reversible he proposed to use a quantum system [18].

In 1985, David Deutsch introduced the first model of QC, *universal quantum Turing machine* [19], that can do certain tasks which are impossible for the universal

[7] In addition to input, a randomized algorithm takes a source of random numbers to make random choices during execution and gain the performance. For example, search over an unsorted database can be completed by an efficient randomized algorithm.

Turing machine. This includes generation of genuine random numbers, parallel calculations with a single register, perfect simulation of quantum systems, etc. David Deutsch reported the second model for quantum computation in 1989, the so-called *quantum circuits model* [20]. Hereafter, the quantum circuits model is referred to as unitary-evolution-based quantum computation model (UQCM) in this work and is discussed in Chapter 2.

In the UQCM, quantum unitary gates can be combined to achieve a QC in the same way as logic gates can be combined to achieve a CC. The UQCM can compute anything that the quantum Turing machine can do, and vice versa. Both are universal. In 1995, Adriano Barenco and others proved that any quantum circuit can be constructed using nothing more than quantum gates on one qubit and the controlled-NOT (CNOT) gates on two qubits. This limited but sufficient set of gates is named a *universal set of gates* [23].

In 2001, Robert Raussendorf and Hans Briegel introduced the measurement-based quantum computation model (MQCM[8]) [37], which is explained in Chapter 3. In the MQCM, a sufficiently large highly entangled multiqubit state, the (two-dimensional square) *graph state*[9] [35, 36], is employed as the central physical resource for (universal) quantum computation on which any quantum algorithm can be simulated by single-qubit projective measurements. The details of an algorithm under simulation lie in the spatiotemporal pattern of single-qubit measurement bases. Also, it is necessary to keep the record of every measurement outcome with a CC for setting the next measurement bases. This is in order to run the computation deterministically and to interpret the final result—called the *classical information processing*[10] in the MQCM [39]. To make the discussion complete, let us now move to quantum algorithms.

[8]The MQCM is also known as one-way quantum computation model, because its resource state can be used only once.

[9]Cluster state is a special case of the graph state.

[10]It is also called as classical feedforward.

After the UQCM, in 1992, David Deutsch and Richard Jozsa proposed the first quantum algorithm[11], which runs faster than its classical analog [31]. In 1994, Daniel Simon introduced a problem[12], which a quantum algorithm can solve exponentially faster than any known classical algorithm [32]. Inspired by this research, Peter Shor invented the polynomial-time algorithms for factorizing large numbers and the discrete logarithms [33]. These problems are widely believed to require an exponential amount of time on a CC. Therefore, Shor's factoring algorithm has been a legitimate threat to the classical cryptography based on the RSA encryption. Later, in 1997, another highly influential quantum algorithm, Grover's algorithm (GA) [34] for the *quantum search* [see Sec. 6.2]—quadratically faster than its classical counterpart—was invented. Hence, a large-scale QC will be able to solve certain problems with quantum algorithms and to simulate physical systems efficiently (much faster and with fewer resources than any CC).

Often, each step of a quantum algorithm is represented by a complex unitary gate. The efficiency of an algorithm is then derived in terms of the number of such gates. Even though, an algorithm does not rely on computation models, the realization of each complex unitary gate (step) of a quantum algorithm with one computation model can be advantageous over others in terms of resources. *Optimization* of resources such as qubits, entanglement, elementary operations and measurements is necessary for an efficient experimental implementation of an algorithm. Let us now review the UQCM [see Chapter 2] and the MQCM [see Chapter 3] by considering experimental optimization.

Both the UQCM and the MQCM are universal, can simulate each other and possess their own advantages. On one hand, no preparation of a resource state and classical information processing is required in the UQCM. On the other hand, measurements in the MQCM are simpler to execute than unitary gates to perform

[11]The Deutsch-Jozsa algorithm determines whether a function f is constant (equals to 1 or 0 over all the inputs) or balanced (equal to 1 for the half of inputs and equal to 0 for the other half).

[12]Basically, the Simon's algorithm is for finding a period under bitwise modulo-2 addition.

the computation. In practice, the difficult part in UQCM is to implement multiqubit gates, while for the MQCM it is to prepare a universal graph state. The bigger the graph state, the more difficult it is to control and protect it from noise. Based on these observations, to fulfill the need for experimental optimization, we introduce hybrid quantum computation model (HQCM) [41] in Chapter 4.

The HQCM employs the MQCM only to implement certain multiqubit gates, which are complicated in the UQCM. These multiqubit gates are realized by preparing small (non-universal) graph states in *one go* followed by *single shot* of measurements in the HQCM [see Sec. 3.1.5]. The implementation of an arbitrary single-qubit operation is rather straightforward in the UQCM, but it requires a chain of five qubits graph state in the MQCM [see Sec. 3.1.3]. Therefore, the HQCM chooses unitary evolution from the UQCM to execute single-qubit gates. Furthermore, the two-qubit controlled-Z (CZ) operations themselves are part of the experimental setup for constructing the graph states [see Sec. 3.1.1], and for this, we have to execute them with unitary evolution.

In conclusion, the set of single-qubit, the CZ and certain multiqubit gates is a set of *elementary gates* for the HQCM. In the HQCM, every complex unitary gate (of an algorithm) is written down in a sequence of the elementary gates, and they are carried out one after the other. The HQCM exploits the MQCM [37, 38, 39, 40] for executing the multiqubit gates and the UQCM [20, 21, 23] for executing single-qubit and the CZ gates.

Wherever measurements are involved in quantum information processing tasks (e.g., the quantum teleportation, the MQCM) classical information processing becomes crucial. Therefore, the second objective for this investigation is to develop a better understanding of the classical information processing in the HQCM, where part of a quantum circuit is simulated by unitary evolution and the rest by measurements on small graph states. The classical information processing in the HQCM turns out rather simple in comparison with the MQCM. It requires only the infor-

mation flow vector and the propagation matrices for the elementary gates. Furthermore, the total number of steps taken by a CC for the classical information processing is the total number of elementary gates in the decomposition of a complex unitary gate. No preprocessing or additional computational steps are required here.

We not only need a universal and scalable computation model but also need a fault-tolerant[13] model [25, 26, 27, 28] for building a proper QC. Chapter 5 contains the basic ideas for a fault-tolerant version of the HQCM. Where, we provide certain methods to implement encoded elementary gates within the hybrid model by taking the Steane 7-qubit code [7]. Besides, the classical information-processing parts of HQCM turns out completely suitable for its fault-tolerant version. These parts need the same information flow vector and the same propagation matrices, nothing more. This completes the introduction of Part I of this thesis. Let us now move to Part II, which is concerned with the quantum search problem.

In the quantum search problem [see Chapter 6], one has to find which one from a permissible set of unitary operators—the *oracles*—is employed by a given black box without actually opening the box. As stated before, the best performance for this search is provided by GA [34, 64] over its classical analog which is based on the hit and trial method. GA shows a quadratic speedup, but the answer from GA is the correct one, only with a high probability, not with certainty. It is, therefore, necessary to verify the answer.

Our prime motive for this investigation is not to speedup, but to design a *test* that confirms the answer produced by GA. This verification can be done with the aid of the *test states*. One such test state for each oracle is introduced in Chapter 7 [42]. The verification is a three-step process called a *single iteration of the test-state approach*. First, the test state corresponding to the GA-outcome is prepared. Sec-

[13] A device that works effectively even when its elementary components are imperfect is said to be fault tolerant.

ond, it is passed through the given black box. Finally, a measurement is performed to get a simple "yes/no" answer. As in the classical case, this measurement says "yes" or "no" if the test state matches the oracle or not. In conclusion, GA with the test-state verification [see Sec. 7.3] successfully terminates the search earlier than the purely GA. Thus, the performance of GA gets improved about 25%.

The test states can also be used for a classical-type search of the quantum data set (that is, the set of oracles)—called the *test-state search* [see Sec. 7.2]. In marked contrast to the purely classical approach, however, there are different "no" answers depending on the actual oracle and the measurement extracts the available information about the most probable oracle. The choice of test state for the next iteration is then guided by this gained information, and this guidance leads to a substantial reduction of the average number of trials needed before the successful termination of the search. The test-state approach to the quantum search is deterministic—it will give the correct answer after a finite number of oracle queries—and 3.41 times faster than the purely classical search. Since the test-state approach [of Chapter 7] and GA look for the same oracle, the average number of the black box queries of the test-state approach is the classical benchmark for GA. Chapter 8 concludes this thesis, and three appendixes contain the required additional material.

Part I

Hybrid Quantum Computation Model

Chapter 2

The unitary-evolution-based quantum computation model

A computer is a machine which stores input data, then processes it according to a set of instructions, and provides the output in a useful format in the end of computation [1, 2, 24]. Every computer is a composition of hardware on which information is processed and software by which information is processed. Hardware is the physical part of a computer, while software is a collection of computer programs (algorithms) designed to perform a required task. A QC is a device for computation that uses the fundamental concepts of quantum mechanics—such as superposition, the Heisenberg uncertainty principle, entanglement, etc., [see Sec. 2.1.1]—to process data. In other words, a QC emerges when the computation is executed under the framework of quantum mechanics [see Sec. 2.1.2].

There are several models for quantum computation. But the most widely used for practical reasons is the quantum circuit model or UQCM [20, 21, 23]. It is the quantum edition of the *reversible classical circuit model* [see Appendix A]. In the step from classical to quantum, the bits are replaced by qubits [see Secs. 2.2.1, 2.3.1, 2.4.1], and the logic gates are replaced by quantum gates (coherent unitary evolution) [see Secs. 2.2.2, 2.3.2, 2.4.2]. Unlike bits, qubits can exist in a superposition of

different computational states. Unlike the logic gates, the quantum gates are able to create and destroy a superposition as well as an entanglement.

Computation in the UQCM is run by a sequence of unitary gates and represented by its circuit diagram, where the connecting wires stand for the (logical) qubits or bits which carry the information, and the information is processed by the sequence of quantum gates. In the end, the result of the computation is read out by the projective measurements [see Sec. 2.2.3] on the qubits. The problem of designing quantum algorithms is largely the task of designing the corresponding quantum circuits.

The task of a QC is to simulate a quantum circuit or realize an arbitrary unitary operation on an input state. The UQCM is a universal quantum computational model in the sense that it can simulate any quantum circuit or realize any unitary operation [see Sec. 2.4.3].

2.1 Overview of quantum mechanics

2.1.1 Properties of quantum systems

- **Superposition:** *A quantum system can exist in all of its possible quantum states simultaneously.* Consequently, one must include every possible state with the associated probability of finding the system in that state to describe the complete state of system. Because of the superposition principle, many quantum algorithms—such as Deutsch's algorithm [31], GA [34] (also, see Sec. 6.2), and Shor's factoring algorithm [33] narrated in terms of the UQCM in the well-known textbook by Nielsen and Chuang [1]—are much faster than their classical analogs to solve some computational problems[1]. The superposition principle reveals the fact that quantum mechanics is a *lin-*

[1]This is also called *quantum parallelism*, where a QC simultaneously calculate the value of a given function for every possible input in a single run without any extra hardware.

ear theory. In quantum mechanics, evolution of a (isolated) system is given by the Schrödinger's equation [see Eq. (2.5)], which is a linear differential equation. Furthermore, physical quantities (observables) in quantum mechanics are represented by linear operators on the Hilbert space.

- **Indeterminism:** *Quantum mechanics can only give the probability of finding a system in a state.* In a deterministic theory, like classical mechanics, if a perfect knowledge of the state of a system is provided, one can (in principle, even without performing a measurement) determine the measurement results with certainty. In classical mechanics, probabilities are used only to describe situations where one's knowledge is incomplete. On the contrary, in quantum mechanics, when the same measurement is performed on several identically prepared systems, then one can not expect the same measurement outcome. This is not because of the lack of information about the state of system; rather, the measurement outcomes are intrinsically random and unpredictable. In a nutshell, quantum mechanics is *indeterministic* but, nevertheless, a *casual theory*[2].

- **Uncertainty:** *"Certain pairs of physical quantities in quantum mechanics, such as the spin of an electron in two orthogonal directions, cannot be simultaneously known to arbitrarily high precision" is the principle of uncertainty.* The more precisely one quantity is measured, the less precisely the other can be measured. This idea is used in the QKD protocol BB84 [11].

- **Quantum entanglement:** *It is possible that the subsystems of a composite quantum system do not have definite "properties*[3]*," whereas the composite system does.* In this situation, the subsystems are said to be *entangled.* Moreover, quantum entanglement cannot be created by *local* operations on the

[2] In a casual theory, the current state of a system implies the future state. In quantum mechanics, causality is given by the unitary evolution of a system [see Eq. (2.4)].

[3] But, of course, the subsystems do have well-defined mixed states.

subsystems. It plays a very crucial role in the field of quantum information [15]—the Ekert's protocol of QKD [12], the superdense coding [13], and the quantum teleportation [14]—as well as in the field of quantum computation—the MQCM [37, 38] given in Chapter 3.

- **Discrete spectra of bound systems:** *When a quantum system is in a static potential, only certain discrete energy levels are allowed*[4]. An isolated hydrogen atom and an electron in static magnetic field are the examples of a bound system with discrete spectrum. This discreteness is very useful in quantum communication and computation. For instance, the simplest quantum system is the two-level quantum system, which we call qubit [see Sec. 2.2]. It is the quantum analogous to a classical bit that can take on one of the two possible values 0, 1.

2.1.2 Postulates of quantum mechanics

The following four postulates of quantum mechanics consider the system in a pure state. Their generalization to mixed states[5] can be found in the well-known textbook by Nielsen and Chuang [1]. Throughout this thesis, Dirac's bra-ket notation is used to describe pure quantum states, and density matrices are used to describe mixed quantum states.

Postulate 1: State space

Every isolated physical system has an associated Hilbert space $\mathcal{H}_N$ of some dimension N, known as the state space of the system. The system is completely described by its state vector (ket) $|\psi\rangle$, which is a normalized vector in $\mathcal{H}_N$: $\langle\psi|\psi\rangle = 1$.

[4] Note that the *scattering states* exist in the *continuum*, of course, not in the *square-integrable* Hilbert space.

[5] A mixed quantum state is a *statistical ensemble of pure states*. A quantum state described by a density operator ρ is pure if $\mathrm{Tr}(\rho^2) = 1$ or mixed if $\mathrm{Tr}(\rho^2) < 1$, where Tr is the trace operation.

An arbitrary state with ket $|\psi\rangle$ of a given system can be written down in a linear combination of an orthonormal basis

$$\mathcal{S}_Q^N := \{|0\rangle, \ldots, |j\rangle, \ldots, |N-1\rangle\} \tag{2.1}$$

of the Hilbert space $\mathcal{H}_N$ in the following form

$$|\psi\rangle = \sum_{j=0}^{N-1} a_j \, |j\rangle \, , \tag{2.2}$$

where a_j are complex numbers[6] called probability amplitudes. The probability of finding the system in the state $|j\rangle$, if a *projective measurement*[7] in the basis $\mathcal{S}_Q^N$ is performed, is given by $|a_j|^2$. Furthermore, all these probabilities add up to one,

$$\sum_{j=0}^{N-1} |a_j|^2 = 1 \, , \tag{2.3}$$

which is nothing but the normalization condition.

For the case of $N = 2^{\mathsf{n}}$, the pure state $|\psi\rangle$ of Eq. (2.2) will be an arbitrary n-qubit state, and the set $\mathcal{S}_Q^N$ of Eq. (2.1) becomes the *computational basis* [see Sec. 2.4.1]. Later on, in Chapters 6 and 7, the elements of $\mathcal{S}_Q^N$ are called "index kets," where the subscript Q stands for quantum.

Postulate 2: Evolution

The time-evolution of a closed quantum system is given by a unitary operator U[8]. This means that the state $|\psi_{\text{in}}\rangle$ of system at time t_1 is related to the state $|\psi_{\text{out}}\rangle$ at a later time $t_2\,(> t_1)$ by a unitary operator $U(t_2, t_1)$ which depends only on the times t_2 and t_1,

$$|\psi_{\text{out}}\rangle := U(t_2, t_1) \, |\psi_{\text{in}}\rangle \, . \tag{2.4}$$

[6] Multiplication of a global phase to any ket has no observable physical consequences.
[7] Projective measurement is discussed in Postulate 3.
[8] $U^\dagger U = U U^\dagger = \mathbf{I}$, where $U^\dagger$ is the adjoint U, and $\mathbf{I}$ is the identity operator in $\mathcal{H}_N$.

The time-evolution of a closed system can also be given by the Schrödinger's wave equation

$$i\hbar\frac{\partial\psi}{\partial t} = \mathbf{H}\,\psi\,, \tag{2.5}$$

where $\hbar$ is the Planck's constant, ψ is the wave function corresponding to the ket $|\psi\rangle$, and $\mathbf{H}$ is a Hermitian operator ($\mathbf{H} = \mathbf{H}^\dagger$) known as the Hamiltonian of system. In the case of time independent Hamiltonian, $\mathbf{H}$ is associated with the unitary operator $U(t_2, t_1)$ of Eq. (2.4) by

$$U(t_2, t_1) := \exp\left[\frac{-i\,\mathbf{H}\,(t_2 - t_1)}{\hbar}\right]. \tag{2.6}$$

The UQCM is largely based on Eq. (2.4), where an initialized input state $|\psi_{\text{in}}\rangle$ is transformed into the output state $|\psi_{\text{out}}\rangle$ by applying a required unitary operation $U(t_2, t_1)$, which is realized by the corresponding Hamiltonian $\mathbf{H}$ of Eq. (2.6) in a laboratory. Finally, the output is read by measurements as described below. In quantum mechanics, unitary evolutions are casual (reversible processes), while measurements are probabilistic (irreversible processes).

Postulate 3: Measurement

Quantum measurements are given by a collection of measurement operators $\{M_m\}$*, which acts on the Hilbert space* $\mathcal{H}_N$ *of the system being measured. The measurement operator* M_m *corresponds to the measurement outcome* m *that may occur in the experiment. If the state of the given system is* $|\psi\rangle$ *immediately before the measurement then the probability of obtaining the outcome* m *is given by*

$$\text{prob}(m) = \langle\psi|\, M_m^\dagger\, M_m\, |\psi\rangle\,, \tag{2.7}$$

and after the measurement the state $|\psi\rangle$ *gets projected onto the state*

$$|\psi_m\rangle = \frac{M_m |\psi\rangle}{\sqrt{\langle\psi| M_m^\dagger M_m |\psi\rangle}} \tag{2.8}$$

in the idealized case of quantum non-demolition measurement[9].

Furthermore, all these probabilities add up to one,

$$\sum_m \text{prob}(m) = \sum_m \langle\psi| M_m^\dagger M_m |\psi\rangle = 1\,. \tag{2.9}$$

And, the completeness relation,

$$\sum_m M_m^\dagger M_m = \mathbf{I}\,, \tag{2.10}$$

is the consequence of Eq. (2.9), where $\mathbf{I}$ is the identity operation in the N-dimensional Hilbert space $\mathcal{H}_N$.

The general description of measurements given above can be rewritten in terms of the probability-operator measurement (POM) formalism [58, 59], where the POM elements Π_m associated with the measurement operators M_m are defined as

$$\Pi_m := M_m^\dagger M_m\,. \tag{2.11}$$

The set $\{\Pi_m\}$ with the completeness relation of Eq. (2.10), $\sum_m \Pi_m = \mathbf{I}$, is known as the POM, and its elements are non-negative self-adjoint operators ($\Pi_m^\dagger = \Pi_m \geq 0$) on the Hilbert space.

Generally, the measurement operators M_m are not orthogonal to each other, whereas a projective measurement is the special case of the POM in the sense that the measurement operators M_m are orthogonal to each other. Hence, the

[9] Quantum non-demolition measurement represents the ideal case of measurement, where the measured system is not destroyed by the measurement, but, of course, the state vector collapses.

operators M_m are Hermitian, $M_m^\dagger = M_m$, and satisfy the additional condition

$$M_m \, M_{m'} = M_m \, \delta_{m,m'} \tag{2.12}$$

with the completeness relation given by Eq. (2.10), where $\delta_{m,m'}$ is the Kronecker delta[10]. Consequently, all the POM elements are the same as the measurement operators in the case of projective measurement: $P_m^\dagger \, P_m = P_m$, it is customary to call the measurement operators P_m of a projective measurement as *projectors*.

Single-qubit projective measurements are discussed in Sec. 2.2.3, which are employed in Chapters 3 and 4 to run the computation. In Chapter 7, the POM and projectors are used to extract information.

Postulate 4: Composite system

The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. For example, if two subsystems a and b are in states with the kets $|\psi\rangle$ and $|\phi\rangle$ which lie in the state spaces $\mathcal{H}^{\mathsf{a}}$ and $\mathcal{H}^{\mathsf{b}}$, respectively. Then, their joint system with its associated state space $\mathcal{H}^{\mathsf{ab}} := \mathcal{H}^{\mathsf{a}} \otimes \mathcal{H}^{\mathsf{b}}$ is in a *product state* with the ket $|\psi\rangle \otimes |\phi\rangle$. Furthermore, if both the subsystems evolve under the influence of a joint Hamiltonian, then in general they will get entangled. The *Bell states* given by Eqs. (2.42) below are the examples of maximally entangled two-qubit quantum states, and the *graph states* used in Chapters 3 and 4 are multiqubit entangled states.

[10] $\delta_{m,m'} = 1$ for $m = m'$, and $\delta_{m,m'} = 0$ for $m \neq m'$.

2.2 Two-level quantum system: Qubit

2.2.1 Single-qubit state vector

The bit is the fundamental unit of classical information, it can either be in a state 0 or 1. Similarly, the qubit is the fundamental unit of quantum information. It is a two-dimensional quantum system, e.g., the two energy levels of the hyperfine splitting, the electron spin, the polarization of a photon, the presence or absence of a photon in a cavity, etc. A natural basis of the two-dimensional state space is $\{|0\rangle, |1\rangle\}$, the so-called computational basis. Unlike bit, qubit can exist in a superposition (linear combination),

$$|\psi(1)\rangle := a_0 |0\rangle + a_1 |1\rangle , \tag{2.13}$$

in the computational basis, where $|a_0|^2$ and $|a_1|^2$ are the probabilities of finding the qubit in the kets $|0\rangle$ and $|1\rangle$, respectively[11]. For a normalized state, $\langle\psi(1)|\psi(1)\rangle = 1$, these probabilities add up to one: $|a_0|^2 + |a_1|^2 = 1$.

Bloch sphere representation of a single-qubit state: Having $a_0 = \cos\left(\frac{1}{2}\theta\right)$ and $a_1 = e^{i\varphi}\sin\left(\frac{1}{2}\theta\right)$ to attach the following geometrical representation to an arbitrary single-qubit (pure) state, the ket $|\psi(1)\rangle$ of Eq. (2.13) can be rewritten as

$$|\uparrow(\theta,\varphi)\rangle := \cos\left(\tfrac{1}{2}\theta\right)|0\rangle + e^{i\varphi}\sin\left(\tfrac{1}{2}\theta\right)|1\rangle . \tag{2.14}$$

The single-qubit (pure) state with ket

$$|\downarrow(\theta,\varphi)\rangle := -\sin\left(\tfrac{1}{2}\theta\right)|0\rangle + e^{i\varphi}\cos\left(\tfrac{1}{2}\theta\right)|1\rangle \tag{2.15}$$

is orthogonal to the ket $|\uparrow(\theta,\varphi)\rangle$, and together they provide an alternative choice

[11] The index 1 of $|\psi(1)\rangle$ represents that the ket corresponds to a single-qubits state.

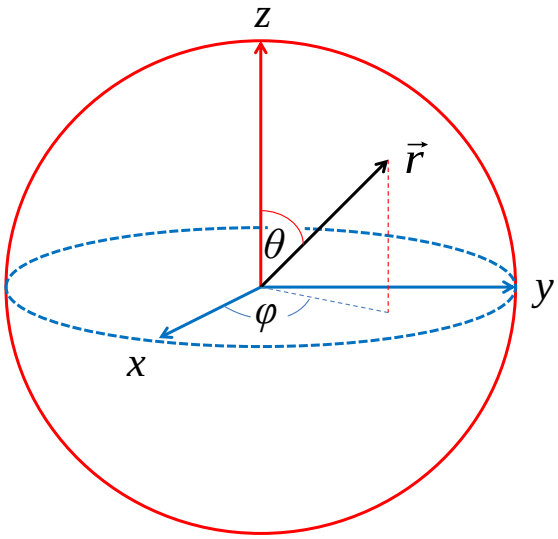


Figure 2.1: The Bloch vector $\vec{r}(\theta, \varphi)$ is depicted by the black arrow in the Bloch sphere.

of an orthonormal basis

$$\mathcal{B}_{\theta,\varphi} := \left\{ |\uparrow(\theta,\varphi)\rangle, |\downarrow(\theta,\varphi)\rangle \right\} \tag{2.16}$$

for a two-dimensional quantum system. These two parameters θ and φ which define the basis $\mathcal{B}_{\theta,\varphi}$, also define a pair of points on the boundary of unit three-dimensional sphere, known as the *Bloch sphere*[12]. The point on the boundary of the Bloch sphere corresponds to the ket $|\uparrow(\theta,\varphi)\rangle$—is given by the unit vector

$$\vec{r}(\theta,\varphi) := \left(\sin\theta\cos\varphi,\ \sin\theta\sin\varphi,\ \cos\theta\right), \tag{2.17}$$

called as the *Bloch vector*—is shown in Fig. 2.1, and its antipodal point represents its orthogonal ket $|\downarrow(\theta,\varphi)\rangle$.

Therefore, the Bloch sphere is a geometrical representation of the state space of a two-dimensional quantum system. This is because, there exist a one-to-one correspondence between the special unitary group SU(2) and the rotation group SO(3). And, that is why any single-qubit unitary operation (up to a global phase)

[12]The points on the boundary and in the interior of the Bloch sphere represent single-qubit pure and mixed states, respectively.

can be thought of a rotation of the Bloch sphere [see Eq. (2.20)]. A discussion of single-qubit unitary operations is given in the next section.

2.2.2 Single-qubit unitary operations

In the case of a single classical bit, there exist only two reversible logic gates: The *trivial* gate, which does not do anything, and the NOT gate (or, the *bit-flip* gate), which changes 0 into 1, and vice versa. In the quantum regime—every gate has to be a unitary operation[13]—the single-qubit identity operator I and the Pauli operator X act as the trivial gate and the bit-flip gate, respectively. In addition to these, there exist many non-trivial single-qubit gates—such as the Pauli operators Z and Y, called the *phase-flip* and *bit-phase-flip* gates, respectively—which do not have any classical analog.

Any single-qubit operation [see Eq. (2.20)] can be described as a linear combination the single-qubit identity operator I and the single-qubit Pauli vector operator

$$\vec{\sigma} := (\sigma_x, \sigma_y, \sigma_z) := (X, Y, Z)\,, \tag{2.18}$$

whose matrix forms in the computational basis representation are

$$I := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad X := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$
$$Y := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad Z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.19}$$

Consequently, every single-qubit operation can be represented by a 2×2 unitary matrix. Properties of the Pauli operators are listed in Table 2.1, where $j, k, l \in \{x, y, z\}$, and ϵ_{jkl} and δ_{jk} are the Levi-Civita symbol[14] and the Kronecker

[13] Strictly speaking, only the *completely positive and trace preserving* maps are allowed in quantum mechanics, which can be thought of unitary operations in a higher dimension Hilbert space.

[14] $\epsilon_{jkl} = 0$ except for $\epsilon_{xyz} = \epsilon_{yzx} = \epsilon_{zxy} = 1$, and $\epsilon_{zyx} = \epsilon_{yxz} = \epsilon_{xzy} = -1$.

Table 2.1: Properties of the single-qubit Pauli operators

$\sigma_j^\dagger$	$=$	σ_j	Hermitian
$\sigma_j^\dagger\,\sigma_j$	$=$	$\sigma_j\,\sigma_j^\dagger = I$	Unitary
$\sigma_j\,\sigma_k - \sigma_k\,\sigma_j$	$=$	$2i\sum_l \epsilon_{jkl}\,\sigma_l$	Noncommutative
$\sigma_j\,\sigma_k + \sigma_k\,\sigma_j$	$=$	$2\,\delta_{jk}\,I$	Anticommutative
$\det(\sigma_j)$	$=$	-1	Determinant
$\mathrm{Tr}(\sigma_j)$	$=$	0	Traceless

delta, respectively. Furthermore, the identity operator I and the Pauli operators of Eq. (2.18) with the multiplicative factors $\pm 1, \pm i$ form the Pauli group on a single qubit.

The most general single-qubit unitary operation (up to a global phase) is the single-qubit rotation around an axis $\vec{r}(\theta,\varphi)$ [as defined in Eq. (2.17) and shown in Fig. 2.1] by an angle υ

$$\begin{aligned} R_{\vec{r}}(\upsilon) &:= \exp\left(-i\frac{\upsilon}{2}\vec{r}\cdot\vec{\sigma}\right) \\ &= \cos\left(\tfrac{1}{2}\upsilon\right) I - i\sin\left(\tfrac{1}{2}\upsilon\right)\vec{r}\cdot\vec{\sigma}\,. \end{aligned} \tag{2.20}$$

The operation $R_{\vec{r}}(\upsilon)$ is called rotation, because its effect on a single-qubit state represented by the Bloch vector $\vec{n}$ is the rotation of $\vec{n}$ by an angle υ about the axis $\vec{r}$ of the Bloch sphere.

The Euler decomposition: This rotation R can be decomposed further into three elementary rotations[15] as

$$R(\alpha,\beta,\gamma) := R_z(\gamma)R_x(\beta)R_z(\alpha)\,, \tag{2.21}$$

where the angle parameters θ, φ of Eq. (2.17) and υ of Eq. (2.20) are related to the

[15] Where $R_z(\alpha) = \exp(-i\alpha Z/2)$, $R_x(\beta) = \exp(-i\beta X/2)$, and $R_z(\gamma) = \exp(-i\gamma Z/2)$.

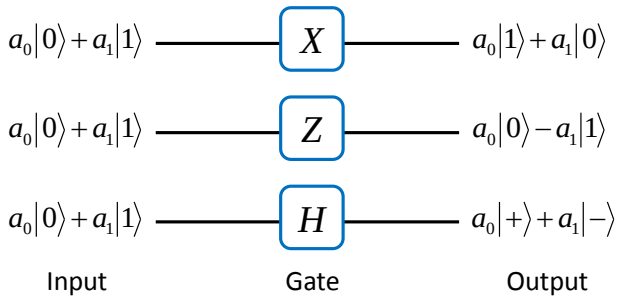


Figure 2.2: The action of quantum gates X, Z and H on the input state $|\psi(1)\rangle$ of Eq. (2.13) is depicted. The kets $|\pm\rangle$ are given by Eq. (2.28) below [1].

angle parameters α, β, γ of Eq. (2.21) by

$$\begin{aligned} \varphi &= \tfrac{1}{2}(\gamma - \alpha)\,, \\ \cos\left(\tfrac{1}{2}\upsilon\right) &= \cos\left(\tfrac{1}{2}\beta\right)\sin\left(\tfrac{1}{2}(\gamma + \alpha)\right)\,, \\ \sin\left(\tfrac{1}{2}\upsilon\right)\sin(\theta) &= \sin\left(\tfrac{1}{2}\beta\right)\,. \end{aligned} \tag{2.22}$$

In addition, the Pauli operators X, Y, and Z are the rotations by angle π around the x, y, and z axis, respectively.

Another very important single-qubit quantum gate—without analog in classical computation and heavily used in quantum computation—is the Hadamard gate

$$H := \frac{X+Z}{\sqrt{2}} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}; \tag{2.23}$$

which interchanges the bases[16] of X and Z:

$$HXH = Z\,, \qquad HYH = -Y\,, \qquad HZH = X\,. \tag{2.24}$$

The functioning of the X, Z, and H gates on a general single-qubit input state $|\psi(1)\rangle$ of Eq. (2.13) is shown in Fig. 2.2.

[16] $H|0\rangle = |+\rangle$, $H|1\rangle = |-\rangle$, and $[H]^2 = I$, where the ket $|\pm\rangle$ is given by Eq. (2.28) below.

2.2.3 Single-qubit projective measurements

As stated in Postulate 3 of Sec. 2.1.2 above, the measurement operator for a projective measurement are projectors. The single-qubit projectors are of the form

$$P_m := \frac{I + (-1)^m \, \vec{r} \cdot \vec{\sigma}}{2} \,, \tag{2.25}$$

where the measurement outcomes $m = 0$ and $m = 1$ mean that the measured qubit is projected onto the states with the kets $|\uparrow(\theta, \varphi)\rangle$ of Eq. (2.14) and $|\downarrow(\theta, \varphi)\rangle$ of Eq. (2.15), respectively. As a side remark, the kets $|\uparrow(\theta, \varphi)\rangle$ and $|\downarrow(\theta, \varphi)\rangle$ are the eigenkets of the observable $\vec{r} \cdot \vec{\sigma}$ which appears on the right-hand side of Eq. (2.25), and the corresponding eigenvalues are $(-1)^m$ for $m = 0$ and $m = 1$. For example, the eigenvalue equations for the single-qubit Pauli operator Z is

$$Z\,|0\rangle := +|0\rangle \,, \quad Z\,|1\rangle := -|1\rangle \,, \tag{2.26}$$

and for the single-qubit Pauli operator X it is

$$X\,|+\rangle := +|+\rangle \,, \quad X\,|-\rangle := -|-\rangle \,, \tag{2.27}$$

where

$$|\pm\rangle := \frac{|0\rangle \pm |1\rangle}{\sqrt{2}} \,. \tag{2.28}$$

The single-qubit projective measurement associated with P_m is called measurement in the basis $\mathcal{B}_{\theta,\varphi}$ of Eq. (2.16), measurement of the $\vec{r} \cdot \vec{\sigma}$ observable, or measurement along the axis specified by the Bloch vector $\vec{r}(\theta, \varphi)$ of Eq. (2.17). In other words, the choice of measurement basis is characterized by the direction (axis) of measurement $\vec{r}(\theta, \varphi)$, which is completely specified by the two parameters θ and φ in the Bloch sphere [see Fig. 2.1]. Single-qubit projectors will be used in Chapters 3 and 4 to execute the computation.

2.3 Two-qubit quantum system

2.3.1 Two-qubit state vector

An arbitrary pure state of a two-qubit system ab with a ket $|\psi(2)\rangle_{\mathsf{ab}}$ lies in the four-dimensional Hilbert space $\mathcal{H}_4^{\mathsf{ab}} := \mathcal{H}_2^{\mathsf{a}} \otimes \mathcal{H}_2^{\mathsf{b}}$, which is made of two copies of the single-qubit Hilbert space $\mathcal{H}_2$. The corresponding computational basis is a set of

$$\begin{aligned} |0\rangle &\equiv |00\rangle_{\mathsf{ab}}\,, \\ |1\rangle &\equiv |01\rangle_{\mathsf{ab}}\,, \\ |2\rangle &\equiv |10\rangle_{\mathsf{ab}}\,, \\ |3\rangle &\equiv |11\rangle_{\mathsf{ab}}\,, \end{aligned} \tag{2.29}$$

where the right-hand sides of Eqs. (2.29) are the binary representation of the left-hand sides, the ket $|00\rangle_{\mathsf{ab}}$ is the short-hand notation for the tensor product $|0\rangle_{\mathsf{a}} \otimes |0\rangle_{\mathsf{b}}$, and the same notation applies elsewhere. An arbitrary ket $|\psi(2)\rangle_{\mathsf{ab}}$ can be written down in a linear combination of these computational basis as

$$\begin{aligned} |\psi(2)\rangle_{\mathsf{ab}} &:= a_0\,|00\rangle_{\mathsf{ab}} + a_1\,|01\rangle_{\mathsf{ab}} + a_2\,|10\rangle_{\mathsf{ab}} + a_3\,|11\rangle_{\mathsf{ab}} \\ &= |0\rangle_{\mathsf{a}} \otimes |\chi_0\rangle_{\mathsf{b}} + |1\rangle_{\mathsf{a}} \otimes |\chi_1\rangle_{\mathsf{b}}\,; \end{aligned} \tag{2.30}$$

where

$$\begin{aligned} |\chi_0\rangle_{\mathsf{b}} &= a_0\,|0\rangle_{\mathsf{b}} + a_1\,|1\rangle_{\mathsf{b}}\,, \\ |\chi_1\rangle_{\mathsf{b}} &= a_2\,|0\rangle_{\mathsf{b}} + a_3\,|1\rangle_{\mathsf{b}}\,, \end{aligned} \tag{2.31}$$

and $|a_0|^2 + |a_1|^2 + |a_2|^2 + |a_3|^2 = 1$.

The Schmidt decomposition of a bipartite pure state [2]: The application of a singe-qubit unitary operator $U^{(\mathsf{a})}$—which operates only on qubit a, and whose action

on the computational basis is of the form

$$U^{(\mathsf{a})}|0\rangle_{\mathsf{a}} := \mu\,|0\rangle_{\mathsf{a}} + \nu\,|1\rangle_{\mathsf{a}}\,,$$
$$U^{(\mathsf{a})}|1\rangle_{\mathsf{a}} := -\nu^*\,|0\rangle_{\mathsf{a}} + \mu^*\,|1\rangle_{\mathsf{a}} \tag{2.32}$$

with $|\mu|^2 + |\nu|^2 = 1$—transforms the ket $|\psi(2)\rangle_{\mathsf{ab}}$ as

$$U^{(\mathsf{a})}|\psi(2)\rangle_{\mathsf{ab}} = |0\rangle_{\mathsf{a}} \otimes |\tilde{\chi}_0\rangle_{\mathsf{b}} + |1\rangle_{\mathsf{a}} \otimes |\tilde{\chi}_1\rangle_{\mathsf{b}}\,; \tag{2.33}$$

where

$$|\tilde{\chi}_0\rangle_{\mathsf{b}} = \mu\,|\chi_0\rangle_{\mathsf{b}} - \nu^*\,|\chi_1\rangle_{\mathsf{b}}\,,$$
$$|\tilde{\chi}_1\rangle_{\mathsf{b}} = \nu\,|\chi_0\rangle_{\mathsf{b}} + \mu^*\,|\chi_1\rangle_{\mathsf{b}}\,. \tag{2.34}$$

The coefficients μ and ν of $U^{(\mathsf{a})}$ are chosen in such a way that the kets $|\tilde{\chi}_0\rangle_{\mathsf{b}}$ and $|\tilde{\chi}_1\rangle_{\mathsf{b}}$ becomes orthogonal to each other, ${}_{\mathsf{b}}\langle\tilde{\chi}_1|\tilde{\chi}_0\rangle_{\mathsf{b}} = 0$, which implies

$$\mu^2\langle\chi_1|\chi_0\rangle - \nu^{*2}\langle\chi_0|\chi_1\rangle + \mu\nu^*\big(\langle\chi_0|\chi_0\rangle - \langle\chi_1|\chi_1\rangle\big) = 0\,. \tag{2.35}$$

If $\langle\chi_1|\chi_0\rangle \neq 0$, then Eq. (2.35) becomes a quadratic equation for $\frac{\mu}{\nu^*}$, which has two complex solutions. If μ of Eqs. (2.32) is a nonzero complex number, then either solution of Eq. (2.35) determines ν with the condition $|\mu|^2 + |\nu|^2 = 1$, and then both μ and ν defines the single-qubit unitary transformation $U^{(\mathsf{a})}$. If $\langle\chi_1|\chi_0\rangle = 0$, then Eqs. (2.30) and (2.33) have the same form, consequently $U^{(\mathsf{a})} = I$.

Subsequently, normalization of the kets $|\tilde{\chi}_0\rangle_{\mathsf{b}}$ and $|\tilde{\chi}_1\rangle_{\mathsf{b}}$ gives $|\bar{\chi}_0\rangle_{\mathsf{b}} = |\tilde{\chi}_0\rangle_{\mathsf{b}}/c_0$ and $|\bar{\chi}_1\rangle_{\mathsf{b}} = |\tilde{\chi}_1\rangle_{\mathsf{b}}/c_1$, where the normalization constants c_0 and c_1 are called the *Schmidt coefficients*. Hence, the set $\big\{|\bar{\chi}_0\rangle_{\mathsf{b}}, |\bar{\chi}_1\rangle_{\mathsf{b}}\big\}$ form the basis for qubit b. They are, therefore, related to the computational basis $\big\{|0\rangle_{\mathsf{b}}, |1\rangle_{\mathsf{b}}\big\}$ by a single-qubit uni-

tary transformation $V^{(\mathsf{b})}$:

$$V^{(\mathsf{b})}|0\rangle_{\mathsf{b}} := |\bar{\chi}_0\rangle_{\mathsf{b}}\,,$$
$$V^{(\mathsf{b})}|1\rangle_{\mathsf{b}} := |\bar{\chi}_1\rangle_{\mathsf{b}}\,. \tag{2.36}$$

Equation (2.33) then gives

$$|\psi(2)\rangle_{\mathsf{ab}} = U^{(\mathsf{a})\dagger}V^{(\mathsf{b})}\big[c_0|00\rangle_{\mathsf{ab}} + c_1|11\rangle_{\mathsf{ab}}\big]\,, \tag{2.37}$$

which is the *Schmidt decomposition* of the pure state $|\psi(2)\rangle_{\mathsf{ab}}$ of Eq. (2.30). The Schmidt decomposition exists for every bipartite pure state, while the unitary transformations $U^{(\mathsf{a})}$, $V^{(\mathsf{b})}$ and the Schmidt coefficients c_0, c_1 depend on the given bipartite pure state. Furthermore, the number of terms in the Schmidt decomposition (or the number of nonzero Schmidt coefficients) is called the *Schmidt number*. If the Schmidt number is more than one, the given bipartite pure state is *entangled* (or nonseparable); otherwise, it is *separable* (or unentangled).

2.3.2 Two-qubit unitary operations

In case of two qubits, controlled-unitary operations,

$$\Lambda^{\mathsf{a}}U^{(\mathsf{b})} := |0\rangle_{\mathsf{a}}\langle 0| \otimes I^{(\mathsf{b})} + |1\rangle_{\mathsf{a}}\langle 1| \otimes U^{(\mathsf{b})}\,, \tag{2.38}$$

are the most useful quantum gates [see Figs. 2.3(i) and 2.4], where the labels a and b are for the control and target qubits, respectively. $\Lambda^{\mathsf{a}}U^{(\mathsf{b})}$ applies the single-qubit unitary operation $U^{(\mathsf{b})}$ on the target qubit b if and only if the control qubit a is in the ket $|1\rangle_{\mathsf{a}}$. When the control is set to the ket $|0\rangle$, then the corresponding gate will be $\Delta^{\mathsf{a}}U^{(\mathsf{b})} := X^{(\mathsf{a})}\big[\Lambda^{\mathsf{a}}U^{(\mathsf{b})}\big]X^{(\mathsf{a})}$; throughout the thesis, the symbols Δ and Λ are used to represent the control is set to the kets $|0\rangle$ and $|1\rangle$, respectively. The

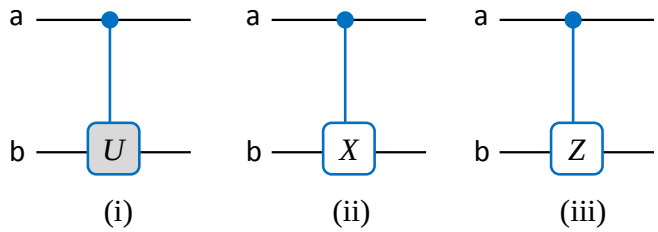


Figure 2.3: (i), (ii), and (iii) represent the two-qubit quantum gates $\Lambda^{\mathsf{a}}U^{(\mathsf{b})}$, $\textsc{cnot}(\mathsf{a},\mathsf{b})$, and $\textsc{cz}(\mathsf{a},\mathsf{b})$, respectively. Where the labels a and b are for the control and target qubits, and the controls are set to $|1\rangle_{\mathsf{a}}$.

two-qubit gate

$$\textsc{cnot}(\mathsf{a},\mathsf{b}) := \Lambda^{\mathsf{a}}X^{(\mathsf{b})} = |0\rangle_{\mathsf{a}}\langle 0| \otimes I^{(\mathsf{b})} + |1\rangle_{\mathsf{a}}\langle 1| \otimes X^{(\mathsf{b})} \tag{2.39}$$

displayed in Fig. 2.3(ii)—has its analog in classical computation—is a special case of $\Lambda^{\mathsf{a}}U^{(\mathsf{b})}$, and $[\textsc{cnot}]^2 = I \otimes I$.

One can generate any two-qubit state (say, the ket given by Eqs. (2.30) and (2.37)) with a combination of the $\textsc{cnot}$ gate and some single-qubit gates. Equation (2.37) can be rewritten as

$$|\psi(2)\rangle_{\mathsf{ab}} = U^{(\mathsf{a})\dagger}V^{(\mathsf{b})}\textsc{cnot}(\mathsf{a},\mathsf{b})\big[c_0|0\rangle_{\mathsf{a}} + c_1|1\rangle_{\mathsf{a}}\big] \otimes |0\rangle_{\mathsf{b}}\,. \tag{2.40}$$

Since $c_0|0\rangle_{\mathsf{a}} + c_1|1\rangle_{\mathsf{a}}$ is a normalized ket, it can be obtained by applying a single-qubit unitary operation $W^{(\mathsf{a})}$ on a standard input ket $|0\rangle_{\mathsf{a}}$:

$$|\psi(2)\rangle_{\mathsf{ab}} = U^{(\mathsf{a})\dagger}V^{(\mathsf{b})}\textsc{cnot}(\mathsf{a},\mathsf{b})W^{(\mathsf{a})}|00\rangle_{\mathsf{ab}}\,. \tag{2.41}$$

In conclusion, a general two-qubit ket $|\psi(2)\rangle_{\mathsf{ab}}$ is constructed, here, out of the standard ket $|00\rangle_{\mathsf{ab}}$ with three single-qubit gates and one $\textsc{cnot}$ gate of Eq. (2.39). As a special case of Eq. (2.41), when the unitary operations $W = H$, $U = I$ and V is either the identity or a Pauli operator of Eq. (2.19), then the two-qubit state

$|\psi(2)\rangle_{\mathsf{ab}}$ becomes one of the Bell states:

$$
\begin{aligned}
|\Phi^+\rangle_{\mathsf{ab}} &:= \frac{1}{\sqrt{2}}\big[|00\rangle_{\mathsf{ab}} + |11\rangle_{\mathsf{ab}}\big] \quad \text{for} \quad V = I\,, \\
|\Phi^-\rangle_{\mathsf{ab}} &:= \frac{1}{\sqrt{2}}\big[|00\rangle_{\mathsf{ab}} - |11\rangle_{\mathsf{ab}}\big] \quad \text{for} \quad V = Z\,, \\
|\Psi^+\rangle_{\mathsf{ab}} &:= \frac{1}{\sqrt{2}}\big[|01\rangle_{\mathsf{ab}} + |10\rangle_{\mathsf{ab}}\big] \quad \text{for} \quad V = X\,, \\
i|\Psi^-\rangle_{\mathsf{ab}} &:= \frac{i}{\sqrt{2}}\big[|01\rangle_{\mathsf{ab}} - |10\rangle_{\mathsf{ab}}\big] \quad \text{for} \quad V = Y\,.
\end{aligned} \tag{2.42}
$$

The Bell states provide an alternative choice of basis for a two-qubit system.

Decomposition of the two-qubit controlled-unitary operation $\Lambda^{\mathsf{a}}U^{(\mathsf{b})}$ [23, 1]: With Eq. (2.21), a general single-qubit operation U can be decomposed (up to a global phase) into three elementary rotations:

$$
\begin{aligned}
U &\equiv R_z(\gamma)R_x(\beta)R_z(\alpha) \\
&= R_z\left(\gamma - \frac{\pi}{4}\right) R_y\left(\beta\right) R_z\left(\alpha + \frac{\pi}{4}\right) \\
&= R_z\left(\gamma - \frac{\pi}{4}\right) R_y\left(\frac{\beta}{2}\right) R_y\left(\frac{\beta}{2}\right) R_z\left(\frac{\alpha+\gamma}{2}\right) R_z\left(\frac{\alpha-\gamma}{2} + \frac{\pi}{4}\right) \\
&= R_z\left(\gamma - \frac{\pi}{4}\right) R_y\left(\frac{\beta}{2}\right) XR_y\left(-\frac{\beta}{2}\right) R_z\left(-\frac{\alpha+\gamma}{2}\right) XR_z\left(\frac{\alpha-\gamma}{2} + \frac{\pi}{4}\right) \\
&= AXBXC\,,
\end{aligned} \tag{2.43}
$$

where the unitary operations

$$
\begin{aligned}
A &:= R_z\left(\gamma - \frac{\pi}{4}\right) R_y\left(\frac{\beta}{2}\right)\,, \\
B &:= R_y\left(-\frac{\beta}{2}\right) R_z\left(-\frac{\alpha+\gamma}{2}\right)\,, \\
C &:= R_z\left(\frac{\alpha-\gamma}{2} + \frac{\pi}{4}\right)
\end{aligned} \tag{2.44}
$$

are such that $ABC = I$. From Eqs. (2.43), we have the decomposition of $\Lambda^{\mathsf{a}}U^{(\mathsf{b})}$—shown in Fig. 2.4—in terms of two CNOT gates and the three single-qubits gates

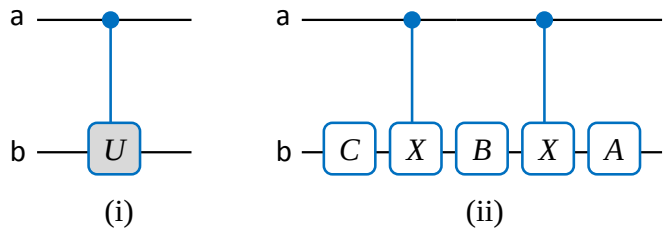


Figure 2.4: (i) represents the two-qubit quantum gate $\Lambda^{\mathsf{a}}U^{(\mathsf{b})}$, and (ii) represents its decomposition in terms of two CNOT gates and the three single-qubits gates A, B, and C. The labels a and b are for the control and target qubits, and the controls are set to $|1\rangle_{\mathsf{a}}$.

A, B, and C. In fact, a general n-qubit quantum gate can be constructed with a combination of single-qubit and the CNOT gates [see Sec. 2.4.3].

Having $\alpha = \beta = 0$ for U in Eqs. (2.43), the two-qubit gate $\Lambda^{\mathsf{a}}U^{(\mathsf{b})}$ of Eq. (2.38) becomes the controlled-PHASE (CPHASE) gate,

$$\text{CPHASE}(\mathsf{a},\mathsf{b}) := \Lambda^{\mathsf{a}}R_z^{(\mathsf{b})}(\gamma) = |0\rangle_{\mathsf{a}}\langle 0| \otimes I^{(\mathsf{b})} + |1\rangle_{\mathsf{a}}\langle 1| \otimes R_z^{(\mathsf{b})}(\gamma)\,. \tag{2.45}$$

The CPHASE gate has no classical analog. In the special cases, where a nonzero γ is an odd multiple of π, the CPHASE gate turns into the two-qubit gate

$$\text{CZ}(\mathsf{a},\mathsf{b}) := \Lambda^{\mathsf{a}}Z^{(\mathsf{b})} = |0\rangle_{\mathsf{a}}\langle 0| \otimes I^{(\mathsf{b})} + |1\rangle_{\mathsf{a}}\langle 1| \otimes Z^{(\mathsf{b})}\,. \tag{2.46}$$

The CZ gate depicted in Fig. 2.3(iii) is the main entangling operation to generate the graph states [35, 36] used in Chapters 3 and 4. Furthermore, it is equivalent to the quantum CNOT gate sandwiched between two Hadamard gates,

$$\text{CZ}(\mathsf{a},\mathsf{b}) = H^{(\mathsf{b})}\,\text{CNOT}(\mathsf{a},\mathsf{b})\,H^{(\mathsf{b})}\,, \tag{2.47}$$

which is a simple consequence of Eqs. (2.24), and $[\text{CZ}]^2 = I \otimes I$.

Another interesting two-qubit gate is the SWAP gate, which interchanges the state of two qubits (bits)[17] and works in both classical and quantum computation.

[17] $\text{SWAP}\,|j_{\mathsf{a}}\,j_{\mathsf{b}}\rangle = |j_{\mathsf{b}}\,j_{\mathsf{a}}\rangle$, where $j_{\mathsf{a}}, j_{\mathsf{b}} \in \{0,1\}$.

It can be constructed with a combination of three CNOT gates,

$$\begin{aligned}\text{SWAP}(\mathsf{a},\mathsf{b}) = \text{SWAP}(\mathsf{b},\mathsf{a}) &:= |00\rangle_{\mathsf{ab}}\langle 00| + |01\rangle_{\mathsf{ab}}\langle 10| + |10\rangle_{\mathsf{ab}}\langle 01| + |11\rangle_{\mathsf{ab}}\langle 11| \\ &= \text{CNOT}(\mathsf{a},\mathsf{b})\,\text{CNOT}(\mathsf{b},\mathsf{a})\,\text{CNOT}(\mathsf{a},\mathsf{b})\,, \end{aligned} \tag{2.48}$$

and $[\text{SWAP}]^2 = I \otimes I$.

2.4 n-qubit quantum system

2.4.1 n-qubit state vector

A digital CC works with the binary-number system and according to Boolean algebra. Where, an integer j in the range $0 \leq j < 2^{\mathsf{n}}$ can be expressed in terms of n bits as

$$j \equiv \sum_{\mathsf{m}=1}^{\mathsf{n}} j_{\mathsf{m}}\, 2^{\mathsf{m}-1}\,; \tag{2.49}$$

where $j_{\mathsf{n}} \cdots j_1$ is the corresponding binary number, and $j_{\mathsf{m}} \in \{0, 1\}$ is the value of mth bit.

The same idea can be utilized for qubits, where a ket $|j\rangle$ for $0 \leq j < N$ represents one of N orthonormal kets. In the case of $N = 2^{\mathsf{n}}$, these orthonormal kets,

$$\begin{aligned} |j\rangle &\equiv \bigotimes_{\mathsf{m}=1}^{\mathsf{n}} |j_{\mathsf{m}}\rangle \\ &\equiv |j_{\mathsf{n}}\, j_{\mathsf{n}-1} \cdots j_2\, j_1\rangle\,, \end{aligned} \tag{2.50}$$

constitute the computational basis [see Eq. (2.1)] for a n-qubit system. Later, in Chapters 6 and 7, they will be called "index kets." A general n-qubit ket can be expressed in a linear combination of the computational basis as given by Eq. (2.2).

2.4.2 n-qubit unitary operations

A general n-qubit quantum gate can be represented by a $2^n \times 2^n$ unitary matrix in the computational basis. Among them, the most useful unitary operations are n-qubit controlled unitary operation of the form

$$\begin{aligned}\Lambda^{1\cdots c}U^{(c+1)\cdots n} := &\left[I^{\otimes c} - |1\cdots 1\rangle_{1\cdots c}\langle 1\cdots 1|\right] \otimes I^{\otimes(n-c)} \\ &+ |1\cdots 1\rangle_{1\cdots c}\langle 1\cdots 1| \otimes U^{(c+1)\cdots n}\,, \end{aligned} \tag{2.51}$$

where the qubits labeled 1 to c are the control qubits and the qubits labeled $c+1$ to n are the target qubits. Only if every control qubit is in the ket $|1\rangle$, then the $(n-c)$-qubit unitary operation $U^{(c+1)\cdots n}$ applies on the target qubits. When every control is set to the ket $|0\rangle$, then $\Delta^{1\cdots c}U^{(c+1)\cdots n} := X^{\otimes c}\left[\Lambda^{1\cdots c}U^{(c+1)\cdots n}\right]X^{\otimes c}$ is the corresponding gate. In the case of $c = n-1$, the n-qubit controlled unitary operation of Eq. (2.51) becomes $\Lambda^{1\cdots(n-1)}U^{(n)}$, a so-called *two-level unitary operation*[18]. Any $2^n \times 2^n$ unitary matrix can be built up as a product of at most $2^{n-1}(2^n-1)$ number of two-level unitary matrices [1, 22].

Two-level unitary operation: Every single-qubit gate and the two-qubit gates $\Lambda^a U^{(b)}$, CNOT, CZ, SWAP are examples of two-level unitary operations. Some important examples of three-qubit two-level unitary operations are

$$\text{CCNOT}(a,b,c) := \Lambda^{ab}X^{(c)} = |0\rangle_a\langle 0| \otimes I^{\otimes 2} + |1\rangle_a\langle 1| \otimes \text{CNOT}(b,c)\,, \tag{2.52}$$

$$\begin{aligned}\text{CCZ}(a,b,c) := \Lambda^{ab}Z^{(c)} &= |0\rangle_a\langle 0| \otimes I^{\otimes 2} + |1\rangle_a\langle 1| \otimes \text{CZ}(b,c) \\ &= H^{(c)}\left[\Lambda^{ab}X^{(c)}\right]H^{(c)}\,, \end{aligned} \tag{2.53}$$

$$\begin{aligned}\text{CSWAP}(a,b,c) &:= |0\rangle_a\langle 0| \otimes I^{\otimes 2} + |1\rangle_a\langle 1| \otimes \text{SWAP}(b,c) \\ &= \left[\Lambda^{ab}X^{(c)}\right]\left[\Lambda^{ac}X^{(b)}\right]\left[\Lambda^{ab}X^{(c)}\right]. \end{aligned} \tag{2.54}$$

[18] Two-level unitary matrices are those which act non-trivially only on two-or-fewer vector components.

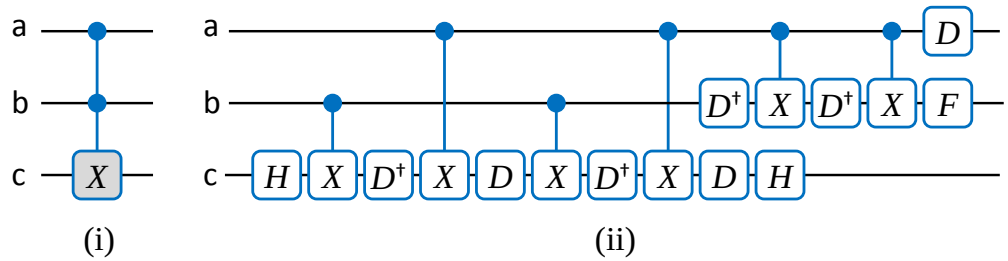


Figure 2.5: (i) represents the quantum Toffoli (CCNOT) gate of Eq. (2.52), and (ii) represents its decomposition in terms of single-qubit and the CNOT gates. In (ii), the single-qubit H, D gates are the Hadamard gate of Eq. (2.23), $R_z(\pi/4)$, respectively, and $D^\dagger = R_z(-\pi/4)$, $F = D^2$. The controls are set to $|1\rangle$ for the Toffoli gate in (i) and for every CNOT gate in (ii).

The controlled-controlled-NOT (CCNOT) and controlled-SWAP (CSWAP) gates are also called Toffoli and Fredkin gates, respectively. Both of them exist in the classical case [see Appendix A] as well. The controlled-controlled-Z (CCZ) gate has no classical analog and is equivalent to the Toffoli (CCNOT) gate [see Eq. (2.53)] sandwiched between two Hadamard gates because of Eq. (2.47). The Fredkin (CSWAP) gate can be written as a combination of three Toffoli (CCNOT) gates [see Eq. (2.54)]. It is a simple consequence of Eqs. (2.48).

The Toffoli gate is universal for *reversible classical computation* [see Appendix A]. On one hand, in the classical regime, single- and two-bit reversible gates are not sufficient to implement the Toffoli gate. On the other hand, in quantum computation, the Toffoli gate can be decomposed further in a sequence of single- and two-qubit gates, such a sequence is given in Fig. 2.5. Every single-qubit gate—the Hadamard gate H, the $\pi/2$-phase gate

$$F := R_z\left(\tfrac{1}{2}\pi\right) = \exp\left(-i\frac{\pi}{4}Z\right) , \tag{2.55}$$

and the $\pi/4$-phase gate

$$D := R_z\left(\tfrac{1}{4}\pi\right) = \exp\left(-i\frac{\pi}{8}Z\right) \tag{2.56}$$

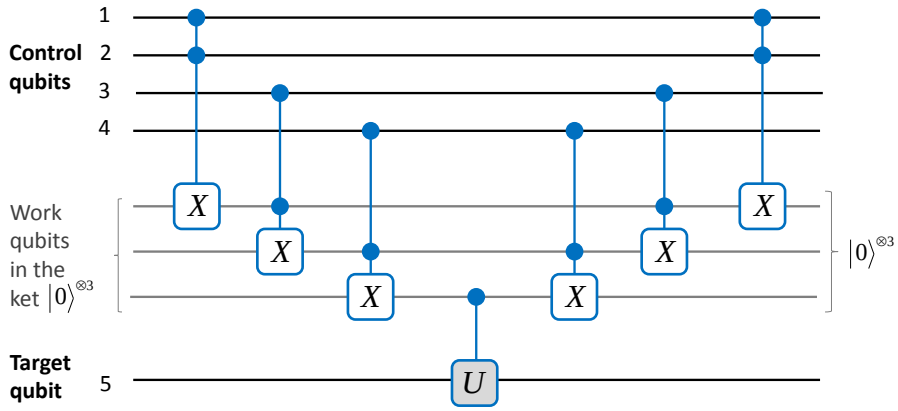


Figure 2.6: The quantum circuit for implementing the five-qubit gate $\Lambda^{1234}U^{(5)}$. From top to bottom, the four black horizontal lines represent control qubits 1 to 4, and the next two, gray horizontal lines represent three work qubits prepared in the ket $|0\rangle^{\otimes 3}$. The black horizontal line at the bottom represents target qubit 5. Here, the two-qubit controlled-U and every three-qubit Toffoli gates are implemented by the circuits shown on Figs. 2.4(ii) and 2.5(ii), respectively.

—of this sequence has no classical analog[19]. Note that if one removes the Hadamard gates from Fig. 2.5(ii), then the circuit executes the CCZ gate.

A very important use of the Toffoli gates is in the implementation of n-qubit two-level unitary operations. For example, the n-qubit gate $\Lambda^{1\cdots(\mathsf{n}-1)}U^{(\mathsf{n})}$ of Eq. (2.51) (for $\mathsf{c}=\mathsf{n}-1$) can be realized by the kind of circuit shown in Fig. 2.6 (for $\mathsf{n}=5$), where $2(\mathsf{n}-2)$ Toffoli gates with $(\mathsf{n}-2)$ work qubits—which start and end in the ket $|0\rangle^{\otimes(\mathsf{n}-2)}$—are used. In Sec. 4.3.3, the implementation of the gate $\Lambda^{1\cdots(\mathsf{n}-1)}Z^{(\mathsf{n})}$ with the HQCM (for $\mathsf{n}=4$) is given [see Fig. 4.3].

2.4.3 Universal set of quantum gates

A general quantum unitary operation can be built up from a set of standard unitary operations called *universal set of gates*. This situation is rather analogous to the situation in classical logics, where any Boolean function can be built up from a set of standard logical gates on one and two bits. In classical computation, {AND, NOT}, {OR, NOT}, {NAND}, {NOR} are examples of universal set of gates [see Appendix A]. But for the "reversible" classical computation, the Toffoli (or Fredkin) gate alone with ancilla qubits is (universal) sufficient to implement any Boolean function. Since

[19]Note that, here, the definition of phase gates F and D is different from Ref. [1].

the Toffoli gate has a direct quantum equivalent, a QC can perform any operation that a CC can do.

As we learned from Sec. 2.4.2, a general operation on a system of n qubits can be represented by a $2^{\mathsf{n}} \times 2^{\mathsf{n}}$ unitary matrix in the computational basis, which can be decomposed in a sequence of at most $2^{\mathsf{n}-1}(2^{\mathsf{n}}-1)$ number of two-level unitary matrices [1, 22]. Furthermore, a sequence of the Toffoli, the CNOT and single-qubit gates with $\mathsf{n}-2$ work qubits is sufficient to implement any two-level unitary operation [see Fig. 2.5]. The Toffoli gate itself can be written as a product of single-qubit and the CNOT gates [see Fig. 2.6]. In few words, single-qubit [see Eq. (2.20)] and the CNOT [see Eq. (2.39)] gates form a universal set of gates, $\{R_{\vec{r}}(\upsilon), \text{CNOT}\}$, for quantum computation [23].

This universal set cannot be reduced further, because the CNOT gate cannot be built up and entanglement cannot be generated with single-qubit operations only. But, the CNOT gate can be transformed into the CZ gate by Eq. (2.47), hence $\{R_{\vec{r}}(\upsilon), \text{CZ}\}$ is another universal set of gate. In these universal sets of gates, $R_{\vec{r}}(\upsilon)$ represents the whole (continuous[20]) family of single-qubit gates. The gates of a universal set form the building blocks for QC. A general unitary operation on n qubits can always be implemented *exactly* with a sequence containing $O(\mathsf{n}^2 4^{\mathsf{n}})$ single-qubit and the CNOT (or CZ) gates [1].

Since any single-qubit operation can be approximated up to an arbitrary accuracy using only the Hadamard and $\pi/4$-phase (D) gates, there exists a discrete set of universal gates: $\{H, F, \text{CNOT}, D\}$ [28]. A general unitary operation is continuous, hence, it can not be implemented "exactly" by a combination of the gates of this discrete set, but can be approximated up to an arbitrary accuracy. This discrete set is very useful from the point of view of fault-tolerant QC [25, 26, 27, 28]. Another discrete set of universal gates $\{H, F, \text{CNOT}, \text{Toffoli}\}$ is also available for fault-tolerant QC [25, 1].

[20] Because, the angle parameters υ and (θ, φ) of $\vec{r}$ of Eq. (2.17) vary continuously.

Chapter 3

The measurement-based quantum computation model

The MQCM is another well-recognized model for QC [37, 38]. Here, a multiqubit entangled state—known as a *cluster state* [35] or, more generally, a *graph state*[1] [36]—is the main ingredient. It provides all the entanglement beforehand for the subsequent computation. Computation in the MQCM is run by a sequence of single-qubit adaptive[2] projective measurements on the graph state. The MQCM [see Sec. 3.1] enables one to simulate any quantum circuit on a sufficiently large two-dimensional square graph [see Fig. 3.1] state by arranging the spatial pattern of measurement bases for the graph qubits according to the temporal order of quantum gates in the circuit. Both the UQCM of Chapter 2 and the MQCM are universal: they can simulate any quantum circuit. Where the UQCM uses the unitary gates, the MQCM uses the measurements for simulating a circuit.

In the MQCM, the measurements on graph qubits are performed in a certain temporal order for the purpose of running the computation deterministically. Fur-

[1] There exists a mathematical graph for every graph state, where the vertices of graph stand for the qubits, and its edges stand for the entangling operations [see Fig. 3.1]. Furthermore, the graph states of one-, two-, and three-dimensional square lattices are called cluster states, so in this sense the cluster state is the special case of the graph state.

[2] Generally, in the MQCM, the measurement bases for the as yet unmeasured qubits are adapted according to the outcomes from the measured qubits.

thermore, the measurement outcomes are recorded classically and are used for setting the measurement bases for the subsequent measurements and for the interpretation of the final result [39]. This is called the classical information processing necessary to the MQCM [see Sec. 3.2]. By contrast, in the UQCM, there is no such temporal order of measurements, but an order in which the unitary gates are executed. Basically, the MQCM can be summarized in the following four steps:

1. A sufficiently large two-dimensional square graph state of qubits is prepared [see Sec. 3.1.1].

2. A *spatial pattern of measurement bases* is assigned to the graph qubits according to the *temporal order of gates* in a quantum circuit under simulation.

3. A sequence of single-qubit adaptive projective measurements is performed in a certain *temporal* order.

4. In parallel, the measurement outcomes—the classical data—are recorded and processed with a CC.

These steps are comprehensively discussed in this chapter, which is made of three sections. In Sec. 3.1, we focus on the realization of individual gates in the framework of the MQCM. The classical information processing is discussed in Sec. 3.2. The results from Secs. 3.1.1, 3.1.5 and 3.2 will be used in Chapter 4. In Sec. 3.3, an efficient measurement scheme for simulating a quantum circuit on a graph state is provided.

3.1 Methodology for computation in the MQCM

In the beginning of this section, a short introduction about the preparation of graph states [35, 36] is given. The kind of single-qubit measurements which are useful for the MQCM, and the realization of some important individual gates [37, 38, 40] are presented in the following parts of this section.

3.1.1 Preparation of graph states

Graph states can be realized in many physical systems by first preparing all the qubits of graph $\mathcal{G}$ in an eigenstate of their respective Pauli operator X. In other words, a qubit a of $\mathcal{G}$ is initialized in the ket

$$|(-1)^{\kappa_{\mathsf{a}}}\rangle_{\mathsf{a}} := \frac{|0\rangle_{\mathsf{a}} + (-1)^{\kappa_{\mathsf{a}}}|1\rangle_{\mathsf{a}}}{\sqrt{2}}\,, \tag{3.1}$$

where $\kappa_{\mathsf{a}} = 0, 1$ and the corresponding kets $|+\rangle_{\mathsf{a}}$, $|-\rangle_{\mathsf{a}}$ are the eigenvalues and the eigenkets [see Eqs. (2.27) and (2.28)] of the Pauli operator X. Then entanglement between each pair of nearest-neighbor qubits is established by the $\textsc{cz}(\mathsf{a},\mathsf{b})$ gate of Eq. (2.46), where the indices a and b stand for the qubits at lattice site a and its nearest-neighbor lattice site b of graph $\mathcal{G}$, respectively. A unitary gate of this kind can be generated by turning on the (controlled) Ising-type nearest-neighbor interaction for an appropriately chosen time period. Experimentally, graph states have been generated using controlled collisions between cold atoms in optical lattices [43] or using linear optics [44, 45, 46, 47, 56, 57].

The graph state associated with a two-dimensional square graph (lattice) as depicted in Fig. 3.1 is sufficient for universal quantum computation. In this figure, the graph qubits are depicted by circles, and the $\textsc{cz}$ operations are depicted by links between the circles.

Mathematically, quantum correlations among the qubits of a graph are specified by correlation operators $K^{(\mathsf{a})}$'s, which are given below. The resultant graph state $|\Phi_{\{\kappa\}}\rangle_{\mathcal{G}}$ is an eigenstate of these operators, and it is completely specified by the set of eigenvalue equations

$$\begin{aligned} K^{(\mathsf{a})}|\Phi_{\{\kappa\}}\rangle_{\mathcal{G}} &:= X^{(\mathsf{a})} \otimes \left(\bigotimes_{\mathsf{b}\,\in\,\mathrm{nbh}(\mathsf{a})} Z^{(\mathsf{b})} \right) |\Phi_{\{\kappa\}}\rangle_{\mathcal{G}} \\ &= (-1)^{\kappa_{\mathsf{a}}}|\Phi_{\{\kappa\}}\rangle_{\mathcal{G}} \end{aligned} \tag{3.2}$$

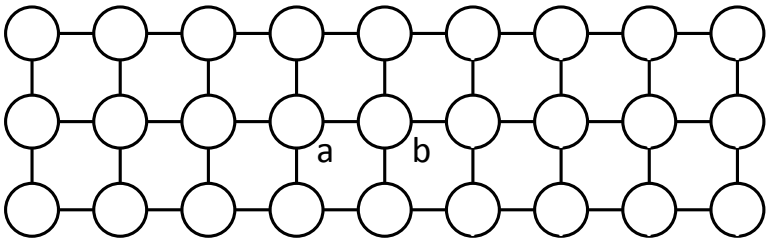


Figure 3.1: A two-dimensional square graph, where the vertices (circles) represent qubits, and the edges (bonds) represent the two-qubit CZ operations of Eq. (2.46).

with the set of eigenvalues $\{\kappa\} := \{\kappa_{\mathsf{a}} \in \{0,1\} | \, \mathsf{a} \in \mathcal{G}\}$. Here, nbh($\mathsf{a}$) stands for the set of all nearest-neighbor qubits which are entangled (connected) to qubit a by the CZ operations. For every qubit a of the graph state $|\Phi_{\{\kappa\}}\rangle_{\mathcal{G}}$, there exists a correlation operator $K^{(\mathsf{a})}$ and an eigenvalue $\kappa_{\mathsf{a}} \in \{0,1\}$. The physical meaning of Eq. (3.2) is that there exists either a correlation ($\kappa_{\mathsf{a}} = 0$) or an anticorrelation ($\kappa_{\mathsf{a}} = 1$) between the outcome of the measurement on qubit a in the X eigenbasis and the outcomes of the measurements on all the qubits of nbh(a) in the Z eigenbasis. These "quantum" correlations provide the framework for "quantum" computation in the MQCM.

3.1.2 Single-qubit measurements on graph state

Once the resource graph state is ready, then the logical qubits—holding the input information—are attached to the resource via the same entangling operations given by Eq. (2.46). Throughout this thesis, excluding Chapter 5, one logical qubit represents one physical qubit. Now, the computation is carried out by a sequence of single-qubit adaptive projective measurements in a certain measurement bases and in a certain temporal order.

As explained in Sec. 2.2.3, the single-qubit projective measurement axis, expressed by the Bloch vector $\vec{r}(\theta,\varphi)$ of Eq. (2.17), is completely characterized by the two real parameters θ and φ. Where, $\mathcal{B}_{\theta,\varphi}$ of Eq. (2.16) and P_m of Eq. (2.25) are the corresponding measurement basis and projector, respectively. The measurement outcomes $m = 0$ and $m = 1$ mean that the measured qubit is projected

onto the states with the kets $|\uparrow(\theta,\varphi)\rangle$ of Eq. (2.14) and $|\downarrow(\theta,\varphi)\rangle$ of Eq. (2.15), respectively.

Three single-qubit projective measurements—for three different sets of values of the angle parameters θ and φ—are exploited in the MQCM [37, 38, 40]. They are given in the following.

Z-measurement:

Measurement along the z axis ($\theta = 0$). It effectively detaches the measured (redundant) qubits from the graph state.

XY-measurement:

Measurement along the Bloch vector $\vec{r}_{xy}(\varphi) := \vec{r}\big(\frac{1}{2}\pi, \varphi\big) = \big(\cos\varphi,\, \sin\varphi,\, 0\big)$—it lies in the x-y plane of the Bloch sphere—processes the information as well as teleporting it from one place to another on the graph. This kind of measurements are employed for implementing the individual gates of Secs. 3.1.3 and 3.1.4.

ZY-measurement:

Measurement along the Bloch vector $\vec{r}_{zy}(\theta) := \vec{r}\big(\theta, \frac{1}{2}\pi\big) = \big(0,\, \sin\theta,\, \cos\theta\big)$—it lies in the z-y plane of the Bloch sphere—only processes the information. Their importance is revealed—for the implementation of the n-qubit gate $U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)$—in Sec. 3.1.5.

In the MQCM, the two outcomes $m = 0, 1$ for every single-qubit measurement on the graph state are equally probable because the reduced density matrix for each qubit is the completely mixed state $I/2$. In the process of getting the desired operations on the logical qubits, one also gets some additional operations because of this randomness in measurement outcomes. These additional operations are called *by-product operators*, and they belong to the Pauli group. These by-product operators depend on the measurement outcomes and the eigenvalues of the graph state $|\Phi_{\{\kappa\}}\rangle_{\mathcal{G}}$ [see Eq. (3.2)]. The measurement outcome $m \in \{0, 1\}$ for every graph

qubit and the eigenvalues $\{\kappa\}$ are binary numbers, so one can record and process them with a CC in order to take care of the by-product operators. The classical information processing of these data makes the computation deterministic and helps to set the measurement bases for the subsequent measurements. This matter is discussed comprehensively in Sec. 3.2.

3.1.3 Arbitrary single-qubit rotation

Simulation of the single-qubit rotation around the z axis, the *phase gate*

$$R_z(\varphi) := \exp\left(-i\frac{\varphi}{2}Z\right) , \tag{3.3}$$

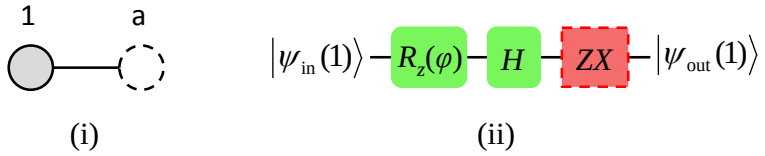
and of an arbitrary single-qubit rotation $R(\alpha, \beta, \gamma)$ of Eq. (2.21) with the MQCM [37, 38] are presented in sequence.

Simulation of the single-qubit rotation $R_z(\varphi)$: A two-qubit graph state corresponding to the graph depicted in Fig. 3.2(i) is sufficient to accomplish the job. The state of the logical qubit 1 [represented by the gray circle in Fig. 3.2(i)] is given in a general input ket $|\psi_{\rm in}(1)\rangle$ of Eq. (2.13), and we want to apply $R_z(\varphi)$ onto this single-qubit state. To generate the required graph state, the qubit a [represented by the dotted circle in Fig. 3.2(i)] is prepared in the ket $|(-1)^{\kappa_{\rm a}}\rangle_{\rm a}$ of Eq. (3.1). Then both the qubits are connected by the CZ operation, which is represented by the bond in Fig. 3.2(i) and given by Eq. (2.46). The resulting graph state with the ket

$$|\phi(1+1)\rangle = \frac{1}{\sqrt{2}}\left[|0\rangle_{\rm a} \otimes |\psi_{\rm in}(1)\rangle + (-1)^{\kappa_{\rm a}}|1\rangle_{\rm a} \otimes \left(Z|\psi_{\rm in}(1)\rangle\right)\right] \tag{3.4}$$

is ready for the simulation. Here, the label $1+1$ indicates that this graph state is made of two qubits, the logical qubit 1 and the ancilla qubit a.

In order to generate the desired effect on the input state, qubit 1 is measured



(i) (ii)

Figure 3.2: (i) Graph associated with the graph state $|\phi(1+1)\rangle$ of Eq. (3.4). The gray circle, bond, and dotted circle represent the logical qubit, which carries the input ket $|\psi_{\rm in}(1)\rangle$, the CZ operation of Eq. (2.46), and the ancilla qubit a, respectively. (ii) The quantum circuit illustrates the effect on the input ket when the qubit 1 is measured in an appropriately chosen basis.

in the basis

$$\mathcal{B}_{\frac{\pi}{2},-\varphi} = \left\{ |\uparrow,\downarrow(\tfrac{1}{2}\pi,-\varphi)\rangle_1 \right\} = \left\{ \frac{|0\rangle_1 + (-1)^{m_1} e^{-i\varphi}|1\rangle_1}{\sqrt{2}} \right\}, \tag{3.5}$$

and the value of m_1 is the result of the measurement. After the measurement, the output state (up to a global phase),

$$|\psi_{\rm out}(1)\rangle = (X)^{m_1}(Z)^{\kappa_{\rm a}} H R_z(\varphi)|\psi_{\rm in}(1)\rangle \tag{3.6}$$

is obtained from qubit a, and qubit 1 gets projected either onto the ket $|\uparrow(\frac{1}{2}\pi,-\varphi)\rangle_1$ (if $m_1=0$) or onto the ket $|\downarrow(\frac{1}{2}\pi,-\varphi)\rangle_1$ (if $m_1=1$). The net effect on the input state is the required operation $R_z(\varphi)$, followed by the Hadamard gate H of Eq. (2.23) [represented by the green boxes in Fig. 3.2(ii)], and the by-product operator $(X)^{m_1}(Z)^{\kappa_{\rm a}}$ [represented by the red box in Fig. 3.2(ii)]. Here, the axis of the measurement lies in the x-y plane of the Bloch sphere, and the input information is not only teleported from one lattice site to the other but also gets processed by the measurement.

Simulation of an arbitrary single-qubit rotation $R(\alpha,\beta,\gamma)$: As we learned in Sec. 2.2.2, every rotation in the Bloch sphere corresponds to a single-qubit unitary operation up to a global phase. Owing to the Euler decomposition of an arbitrary rotation $R(\alpha,\beta,\gamma)$ [see Eq. (2.21)], one can simulate an arbitrary single-qubit operation on a chain of five qubits graph state with four single-qubit measurements,

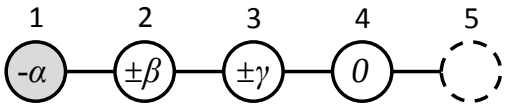


Figure 3.3: Measurement pattern on a five-qubit one-dimensional graph for realizing an arbitrary rotation $R(\alpha, \beta, \gamma)$, where the 1st qubit shown by gray circle and the 5th qubit shown by dotted circle are the input and output qubits, respectively. Since the measurement axises for qubits 1st to 4th lie in the x-y plane of the Bloch sphere, only the azimuthal angles of measurement bases are shown here.

where the measurement direction for each qubit (angles α, β, γ) lies in the x-y plane of the Bloch sphere [37, 38]. The associated five-qubit graph and the measurement pattern for $R(\alpha, \beta, \gamma)$ are depicted in Fig. 3.3.

The 1st qubit shown by the gray circle in Fig. 3.3 carries a general input ket $|\psi_{\rm in}(1)\rangle$, and the rest of the qubits are initialized in an eigenket of the Pauli operator X, say, in the ket $|+\rangle$ of Eq. (2.27). Then entanglement between each pair of nearest-neighbor qubits is established by the CZ operations [represented by the bonds in Fig. 3.3 and given by Eq. (2.46)] to realize the required graph state. A general rotation $R(\alpha, \beta, \gamma)$ is executed by measuring the qubits from 1 to 4 in the following manner[3]:

1. The 1st qubit is measured in the basis $\mathcal{B}_{\frac{\pi}{2},-\alpha}$
2. The 2nd qubit is measured in the basis $\mathcal{B}_{\frac{\pi}{2},-(-1)^{m_1}\beta}$
3. The 3rd qubit is measured in the basis $\mathcal{B}_{\frac{\pi}{2},-(-1)^{m_2}\gamma}$
4. The 4th qubit is measured in the basis $\mathcal{B}_{\frac{\pi}{2},0}$

Let us recall that the definition of single-qubit bases from Eqs. (2.14), (2.15), and (2.16). Here, m_1, m_2, m_3, and m_4 are the outcomes of the measurements on the 1st, 2nd, 3rd, and 4th qubits.

The measurement bases of the 2nd and 3rd qubits are adjusted according to the outcomes m_1 and m_2, respectively. Therefore, these measurements have to be preformed in the required order. Hence, the realization of an arbitrary rotation

[3]Here, the order of measurements follows the numbering of qubits.

$R(\alpha, \beta, \gamma)$ by such a sequence of two z rotations sandwiching an x rotation [see Eq. (2.21)] illustrates the importance of the temporal ordering of the measurements in the MQCM. After these four measurements, the 5th qubit [shown by the dotted circle in Fig. 3.3] will be (up to a global phase) in the output state

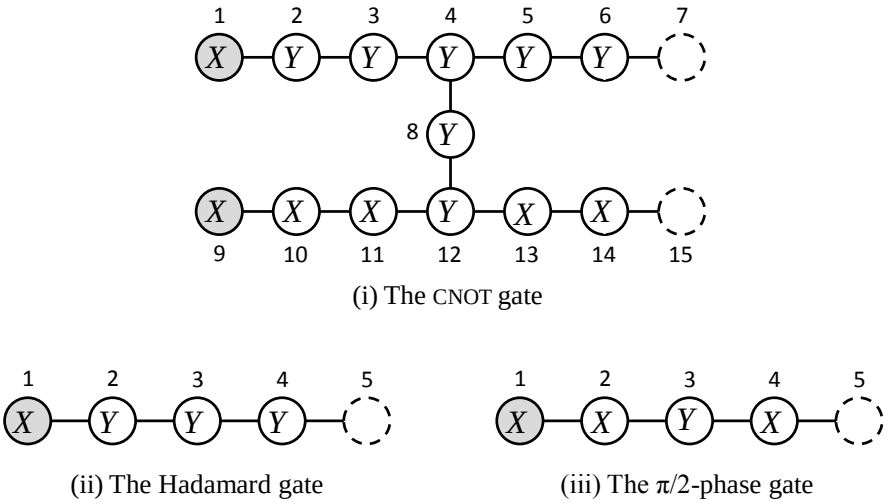
$$|\psi_{\text{out}}(5)\rangle = (X)^{m_2+m_4}(Z)^{m_1+m_3} R(\alpha, \beta, \gamma)|\psi_{\text{in}}(1)\rangle\,, \tag{3.7}$$

where $(X)^{m_2+m_4}(Z)^{m_1+m_3}$ is the by-product operator, and $R(\alpha, \beta, \gamma)$ is the desired operation on the input ket $|\psi_{\text{in}}(1)\rangle$.

3.1.4 Gates from the Clifford group

Every quantum gate from the generating set of the Clifford group—the CNOT gate of Eq. (2.39), the Hadamard gate H of Eq. (2.23), and the $\pi/2$-phase gate F of Eq. (2.55)—can be executed in a single time step in the MQCM [39]. This holds because every measurement in these cases is performed either in the X eigenbasis or in the Y eigenbasis, and is not influenced by the result of any other measurement. Therefore, all the measurements can be performed simultaneously. The CNOT gate can be achieved by thirteen single-qubit measurements on a 15-qubit graph state. Moreover, both the Hadamard and the $\pi/2$-phase gates can be implemented by four single-qubit measurements on a chain of five qubits graph state [38].

The associated graphs and measurement patterns for the CNOT, the Hadamard, and the $\pi/2$-phase gates are shown in Fig. 3.4. Single-qubit and CNOT gates together constitute a universal set of gates [see Sec. 2.4.3], and they are realizable in the MQCM. In this sense, like the UQCM in Chapter 2, the MQCM is also universal for quantum computation.



(i) The CNOT gate

(ii) The Hadamard gate

(iii) The $\pi/2$-phase gate

Figure 3.4: The measurement patterns (i), (ii), and (iii) on the 15-qubit, five-qubit, and five-qubit graphs are for simulating the CNOT, the Hadamard, and the $\pi/2$-phase gates, respectively. Qubits shown by gray and dotted circles are the input and the output qubits, respectively. The label X or Y on a qubit illustrates that the respective qubit will be measured in the X eigenbasis or in the Y eigenbasis.

3.1.5 n-qubit rotation $U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)$

The unitary operation for the n-qubit rotation around the z axis is

$$U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta) := \exp\left(-i\frac{\theta}{2}Z^{\otimes \mathsf{n}}\right), \tag{3.8}$$

where the superscripts $12\cdots\mathsf{n}$ symbolize the logical qubits on which this operation will be carried out [40]. One can accomplish this operation by performing a single measurement on a $(1+\mathsf{n})$-qubit star-graph state. The associated star graph is shown in Fig. 3.5(i), where the input quantum register of n qubits is displayed by the dotted gray circles, and the ancilla qubit[4] a by the black diamond. The input register is given in a general n-qubit input ket $|\psi_{\text{in}}(\mathsf{n})\rangle$, and the ancilla qubit is prepared in the ket $|(-1)^{\kappa_{\mathsf{a}}}\rangle_{\mathsf{a}}$ of Eq. (3.1). Then n CZ operations [represented by bonds in the figure and given by Eq. (2.46)] between qubit a and every logical qubit are performed. In principle, all the CZ operations can be performed in a "single shot," because they commute with each other. This series of steps leads to the

[4]Note that a is just the label of the ancilla qubit. Like n, it does not represent any number.

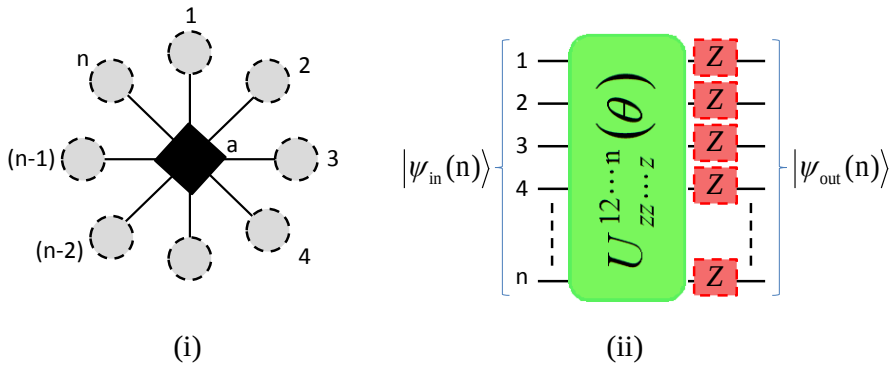


Figure 3.5: (i) is called star graph because of its appearance, and the associated graph state $|\phi(1+\mathsf{n})\rangle$ is given by Eq. (3.9). Here, the logical qubits which carry the input information, the CZ operations of Eq. (2.46), and the ancilla qubit a are represented by the dotted gray circles, bonds, and black diamond, respectively. (ii) shows the effect on the input register, when the qubit a of the graph state $|\phi(1+\mathsf{n})\rangle$ is measured in an appropriately chosen basis.

resultant star-graph state

$$|\phi(1+\mathsf{n})\rangle = \frac{1}{\sqrt{2}}\left[|0\rangle_{\mathsf{a}} \otimes |\psi_{\text{in}}(\mathsf{n})\rangle + (-1)^{\kappa_{\mathsf{a}}}|1\rangle_{\mathsf{a}} \otimes \left(Z^{\otimes \mathsf{n}}|\psi_{\text{in}}(\mathsf{n})\rangle\right)\right]. \tag{3.9}$$

The label $1+\mathsf{n}$ reveals that the final graph state is of one ancilla qubit and n logical qubits.

A measurement on ancilla qubit a in the basis

$$\mathcal{B}_{\theta,(-1)^{\kappa_{\mathsf{a}}}\frac{\pi}{2}} = \left\{|\uparrow,\downarrow(\theta,(-1)^{\kappa_{\mathsf{a}}}\tfrac{1}{2}\pi)\rangle_{\mathsf{a}}\right\}, \tag{3.10}$$

transforms the input ket into the output ket

$$|\psi_{\text{out}}(\mathsf{n})\rangle = (Z^{\otimes \mathsf{n}})^{m_{\mathsf{a}}} U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)|\psi_{\text{in}}(\mathsf{n})\rangle. \tag{3.11}$$

In this case, the direction of measurement lies in the z-y plane of the Bloch sphere, and $m_{\mathsf{a}} \in \{0,1\}$ is the measurement outcome. $(Z^{\otimes \mathsf{n}})^{m_{\mathsf{a}}}$ is the by-product operator, which is represented by the red boxes on all the logical qubits in Fig. 3.5(ii). After the measurement, all bonds [illustrated in Fig. 3.5(i)] gets broken, and qubit a gets projected either onto the ket $|\uparrow(\theta,(-1)^{\kappa_{\mathsf{a}}}\frac{1}{2}\pi)\rangle_{\mathsf{a}}$ (if $m_{\mathsf{a}}=0$) or onto the ket

$|\downarrow(\theta,(-1)^{\kappa_{\mathsf{a}}}\frac{1}{2}\pi)\rangle_{\mathsf{a}}$ (if $m_{\mathsf{a}}=1$). This kind of multiqubit rotations will be used for the HQCM in Chapter 4.

In contrast to the rotation $R_z(\varphi)$ of Sec. 3.1.3 where the qubits used for input and output are different, in the case of $U_{zz\cdots z}^{12\cdots\mathsf{n}}(\theta)$ the input and output states reside in the same n logical qubits. In other words, here the information gets processed but does not get transferred from one place to another on the graph. As a side remark, the resultant by-product operator $(X)^{m_1}(Z)^{\kappa_{\mathsf{a}}}$ in the case of $R_z(\varphi)$ [see Eq. (3.6)] and the measurement basis $\mathcal{B}_{\theta,(-1)^{\kappa_{\mathsf{a}}}\frac{\pi}{2}}$ [see Eq. (3.10)] in the case of $U_{zz\cdots z}^{12\cdots\mathsf{n}}(\theta)$ depend on the eigenvalue κ_{a}.

Generalization of this procedure is given as follows. If, instead of the CZ operations, one performs the $(1+\mathsf{n})$-qubit unitary operation

$$\Lambda^{\mathsf{a}}\mathbf{A} := |0\rangle_{\mathsf{a}}\langle 0|\otimes\mathbf{I}+|1\rangle_{\mathsf{a}}\langle 1|\otimes\mathbf{A} \tag{3.12}$$

between the ancilla qubit and the input register of n qubits to prepare a graph state; where $\mathbf{I}=I^{\otimes\mathsf{n}}$, and the n-qubit operation[5] $\mathbf{A}$ is such that $\mathbf{A}^2=\mathbf{I}$. Then, the resultant graph state can be used to implement the n-qubit unitary operation

$$U_{\mathbf{A}}(\theta) := \exp\left(-i\frac{\theta}{2}\mathbf{A}\right) \tag{3.13}$$

with the procedure given above. In this case, the output ket will be

$$|\psi_{\text{out}}(\mathsf{n})\rangle = (\mathbf{A})^{m_{\mathsf{a}}}U_{\mathbf{A}}(\theta)|\psi_{\text{in}}(\mathsf{n})\rangle\,, \tag{3.14}$$

where $(\mathbf{A})^{m_{\mathsf{a}}}$ is the by-product operator. An example of this generalization is given in Appendix C, where $\mathbf{A}=H^{\otimes\mathsf{n}}$.

To simulate a complex unitary gate in the MQCM, it is customary to first decompose it efficiently into a sequence of gates from the universal set [see Sec. 2.4.3].

[5]Note that $\mathbf{A}$ is both unitary and Hamiltonian operator.

Then the temporal order of gates is transformed into the spatial pattern of measurement bases for the graph qubits. Afterwards, the measurements are performed in the required order.

Up to now, the simulations of individual gates are studied. Until now, only the *production* of the by-product operators appears, and there was no need to worry about the *propagation* of the by-product operators. But the next section is focused on the simulation of a sequence of gates, where the study of classical information processing and the temporal order of the measurements become necessary. Classical information processing is needed to record the production and monitor the propagation of the by-product operators.

3.2 Classical information processing in the MQCM

This section serves as a summary of the results discussed in Ref. [39]. When a sequence of gates is simulated in the MQCM, the by-product operator which originates from the implementation of gates propagates through the sequence. Either the propagation of the by-product operator transforms the subsequent gates in the sequence or the by-product operator in itself gets transformed. The first part of this section is about *propagation relations* for some elementary gates. The second part is reserved for defining an *information flow vector* and the *propagation matrices* for some elementary gates based on their propagation relations. This material will be used in Chapter 4.

3.2.1 Propagation relation

The structure of the by-product operator on the logical qubit $\mathsf{j} \in \{1, \cdots, \mathsf{n}\}$ is $(X^{(\mathsf{j})})^{x_\mathsf{j}}(Z^{(\mathsf{j})})^{z_\mathsf{j}}$, where x_j and z_j are non-negative integers. Both x_j and z_j depend on

the outcomes of measured qubits and the eigenvalues $\{\kappa\}$ of graph state [39]. Their dependence on $\{\kappa\}$ is in our control. For example, the $\{\kappa\}$-dependency disappears from the calculation if one prepares a graph state with $\kappa = 0$ for all the graph qubits. But we cannot control the dependence of the by-product operators on the measurement outcomes, which are intrinsically random.

In Ref. [39], the authors took $x_{\rm j}, z_{\rm j} \in \{0, 1\}$, but here both $x_{\rm j}$ and $z_{\rm j}$ are taken as non-negative integers. This is permissible because only the modulo-2 values of $x_{\rm j}$ and $z_{\rm j}$ matter in $(X^{({\rm j})})^{x_{\rm j}}$ and $(Z^{({\rm j})})^{z_{\rm j}}$. Throughout this thesis, the signs $+$ and $\oplus$ are reserved for the ordinary and modulo-2 addition, respectively.

In principle, the by-product operators can be corrected—step by step—after completing each gate of a sequence under simulation. But it is more convenient to choose not to correct them and let them pass through the gates, then just keep track of the measurement outcomes in a systematic way using simple classical information processing. At the end of the computation, either the measurement bases for the final readout are set according to the history of outcomes or, alternatively, the final measurements are performed in the computational basis and interpretation of the result is done with the help of the recorded outcomes.

Propagation of the by-product operator through a gate is given by the propagation relation. The propagation relation for an arbitrary single-qubit rotation $R(\alpha, \beta, \gamma)$ of Eq. (2.21) is

$$R(\alpha, \beta, \gamma)(X)^x(Z)^z = (X)^x(Z)^z \tilde{R}\big((-1)^x\alpha, (-1)^z\beta, (-1)^x\gamma\big)\,. \tag{3.15}$$

The rotation $R(\alpha, \beta, \gamma)$ gets transformed into $\tilde{R}\big((-1)^x\alpha, (-1)^z\beta, (-1)^x\gamma\big)$, while the by-product operator stays as it is. Equation (3.15) can be taken as an illustration of the importance of "the temporal order of the measurements." This is because, when $R(\alpha, \beta, \gamma)$ is a part of a circuit, the superscripts x and z are functions of the earlier measurement outcomes, and to determine the right sign for the measurement angles

α, β, and γ [see, Sec. 3.1.3], we have to wait until the necessary measurements are completed [37].

Equation (3.15) also justifies the following points. The measurement directions for those qubits lie in the x-y plane of the Bloch sphere, $\vec{r}_{xy}(\varphi) = \big(\cos\varphi, \sin\varphi, 0\big)$ with $\varphi \notin \big\{0, \pm\frac{1}{2}\pi\big\}$, their measurement bases depend on the results of previous measurements [see Sec. 3.1.3] [37, 38]. When $\varphi \in \big\{0, \pm\frac{1}{2}\pi\big\}$, then the directions for $+\varphi$ and $-\varphi$ coincide and do not get influenced by the outcome of any other measurement. Measurements of this kind are either in the X ($\varphi = 0$) or the Y ($\varphi = \pm\frac{1}{2}\pi$) eigenbasis. The gates from the generating set of the Clifford group are realized by such measurements [see Sec. 3.1.4], and their propagation relations are given in the following.

The propagation relation for the $\textsc{cnot}(\mathsf{a}, \mathsf{b})$ gate of Eq. (2.39) is

$$\textsc{cnot}(\mathsf{a}, \mathsf{b})\, \boldsymbol{U}_B^{\textsc{cnot}} = \tilde{\boldsymbol{U}}_B^{\textsc{cnot}} \textsc{cnot}(\mathsf{a}, \mathsf{b})\,, \tag{3.16}$$

where

$$\boldsymbol{U}_B^{\textsc{cnot}} := (X^{(\mathsf{a})})^{x_\mathsf{a}} (Z^{(\mathsf{a})})^{z_\mathsf{a}} (X^{(\mathsf{b})})^{x_\mathsf{b}} (Z^{(\mathsf{b})})^{z_\mathsf{b}}\,, \tag{3.17}$$

and

$$\tilde{\boldsymbol{U}}_B^{\textsc{cnot}} = (X^{(\mathsf{a})})^{x_\mathsf{a}} (Z^{(\mathsf{a})})^{z_\mathsf{a}+z_\mathsf{b}} (X^{(\mathsf{b})})^{x_\mathsf{a}+x_\mathsf{b}} (Z^{(\mathsf{b})})^{z_\mathsf{b}}\,. \tag{3.18}$$

In case of the $\textsc{cnot}(\mathsf{a}, \mathsf{b})$ gate, Eq. (3.16), the gate stays as it is, but the by-product operator $\boldsymbol{U}_B^{\textsc{cnot}}$ gets transformed into $\tilde{\boldsymbol{U}}_B^{\textsc{cnot}}$. This is also the case for the other two gates from the generating set of the Clifford group. The propagation relation for the Hadamard gate H of Eq. (2.23) is

$$H(X)^x(Z)^z = (X)^z(Z)^x H\,, \tag{3.19}$$

and for the $\pi/2$-phase gate F of Eq. (2.55) it is (up to a phase factor $\pm i$)[6]

$$F(X)^x(Z)^z = (X)^x(Z)^{z+x}F\,. \tag{3.20}$$

The propagation relations (3.16), (3.19), and (3.20) can also be understood from the definition of the Clifford group, which maps the Pauli group into itself under conjugation.

3.2.2 Information flow vector and propagation matrix

Information flow vector: At every stage of the computation, the by-product operator $\boldsymbol{U}_B$ upon the logical qubits $1, \cdots, \mathsf{n}$ is of the form $\prod_{\mathsf{j}=1}^{\mathsf{n}}(X^{(\mathsf{j})})^{x_\mathsf{j}}(Z^{(\mathsf{j})})^{z_\mathsf{j}}$. After the implementation of a gate, only the values $\{x_\mathsf{j}\}$ and $\{z_\mathsf{j}\}$ get changed, and the new values determine the modifications in the measurement bases for the subsequent gates. These values are processed by a CC. There is a one-to-one correspondence between the by-product operator $\boldsymbol{U}_B$ (ignoring the global phase ± 1) and a $2\mathsf{n}$-component information flow vector $\mathcal{I}$, which is given as follows:

$$\boldsymbol{U}_B := \prod_{\mathsf{j}=1}^{\mathsf{n}}(X^{(\mathsf{j})})^{x_\mathsf{j}}(Z^{(\mathsf{j})})^{z_\mathsf{j}} \Longleftrightarrow \mathcal{I} := \begin{pmatrix} \mathcal{I}_x \\ \mathcal{I}_z \end{pmatrix}, \tag{3.21}$$

where

$$\mathcal{I}_x := \begin{pmatrix} x_1 \\ \vdots \\ x_\mathsf{n} \end{pmatrix}, \quad \mathcal{I}_z := \begin{pmatrix} z_1 \\ \vdots \\ z_\mathsf{n} \end{pmatrix}. \tag{3.22}$$

Here, the multiplication of by-product operators (up to a phase factor ± 1) corresponds to the component-wise addition of information flow vectors. The information flow vector $\mathcal{I}$ represents the flow of classical information $\{x_\mathsf{j}\}$ and $\{z_\mathsf{j}\}$. Also, it keeps track of the sign(s) of the measurement angle(s) for a gate. In accordance

[6] $F(X)^x(Z)^z = (Y)^x(Z)^z F$.

with Eq. (3.15), the signs of the measurement angles for the operation $R^{(\mathsf{j})}(\alpha, \beta, \gamma)$ on the qubit j are determined by the current value of x_{j} and z_{j} in $\mathcal{I}$. The propagation relations (3.16), (3.19), and (3.20) suggest that none of the gates from the generating set of the Clifford group gets altered under the propagation of the by-product operator. Therefore, the measurement angles for these gates are independent of the values stored in $\mathcal{I}$.

Propagation matrix: For every gate g a $2\mathsf{n} \times 2\mathsf{n}$ propagation matrix $\mathbf{C}(g)$ can be defined [see Eq. (3.27)], which represents the transformation in the information flow vector when the corresponding by-product operator passes through the gate g. The propagation matrices given below are derived from the propagation relations (3.15), (3.16), (3.19), and (3.20) with the help of the one-to-one correspondence given by Eq. (3.21), and the entries in the information flow vectors and the propagation matrices are given only for relevant qubits.

The by-product operator passes through a single-qubit rotation $R(\alpha, \beta, \gamma)$ without getting transformed. Hence, the information flow vector stays as it is:

$$\begin{pmatrix} x \\ z \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{\mathbf{C}(R)} \begin{pmatrix} x \\ z \end{pmatrix}. \tag{3.23}$$

The information flow vector gets transformed when the associated by-product operator passes through the $\textsc{cnot}(\mathsf{a}, \mathsf{b})$ gate of Eq. (2.39) in the following way:

$$\begin{pmatrix} x_{\mathsf{a}} \\ x_{\mathsf{a}} + x_{\mathsf{b}} \\ z_{\mathsf{a}} + z_{\mathsf{b}} \\ z_{\mathsf{b}} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{C}(\textsc{cnot})} \begin{pmatrix} x_{\mathsf{a}} \\ x_{\mathsf{b}} \\ z_{\mathsf{a}} \\ z_{\mathsf{b}} \end{pmatrix}. \tag{3.24}$$

Under the one-to-one correspondence given by Eq. (3.21), the propagation relation

(3.19) for the Hadamard gate H becomes

$$\begin{pmatrix} z \\ x \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}}_{\mathbf{C}(H)} \begin{pmatrix} x \\ z \end{pmatrix}, \tag{3.25}$$

and the propagation relation (3.20) for the $\pi/2$-phase gate F becomes

$$\begin{pmatrix} x \\ z+x \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}}_{\mathbf{C}(F)} \begin{pmatrix} x \\ z \end{pmatrix}. \tag{3.26}$$

The propagation matrices for the R, CNOT, H, and F gate for the case of n logical qubits are given in the following. The propagation matrix $\mathbf{C}$ is a $2\mathsf{n} \times 2\mathsf{n}$ matrix of the form

$$\mathbf{C} := \left(\begin{array}{c|c} \mathbf{C}_{xx} & \mathbf{C}_{zx} \\ \hline \mathbf{C}_{xz} & \mathbf{C}_{zz} \end{array}\right), \tag{3.27}$$

where $\mathbf{C}_{xx}, \mathbf{C}_{zx}, \mathbf{C}_{xz}$ and $\mathbf{C}_{zz}$ are $\mathsf{n} \times \mathsf{n}$ matrices with binary-valued entries [39]. One can generate the propagation matrices for an arbitrary single-qubit rotation[7] $R^{(\mathsf{j})}$ on the logical qubit j with

$$\begin{aligned} \left[\mathbf{C}_{xx}(R^{(\mathsf{j})})\right]_{\mathsf{kl}} &:= \left[\mathbf{C}_{zz}(R^{(\mathsf{j})})\right]_{\mathsf{kl}} := \delta_{\mathsf{kl}}\,, \\ \left[\mathbf{C}_{zx}(R^{(\mathsf{j})})\right]_{\mathsf{kl}} &:= \left[\mathbf{C}_{xz}(R^{(\mathsf{j})})\right]_{\mathsf{kl}} := 0\,. \end{aligned} \tag{3.28}$$

The notation $\left[\mathbf{C}_{xx}(R^{(\mathsf{j})})\right]_{\mathsf{kl}}$ stands for the entry in kth row and lth column of the matrix $\mathbf{C}_{xx}$ corresponding to the $R^{(\mathsf{j})}$ gate, and the same nomenclature applies elsewhere.

The propagation matrix for the CNOT(a, b) gate (both the control qubit a and

[7] $\mathbf{C}(R)$ is the $2\mathsf{n} \times 2\mathsf{n}$ identity matrix.

the target qubit b belong to the set of n logical qubits; $\mathsf{a} \neq \mathsf{b}$) is given by

$$\begin{aligned}
&[\mathbf{C}_{xx}(\mathrm{CNOT}(\mathsf{a},\mathsf{b}))]_{\mathsf{kl}} := \delta_{\mathsf{kl}} + \delta_{\mathsf{kb}}\delta_{\mathsf{la}}\,, \\
&[\mathbf{C}_{zz}(\mathrm{CNOT}(\mathsf{a},\mathsf{b}))]_{\mathsf{kl}} := \delta_{\mathsf{kl}} + \delta_{\mathsf{ka}}\delta_{\mathsf{lb}}\,, \\
&[\mathbf{C}_{zx}(\mathrm{CNOT}(\mathsf{a},\mathsf{b}))]_{\mathsf{kl}} := [\mathbf{C}_{xz}(\mathrm{CNOT}(\mathsf{a},\mathsf{b}))]_{\mathsf{kl}} := 0\,;
\end{aligned} \tag{3.29}$$

the corresponding matrix for the Hadamard gate $H^{(\mathsf{j})}$ on the logical qubit j is given by

$$\begin{aligned}
&\left[\mathbf{C}_{xx}(H^{(\mathsf{j})})\right]_{\mathsf{kl}} := \left[\mathbf{C}_{zz}(H^{(\mathsf{j})})\right]_{\mathsf{kl}} := \delta_{\mathsf{kl}} \oplus \delta_{\mathsf{kj}}\delta_{\mathsf{lj}}\,, \\
&\left[\mathbf{C}_{zx}(H^{(\mathsf{j})})\right]_{\mathsf{kl}} := \left[\mathbf{C}_{xz}(H^{(\mathsf{j})})\right]_{\mathsf{kl}} := \delta_{\mathsf{kj}}\delta_{\mathsf{lj}}\,;
\end{aligned} \tag{3.30}$$

and the matrix for the $\pi/2$-phase gate $F^{(\mathsf{j})}$ on the logical qubit j is given by

$$\begin{aligned}
&\left[\mathbf{C}_{xx}(F^{(\mathsf{j})})\right]_{\mathsf{kl}} := \left[\mathbf{C}_{zz}(F^{(\mathsf{j})})\right]_{\mathsf{kl}} := \delta_{\mathsf{kl}}\,, \\
&\left[\mathbf{C}_{zx}(F^{(\mathsf{j})})\right]_{\mathsf{kl}} := 0\,, \\
&\left[\mathbf{C}_{xz}(F^{(\mathsf{j})})\right]_{\mathsf{kl}} := \delta_{\mathsf{kj}}\delta_{\mathsf{lj}}\,.
\end{aligned} \tag{3.31}$$

It is advantageous to deal with the information flow vector $\mathcal{I}$ together with the propagation matrices [Eqs. (3.28)–(3.31)] by a CC, rather than dealing directly with the corresponding by-product operator $\boldsymbol{U}_B$ together with the propagation relations [Eqs. (3.15), (3.16), (3.19), and (3.20)].

As a side remark, the temporal order of the measurements does not typically follow the temporal order of gates in a circuit which we want to simulate with the MQCM. Indeed, there exists an *efficient measurement scheme* where measurements are performed round by round, and in each round all the measurements are executed at the same time [39]. The information flow vector is updated after every round. After the final round, the result of the computation is interpreted from the x part

of the information flow vector $\mathcal{I}_x$ [according to Eq. (4.5)]. An extended discussion of this efficient measurement scheme is provided in the next section.

3.3 Efficient measurement scheme of the MQCM

This section holds a discussion an efficient measurement scheme of the MQCM where the temporal order of the measurements plays an important role [39]. On one hand in the UQCM, any two gates of a sequence that do not commute cannot be parallelized [see Chapter 2]. On the other hand in the MQCM, all the gates from the Clifford group can be executed in a single time step, irrespective of their positions in the circuit [see Sec. 3.1.4]. In other words, the temporal order of the measurements in the MQCM is not preimposed by the temporal order of the gates. So, the most efficient scheme for the measurements does not necessarily follow the temporal order of the gates in a circuit under simulation. Initially, the spatial pattern of the measurement bases is assigned to the graph qubits according to the sequence of gates. Then the measurements are performed round by round according to the scheme which is given as follows.

First, the graph $\mathcal{G}$ is divided into disjoint subsets of qubits $\mathcal{Q}_t$, where index t stands for the round of measurements and $0 \leq t \leq t_{\max}$. Mathematically, $\mathcal{G} := \bigcup_{t=0}^{t_{\max}} \mathcal{Q}_t$ and $\mathcal{Q}_s \cap \mathcal{Q}_t := \emptyset$ for all $s \neq t$. The subset $\mathcal{Q}_t$ is a collection of all those qubits which will be measured simultaneously in tth round. All the measurements in the X, Y and Z eigenbasis are put together in the very first round (zeroth round), and there is no need to adjust the measurement bases according to the previous measurement results for the qubits of $\mathcal{Q}_0$. In the first measurement round, the *redundant graph qubits* are removed by the Z-measurements [see Sec. 3.1.2], the *readout qubits* are measured in the Z eigenbasis, and the part of the circuit belonging to the Clifford group is simulated by measurements in the X, Y eigenbasis [39]. In the MQCM, the readout qubits, which play the role of *output register*, are not

the last ones to be measured; they are among the first.

The XY-measurements [see Sec. 3.1.2] are used for all the subsequent measurement rounds, where the measurement observables are of the form $\cos(\varphi)X \pm \sin(\varphi)Y$ with $\varphi \notin \{0, \pm\frac{1}{2}\pi\}$. The changes in measurement bases for these qubits are decided by the measurement outcomes from the previous rounds. All those qubits whose measurement bases depend on the outcomes from the first measurement round belong to the subset $\mathcal{Q}_1$. Similarly, the measurement outcomes from the subset $\mathcal{Q}_1$ together with $\mathcal{Q}_0$ decide the alterations in measurement bases for the qubits in $\mathcal{Q}_2$, and so on. These subsets are measured one by one up to the final measurement round $t_{\max}$. One can think of the total number of measurement rounds $(t_{\max}+1)$ as the logical depth (temporal complexity) in the MQCM.

Parallel to the measurement rounds, the classical data-processing parts are taken care of by a CC. After preparing the graph state and just before starting the measurements, the information vector is initialized to $\mathcal{I}_{\text{init}}^{\text{MQCM}}$. $\mathcal{I}_{\text{init}}^{\text{MQCM}}$ depends on the eigenvalues $\{\kappa\}$ of the graph state and some particular gates (like CNOT, F) which appear in a quantum circuit under simulation [39]. After executing the first measurement round on the set $\mathcal{Q}_0$, $\mathcal{I}_{\text{init}}^{\text{MQCM}}$ gets updated to $\mathcal{I}(0)$ through the measurement results. $\mathcal{I}(0)$ then determines the adjustments in measurement bases for the qubits of $\mathcal{Q}_1$. Similarly, the measurement outcomes from round t update the information flow vector from $\mathcal{I}(t-1)$ to $\mathcal{I}(t)$. The corresponding by-product operator is given by

$$\boldsymbol{U}_B(t) = \prod_{\mathsf{j}=1}^{\mathsf{n}} (X^{(\mathsf{j})})^{x_{\mathsf{j}}(t)} (Z^{(\mathsf{j})})^{z_{\mathsf{j}}(t)} \,. \tag{3.32}$$

Then $\mathcal{I}(t) = \mathcal{I}\big(x_{\mathsf{j}}(t), z_{\mathsf{j}}(t)\big)$ sets the measurement bases for the $(t+1)$th round. After the final measurement round $t_{\max}$, the x part of the information flow vector $\mathcal{I}_x(t_{\max})$ enables us to interpret the result of the computation [see Eq. (4.5)].

In this measurement scheme, the following technical points are worth emphasizing; they are discussed in Ref. [39]. (1) In order to construct the subsets of graph

qubits $\{\mathcal{Q}_t\}$, a CC needs the forward cones for all the graph qubits. The forward cones decide a strict partial ordering among the qubits, and the sets $\{\mathcal{Q}_t\}$ are constructed accordingly. (2) To account for the influence of the measurement outcomes and the set of eigenvalues $\{\kappa\}$ on $\mathcal{I}(t)$, a CC needs the by-product images for all the graph qubits. (3) $\{\kappa\}$, the by-product images, and $\mathcal{I}(t)$ are required to set the measurement bases for the as yet unmeasured qubits. A CC uses the symplectic scalar product for doing this.

Chapter 4

The hybrid quantum computation model

Both the UQCM [see Chapter 2] and the MQCM [see Chapter 3] are universal for quantum computation, nevertheless, it is beneficial to employ one rather than other in certain experimental scenarios. In the case of UQCM, no preparation of a graph state and classical information processing is needed. However, to perform the computation, measurements in the MQCM are easier to execute than quantum gates in the UQCM. The practical difficulties come from the implementation of multiqubit gates in the UQCM and from the preparation of universal graph state in the MQCM. The larger the graph state, the more difficult it is to control and protect it from the noise.

These observation led us to design a hybrid model of the UQCM and the MQCM, the HQCM [41], with the aim of exploiting the strengths of both models. Since both the UQCM and the MQCM are universal, the HQCM is universal too. There are two main tasks to achieve the HQCM.

The first task is to establish a *set of elementary gates* [see Sec. 4.1.1] for this hybrid model using an optimal amount of resources[1] to get an efficient experimental

[1]Here, the resources are qubits, entanglement, elementary operations and measurements.

implementation. Since every member of this elementary gate set can be executed in a single shot, each one of them is considered as a single unit in the HQCM. The HQCM employs the MQCM for executing certain multiqubit gates and the UQCM for others.

The second task in this investigation is to work out the *classical information-processing parts* of the HQCM [see Sec. 4.2]. Indeed, where measurements are involved in quantum information processing (e.g., the quantum teleportation [14], the MQCM [39]), the classical information processing is required side by side. In the hybrid model, part of a quantum circuit is simulated by unitary evolution and the rest by measurements on small (non-universal) graph states.

Simulation of a complicated unitary gate with the HQCM [for examples, see Sec. 4.3] can be summarized in the following four steps:

1. Like the UQCM, a given unitary gate is efficiently decomposed in terms of a sequence of—single-qubit gates, the CZ gates, and the multiqubit rotations around the z axis $U_{zz\cdots z}^{12\cdots n}(\theta)$—the elementary gates of the HQCM [see Sec. 4.1.1].

2. These elementary gates are executed one by one in the sequence. In the HQCM, single-qubit and the CZ gates are realized by their respective unitary evolution, and every multiqubit rotation is implemented by a single measurement on a required star-graph state according to the procedure given in Sec. 3.1.5.

3. Like the MQCM, the classical information is processed in parallel. In here, the classical information processing only needs the information flow vector and the propagation matrices [see Sec. 3.2.2] for the elementary gates.

4. After the last gate of the sequence, the x part of the information flow vector enables us to interpret the final result of the computation [see Eq. (4.5)].

These steps are comprehensively discussed in this chapter. Sections 4.1 and 4.2 present the methodology for computation and classical information processing in the HQCM, respectively. The results from these sections are used in Sec. 4.3, which explains the simulation of multi-control gates with the HQCM. These multi-control gates will be utilized for implementing GA in Sec. 6.2.1.

4.1 Methodology for computation in the HQCM

In this section, the methods for computation are formulated. Before going into the details, let us first focus on what benefits one can get from the UQCM and the MQCM in different situations. Here, the preparation of graph states [see Sec. 3.1.1], the set of elementary gates for the HQCM, and the simulation of a quantum circuit with the HQCM are considered one by one.

The very first experimental step in the MQCM is the preparation of universal graph state, whereas in the UQCM no such preparation is needed. While preparing a graph state, in principle, the initialization of every graph qubit in the X eigenbasis can be completed in a single shot. To this end, we have to address every graph qubit simultaneously. Consequently, this requires a lot of experimental resources, and that many interactions are difficult to control. Likewise, the subsequent two-qubit entangling operations [$\mathrm{CZ}(\mathsf{a},\mathsf{b})$ defined by Eq. (2.46)] to create the resource graph state can be performed in a single step, because they commute with each other. Thus, it is more difficult to prepare and control a larger graph state and to protect it against decoherence. So, for the HQCM, we choose not to prepare the whole two-dimensional universal graph [see Fig. 3.1] state at once but, instead, prepare small (nonuniversal) graph states step by step as we need them when the computation progresses. Only the star-graph states, such as $|\phi(1+\mathsf{n})\rangle$ given in Eq. (3.9), are required for the HQCM.

4.1.1 Set of elementary gates for the HQCM

Single-qubit rotation $R_{\vec{r}}(\upsilon)$ of Eq. (2.20), the CZ gate of Eq. (2.46) and the multi-qubit rotation around the z axis $U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)$ for an arbitrary value of θ of Eq. (3.8) are chosen as the elementary gates for the HQCM. In analogy to the procedure for the UQCM, first a complex unitary gate under simulation is *efficiently* decompose into a sequence of elementary gates in such a way that the number of elementary gates grows *polynomially* with the number of logical qubits. Then every elementary gate is implemented one after another. Every single-qubit and the CZ gates are carried out by the unitary evolution under the formalism of UQCM. The rotations $U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)$ are implemented by the method described in Sec. 3.1.5 under the formalism of MQCM. The motivation behind these choices is explained in the following.

Simulation of an arbitrary single-qubit rotation in the MQCM [see Sec. 3.1.3] costs at least a chain of five qubits and four measurements [37, 38]. But it can be realized quite simply by the unitary evolution of the respective single qubit. Furthermore, the Euler decomposition for an arbitrary single-qubit rotation $R(\alpha, \beta, \gamma)$ [see Eq. (2.21)] is not needed.

The CZ operations themselves are part of the experimental setup for constructing the graph states. Therefore, we have to execute unitary evolutions to construct them. That is why the CZ gate is considered as an elementary gate for the HQCM. Furthermore, it is more economical to implement $\textsc{cnot}(\mathsf{a}, \mathsf{b})$ by the unitary evolution $H^{(\mathsf{b})}\, \textsc{cz}(\mathsf{a}, \mathsf{b})\, H^{(\mathsf{b})}$ [see Eq. (2.47)] instead of first preparing a $\mathsf{15}$-qubit graph state and then implement it with the MQCM [see Sec. 3.1.4].

Although the HQCM already has the universal set of gates (single-qubit and CZ gates) [see Sec. 2.4.3], the rotation $U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)$ is taken as an elementary gate because of the following two reasons. The first reason is the experimental optimization in terms of resources. The resource $(\mathsf{1}+\mathsf{n})$-qubit graph state $|\phi(\mathsf{1}+\mathsf{n})\rangle$ [given by Eq. (3.9)] needed for the implementation of $U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)$ is relatively easy to

create experimentally. It has only one ancilla qubit, and the entanglement can be established in one go. Furthermore, a single measurement on the ancilla qubit is enough to realize $U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)$ all together on n logical qubits. While it is also possible to decompose the rotation $U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)$ in terms of the gates from the universal gate set and implement it under the formalism of UQCM, its implementation there will not be so optimal, and it cannot be regarded as a single unit.

The second reason for including $U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)$ as an elementary gate in the HQCM is to investigate the classical information processing. Generally, one uses either unitary evolution (UQCM) or measurements on the graph state (MQCM) to simulate a quantum circuit. The classical processing does not come into the picture of UQCM where measurements are used only for the readout of the final result of computation. In all those schemes where measurements are needed for the computation (e.g., the quantum teleportation [14], the MQCM [39]), the classical information processing in parallel is essential.

In the HQCM also, classical information processing is needed, because the rotations $U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)$ are executed by the measurements. But here the classical information processing is simpler than the MQCM. It requires only the information flow vector and the propagation matrices. A comprehensive discussion of this is given in the following section.

4.2 Classical information processing in the HQCM

We now focus on the classical information-processing parts of the HQCM, where only the information flow vector and the propagation matrices for the elementary gates are required. First the information flow vector in the context of HQCM is redefined, and, second, the propagation relations followed by the propagation matrices for the elementary gates are given.

4.2.1 Information flow vector in the HQCM

At every computation step τ in the hybrid model, the by-product operator still has the same form as given in Eqs. (3.21) and (3.22),

$$\boldsymbol{U}_B(\tau) = \prod_{\mathsf{j}=1}^{\mathsf{n}} (X^{(\mathsf{j})})^{x_{\mathsf{j}}(\tau)} (Z^{(\mathsf{j})})^{z_{\mathsf{j}}(\tau)} \,. \tag{4.1}$$

Hence, the structure of the related information flow vector $\mathcal{I}(\tau) = \mathcal{I}\big(x_{\mathsf{j}}(\tau), z_{\mathsf{j}}(\tau)\big)$ is unchanged [see Eqs. (3.21) and (3.22)]. However, in the HQCM, there exist a few differences in comparison to the efficient measurement scheme of MQCM given in Sec. 3.3.

In that scheme of MQCM, the index t of $\mathcal{I}(t)$ stands for the measurement round. However, in the HQCM, where implementation of every elementary gate takes only one computational step, the index τ of $\mathcal{I}(\tau)$ is the label of the elementary gate. In the MQCM, $\mathcal{I}(t)$ gets updated after each round, but in the HQCM it is updated after each gate.

Furthermore, in the MQCM, the initial value of the information flow vector $\mathcal{I}_{\rm init}^{\rm MQCM}$ is determined by the set of eigenvalues $\{\kappa\}$ plus some particular gates. Whereas in the HQCM, just before starting the computation all the entries of $\mathcal{I}(\mathbf{0}) := \mathcal{I}_{\rm init}^{\rm HQCM}$ are zeros, that is, both $x_{\mathsf{j}}(\mathbf{0}) = 0$ and $z_{\mathsf{j}}(\mathbf{0}) = 0$ for all $\mathsf{j} = 1, 2, \cdots, \mathsf{n}$. This means that the by-product operator at $\tau = \mathbf{0}$ is the identity operator I on every logical qubit. In fact, the first relevant by-product operator appears in the computation when the first multiqubit rotation is implemented, and then the information flow vector gets some nonzero entries. In the MQCM, the information flow vector gets updated from $\mathcal{I}(t-1)$ to $\mathcal{I}(t)$ after the tth measurement round. In the HQCM, the information flow vector gets updated from $\mathcal{I}(\tau-1)$ to $\mathcal{I}(\tau)$ after the implementation of τth gate. $\mathcal{I}(\tau)$ then influences the $(\tau+1)$th gate of a quantum circuit under simulation. Similar to the case of UQCM, the total number of computation steps, or logical depth, is denoted by $\tau_{\rm max}$. This is the total number

of elementary gates used for the computation. Furthermore, $\tau_{\max}$ is also the total number of steps taken by a CC for the classical information processing in parallel.

Interpretation of the final computational result from $\mathcal{I}_x(\tau_{\max})$: In the UQCM, every gate of a circuit is executed by its respective unitary evolution, and the final readout measurements are performed in the computational basis. In this case, the output state $|\text{out}\rangle$ gets projected onto the state $|M_{\text{UQCM}}\rangle := \otimes_{\mathsf{j}=1}^{\mathsf{n}} |\acute{s}_{\mathsf{j}}\rangle$ after the final readout measurements,

$$|M_{\text{UQCM}}\rangle = \prod_{\mathsf{j}=1}^{\mathsf{n}} \frac{I^{(\mathsf{j})} + (-1)^{\acute{s}_{\mathsf{j}}} Z^{(\mathsf{j})}}{2} |\text{out}\rangle \,, \tag{4.2}$$

where $\acute{s}_{\mathsf{j}} \in \{0, 1\}$ are the readout measurement outcomes for the logical qubits $\mathsf{j} = 1, 2, \cdots, \mathsf{n}$.

In the hybrid model, however, the final state of the output register will be $\boldsymbol{U}_B(\tau_{\max})|\text{out}\rangle$ after performing the last gate of the same circuit. Without loss of generality, as above, we consider the computational basis for the final readout, where $s_{\mathsf{j}} \in \{0, 1\}$ are the readout measurement outcomes for the logical qubits $\mathsf{j} = 1, 2, \cdots, \mathsf{n}$. It means that the output state $\boldsymbol{U}_B(\tau_{\max})|\text{out}\rangle$ gets projected onto the state with the ket $|M_{\text{HQCM}}\rangle := \otimes_{\mathsf{j}=1}^{\mathsf{n}} |s_{\mathsf{j}}\rangle$ after the readout measurements, that is,

$$|M_{\text{HQCM}}\rangle = \prod_{\mathsf{j}=1}^{\mathsf{n}} \frac{I^{(\mathsf{j})} + (-1)^{s_{\mathsf{j}}} Z^{(\mathsf{j})}}{2} \boldsymbol{U}_B(\tau_{\max})|\text{out}\rangle \,. \tag{4.3}$$

The above Eq. (4.3) can be transformed with the help of Eq. (4.1) into

$$|M_{\text{HQCM}}\rangle = \boldsymbol{U}_B(\tau_{\max}) \prod_{\mathsf{j}=1}^{\mathsf{n}} \frac{I^{(\mathsf{j})} + (-1)^{s_{\mathsf{j}} + x_{\mathsf{j}}(\tau_{\max})} Z^{(\mathsf{j})}}{2} |\text{out}\rangle \,. \tag{4.4}$$

The inference we obtain by comparing Eq. (4.2) and (4.4) is that the readout measurements on the state $|\text{out}\rangle$ with the results $\{\acute{s}_{\mathsf{j}}\}$ give the same circuit output as the readout measurements on the state $\boldsymbol{U}_B(\tau_{\max})|\text{out}\rangle$ with the results $\{s_{\mathsf{j}}\}$, and

these sets of results are related by

$$\acute{s}_{\mathsf{j}} \equiv s_{\mathsf{j}} + x_{\mathsf{j}}(\tau_{\max}) \quad \text{for all} \quad \mathsf{j} \in \{1, 2, \cdots, \mathsf{n}\}\,. \tag{4.5}$$

That is how one can interpret the final result of the computation with the help of $\mathcal{I}_x(\tau_{\max})$ in the HQCM.

4.2.2 Propagation relations and propagation matrices for the elementary gates

Let us consider an arbitrary single-qubit rotation $R_{\vec{r}}(\upsilon)$ [of Eq. (2.20)] around an axis $\vec{r}(\theta, \varphi)$ [of Eq. (2.17)] by an angle υ. The by-product operator passes through this gate without any change, but the axis of rotation of the gate is changed from $\vec{r}$ to $\vec{r}\,'$. The propagation relation for $R_{\vec{r}}(\upsilon)$ is given by

$$R_{\vec{r}}(\upsilon)(X)^x(Z)^z = (X)^x(Z)^z R_{\vec{r}\,'}(\upsilon)\,, \tag{4.6}$$

where

$$\vec{r}\,' = \big((-1)^z \sin\theta\cos\varphi,\ (-1)^{x+z}\sin\theta\sin\varphi,\ (-1)^x\cos\theta\big)\,. \tag{4.7}$$

In other words, the angles θ, φ that define the axis of rotation $\vec{r}$ get transformed as

$$\begin{aligned} \theta &\to (x\pi - \theta)\,, \\ \varphi &\to (-1)^x(z\pi + \varphi)\,. \end{aligned} \tag{4.8}$$

The by-product operator passes through $R_{\vec{r}}(\upsilon)$ without getting transformed, which means that the propagation matrix $\mathbf{C}(R)$ is the same $2\mathsf{n} \times 2\mathsf{n}$ identity matrix as defined by Eq. (3.28).

Every single-qubit unitary operator in SU(2) follows this propagation relation. For $\upsilon = \frac{1}{2}\pi$, it becomes the propagation relation (3.19) for the Hadamard gate when

$\theta = \varphi = \frac{1}{2}\pi$ and the propagation relation (3.20) for the $\pi/2$-phase gate when $\theta = 0$. However, the Hadamard and the $\pi/2$-phase gate remain special cases in the sense that the propagation changes the by-product operator, but these gates stay as they are. Both of them are executed by the unitary evolution like any other single-qubit gate, but for the classical information-processing parts their propagation matrices defined by Eqs. (3.30) and (3.31) have to be used in the HQCM.

The propagation relation for the next elementary gate, $\mathrm{CZ}(\mathsf{a},\mathsf{b})$ of Eq. (2.46), is

$$\mathrm{CZ}(\mathsf{a},\mathsf{b})\,\boldsymbol{U}_B^{\mathrm{CZ}} = \tilde{\boldsymbol{U}}_B^{\mathrm{CZ}}\,\mathrm{CZ}(\mathsf{a},\mathsf{b})\,, \tag{4.9}$$

where

$$\boldsymbol{U}_B^{\mathrm{CZ}} = (X^{(\mathsf{a})})^{x_\mathsf{a}}(Z^{(\mathsf{a})})^{z_\mathsf{a}}(X^{(\mathsf{b})})^{x_\mathsf{b}}(Z^{(\mathsf{b})})^{z_\mathsf{b}}\,, \tag{4.10}$$

and

$$\tilde{\boldsymbol{U}}_B^{\mathrm{CZ}} = (X^{(\mathsf{a})})^{x_\mathsf{a}}(Z^{(\mathsf{a})})^{z_\mathsf{a}+x_\mathsf{b}}(X^{(\mathsf{b})})^{x_\mathsf{b}}(Z^{(\mathsf{b})})^{z_\mathsf{b}+x_\mathsf{a}}\,. \tag{4.11}$$

Under the one-to-one correspondence given in Eq. (3.21) the propagation relation (4.9) becomes

$$\begin{pmatrix} x_\mathsf{a} \\ x_\mathsf{b} \\ z_\mathsf{a}+x_\mathsf{b} \\ z_\mathsf{b}+x_\mathsf{a} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{C}(\mathrm{CZ})} \begin{pmatrix} x_\mathsf{a} \\ x_\mathsf{b} \\ z_\mathsf{a} \\ z_\mathsf{b} \end{pmatrix}. \tag{4.12}$$

When the control qubit a and the target qubit b belong to the set of n logical qubits ($\mathsf{a} \neq \mathsf{b}$), then the propagation matrix $\mathbf{C}(\mathrm{CZ}(\mathsf{a},\mathsf{b}))$ can be generated by the following

relations[2]:

$$
\begin{aligned}
&[\mathbf{C}_{xx}(\textsc{cz}(\mathsf{a},\mathsf{b}))]_{\mathsf{kl}} := [\mathbf{C}_{zz}(\textsc{cz}(\mathsf{a},\mathsf{b}))]_{\mathsf{kl}} := \delta_{\mathsf{kl}}\,,\\
&[\mathbf{C}_{xz}(\textsc{cz}(\mathsf{a},\mathsf{b}))]_{\mathsf{kl}} := \delta_{\mathsf{ka}}\delta_{\mathsf{lb}} + \delta_{\mathsf{kb}}\delta_{\mathsf{la}}\,,\\
&[\mathbf{C}_{zx}(\textsc{cz}(\mathsf{a},\mathsf{b}))]_{\mathsf{kl}} := 0\,.
\end{aligned}
\tag{4.13}
$$

Note that Eqs. (4.12) and (4.13) are different from Eqs. (3.24) and (3.29). The CZ and CNOT gates are interconvertible by using the Hadamard gate [see Eq. (2.47)], and the same is true for their propagation matrices, that is,

$$
\mathbf{C}(H^{(\mathsf{b})})\,\mathbf{C}\big(\textsc{cz}(\mathsf{a},\mathsf{b})\big)\,\mathbf{C}(H^{(\mathsf{b})}) = \mathbf{C}\big(\textsc{cnot}(\mathsf{a},\mathsf{b})\big)\,. \tag{4.14}
$$

The propagation relation for $U^{12\cdots\mathsf{n}}_{zz\cdots z}(\theta)$ is

$$
U^{12\cdots\mathsf{n}}_{zz\cdots z}(\theta)\,\boldsymbol{U}_B = \boldsymbol{U}_B\,U^{12\cdots\mathsf{n}}_{zz\cdots z}\big((-1)^x\theta\big), \tag{4.15}
$$

where $\boldsymbol{U}_B$ is the same as given in Eq. (3.21), and

$$
x = \sum_{\mathsf{j}=1}^{\mathsf{n}} x_{\mathsf{j}}\,. \tag{4.16}
$$

In this case, the measurement angle θ gets modified under the propagation, but the by-product operator stays as it is. Therefore, the propagation matrix $\mathbf{C}\big(U^{12\cdots\mathsf{n}}_{zz\cdots z}(\theta)\big)$ will be the $2\mathsf{n}\times 2\mathsf{n}$ identity matrix, which can be defined in the same way as the $\mathbf{C}(R)$ is defined in Eq. (3.28):

$$
\begin{aligned}
&\big[\mathbf{C}_{xx}\big(U^{12\cdots\mathsf{n}}_{zz\cdots z}(\theta)\big)\big]_{\mathsf{kl}} := \big[\mathbf{C}_{zz}\big(U^{12\cdots\mathsf{n}}_{zz\cdots z}(\theta)\big)\big]_{\mathsf{kl}} := \delta_{\mathsf{kl}}\,,\\
&\big[\mathbf{C}_{zx}\big(U^{12\cdots\mathsf{n}}_{zz\cdots z}(\theta)\big)\big]_{\mathsf{kl}} := \big[\mathbf{C}_{xz}\big(U^{12\cdots\mathsf{n}}_{zz\cdots z}(\theta)\big)\big]_{\mathsf{kl}} := 0\,.
\end{aligned}
\tag{4.17}
$$

[2]The same notation, as given in Sec. 3.2.2, for the propagation matrices applies here.

Now, one can draw the following conclusions. (1) The Hadamard, the $\pi/2$-phase and the CZ gates remain unchanged under the propagation, while the by-product operator gets altered. (2) Single- and multi-qubit rotations (with nontrivial angles) get transformed, while the by-product operator stays unaltered under the propagation. This completes the discussion of all the basic tools required for the HQCM.

Let us now turn into some important examples, which are useful for the implementation of GA within the framework of HQCM [see Sec. 6.2.1].

4.3 Controlled operations with the HQCM

In this section we are considering n-qubit controlled rotations around the z axis, which are defined by

$$\begin{aligned}\Lambda^{1\cdots\mathsf{c}}U_{z\cdots z}^{(\mathsf{c}+1)\cdots\mathsf{n}}(\theta) := \left[I^{\otimes\mathsf{c}} - |1\cdots1\rangle_{1\cdots\mathsf{c}}\langle1\cdots1|\right]\otimes I^{\otimes(\mathsf{n}-\mathsf{c})} \\ + |1\cdots1\rangle_{1\cdots\mathsf{c}}\langle1\cdots1|\otimes U_{z\cdots z}^{(\mathsf{c}+1)\cdots\mathsf{n}}(\theta)\,,\end{aligned} \tag{4.18}$$

where the qubits labeled 1 to c are the control qubits and the qubits labeled $\mathsf{c}+1$ to n are the target qubits [also see Eq. (2.51)]. Only when every control qubit is in the ket $|1\rangle$, then the $(\mathsf{n}-\mathsf{c})$-qubit rotation $U_{z\cdots z}^{(\mathsf{c}+1)\cdots\mathsf{n}}(\theta)$ operates on the target qubits. The HQCM implementation of these controlled rotations for three values—$\mathsf{c}=1$ (single control), $\mathsf{c}=2$ (double control) and $\mathsf{c}=3$ (triple control)—is presented here.

4.3.1 The single-control gate $\Lambda^1U_{z\cdots z}^{2\cdots\mathsf{n}}(-2\theta)$

First, $\Lambda^1U_{z\cdots z}^{2\cdots\mathsf{n}}(-2\theta)$ is decomposed in terms of multiqubit rotations like $U_{zz\cdots z}^{12\cdots\mathsf{n}}(\theta)$ of Eq. (3.8). To have the logical qubit 1 as the control and the rest as the target

qubits, the n-qubit rotation about the z axis is expressed as

$$U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta) = |0\rangle_1\langle 0| \otimes U_{z\cdots z}^{2\cdots \mathsf{n}}(\theta) + |1\rangle_1\langle 1| \otimes U_{z\cdots z}^{2\cdots \mathsf{n}}(-\theta)\,.$$

Consequently, the required decomposition is obtained:

$$\Lambda^1 U_{z\cdots z}^{2\cdots \mathsf{n}}(-2\theta) = U_{z\cdots z}^{2\cdots \mathsf{n}}(-\theta) U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)\,. \tag{4.19}$$

Since $U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)$ is symmetric under permutation of the qubits, one can take any qubit as the control and the remaining qubits as targets. In the HQCM, a "multiqubit rotation about the z axis" is considered as a *single unit*, and then $\Lambda^1 U_{z\cdots z}^{2\cdots \mathsf{n}}(-2\theta)$ costs only two units of this kind.

The circuit representation of Eq. (4.19) is given in Fig. 4.1. In Fig. 4.1(i), the left and the right rectangular boxes depict $U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)$ (the first rotation) and $U_{z\cdots z}^{2\cdots \mathsf{n}}(-\theta)$ (the second rotation), respectively. In practice, both of them are realized—by using a single ancilla qubit and a single measurement—with the methodology given in Sec. 3.1.5. Moreover, after executing the first rotation, the ancilla qubit is brought back into an eigenstate of X [given by Eq. (3.1)] and used for the second rotation. The implementation of $U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)$ and $U_{z\cdots z}^{2\cdots \mathsf{n}}(-\theta)$ require $(1+\mathsf{n})$-qubit and n-qubit star-graph states, respectively, where the ancilla qubit is connected to the relevant logical qubits [see Fig. 3.5(i) and Eq. (3.9)]. The eigenvalues of the ancilla qubit corresponding to the first and the second rotation are κ_1 and κ_2, and the measurement outcomes are m_1 and m_2, respectively.

The classical information processing for this gate can be decomposed into three parts. The first part deals with the changes in the measurement angles due to the by-product operator $\boldsymbol{U}_{B,\text{in}}$. This operator appears just before implementing the first rotation and is denoted by the dashed vertical line in the input section in Fig. 4.1. When the gate $\Lambda^1 U_{z\cdots z}^{2\cdots \mathsf{n}}(-2\theta)$ itself is a part of a circuit un-

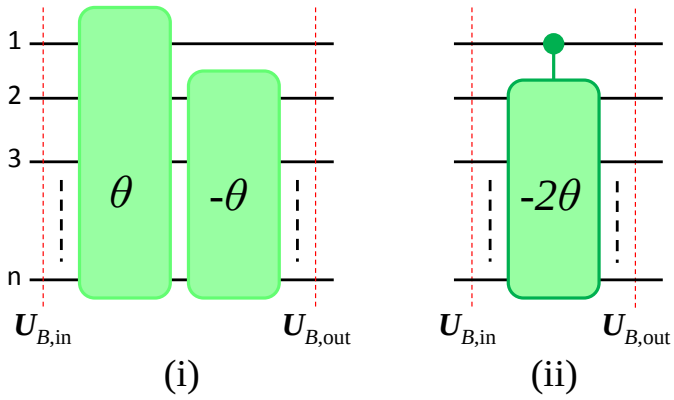


Figure 4.1: Quantum circuit (i) merely represents the temporal order of rotations for $\Lambda^1 U_{z\cdots z}^{2\cdots \mathsf{n}}(-2\theta)$. Horizontal lines represent n logical qubits. The left (green) rectangular box symbolizes the n-qubit rotation $U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)$, and the right box symbolizes the $(\mathsf{n}-1)$-qubit rotation $U_{z\cdots z}^{2\cdots \mathsf{n}}(-\theta)$. Both of them are executed under the scheme described in Sec. 3.1.5. Circuit (ii) represents $\Lambda^1 U_{z\cdots z}^{2\cdots \mathsf{n}}(-2\theta)$, where qubit 1 is the control qubit and the other qubits are targets. The dashed (red) vertical lines in the input and the output section represent the by-product operators $\boldsymbol{U}_{B,\text{in}}$ and $\boldsymbol{U}_{B,\text{out}}$, respectively. Circuits (i) and (ii) are equivalent.

der simulation, then the by-product $\boldsymbol{U}_{B,\text{in}}$ has emerged prior to the execution of $\Lambda^1 U_{z\cdots z}^{2\cdots \mathsf{n}}(-2\theta)$ due to the implementation of previous gates. Without loss of generality, $\boldsymbol{U}_{B,\text{in}} = \prod_{\mathsf{j}=1}^{\mathsf{n}} (X^{(\mathsf{j})})^{x_\mathsf{j}} (Z^{(\mathsf{j})})^{z_\mathsf{j}}$ is taken the same as given in Eq. (3.21). Only the x part of the corresponding information flow vector $\mathcal{I}_{x,\text{in}}$ influences the measurement bases $\mathcal{B}_{\pm\theta,(-1)^\kappa \frac{\pi}{2}}$ of Eq. (3.10) for both rotations. According to Eq. (4.15), the angles θ for the first and $-\theta$ for the second rotation get altered as follows

$$\begin{aligned} \theta &\to (-1)^x \theta \qquad \text{for} \quad U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)\,, \\ -\theta &\to -(-1)^{x-x_1}\theta \quad \text{for} \quad U_{z\cdots z}^{2\cdots \mathsf{n}}(-\theta)\,, \end{aligned} \tag{4.20}$$

where x is given by Eq. (4.16).

The second part of classical information processing deals with the eigenvalues κ_1 and κ_2, which influence the azimuthal angle $\frac{1}{2}\pi$ of the measurement bases in the following way:

$$\begin{aligned} \tfrac{1}{2}\pi &\to (-1)^{\kappa_1}\tfrac{1}{2}\pi \quad \text{for} \quad U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)\,, \\ \tfrac{1}{2}\pi &\to (-1)^{\kappa_2}\tfrac{1}{2}\pi \quad \text{for} \quad U_{z\cdots z}^{2\cdots \mathsf{n}}(-\theta)\,. \end{aligned} \tag{4.21}$$

Finally, the third part manages the contribution of measurement outcomes m_1 and m_2 to the by-product operator $\boldsymbol{U}_{B,\text{in}}$. The implementation of both the first and

the second rotations cause the by-product operators $(Z^{\otimes \mathsf{n}})^{m_1}$ and $(Z^{\otimes(\mathsf{n}-1)})^{m_2}$ on the relevant logical qubits. Furthermore, these by-product operators update $\boldsymbol{U}_{B,\text{in}}$ to $\boldsymbol{U}_{B,\text{out}}$. $\boldsymbol{U}_{B,\text{out}}$ is represented by the dashed vertical line in the output section in Fig. 4.1. Only the z part of the information flow vector $\mathcal{I}_{z,\text{in}}$ gets changed, while the x part remains as it is, that is, $\mathcal{I}_{x,\text{out}} = \mathcal{I}_{x,\text{in}}$,

$$\boldsymbol{U}_{B,\text{out}} = (X^{(1)})^{x_1}(Z^{(1)})^{z_1+m_1}\prod_{\mathsf{j}=2}^{\mathsf{n}}(X^{(\mathsf{j})})^{x_\mathsf{j}}(Z^{(\mathsf{j})})^{z_\mathsf{j}+m_1+m_2}\,. \tag{4.22}$$

4.3.2 The double-control gate $\Lambda^{12}U^{3\cdots\mathsf{n}}_{z\cdots z}(4\theta)$

One has to combine two additional units $U^{13\cdots\mathsf{n}}_{zz\cdots z}(-\theta)$ (the third rotation) and $U^{3\cdots\mathsf{n}}_{z\cdots z}(\theta)$ (the fourth rotation) with $\Lambda^1U^{2\cdots\mathsf{n}}_{z\cdots z}(-2\theta)$ for the purpose of getting the $\Lambda^{12}U^{3\cdots\mathsf{n}}_{z\cdots z}(4\theta)$ gate. In other words, $\Lambda^{12}U^{3\cdots\mathsf{n}}_{z\cdots z}(4\theta)$ with two control qubits, 1 and 2, is made of four rotations, and its decomposition is given by

$$\Lambda^{12}U^{3\cdots\mathsf{n}}_{z\cdots z}(4\theta) = U^{3\cdots\mathsf{n}}_{z\cdots z}(\theta)\,U^{13\cdots\mathsf{n}}_{zz\cdots z}(-\theta)\,U^{2\cdots\mathsf{n}}_{z\cdots z}(-\theta)\,U^{12\cdots\mathsf{n}}_{zz\cdots z}(\theta)\,. \tag{4.23}$$

Figure 4.2(i) illustrates the temporal ordering of the multiqubit rotations given in Eq. (4.23) by the rectangular boxes.

The treatment for $\Lambda^{12}U^{3\cdots\mathsf{n}}_{z\cdots z}(4\theta)$ is similar to that of Sec. 4.3.1. All the rotations—$U^{12\cdots\mathsf{n}}_{zz\cdots z}(\theta)$ (first), $U^{2\cdots\mathsf{n}}_{z\cdots z}(-\theta)$ (second), $U^{13\cdots\mathsf{n}}_{zz\cdots z}(-\theta)$ (third), and $U^{3\cdots\mathsf{n}}_{z\cdots z}(\theta)$ (fourth)—are performed one after another under the scheme presented in Sec. 3.1.5. After initializing the ancilla qubit in the X eigenbasis [of Eq. (3.1)], a necessary star-graph state for the first rotation is prepared, and then the ancilla qubit is measured in the appropriate basis. The measurement outcome is then recorded, and the ancilla qubit is brought back again into an eigenstate of X (recycled) for executing the next rotation. In this way, one can use the same ancilla qubit for all the rotations. κ_1, κ_2, κ_3, and κ_4 are the eigenvalues of the ancilla qubit, while m_1, m_2, m_3, and m_4 are the measurement outcomes corresponding to the first, second, third, and fourth

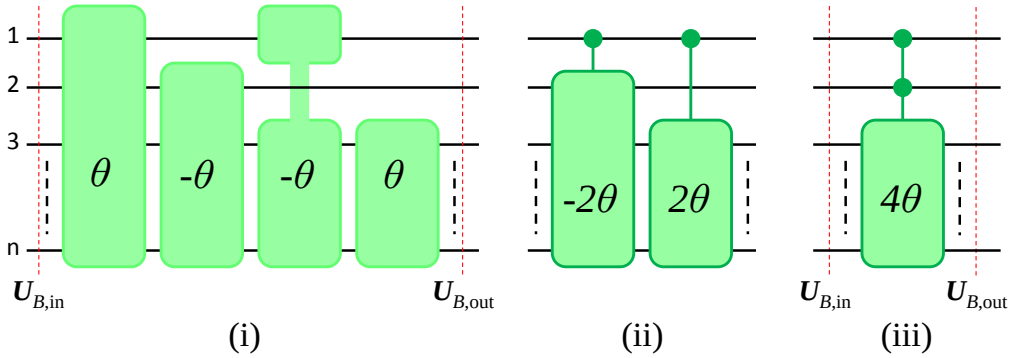


Figure 4.2: Horizontal lines represent n logical qubits. (i) The four (green) rectangular boxes (from left to right) represent $U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)$, $U_{z\cdots z}^{2\cdots \mathsf{n}}(-\theta)$, $U_{zz\cdots z}^{13\cdots \mathsf{n}}(-\theta)$, and $U_{z\cdots z}^{3\cdots \mathsf{n}}(\theta)$, respectively. Each rotation is realized under the scheme described in Sec. 3.1.5. (ii) The left (green) rectangular box symbolizes the n-qubit operation $\Lambda^1 U_{z\cdots z}^{2\cdots \mathsf{n}}(-2\theta)$, and the right box symbolizes the $(\mathsf{n}-1)$-qubit operation $\Lambda^1 U_{z\cdots z}^{3\cdots \mathsf{n}}(2\theta)$. Both of them have qubit 1 as the control. The diagram (iii) represents $\Lambda^{12} U_{z\cdots z}^{3\cdots \mathsf{n}}(4\theta)$, where qubits 1 and 2 are the control qubits. In (i) and (iii), the dashed (red) vertical lines in the input and output sections represent the by-product operators $\boldsymbol{U}_{B,\text{in}}$ and $\boldsymbol{U}_{B,\text{out}}$, respectively. Circuit (ii) merely depicts the intermediate stage of circuits (i) and (iii), and they all are mutually equivalent.

rotations. As a side remark, one can also choose to perform these four rotations at the same time by using four different ancilla qubits, but this would require more hardware resources. In the HQCM, $\Lambda^{12} U_{z\cdots z}^{3\cdots \mathsf{n}}(4\theta)$ can be completed in a *single time step*, because all the four rotations can be executed at the same time.

The classical information processing for this case can also be decomposed into three parts. The first part deals with the modification in the measurement angles because of the by-product operator $\boldsymbol{U}_{B,\text{in}} = \prod_{\mathsf{j}=1}^{\mathsf{n}} (X^{(\mathsf{j})})^{x_\mathsf{j}} (Z^{(\mathsf{j})})^{z_\mathsf{j}}$ [same as of Eq. (3.21)], which is represented by the dashed vertical line in the input section in Figs. 4.2(i) and 4.2(iii). Here also, only $\mathcal{I}_{x,\text{in}}$ influences the measurement angle $\pm\theta$ for every rotation.

$$\begin{aligned}
\theta &\to (-1)^x \theta && \text{for} \quad U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)\,, \\
-\theta &\to -(-1)^{x-x_1} \theta && \text{for} \quad U_{z\cdots z}^{2\cdots \mathsf{n}}(-\theta)\,, \\
-\theta &\to -(-1)^{x-x_2} \theta && \text{for} \quad U_{zz\cdots z}^{13\cdots \mathsf{n}}(-\theta)\,, \\
\theta &\to (-1)^{x-x_1-x_2} \theta && \text{for} \quad U_{z\cdots z}^{3\cdots \mathsf{n}}(\theta),
\end{aligned} \tag{4.24}$$

where x is the same as given by Eq. (4.16).

The second part manages the influence of the eigenvalues κ_1, κ_2, κ_3, and κ_4 on the azimuthal angle $\frac{1}{2}\pi$ of the measurement bases $\mathcal{B}_{\pm\theta,(-1)^{\kappa}\frac{\pi}{2}}$ of Eq. (3.10) in the following way:

$$\begin{aligned}
&\tfrac{1}{2}\pi \to (-1)^{\kappa_1}\tfrac{1}{2}\pi \quad \text{for} \quad U^{12\cdots \mathsf{n}}_{zz\cdots z}(\theta)\,,\\
&\tfrac{1}{2}\pi \to (-1)^{\kappa_2}\tfrac{1}{2}\pi \quad \text{for} \quad U^{2\cdots \mathsf{n}}_{z\cdots z}(-\theta)\,,\\
&\tfrac{1}{2}\pi \to (-1)^{\kappa_3}\tfrac{1}{2}\pi \quad \text{for} \quad U^{13\cdots \mathsf{n}}_{zz\cdots z}(-\theta)\,,\\
&\tfrac{1}{2}\pi \to (-1)^{\kappa_4}\tfrac{1}{2}\pi \quad \text{for} \quad U^{3\cdots \mathsf{n}}_{z\cdots z}(\theta)\,.
\end{aligned} \tag{4.25}$$

The third part handles the random measurement outcomes m_1, m_2, m_3, and m_4, which cause the by-product operators $(Z^{\otimes \mathsf{n}})^{m_1}$, $(Z^{\otimes(\mathsf{n}-1)})^{m_2}$, $(Z^{\otimes(\mathsf{n}-1)})^{m_3}$, and $(Z^{\otimes(\mathsf{n}-2)})^{m_4}$, respectively, on the relevant logical qubits. Furthermore, they change $\boldsymbol{U}_{B,\text{in}}$ into $\boldsymbol{U}_{B,\text{out}}$ by their contribution. $\boldsymbol{U}_{B,\text{out}}$ is represented by the dashed vertical line in the output section in Figs. 4.2(i) and 4.2(iii). Consequently, only the z part of the corresponding information flow vector $\mathcal{I}_{z,\text{in}}$ gets changed into $\mathcal{I}_{z,\text{out}}$, while $\mathcal{I}_{z,\text{out}} = \mathcal{I}_{z,\text{in}}$,

$$\mathcal{I}_{z,\text{out}} = \begin{pmatrix} z_1 + m_1 + m_3 \\ z_2 + m_1 + m_2 \\ z_3 + m \\ \vdots \\ z_{\mathsf{n}} + m \end{pmatrix}, \tag{4.26}$$

where

$$m = m_1 + m_2 + m_3 + m_4\,.$$

Two points are worth emphasizing here. First, $\Lambda^{12}U^{3\cdots \mathsf{n}}_{z\cdots z}(4\theta)$ is symmetric under permutation of the qubits. Hence, we can take any two qubits as controls and the rest of qubits as targets by using only four multiqubit rotations. If we use the same argument further, then the controlled rotation $\Lambda^{1\cdots \mathsf{c}}U^{(\mathsf{c}+1)\cdots \mathsf{n}}_{z\cdots z}(\theta)$ of Eq. (4.18), with c control qubits, requires 2^{c} units of rotation. The number 2^{c} is independent of the number of target qubits $(\mathsf{n}-\mathsf{c})$, but when c becomes of the order of n, then

2^{c} becomes exponential in n. To fix this exponential growth problem, some extra work qubits are needed [23]. The next subsection will exemplify this remark.

Second, $\Lambda^{12}U_{z\cdots z}^{3\cdots \mathsf{n}}(4\theta)$ becomes $\Lambda^{12}U_z^{(3)}(4\theta)$ for $\mathsf{n}=3$, where $U_z^{(3)}(4\theta)$ on the logical qubit 3 is the phase gate $R_z^{(3)}(4\theta)$ defined by Eq. (3.3). The gate $\Lambda^{12}U_z^{(3)}(4\theta)$ is equivalent to Deutsch's universal[3] gate $\Lambda^{12}\big[iR_x^{(3)}(4\theta)\big]$ [20] up to some single-qubit unitary operations. In the next section, the three-qubit gates like $\Lambda^{12}U_z^{(3)}(\pm\pi)$ with two work qubits are used to implement the four-qubit gate $\Lambda^{123}Z^{(6)}$.

4.3.3 The triple-control gate $\Lambda^{123}Z^{(6)}$

We put every bit and piece of the HQCM together in the implementation of the gate $\Lambda^{123}Z^{(6)}$. The complete scheme about its implementation in terms of its circuit diagram is shown in Fig. 4.3, and the associated classical information-processing parts are listed in Table 4.1.

First, the gate $\Lambda^{123}Z^{(6)}$ is efficiently decomposed into a sequence of the elementary gates—single-qubit, the CZ, and $U_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)$ gates—of Sec. 4.1.1. The temporal order of the elementary gates for $\Lambda^{123}Z^{(6)}$ is depicted by the circuit diagram in Fig. 4.3, where qubits 1, 2, and 3 act as the control qubits and qubit 6 acts as the target qubit. They all are represented by the black horizontal lines. The work qubits 4 and 5 start and end in the ket $|+\rangle^{\otimes 2}$ [see Eq. (2.28)], and they are represented by the gray horizontal lines in Fig. 4.3. Like in Sec. 2.4.2, the work qubits are used here to make the decomposition of $\Lambda^{123}Z^{(6)}$ economical.

The three-qubit gates $\Lambda^{12}U_z^{(4)}(\pi)$ (first), $\Lambda^{34}U_z^{(5)}(\pi)$ (third), $\Lambda^{34}U_z^{(5)}(-\pi)$ (seventh), and $\Lambda^{12}U_z^{(4)}(-\pi)$ (ninth) are represented by rectangular boxes with double control. Every three-qubit gate is further decomposed into four rotations around the z axis according to Eq. (4.23); here $\mathsf{n}=3$ and $\theta=\pm\frac{1}{4}\pi$. Furthermore, each rotation is executed by preparing a required star-graph state, followed by the measurement in the appropriate basis. The detailed methodology is mentioned in Sec. 4.3.2. The

[3]Deutsch's gate is universal, provided the angle 4θ is incommensurate with π.

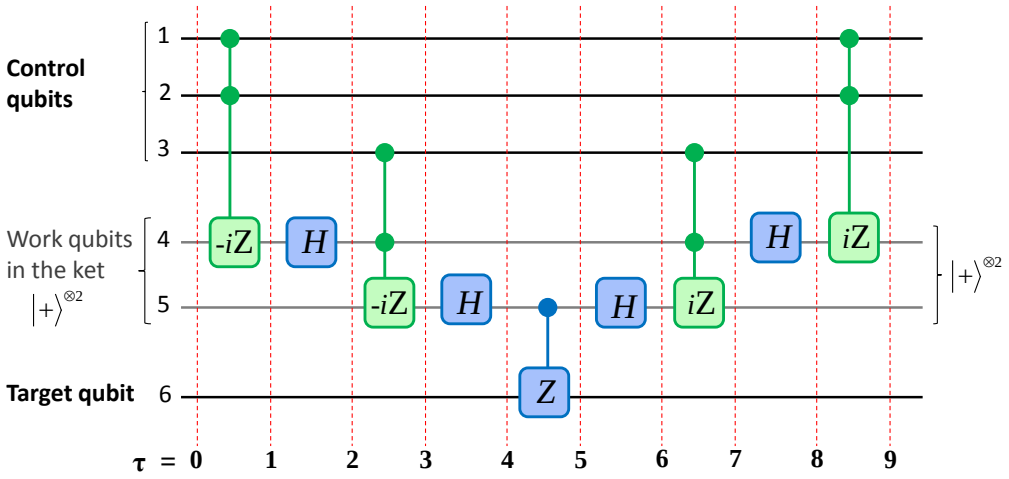


Figure 4.3: The quantum circuit—similar to the circuit of Fig. 2.6—for implementing the four-qubit gate $\Lambda^{123}Z^{(6)}$. From top to bottom, the three black horizontal lines represent control qubits 1, 2, and 3, and the next two, gray horizontal lines represent work qubits 4 and 5, which are prepared in the ket $|+\rangle^{\otimes 2}$. The black horizontal line at the bottom represents target qubit 6. Every three-qubit gate is the special case of $\Lambda^{12}U_{z\cdots z}^{3\cdots \mathsf{n}}(4\theta)$ (for $\mathsf{n}=3$ and $\theta=\pm\frac{1}{4}\pi$), and they are realized by the procedure given in Sec. 4.3.2. The Hadamard H and $\mathrm{CZ}(5,6)$ gates are executed by the unitary evolution. The computation steps (τ) are expressed by dashed (red) vertical lines for the classical information-processing parts listed in Table 4.1.

Hadamard gates H of Eq. (2.23) are displayed by the rounded rectangles, and the two-qubit gate $\mathrm{CZ}(5,6)$ of Eq. (2.46) is shown by the rounded rectangle on qubit 6 with qubit 5 as the control in Fig. 4.3. The Hadamard and the $\mathrm{CZ}(5,6)$ gates are executed by the unitary evolution.

The classical information-processing parts for $\Lambda^{123}Z^{(6)}$ are handled by a CC according to Table 4.1. In this table, the first column is for the computational steps τ, which are represented by the dashed vertical lines in Fig. 4.3. There are 10 vertical lines in the figure and 10 rows in the table for the 10 computational steps from **0** to **9**. At each vertical line the information flow vector $\mathcal{I}(\tau)$ gets updated. The second and third columns are reserved for $\mathcal{I}_x(\tau)$ and $\mathcal{I}_z(\tau)$, respectively. If required, the changes in the measurement angles for the next gate based on the updated value of $\mathcal{I}(\tau)$ is calculated; they are given in the fourth column. After performing the measurements in the appropriate bases, the measurement outcomes are recorded in the fifth column.

Let us go through Table 4.1 row by row. Before starting the computation (in the

Table 4.1: Classical information-processing parts for $\Lambda^{123}Z^{(6)}$

τ	$\mathcal{I}_x(\tau)$	$\mathcal{I}_z(\tau)$	Angle $\pm\theta$ (here $\theta = \frac{1}{4}\pi$)	Measurement outcomes
0	$\begin{pmatrix}0\\0\\0\\0\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\0\\0\\0\\0\\0\end{pmatrix}$	No change in angle for $U^{124}_{zzz}(\theta)$ No change in angle for $U^{24}_{zz}(-\theta)$ No change in angle for $U^{14}_{zz}(-\theta)$ No change in angle for $U^{4}_{z}(\theta)$	m_{11} for $U^{124}_{zzz}(\theta)$ m_{12} for $U^{24}_{zz}(-\theta)$ m_{13} for $U^{14}_{zz}(-\theta)$ m_{14} for $U^{4}_{z}(\theta)$ $m_1 = m_{11} + m_{12} + m_{13} + m_{14}$
1	$\begin{pmatrix}0\\0\\0\\0\\0\\0\end{pmatrix}$	$\begin{pmatrix}m_{11}+m_{13}\\m_{11}+m_{12}\\0\\m_1\\0\\0\end{pmatrix}$		
2	$\begin{pmatrix}0\\0\\0\\m_1\\0\\0\end{pmatrix}$	$\begin{pmatrix}m_{11}+m_{13}\\m_{11}+m_{12}\\0\\0\\0\\0\end{pmatrix}$	$\theta \to (-1)^{m_1}\theta$ for $U^{345}_{zzz}(\theta)$ $-\theta \to -(-1)^{m_1}\theta$ for $U^{45}_{zz}(-\theta)$ No change in angle for $U^{35}_{zz}(-\theta)$ No change in angle for $U^{5}_{z}(\theta)$	m_{31} for $U^{345}_{zzz}(\theta)$ m_{32} for $U^{45}_{zz}(-\theta)$ m_{33} for $U^{35}_{zz}(-\theta)$ m_{34} for $U^{5}_{z}(\theta)$ $m_3 = m_{31} + m_{32} + m_{33} + m_{34}$
3	$\begin{pmatrix}0\\0\\0\\m_1\\0\\0\end{pmatrix}$	$\begin{pmatrix}m_{11}+m_{13}\\m_{11}+m_{12}\\m_{31}+m_{33}\\m_{31}+m_{32}\\m_3\\0\end{pmatrix}$		
4	$\begin{pmatrix}0\\0\\0\\m_1\\m_3\\0\end{pmatrix}$	$\begin{pmatrix}m_{11}+m_{13}\\m_{11}+m_{12}\\m_{31}+m_{33}\\m_{31}+m_{32}\\0\\0\end{pmatrix}$		
5	$\begin{pmatrix}0\\0\\0\\m_1\\m_3\\0\end{pmatrix}$	$\begin{pmatrix}m_{11}+m_{13}\\m_{11}+m_{12}\\m_{31}+m_{33}\\m_{31}+m_{32}\\0\\m_3\end{pmatrix}$		
6	$\begin{pmatrix}0\\0\\0\\m_1\\0\\0\end{pmatrix}$	$\begin{pmatrix}m_{11}+m_{13}\\m_{11}+m_{12}\\m_{31}+m_{33}\\m_{31}+m_{32}\\m_3\\m_3\end{pmatrix}$	$-\theta \to -(-1)^{m_1}\theta$ for $U^{345}_{zzz}(-\theta)$ $\theta \to (-1)^{m_1}\theta$ for $U^{45}_{zz}(\theta)$ No change in angle for $U^{35}_{zz}(\theta)$ No change in angle for $U^{5}_{z}(-\theta)$	m_{71} for $U^{345}_{zzz}(-\theta)$ m_{72} for $U^{45}_{zz}(\theta)$ m_{73} for $U^{35}_{zz}(\theta)$ m_{74} for $U^{5}_{z}(-\theta)$ $m_7 = m_{71} + m_{72} + m_{73} + m_{74}$
7	$\begin{pmatrix}0\\0\\0\\m_1\\0\\0\end{pmatrix}$	$\begin{pmatrix}m_{11}+m_{13}\\m_{11}+m_{12}\\m_{31}+m_{33}+m_{71}+m_{73}\\m_{31}+m_{32}+m_{71}+m_{72}\\m_3+m_7\\m_3\end{pmatrix}$		
8	$\begin{pmatrix}0\\0\\0\\\tilde{m}\\0\\0\end{pmatrix}$	$\begin{pmatrix}m_{11}+m_{13}\\m_{11}+m_{12}\\m_{31}+m_{33}+m_{71}+m_{73}\\m_1\\m_3+m_7\\m_3\end{pmatrix}$	$-\theta \to -(-1)^{\tilde{m}}\theta$ for $U^{124}_{zzz}(-\theta)$ $\theta \to (-1)^{\tilde{m}}\theta$ for $U^{24}_{zz}(\theta)$ $\theta \to (-1)^{\tilde{m}}\theta$ for $U^{14}_{zz}(\theta)$ $-\theta \to -(-1)^{\tilde{m}}\theta$ for $U^{4}_{z}(-\theta)$ $\tilde{m} = m_{31} + m_{32} + m_{71} + m_{72}$	m_{91} for $U^{124}_{zzz}(-\theta)$ m_{92} for $U^{24}_{zz}(\theta)$ m_{93} for $U^{14}_{zz}(\theta)$ m_{94} for $U^{4}_{z}(-\theta)$ $m_9 = m_{91} + m_{92} + m_{93} + m_{94}$
9	$\begin{pmatrix}0\\0\\0\\\tilde{m}\\0\\0\end{pmatrix}$	$\begin{pmatrix}m_{11}+m_{13}+m_{91}+m_{93}\\m_{11}+m_{12}+m_{91}+m_{92}\\m_{31}+m_{33}+m_{71}+m_{73}\\m_1+m_9\\m_3+m_7\\m_3\end{pmatrix}$		

first row $\tau = \mathbf{0}$), all the entries of both $\mathcal{I}_x(\mathbf{0})$ and $\mathcal{I}_z(\mathbf{0})$ are zeros (initialization). So, there is no change in the measurement angle for each of the four rotations associated with the gate $\Lambda^{12}U_z^{(4)}(\pi)$ (first gate). The measurement outcomes m_{11}, m_{12}, m_{13}, and m_{14} corresponding to the four rotations $U_{zzz}^{124}(\theta)$, $U_{zz}^{24}(-\theta)$, $U_{zz}^{14}(-\theta)$, and $U_z^4(\theta)$ are recorded. These outcomes give some nonzero entries to $\mathcal{I}_z(\mathbf{1})$ according to Eq. (4.26). The measurement outcome m_{jk} corresponds to the kth rotation of the jth three-qubit gate. The next gate in the circuit is the Hadamard gate $H^{(4)}$, which does not change under the propagation of the by-product operator; therefore the fourth column of the second row $\tau = \mathbf{1}$ is empty. The H gate is realized by the unitary evolution; therefore the fifth column of the second row is also empty. However, the H gate changes the information flow vector $\mathcal{I}(\mathbf{1})$ into $\mathcal{I}(\mathbf{2})$ under the propagation relation given by Eq. (3.19), and the propagation matrix for the H gate is defined by Eq. (3.30) [also see Eq. (3.25)]. The third gate is $\Lambda^{34}U_z^{(5)}(\pi)$. The measurement angles $\pm\theta$ only for the rotations $U_{zzz}^{345}(\theta)$ and $U_{zz}^{45}(-\theta)$ get influenced by $\mathcal{I}_x(\mathbf{2})$ according to Eq. (4.24). The measurement outcomes m_{31}, m_{32}, m_{33}, and m_{34} only transform $\mathcal{I}_z(\mathbf{2})$ into $\mathcal{I}_z(\mathbf{3})$. In this way going through Table 4.1 along with Fig. 4.3 explains the whole scheme, and the final output result is interpreted according to Eq. (4.5) with the help of $\mathcal{I}_x(\mathbf{9})$.

Here, the z part of the information flow vector $\mathcal{I}_z(\tau)$ gets the new entries from the implementation of three-qubit gates only according to Eq. (4.26). The entries of $\mathcal{I}(\tau)$ get manipulated under the propagation of the by-product operator through the Hadamard gates and the $\textsc{cz}(5,6)$ gate according to the propagation relations (3.19) and (4.9), respectively. However, the propagation of the by-product operator does not change the H and CZ gates. The x part of the information flow vector $\mathcal{I}_x(\tau)$ influences the measurement angles $\pm\theta$ of the rotations for every three-qubit gate according to Eq. (4.24). As a side remark, the sign of the azimuthal angle $\frac{1}{2}\pi$ of the measurement bases for the rotations also depends on the eigenvalues of the ancilla qubit according to Eq. (4.25). This is not mentioned in Table 4.1.

One can easily generalize this example up to the n-qubit two-level unitary gate $\Lambda^{12\cdots(\mathsf{n}-1)}U^{(\mathsf{n})}$ [see Sec. 2.4.2], where the $\mathsf{n}-1$ logical qubits $1, 2, \cdots, \mathsf{n}-1$ are the control qubits and the last qubit n is the target qubit on which the single-qubit gate U is applied. To implement this gate with the HQCM, one needs $\mathsf{n}-2$ work qubits, which are initialized in the key $|+\rangle^{\otimes(\mathsf{n}-2)}$. We already know from Sec. 2.4.2 that a general $2^{\mathsf{n}} \times 2^{\mathsf{n}}$ unitary gate can be written down as a product of two-level unitary gates. Hence, the HQCM can realize any unitary operation. Let us add here that the gate $\Lambda^{12\cdots(\mathsf{n}-1)}Z^{(\mathsf{n})}$ plays a very important role in GA. Simulation of GA within the framework of HQCM will be discussed in Sec. 6.2.1.

Chapter 5

Encoded gates within the hybrid quantum computation model

One of the biggest challenges in a practical implementation of quantum communication and computation is to protect the quantum information from the noise. This challenge can be overcome by encoding a *logical qubit* into a number of *physical qubits* with the aid of quantum error-correcting codes [6, 7, 8, 9, 10]. These error-correcting codes facilitate reliable storage, communication and computation of the information.

Another challenge is due to imperfect unitary gates and measurements. As unitary operations [see Eq. (2.4)] and measurement bases [see Eq. (2.16)] depend on *continuous* parameters, they cannot be executed with *perfect accuracy*. Small imperfections in the gates and measurements cause errors in the computation. These errors can *propagate* and *accumulate* over the course of a computation, become no longer correctable and eventually causing a failure. To overcome this issue, one must perform the elementary components—gates and measurements—in such a way that brings the accumulated errors below a certain threshold [29, 30]. A computer that works effectively even when its elementary components are imperfect is said to be *fault tolerant* [25, 26, 27, 28].

Primary steps in the development of fault-tolerant quantum computation[1] are to choose an appropriate quantum error-correcting code and to design a procedure to get the corresponding *encoded gates*. Encoded gates are those which can operate *directly* (without decode, perform a gate, and then re-encode) on logical qubits[2]. In this chapter, we only discuss these primary steps for the HQCM. To achieve a proper fault-tolerant HQCM, encoding, processing with encoded gates and decoding of the information have to be performed in a fault-tolerant manner. This will be the subject of further studies.

Due to its interesting properties, researcher have exploited the Steane 7-qubit code [7] in fault-tolerant quantum computation [25, 26, 27, 28]. We also choose to use this code [see Sec. 5.1] in the following discussion. In Secs. 5.2 and 5.3, we present certain methods to obtain encoded elementary gates [of Sec. 4.1.1] within the HQCM.

5.1 Steane 7-qubit code

The Steane 7-qubit code[3] enables us to encode a single-qubit state using altogether 7 physical qubits. Here, an arbitrary single-qubit ket $|\psi(1)\rangle$ of Eq. (2.13) is encoded in a two-dimensional subspace $\mathcal{C}$ spanned by the two logical kets

$$
\begin{aligned}
|0\rangle_L &:= \frac{1}{2}\big[|000\rangle|\eta_1\rangle + |011\rangle|\eta_2\rangle + |101\rangle|\eta_3\rangle + |110\rangle|\eta_4\rangle\big] , \\
|1\rangle_L &:= \frac{1}{2}\big[|111\rangle|\eta_1\rangle + |100\rangle|\eta_2\rangle + |010\rangle|\eta_3\rangle + |001\rangle|\eta_4\rangle\big] ,
\end{aligned} \tag{5.1}
$$

[1] In contrast to a QC, a modern digital CC is so much reliable that its fault-tolerant version is not required for operating it reliably.

[2] A valid encoded gate on logical qubits produces the same effect as the corresponding unencoded gate on unencoded qubits [for examples, see Secs. 5.2 and 5.3].

[3] It is the quantum version of the Hamming 7-bit code which is used to encode 4 bits of classical information [1, 7, 26].

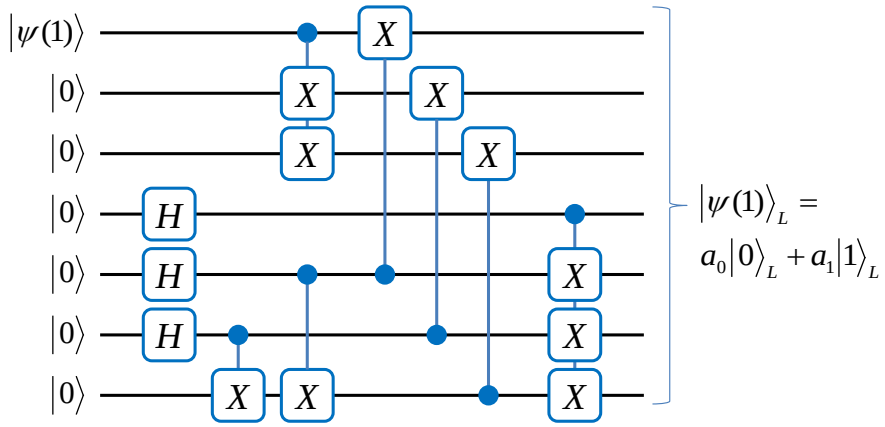


Figure 5.1: An encoding circuit for the Steane 7-qubit code, where the data qubit 1 is given in the ket $|\psi(1)\rangle$ of Eq. (2.13), and the rest of 6 qubits are initialized in the ket $|0\rangle^{\otimes 6}$. Only the Hadamard H of Eq. (2.23) and CNOT of Eq. (2.39) gates are employed here for encoding a single qubit, and every control is set to $|1\rangle$. One can use the same circuit for decoding by running it in the reverse order.

where

$$
\begin{aligned}
|\eta_1\rangle &:= \frac{1}{\sqrt{2}}\big[|0000\rangle + |1111\rangle\big]\,, \\
|\eta_2\rangle &:= \frac{1}{\sqrt{2}}\big[|0011\rangle + |1100\rangle\big]\,, \\
|\eta_3\rangle &:= \frac{1}{\sqrt{2}}\big[|0101\rangle + |1010\rangle\big]\,, \\
|\eta_4\rangle &:= \frac{1}{\sqrt{2}}\big[|0110\rangle + |1001\rangle\big]\,.
\end{aligned}
\tag{5.2}
$$

These logical kets are called *codewords*, and the space $\mathcal{C} := \{|0\rangle_L, |1\rangle_L\}$ that they span is called *code space*. The encoding can be accomplished by using a quantum circuit shown in Fig. 5.1. Note that this circuit is not fault tolerant.

This code enable us to recover from a single error occurring in any of the 7 qubits. If an error occurs, the measurement of specific operators, the *stabilizer*[4] [10, 27], will reveal it. The error, once identified by the measurement outcome, or *syndrome*, can then be corrected. A detailed description of error-correcting procedure for the 7-qubit code is given in Ref. [26].

Let us now turn to design *encoded operations* for this code. A valid encoded

[4]The logical kets $|0\rangle_L$ and $|1\rangle_L$ are eigenkets of a set of operators with the eigenvalue $+1$. This set is called stabilizer.

operation transforms the logical kets in such a way that the resultant kets stay in the code space. We want the encoded Pauli operators transform the logical kets in the same way as the Pauli operators X, Y and Z transform the computational basis $\{|0\rangle, |1\rangle\}$. Therefore, the encoded Pauli operators X_L, Y_L and Z_L for this code simply are

$$\mathsf{X}_L := X^{\otimes 7}, \qquad \mathsf{Y}_L := -Y^{\otimes 7}, \qquad \mathsf{Z}_L := Z^{\otimes 7} \tag{5.3}$$

whose actions on the logical kets are by construction

$$\begin{aligned} &\mathsf{X}_L\,|0\rangle_L = |1\rangle_L\,, \qquad &&\mathsf{Y}_L\,|0\rangle_L = +i|1\rangle_L\,, \qquad &&\mathsf{Z}_L\,|0\rangle_L = +|0\rangle_L\,, \\ &\mathsf{X}_L\,|1\rangle_L = |0\rangle_L\,, \qquad &&\mathsf{Y}_L\,|1\rangle_L = -i|0\rangle_L\,, \qquad &&\mathsf{Z}_L\,|1\rangle_L = -|0\rangle_L\,. \end{aligned} \tag{5.4}$$

The action of X_L on the logical kets is almost self-evident, and the actions of Y_L and Z_L can be easily understood by noticing that every ket in the superposition of $|0\rangle_L$ and $|1\rangle_L$ has 0(mod 4) and 3(mod 4) number of ones, respectively.

One can note that the application of the Pauli gates X, Y and Z to each of the 7 qubits according to Fig. 5.2(i) implement the encoded gates X_L, $-\mathsf{Y}_L$ and Z_L, respectively. This method of implementation is called *transversal* or *bitwise* implementation, which fulfills the two basic criteria of fault tolerance [25, 26, 27, 28].

First, the transversal implementation, in which an operation acts independently on each qubit in a block, minimizes the spread of existing errors[5] within the encoded block. It is necessary because the chosen 7-qubit code enables us to correct only one error per block. The spread of more than one error during the computation rapidly reduces the code's tolerance for errors and causes failure of the computation.

Second, the bitwise application of certain gates implements their respective encoded operations *directly* (without decoding and encoding afterwards) on the en-

[5] In Sec. 5.2, the spread of errors is explained by taking the CNOT gate as an example.

coded data. Of course, one can decode, perform a gate, and then re-encode, but that procedure will temporarily expose the quantum information to the noise. Consequently, it can cause errors in the information processing.

Unfortunately, it is not possible to realize every encoded gate transversally for a fix (or chosen) code, because the transversal implementation relays on the properties of the code. In the next section, we present the gates which can be implemented transversally in the case of the 7-qubit code [26, 27].

5.2 Encoded gates on one and two logical qubits

In this section, we present the transversal implementation of the gates from the Clifford group. As mentioned in Sec. 3.1.4, the Hadamard H, $\pi/2$-phase F and CNOT gates from the generating set of the Clifford group. In the case of 7-qubit code, the bitwise application of these gates executes the corresponding encoded operations. Furthermore, we can obtain other encoded gates of the Clifford group with this generating set.

Similar to the case of encoded Pauli gates, the encoded Hadamard $\mathsf{H}_L := H^{\otimes 7}$ gate can be achieved by applying H [of Eq. (2.23)] transversally according to Fig. 5.2(i). The encoded gate H_L transforms the logical kets as

$$\begin{aligned} \mathsf{H}_L\,|0\rangle_L = |+\rangle_L := \frac{|0\rangle_L + |1\rangle_L}{\sqrt{2}}\,, \\ \mathsf{H}_L\,|1\rangle_L = |-\rangle_L := \frac{|0\rangle_L - |1\rangle_L}{\sqrt{2}}\,, \end{aligned} \tag{5.5}$$

and it transforms the encodes Pauli gates as

$$\mathsf{H}_L\,\mathsf{X}_L = \mathsf{Z}_L\,\mathsf{H}_L\,, \qquad \mathsf{H}_L\,\mathsf{Y}_L = -\mathsf{Y}_L\,\mathsf{H}_L\,, \qquad \mathsf{H}_L\,\mathsf{Z}_L = \mathsf{X}_L\,\mathsf{H}_L\,. \tag{5.6}$$

One can verify that H_L is a legitimate encoded operation by comparing Eqs. (5.6)

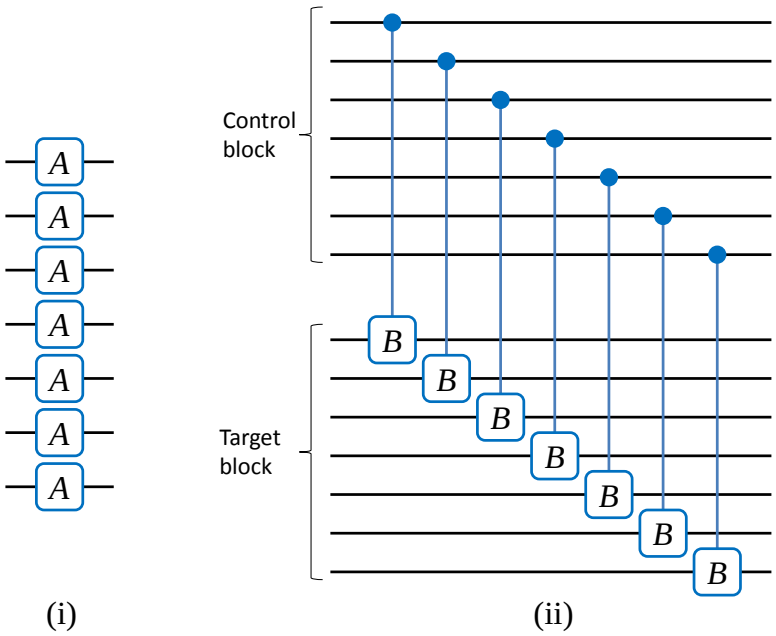


Figure 5.2: Transversal implementation of the Clifford gates on qubits encoded in the Steane code. In (i), when a single-qubit gate A belongs to the set $\{X, Y, Z, H, F\}$ of gates then this circuit realizes the corresponding set $\{\mathsf{X}_L, -\mathsf{Y}_L, \mathsf{Z}_L, \mathsf{H}_L, -\mathsf{F}_L\}$ of encoded operations. In (ii), when a single-qubit gate B belongs to the set $\{X, Z, H\}$ of gates then the circuit implements the corresponding set $\{\mathsf{CNOT}_L, \mathsf{CZ}_L, \mathsf{CH}_L\}$ of encoded gates. Here, every control is set to $|1\rangle$.

and Eq. (3.19) (or Eqs. (5.6)).

Likewise, the bitwise application of the gate $F^\dagger = R_z(-\frac{1}{2}\pi)$ [see Eq. (2.55)] to each one of the physical qubit [see Fig. 5.2(i)] realizes the encoded $\pi/2$-phase gate $\mathsf{F}_L := \left[F^\dagger\right]^{\otimes 7}$. This encoded gate brings the following transformations to the logical kets

$$\begin{aligned} \mathsf{F}_L\,|0\rangle_L &= \ \ |0\rangle_L\,, \\ \mathsf{F}_L\,|1\rangle_L &= i|1\rangle_L\,. \end{aligned} \tag{5.7}$$

Since we know that every ket in the superposition of $|0\rangle_L$ and $|1\rangle_L$ has $0(\mathrm{mod}\ 4)$ and $3(\mathrm{mod}\ 4)$ number of ones, respectively, it is not difficult to understand Eqs. (5.7). Furthermore, the encoded $\pi/2$-phase gate transforms the encoded Pauli gates in

the same was as given in Eq. (3.20):

$$\mathsf{F}_L \mathsf{X}_L = \mathsf{Y}_L \mathsf{F}_L \,, \qquad \mathsf{F}_L \mathsf{Y}_L = -\mathsf{X}_L \mathsf{F}_L \,, \qquad \mathsf{F}_L \mathsf{Z}_L = \mathsf{Z}_L \mathsf{F}_L \,. \tag{5.8}$$

We can attain other single-qubit gates of the Clifford group by combining the H_L and F_L gates[6].

Let us now come to design encoded two-qubit gates. Here, we have to be careful about the spread of existing errors. If an error occurs in one qubit which afterwards interacts with another qubit through a two-qubit gate, then the error is likely to spread to the second qubit. For example, in the case of two-qubit CNOT gate [of Eq. (2.39)], if a bit-flip error (X) occurs in the control qubit, before the operation, then the error propagates to both the control and target qubits. Similarly, if a phase-flip error (Z) occurs in the target qubit then the error propagates to the two qubits. This fact can be verified by Eq. (3.16). The spread of existing errors also occurs in the case of other two-qubit gates such as the CZ and controlled-Hadamard[7] (CH) gates.

In the case of the Steane code, the encoded two-qubit CNOT_L gate can be archived by applying bitwise seven CNOT gates between each qubit of the control block and the corresponding qubit of the target block according to Fig. 5.2(ii). This transversal implementation does not introduce more than one error per block, and the Steane code can handle one error per block. Similarly, we can realize the encoded two-qubit gates CZ_L and CH_L.

[6] In the case of the 7-qubit code, the transversal implementation of H_L and F_L is possible because this quantum code is derived from a *punctured doubly-even self-dual* classical code (the Hamming 7-bit code). In the case of a binary doubly-even self-dual code, the number of ones in all codewords is divisible by 4. Reader can find details about these properties—doubly-even and self-dual—in Refs. [25, 27].

[7] $\mathrm{CH}(\mathsf{a},\mathsf{b}) := |0\rangle_{\mathsf{a}}\langle 0| \otimes I^{(\mathsf{b})} + |1\rangle_{\mathsf{a}}\langle 1| \otimes H^{(\mathsf{b})}$.

Thus these encoded gates can be written as

$$\mathsf{CNOT}_L(\mathsf{a},\mathsf{b}) := \otimes_{\mathsf{k}=1}^{7}\left[\mathrm{CNOT}(\mathsf{k}_\mathsf{a},\mathsf{k}_\mathsf{b})\right], \tag{5.9}$$

$$\mathsf{CZ}_L(\mathsf{a},\mathsf{b}) := \otimes_{\mathsf{k}=1}^{7}\left[\mathrm{CZ}(\mathsf{k}_\mathsf{a},\mathsf{k}_\mathsf{b})\right], \tag{5.10}$$

$$\mathsf{CH}_L(\mathsf{a},\mathsf{b}) := \otimes_{\mathsf{k}=1}^{7}\left[\mathrm{CH}(\mathsf{k}_\mathsf{a},\mathsf{k}_\mathsf{b})\right], \tag{5.11}$$

where label a (b) stands for the control (target) block, and label k stands for the physical qubit of a block. The $\mathsf{CNOT}_L(\mathsf{a},\mathsf{b})$ gate transforms the encoded Pauli operators in the following way

$$\begin{aligned}
\mathsf{CNOT}_L(\mathsf{a},\mathsf{b})\,\mathsf{X}_L^{(\mathsf{a})} &= \mathsf{X}_L^{(\mathsf{a})}\mathsf{X}_L^{(\mathsf{b})}\,\mathsf{CNOT}_L(\mathsf{a},\mathsf{b}),\\
\mathsf{CNOT}_L(\mathsf{a},\mathsf{b})\,\mathsf{X}_L^{(\mathsf{b})} &= \mathsf{X}_L^{(\mathsf{b})}\,\mathsf{CNOT}_L(\mathsf{a},\mathsf{b}),\\
\mathsf{CNOT}_L(\mathsf{a},\mathsf{b})\,\mathsf{Z}_L^{(\mathsf{a})} &= \mathsf{Z}_L^{(\mathsf{a})}\,\mathsf{CNOT}_L(\mathsf{a},\mathsf{b}),\\
\mathsf{CNOT}_L(\mathsf{a},\mathsf{b})\,\mathsf{Z}_L^{(\mathsf{b})} &= \mathsf{Z}_L^{(\mathsf{a})}\mathsf{Z}_L^{(\mathsf{b})}\,\mathsf{CNOT}_L(\mathsf{a},\mathsf{b}),
\end{aligned} \tag{5.12}$$

and the $\mathsf{CZ}_L(\mathsf{a},\mathsf{b})$ gate brings the following transformations

$$\begin{aligned}
\mathsf{CZ}_L(\mathsf{a},\mathsf{b})\,\mathsf{X}_L^{(\mathsf{a})} &= \mathsf{X}_L^{(\mathsf{a})}\mathsf{Z}_L^{(\mathsf{b})}\,\mathsf{CZ}_L(\mathsf{a},\mathsf{b}),\\
\mathsf{CZ}_L(\mathsf{a},\mathsf{b})\,\mathsf{X}_L^{(\mathsf{b})} &= \mathsf{Z}_L^{(\mathsf{a})}\mathsf{X}_L^{(\mathsf{b})}\,\mathsf{CZ}_L(\mathsf{a},\mathsf{b}),\\
\mathsf{CZ}_L(\mathsf{a},\mathsf{b})\,\mathsf{Z}_L^{(\mathsf{a})} &= \mathsf{Z}_L^{(\mathsf{a})}\,\mathsf{CZ}_L(\mathsf{a},\mathsf{b}),\\
\mathsf{CZ}_L(\mathsf{a},\mathsf{b})\,\mathsf{Z}_L^{(\mathsf{b})} &= \mathsf{Z}_L^{(\mathsf{b})}\,\mathsf{CZ}_L(\mathsf{a},\mathsf{b}).
\end{aligned} \tag{5.13}$$

Validity of the CNOT_L and CZ_L gates can be proved by comparing Eq. (5.12) with Eq. (3.16) and Eq. (5.13) with Eq. (4.9), respectively. Similarly, one can also prove that the CH_L gate is a consistent encoded operation.

So far each one of the encoded gates (the Pauli and Clifford gates) we presented is fault tolerant [25, 26, 27, 28], but they are not sufficient for universal quantum computation. To obtain a universal set of fault-tolerant gates, we present the en-

coded rotation $\mathsf{U}^{12\cdots n}_{zz\cdots z}(\theta)_L$ in the next section.

5.3 Encoded gates on n logical qubits

In this section, we describe a procedure to achieve within the HQCM the encoded n-qubit rotation around the z axis,

$$\mathsf{U}^{12\cdots n}_{zz\cdots z}(\theta)_L := \cos\big(\tfrac{1}{2}\theta\big)\mathsf{I}_L^{\otimes n} - i\sin\big(\tfrac{1}{2}\theta\big)\mathsf{Z}_L^{\otimes n}\,, \tag{5.14}$$

where I_L is the encoded identity operation $I^{\otimes 7}$. Note that the superscripts $12\cdots n$, here, symbolize the logical qubits. Implementation of the n-qubit rotation is accomplished by the same procedure as given in Sec. 3.1.5.

First, the ancilla block (of seven physical qubits) a is initialized in the logical ket $|0\rangle_L$ with the aid of the encoding circuit shown in Fig. 5.1. Then, we perform H_L on $|0\rangle_L$ to get the ket $|+\rangle_L$ [see Eq. (5.5)]. In Fig. 5.3, the encoding circuit is displayed by the large rounded rectangle, and the H_L gate is represented by the H gates on the physical qubits of the ancilla.

To execute the encoded gate $\mathsf{U}^{12\cdots n}_{zz\cdots z}(\theta)_L$ on a general n-qubit encoded state $|\psi_{\rm in}(n)\rangle_L$, we have to entangle the input quantum register of the n logical qubits to the ancilla a. To do so we perform n CZ_L operations between the ancilla qubit a and every logical qubit. In the quantum circuit of Fig. 5.3, only the gate $\mathsf{CZ}_L(a,1) = \otimes_{k=1}^{7}\big[\mathrm{CZ}(k_a,k_1)\big]$ is shown. Furthermore, all the CZ_L gates can be performed in a single shot, because they all commute with each other.

Consequently, we get the required star-graph state,

$$|\phi(1+n)\rangle_L = \frac{1}{\sqrt{2}}\big[|0\rangle_L\otimes|\psi_{\rm in}(n)\rangle_L + |1\rangle_L\otimes\big(\mathsf{Z}_L^{\otimes n}|\psi_{\rm in}(n)\rangle_L\big)\big]\,, \tag{5.15}$$

of $1+n$ qubits. The associated star graph is shown in Fig. 5.3, where the input quantum register of the n logical qubits is displayed with circles, the ancilla a with

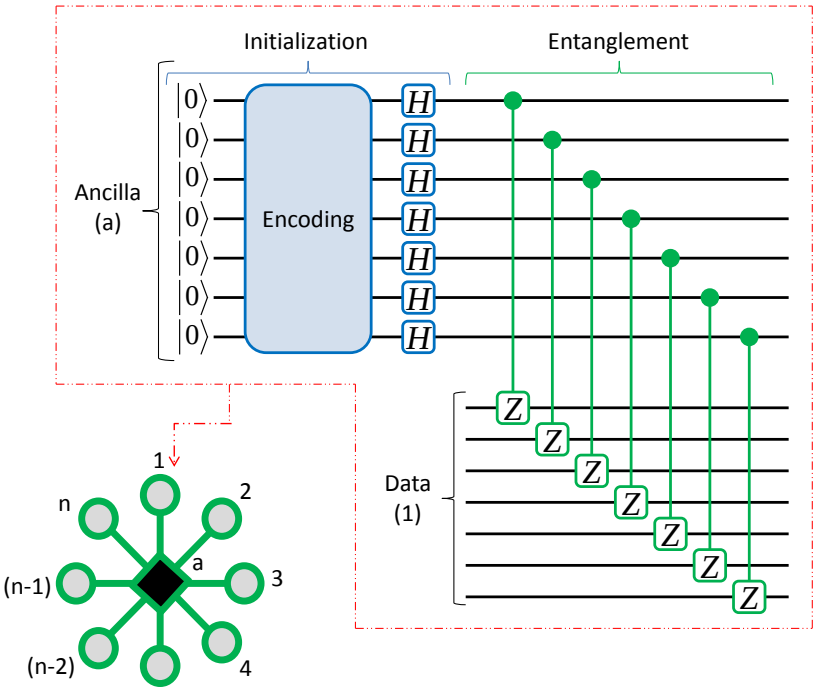


Figure 5.3: The star graph at the bottom left corner of this figure corresponds to the encoded state $|\phi(1+\mathsf{n})\rangle_L$. In the graph, the circles represent the n logical qubits, the bonds represent the CZ_L gates, and the diamond represents the ancilla qubit a. The circuit enclosed in dotted red lines represents the initialization of the ancilla qubit in the ket $|+\rangle_L$ followed be the encoded $\mathsf{CZ}_L(\mathsf{a},1)$ gate. This encoded gate is performed by the seven CZ gates [see Eq. (5.10)].

a diamond, and the CZ_L gates with green bonds.

First, let us note that the procedure to perform measurement given below is not fault tolerant and would therefore require further investigation. After preparing the graph state, we only decode the ancilla block. The decoding can be perform by running the circuit of Fig. 5.1 in the reverse order. As a result, the ket $|\phi(1+\mathsf{n})\rangle_L$ is transformed into the ket

$$|\tilde{\phi}(1+\mathsf{n})\rangle_L = \frac{1}{\sqrt{2}}\big[|0000000\rangle \otimes |\psi_{\rm in}(\mathsf{n})\rangle_L + |1000000\rangle \otimes \big(\mathsf{Z}_L^{\otimes \mathsf{n}}|\psi_{\rm in}(\mathsf{n})\rangle_L\big)\big]\,, \quad (5.16)$$

Like in Sec. 3.1.5, a projective measurement on the first physical qubit of the ancilla block in the basis $\mathcal{B}_{\theta,\frac{\pi}{2}}$ [see Eq. (2.16)] transforms the input ket into the output ket

$$|\psi_{\rm out}(\mathsf{n})\rangle_L = (\mathsf{Z}_L^{\otimes \mathsf{n}})^{m_{\mathsf{a}}}\, \mathsf{U}^{12\cdots \mathsf{n}}_{zz\cdots z}(\theta)_L |\psi_{\rm in}(\mathsf{n})\rangle_L\,. \quad (5.17)$$

Here, $m_{\mathsf{a}} \in \{0,1\}$ is the measurement outcome, and $(\mathsf{Z}_L^{\otimes \mathsf{n}})^{m_{\mathsf{a}}}$ is the by-product operator. After the measurement, the bonds [illustrated in Fig. 5.3] are broken, and the first qubit of the ancilla block is projected either onto the ket $|\uparrow(\theta, \frac{1}{2}\pi)\rangle_{\mathsf{a}}$ (if $m_{\mathsf{a}} = 0$) or onto the ket $|\downarrow(\theta, \frac{1}{2}\pi)\rangle_{\mathsf{a}}$ (if $m_{\mathsf{a}} = 1$). Every other physical qubit of the ancilla is already in the ket $|0\rangle$. Thus the implementation of the encoded n-qubit rotation is achieved within the HQCM.

With the same procedure, we can also get the encoded operations

$$\mathsf{U}_{xx\cdots x}^{12\cdots \mathsf{n}}(\theta)_L := \cos\left(\tfrac{1}{2}\theta\right)\mathsf{I}_L^{\otimes \mathsf{n}} - i\sin\left(\tfrac{1}{2}\theta\right)\mathsf{X}_L^{\otimes \mathsf{n}} \tag{5.18}$$

and

$$\mathsf{U}_{HH\cdots H}^{12\cdots \mathsf{n}}(\theta)_L := \cos\left(\tfrac{1}{2}\theta\right)\mathsf{I}_L^{\otimes \mathsf{n}} - i\sin\left(\tfrac{1}{2}\theta\right)\mathsf{H}_L^{\otimes \mathsf{n}} \tag{5.19}$$

just by replacing CZ_L with CNOT_L and CH_L, respectively. The encoded operation $\mathsf{U}_{HH\cdots H}^{12\cdots \mathsf{n}}(\theta)_L$ is useful to construct the test states [see Appendix C] employed in Chapter 7 for a quantum search. In the case of $\mathsf{n} = 1$ and $\theta = \frac{1}{4}\pi$, the encoded rotation $\mathsf{U}_{zz\cdots z}^{12\cdots \mathsf{n}}(\theta)_L$ becomes the encoded single-qubit $\pi/4$-phase gate

$$\mathsf{D}_L := \exp\left(-i\frac{\pi}{8}\mathsf{Z}_L\right) \tag{5.20}$$

This gate together with certain Clifford gates constitutes a discrete set of universal gates $\{\mathsf{D}_L, \mathsf{F}_L, \mathsf{H}_L, \mathsf{CZ}_L\}$ [28].

Let us now briefly comment on the classical information processing of HQCM, which fits perfectly in the picture of quantum error-correcting code. Indeed, the structure of by-product operator

$$\boldsymbol{U}_B(\tau) = \prod_{\mathsf{j}=1}^{\mathsf{n}} (\mathsf{X}_L^{(\mathsf{j})})^{x_{\mathsf{j}}(\tau)} (\mathsf{Z}_L^{(\mathsf{j})})^{z_{\mathsf{j}}(\tau)} \tag{5.21}$$

will stay the same as given by Eq. (4.1). Here, only the single-qubit Pauli operators

X and Z are replaced by their respective encoded operators X_L and Z_L. While, the classical entries x_j and z_j for jth logical qubit remain unchanged. Consequently, one can still work with the same $2\mathsf{n}$-component information flow vector given by Eq. (3.21). Furthermore, the propagation relations for the H_L, F_L, and CZ_L are given by Eqs. (5.6), (5.8), and (5.13), respectively. They are of the same form as given by Eqs. (3.19), (3.20), and (4.9). Also, one can obtain the same propagation relation for $\mathsf{U}^{12\cdots\mathsf{n}}_{zz\cdots z}(\theta)_L$ as given by Eq. (4.15). Hence, we can use the same $2\mathsf{n} \times 2\mathsf{n}$ propagation matrices [defined by Eqs. (3.30), (3.31), (4.13), and (4.17)] for the classical information processing in a fault-tolerant HQCM.

This completes the first step towards the establishment of a fault-tolerant HQCM. To make the picture of fault tolerance complete, we would have to implement the $\mathsf{U}^{12\cdots\mathsf{n}}_{zz\cdots z}(\theta)_L$ gate in a fault-tolerant manner. In other words, we would have to perform the encoding, decoding[8] and measurement in a fault-tolerant manner. This will be the subject of further studies.

[8] We know that the quantum circuit for encoding/decoding given in Fig. 5.1 is not fault tolerant. In the implementation of $\mathsf{U}^{12\cdots\mathsf{n}}_{zz\cdots z}(\theta)_L$ given above, the same circuit is used to prepare the ancilla qubit in the logical ket $|0\rangle_L$ and, before the measurement, to decode the ancilla qubit.

Part II

Test-State Approach to the Quantum Search

Chapter 6

Search problem

Suppose someone gives us a list of one hundred names of different animals on a piece of paper, and ask where "Lion" appears on this list. If "Lion" appears exactly once on the list, and the list is not ordered in any obvious way, then we have to go through about fifty names on average before we find "Lion." For a search of this kind, neither a CC nor a QC can directly helps us, because the data (names) are given on a piece of paper.

In order to make use of a CC or a QC for this kind of database search, first we have to convert the data into an accessible format. For example, in case of a CC for such a search, first we have to load the data (the given list) into the memory of a CC. However, we can find the name Lion in the process of converting the list of names into an electronic format (in terms of strings of bits) and storing them in the memory. So, neither a CC nor a QC is very helpful for a search of this kind. In other words, a CC (QC) is helpful for a database search only when the database is given in an electronic format (quantum format).

Similarly, a QC cannot search a classical database without a "quantum addressing scheme" [1] where the classical database is converted into a quantum format (in terms of quantum kets). So, the process of searching a marked string of bits with a CC in a classical database which is stored in the memory of a given computer is

called as *classical search* [see Sec. 6.1]. Likewise, *quantum search* [see Sec. 6.2] is a process where a QC searches a marked quantum state (or, rather, a particular unitary operation) out of a given set of quantum states (or, rather, a given set of unitary operations). Classical and quantum searches are analogous but not the same, their detailed description is given in the following sections.

6.1 Classical search and classical algorithm

Suppose we have an unsorted database as a set

$$\mathcal{S}_C^N := \{0, \cdots, j, \cdots, N-1\} \tag{6.1}$$

of a total of N items stored electronically in the memory of a given CC. Each item is labeled by an index from 0 to $N-1$ and further represented by a n-bit string in binary representation [see Eq. (2.49)]. Only one particular item matches with our query, and the task of the search problem is to recover the corresponding index (n-bit string) to the marked item at the end of computation. For convenience, the case of $N = 2^{\mathsf{n}}$ ($0 \equiv 00\cdots0, N-1 \equiv 11\cdots1$) is taken here, but the following algorithms and the test-state approach given in Chapter 7 can be applied for an arbitrary value of N.

The method employed by a CC to solve the search problem is to check every element of $\mathcal{S}_C^N$ one by one in a sequence till a match is found [3]. A single iteration of this classical search algorithm is a three-step process given as follows:

Step 1: The CC picks a n-bit string at random from the set $\mathcal{S}_C^N$ as an input.

Step 2: The CC checks, whether or not this string matches with our query.

Step 3: It returns a "yes" or "no" answer to the question.

If the answer is "yes," then the CC stops the computation and produces the string

as the result, and the corresponding item will be the marked item. If the answer turns out to be "no," then the CC picks another string at random from the set $\mathcal{S}_C^N$ as an input, with items tested earlier excluded, and asks the same question given above in Step 2. If the answer is again "no," then the CC repeats the above iteration until it hits the marked item.

One of the main points in this classical algorithm is, "Every time the CC picks at random only one n-bit string, and its current guess does not depend on previous guesses" other than excluding them. In this way, a CC needs, on average, as many as

$$G_C(N) = \frac{N+1}{2} - \frac{1}{N} \tag{6.2}$$

queries of the database before it finds the matching item. This is an immediate consequence of the recurrence relation

$$G_C(N+1) = 1 + \frac{N}{N+1} G_C(N) \quad \text{for} \quad N > 1 \tag{6.3}$$

that commences with $G_C(1) = 0$.

Since $G_C(N) \propto N$ for $N \gg 1$, this classical search algorithm is *linear* in the number of candidate items. If, rather than being unstructured, the data were sorted beforehand, then the problem could be completed by a *binary search*[1] in approximately $\log_2(N)$ iterations [3].

6.2 Quantum search and Grover's algorithm

In this section, a quantum search problem analogous to the classical one and a brief description of GA [34] is provided. In the transition from classical to quantum, bits are replaced by qubits. So, for each index (n-bit string) j [see Eq. (2.49)] of $\mathcal{S}_C^N$ defined by Eq. (6.1) there exist a n-qubit quantum ket $|j\rangle$ [see Eq. (2.50)], the

[1] Binary search algorithm is a *divide and conquer search algorithm*, where the size of search space reduces into it half after each iteration.

so-called *index ket*. There is then a unitary operation $\mathcal{O}^j$—the jth *oracle*—which gives a conditional phase shift of π to the index ket $|j\rangle$ only,

$$\mathcal{O}^j := (-\mathbf{I})^{|j\rangle\langle j|} = \mathbf{I} - 2|j\rangle\langle j|\,, \tag{6.4}$$

where $\mathbf{I} = I^{\otimes n}$ is the identity operator in the N-dimensional Hilbert space.

One can define a quantum search problem analogous to the classical search problem of Sec. 6.1 in the following way: Suppose someone gives us a quantum black box, which implements one of these N different oracles, and asks us to find out which of the oracles is the case without actually opening the box and looking inside. Clearly, we are not using QC to search a marked item in a classical database, but we are searching the index ket corresponding to the given oracle. The question of how many queries of the database are now needed, reads "How many times must one use the quantum black box to find out the correct result?"

The most efficient way of finding out which oracle is the actual one is GA [64]. GA begins by applying the Hadamard gate H of Eq. (2.23) to each qubit, after initially preparing the state with index ket $|0\rangle \equiv |0\rangle^{\otimes \mathsf{n}}$. The operation $H^{\otimes \mathsf{n}}$ creates the input state

$$\begin{aligned} |\psi_0\rangle &= \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle \\ &= A_0\,|j\rangle + \frac{B_0}{\sqrt{N-1}} \sum_{l(\neq j)} |l\rangle \end{aligned} \tag{6.5}$$

with

$$A_0 = \frac{1}{\sqrt{N}} := \sin\big(\theta_N\big) \tag{6.6}$$

and

$$B_0 = \frac{\sqrt{N-1}}{\sqrt{N}} := \cos\big(\theta_N\big)\,. \tag{6.7}$$

The ket $|\psi_0\rangle$ is a superposition of all the index kets of the set $\mathcal{S}_Q^N$ of Eq. (2.1) with

equal amplitude $1/\sqrt{N}$. The next step is an application of the Grover iteration operation $\mathcal{G}$. Geometrically it is a rotation composed of two reflection operations as $\mathcal{G} = \mathcal{D}\mathcal{O}$. The operator $\mathcal{O}$ is the same quantum oracle (black box) defined by Eq. (6.4), whose unknown index we have to find. The diffusion operator $\mathcal{D}$ gives an inversion about the average [34],

$$\begin{aligned}\mathcal{D} &= 2|\psi_0\rangle\langle\psi_0| - \mathbf{I} \\ &= -H^{\otimes \mathsf{n}}\big(\mathbf{I} - 2|0\rangle\langle 0|\big)H^{\otimes \mathsf{n}}\,,\end{aligned} \tag{6.8}$$

its central piece is the 0th oracle $\mathcal{O}^0$.

If the black box implements the jth oracle, then after k applications of $\mathcal{G}$ the resultant state will be

$$|\psi_k\rangle = A_k\,|j\rangle + \frac{B_k}{\sqrt{N-1}}\sum_{l(\neq j)}|l\rangle \tag{6.9}$$

with

$$A_k = \left(1 - \frac{2}{N}\right)A_{k-1} + 2\frac{\sqrt{N-1}}{N}B_{k-1} = \sin\big((2k+1)\theta_N\big) \tag{6.10}$$

and

$$B_k = -2\frac{\sqrt{N-1}}{N}A_{k-1} + \left(1 - \frac{2}{N}\right)B_{k-1} = \cos\big((2k+1)\theta_N\big)\,. \tag{6.11}$$

GA is probabilistic in nature in the sense that, after applying $\mathcal{G}$ k times, the probability of the privileged index ket

$$p_k^{(N)} = \sin\big((2k+1)\theta_N\big)^2 \tag{6.12}$$

becomes significantly larger than the probability of other index kets. Upon opti-

mizing k, GA solves the quantum search problem by using the black box only

$$G_Q(N) = 0.69\sqrt{N} \tag{6.13}$$

times when $N \gg 1$ [see Sec. 7.3]. As a side remark, after $G_Q(N)$ number of Grover iterations the success probability $p_k^{(N)}$ goes down again. Finally, the output is read by performing projective measurements on each qubit, and so one of the index kets is obtained. Most likely the oracle associated with the final output index ket is the one which the black box is executing.

The quadratic speedup of $G_Q(N) \propto \sqrt{N}$ versus $G_C(N) \propto N$ is owed to the computational power of quantum physics; specifically, the superposition principle is at work. But, it is worth emphasizing that the outcome of GA is not guaranteed to be the correct answer; it can be incorrect with a probability of the order of $1/N$ for $N \gg 1$, that is very small but definitely nonzero.

In passing, one can make a note of the following. A general treatment of GA for multiple targets and for an arbitrary value of N is given in Ref. [65]. Moreover, GA is a special case of the quantum amplitude amplification [66] and can be used in the quantum counting problem [67] with the help of the discrete quantum Fourier transformation [33, 1]. In addition, one can get rid of the probabilistic nature of GA if one has the option of changing the structure of the diffusion operator $\mathcal{D}$ and the oracle $\mathcal{O}$ [68]. When one is only allowed to use the given black box, namely the oracles of Eq. (6.4), but not to open it up and change its setting, then GA remains probabilistic in nature.

So, one needs a confirmation step to be sure of the result obtained by GA. A single iteration of the test-state search introduced in the next chapter acts as a confirmation step for GA, where the verification matter is discussed after Eqs. (7.6) in Sec. 7.1. In the next section, the implementation of GA with the HQCM is given.

6.2.1 Grover's search algorithm within the HQCM

There have been many successful attempts to implement of GA [34], for $N = 4$ or $N = 8$, in different physical setups such as the NMR systems [48, 49, 50], cavity quantum electrodynamics [51, 52, 53], optics [54, 55]. Furthermore, the implementation of GA within the MQCM of Chapter 3 is demonstrated in Refs. [56, 57]. In this section, the implementation of GA—where the n-qubit $\Lambda^{12\cdots(\mathsf{n}-1)}Z^{(\mathsf{n})}$ and some single-qubit gates are sufficient—with the HQCM of Chapter 4 is illustrated.

Let us take a look at the structures of the two reflection operators—$\mathcal{O}^j$ of Eq. (6.4) and $\mathcal{D}$ of Eqs. (6.8)—of Grover iteration operator $\mathcal{G}$. If the given black box is executing only one out of $N = 2^{\mathsf{n}}$ different oracles, then the oracle for the case of $j = N - 1$ will be

$$\mathcal{O}^{N-1} = \mathbf{I} - 2|N-1\rangle\langle N-1| = \Lambda^{12\cdots(\mathsf{n}-1)}Z^{(\mathsf{n})}\,. \tag{6.14}$$

Its simulation with the HQCM for the case of $\mathsf{n} = 4$ is already discussed in Sec. 4.3.3. Furthermore, any other oracle can be derived by performing the single-qubit X gate(s) on the relevant qubit(s) before and after performing the gate $\Lambda^{12\cdots(\mathsf{n}-1)}Z^{(\mathsf{n})}$, as exemplified by

$$\mathcal{O}^0 = X^{\otimes\mathsf{n}}\left[\Lambda^{12\cdots(\mathsf{n}-1)}Z^{(\mathsf{n})}\right]X^{\otimes\mathsf{n}}\,. \tag{6.15}$$

Hence, the gate $\Lambda^{12\cdots(\mathsf{n}-1)}Z^{(\mathsf{n})}$ is a necessary part of the circuitry of the black box and is employed in every case of oracle. However, the answer to "Which of the oracle is executed by the black box?" is hidden from us. In other words, we do not know on which qubit(s) the black box implements the X gate(s) along with $\Lambda^{12\cdots(\mathsf{n}-1)}Z^{(\mathsf{n})}$.

Similarly, one can realize the diffusion operator $\mathcal{D}$ of Eqs. (6.8),

$$\mathcal{D} = -H^{\otimes\mathsf{n}}X^{\otimes\mathsf{n}}\left[\Lambda^{12\cdots(\mathsf{n}-1)}Z^{(\mathsf{n})}\right]X^{\otimes\mathsf{n}}H^{\otimes\mathsf{n}}\,, \tag{6.16}$$

by performing the Hadamard H and the Pauli X gates on every logical qubit before and after performing the gate $\Lambda^{12\cdots(n-1)}Z^{(n)}$. In conclusion, the gate $\Lambda^{12\cdots(n-1)}Z^{(n)}$ together with the Hadamard H and the Pauli X gates is sufficient to run GA in the HQCM.

Chapter 7

Test-state approach to the quantum search

The search for *a quantum needle in a quantum haystack*—quantum search [see Sec. 6.2]—is a metaphor for the problem of finding out which one of the permissible set of oracles [of Eq. (6.4)] is implemented by a given black box. GA [34] solves this problem with quadratic speedup [see Eq. (6.13)] as compared with the analogous search for *a classical needle in a classical haystack*—classical search [see Sec. 6.1]. Since the outcome of GA is probabilistic—it gives the correct answer with a high probability, not with certainty—the answer requires verification. For this purpose, specific test states [42] are introduced, one test state for each oracle. These test states can also be used to realize *a classical search for the quantum needle* which is deterministic—it always gives the correct answer after a finite number of iterations—and 3.41 times faster than the purely classical search of Sec. 6.1. Since the test-state search [42] and GA look for the same quantum needle, the average number of oracle queries of the test-state search is the classical benchmark for GA.

Features of both the classical and quantum approaches are embodied in this approach. A *single iteration* in the test-state approach [see Sec. 7.1] can be summarized in the following three steps.

Step 1: An index ket $|j\rangle$ from the set $\mathcal{S}_Q^N$ [see Eq. (2.1)] is picked, and the corresponding test state is prepared.

Step 2: The test state is then passed through the given quantum black box which is executing one of the oracles of Eq. (6.4).

Step 3: The available information is extracted from the processed test state with the help of a POM[1] [58, 59].

Here, a single iteration comprises of these three steps, which are similar to the classical search algorithm of Sec. 6.1. As is the case in the classical search, the result of the POM gives an answer to the same question—whether or not the black box is executing the oracle $\mathcal{O}^j$—in terms of "yes" or "no." The answer "yes" tells us that the black box is executing the corresponding oracle to the index ket we have picked and terminates the search.

Even if the answer is "no," the result of the POM gives us some information about the actual oracle. This information facilitates an educated guess and a judicious choice of the test state for the next iteration.

The correct result is obtained after a finite number of iterations. In other words, the test-state search is deterministic, rather than probabilistic. And, the systematic educated guessing makes the test-state search more efficient than a truly classical search, in which all the test states would be chosen at random: For $N \gg 1$, the test-state search needs fewer guesses by a factor of $1/3.41 = 0.293$.

The structure of this chapter is as follows. A comprehensive description of the test-state approach to the quantum search problem is provided in Sec. 7.1 and Sec. 7.2. GA with test-state verification is then discussed in Sec. 7.3, and Sec. 7.4 deals with alternative test-state search strategies. In Sec. 7.5, the quantum circuits for the construction of the test states and for realizing the measurements are described.

[1] A discussion on POM is given in Postulate 3 of Sec. 2.1.2.

7.1 A single iteration in the test-state approach

In this section, the construction of the test states for the verification of the outcome of GA and the three steps of one iteration round in the test-state approach for determining the oracle of the quantum search problem are discussed. The narrative follows the steps in sequence.

Step 1—Preparing the test state: An index ket $|j\rangle$ from the set $\mathcal{S}_Q^N$ given by Eq. (2.1) is picked. For the very first round of iteration, the choice of $|j\rangle$ is random, but for all subsequent rounds the choice is dictated by the result of the measurement in Step 3, as discussed in Sec. 7.2.

Then the corresponding test state $|t_j\rangle$ of the form

$$|t_j\rangle := a|j\rangle + b \sum_{l(\neq j)} |l\rangle \tag{7.1}$$

is prepared, where a is the amplitude of the privileged index ket $|j\rangle$ and b is the common amplitude of all other index kets. Both a and b are functions of N; it suffices to consider only real positive values for a and b, but this is a restriction of convenience, not of necessity.

In Sec. 7.5.1, a method for constructing the test state $|t_0\rangle$ for the case of $N = 2^{\mathsf{n}}$ is presented. The n-qubit test state $|t_0\rangle$ can be transformed into any other test state $|t_j\rangle$ by applying the X operations on the relevant qubits. In other words, each $|t_j\rangle$ is equivalent to $|t_0\rangle$ up to some single-qubit operations.

Step 2—Processing the test state: The test state $|t_j\rangle$ is passed through the given quantum black box. As recalled in Sec. 6.2, the black box implements one out of N different oracles of Eq. (6.4); but we do not know which one oracle is the case. Hence, there is a *priori* probability $1/N$ for every oracle. If the black box

implements the jth oracle, then the resultant state is

$$\mathcal{O}^j|t_j\rangle = |t_j^j\rangle = -a|j\rangle + b|k\rangle + b\sum_{l(\neq j,k)}|l\rangle\,. \tag{7.2}$$

If the black box is not implementing the jth oracle, but some other one, the kth oracle, say, then the resultant state is

$$k \neq j\ :\quad \mathcal{O}^k|t_j\rangle = |t_j^k\rangle = a|j\rangle - b|k\rangle + b\sum_{l(\neq j,k)}|l\rangle\,. \tag{7.3}$$

Result $|t_j^j\rangle$ says "yes, it is the jth oracle" whereas each $|t_j^k\rangle$ with $k \neq j$ says "no, it is not the jth oracle." Note that there is one "yes" but $N-1$ different "no"s.

The "no" set $\mathcal{C}_j^N$ to the index ket $|j\rangle$ as the collection of all $N-1$ "no" states of Eq. (7.3) is defined as

$$\mathcal{C}_j^N := \left\{|t_j^0\rangle, \cdots, |t_j^{j-1}\rangle, |t_j^{j+1}\rangle, \cdots, |t_j^{N-1}\rangle\right\}. \tag{7.4}$$

In order to be able to completely discriminate the "yes" ket $|t_j^j\rangle$ from the "no" kets in $\mathcal{C}_j^N$, we demand that

$$\langle t_j^k|t_j^j\rangle = 0 \quad \text{for} \quad k \neq j\,, \tag{7.5}$$

so that the "yes" ket is orthogonal to all the "no" kets. Together with the normalization of the test-state ket $|t_j\rangle$, this gives

$$\begin{aligned} a &= \sqrt{\frac{N-3}{2N-4}}\,, \\ b &= \frac{1}{\sqrt{2N-4}}\,, \end{aligned} \tag{7.6}$$

for the amplitudes in Eq. (7.1).

The use of the test states for the verification of the outcome of GA, is quite obvious: After GA identifies the jth oracle, we prepare the jth test state $|t_j\rangle$ and

let the oracle act on it. Then we perform a measurement that determines whether the resulting ket is proportional to the "yes" ket $|t_j^j\rangle$ or resides in the orthogonal subspace spanned by the $N-1$ "no" kets. If we find the "yes" ket, the search is over; otherwise, we have to execute GA another time. An alternative confirmation step for GA, where one has to use the black box at most two times, is described in Appendix B.

As Eqs. (7.6) show, there are test states for $N > 2$, but none for $N = 2$. It is as it should be. Indeed, the two oracles $\mathcal{O}^0 = |1\rangle\langle 1| - |0\rangle\langle 0|$ and $\mathcal{O}^1 = |0\rangle\langle 0| - |1\rangle\langle 1|$ for $N = 2$ are simply indistinguishable; they do not tell the index kets $|0\rangle$ and $|1\rangle$ apart.

Turning our attention to the "no" kets, one can observe that they are the edges of a $(N-1)$-dimensional pyramid,

$$k \neq j,\ l \neq j\ :\quad \langle t_j^k | t_j^l \rangle = \lambda + (1-\lambda)\delta_{kl} \tag{7.7}$$

with the common overlap

$$\lambda = \frac{N-4}{N-2} \tag{7.8}$$

shared by each pair of "no" kets. In the terminology of Ref. [61], the pyramid is acute ($\lambda > 0$) for $N > 4$, orthogonal ($\lambda = 0$) for $N = 4$, and flat ($\lambda = -1$) for $N = 3$.

The case $N = 4$ is particular: we have $a = b = 1/2$ and all four test states are identical. The "no" states for one index ket are pairwise orthogonal; they are "yes" states for the other index kets. As a consequence, testing the oracle once with the one common test state will reveal its identity. This observation is sometimes stated as "GA needs to query the oracle only once for $N = 4$." Indeed, we have $p_1^{(4)} = 1$ in Eq. (6.12). This peculiarity of GA comes about because the common $N = 4$ test state is also the $N = 4$ initial state of GA, and the $N = 4$ version of the diffusion operator $\mathcal{D}$ of Eq. (6.8) maps the square-root measurement (SRM) kets of Eq. (7.18)

below onto the computational basis, in which the outcome of GA is obtained.

For the case of $N > 4$, the "no" kets are not orthogonal to each other ($\lambda > 0$), so the POM is a more efficient mean to obtain the information from the non-orthogonal quantum states [58, 59]. In addition, these "no" kets possess a symmetry in the sense that described in Ref. [60]. To understand this property, one can take the test state $|t_0\rangle$ and the corresponding set of "no" kets $\mathcal{C}_0^N$ as an example. There is then a corresponding unitary operation

$$\mathbf{P}_0 := |0\rangle\langle 0| + \sum_{l=1}^{N-2} |l+1\rangle\langle l| + |1\rangle\langle N-1| \tag{7.9}$$

which performs a cyclic permutation over a subset of the computational basis, and its periodicity is $N-1$: $(\mathbf{P}_0)^{N-1} = \mathbf{I}$. The symmetry possessed by the "no" kets of $\mathcal{C}_0^N$ can be stated as:

$$\mathbf{P}_0 \, |t_0^l\rangle = \begin{cases} |t_0^0\rangle & \text{for} \quad l = 0 \\ |t_0^{l+1}\rangle & \text{for} \quad 0 < l < N-1 \\ |t_0^1\rangle & \text{for} \quad l = N-1 \end{cases} \tag{7.10}$$

As we know that any test state $|t_j\rangle$ of Eq. (7.1) is locally equivalent to $|t_0\rangle$, so one can easily generalize this property for any other "no" set $\mathcal{C}_j^N$. In the case of symmetric non-orthogonal quantum state discrimination, the SRM is the optimum measurement in the sense of minimization of the average probability of error [60]. In the next step, therefore, the SRM are used to get the available information.

Step 3—Measuring the result: When measuring the state that results from applying the black-box oracle to the jth test state $|t_j\rangle$, we not only need to distinguish between "yes" and "no" but also want to acquire information about which of the "no"s is the case, so that we can make a judicious choice for the next test state. Thanks to the pyramidal structure of the "no" kets [61], the POM that maximizes our odds of guessing right is the SRM [60].

For $N = 3$, there is no useful POM of this kind because the two "no" states are the same, as is exemplified by $|t_0^1\rangle = -|t_0^2\rangle$. For $N > 3$, the SRM

$$\sum_{l=0}^{N-1} \Pi_j^l = \mathbf{I} \tag{7.11}$$

has the elements of POM $\Pi_j^l := |T_j^l\rangle\langle T_j^l|$ with

$$|T_j^l\rangle = \rho_j^{-1/2} |t_j^l\rangle\,, \tag{7.12}$$

where $\rho_j := \sum_{k=0}^{N-1} |t_j^k\rangle\langle t_j^k|$. In order to narrate the kets $|T_j^l\rangle$ in terms of the computational basis, one has to look into the structure of $\rho_j^{-1/2}$.

Diagonalization of the $N \times N$ density matrix[2],

$$\rho_j = |t_j^j\rangle\langle t_j^j| + \big(1 + (N-2)\lambda\big)|T\rangle\langle T| + \big(1-\lambda\big)|T^\perp\rangle\langle T^\perp|\,, \tag{7.13}$$

divides the N-dimensional space into three mutually orthogonal subspaces, which are defined by the eigenkets $\big\{|t_j^j\rangle, |T\rangle, |T^\perp\rangle\big\}$ and the corresponding eigenvalues $\big\{1, \big(1 + (N-2)\lambda\big), \big(1-\lambda\big)\big\}$ of ρ_j. Where, the eigenket

$$|T\rangle := \frac{1}{\sqrt{(N-1)\big(1 + (N-2)\lambda\big)}} \sum_{k(\neq j)} |t_j^k\rangle \tag{7.14}$$

is an equal superposition of all the "no" kets of $\mathcal{C}_j^N$, the projector onto its orthogonal subspace is $|T^\perp\rangle\langle T^\perp| + |t_j^j\rangle\langle t_j^j| = \mathbf{I} - |T\rangle\langle T|$, and λ is of Eq. (7.8). Hence, the required representation of $\rho_j^{-1/2}$,

$$\rho_j^{-1/2} = \tfrac{1}{\sqrt{1-\lambda}} \left[\mathbf{I} + \big(\sqrt{1-\lambda} - 1\big)\,|t_j^j\rangle\langle t_j^j| + \left(\tfrac{\sqrt{1-\lambda}}{\sqrt{1+(N-2)\lambda}} - 1\right)|T\rangle\langle T|\right], \tag{7.15}$$

[2]Note that the density matrix ρ_j is not normalized: $\mathrm{Tr}(\rho_j) = N$.

is obtained from Eq. (7.13). And, from Eqs. (7.5) and (7.7), we have:

$$\langle T|t_j^l\rangle = \begin{cases} 0 & \text{for } \; l = j \\ \sqrt{\frac{1+(N-2)\lambda}{N-1}} & \text{for } \; l \neq j \end{cases} \tag{7.16}$$

Considering Eqs. (7.15) and (7.16), the ket $|T_j^l\rangle$ of Eq. (7.12) is reviewed:

$$|T_j^l\rangle = \begin{cases} |t_j^j\rangle & \text{for } \; l = j \\ \frac{1}{\sqrt{1-\lambda}}\left[|t_j^l\rangle + \frac{\sqrt{1-\lambda}-\sqrt{1+(N-2)\lambda}}{\sqrt{N-1}}|T\rangle\right] & \text{for } \; l \neq j \end{cases} \tag{7.17}$$

The Final form of $|T_j^l\rangle$ in the computational basis follows form Eqs. (7.3) and (7.14):

$$|T_j^l\rangle = \begin{cases} -a|j\rangle + b|l\rangle + b\sum_{k(\neq j,l)}|k\rangle & \text{for } \; l = j \\ b|j\rangle - x|l\rangle + y\sum_{k(\neq j,l)}|k\rangle & \text{for } \; l \neq j \end{cases} \tag{7.18}$$

where

$$y = \frac{1+a}{N-1}\,, \quad x = 1-y \tag{7.19}$$

and a, b are the coefficients of Eqs. (7.1)-(7.3) and (7.6).

The set of kets $\left\{|T_j^{l\neq j}\rangle\right\}$ constitutes another acute pyramid of $N-1$ legs associated with the SRM [61] and, together with the ket $|T_j^j\rangle$, they form a complete basis for the SRM. Since

$$\langle T_j^k|T_j^l\rangle = \delta_{kl}\,, \tag{7.20}$$

the SRM is an orthogonal measurement, a standard von Neumann measurement, not a POM proper. Therefore, the SRM can be implemented by a unitary transformation followed by measuring the computational basis. One quantum circuit for such a unitary transformation is given in Sec. 7.5.2.

7.2 Conditional probabilities in the test-state approach

The probability of getting the lth outcome if the processed jth test state is $|t_j^k\rangle$ is given by

$$\text{prob}(t_j^k \to \Pi_j^l) = \langle t_j^k|\Pi_j^l|t_j^k\rangle = \left|\langle T_j^l|t_j^k\rangle\right|^2 . \tag{7.21}$$

It follows from Eqs. (7.2), (7.3), and (7.18) that there are three cases,

$$\text{prob}(t_j^k \to \Pi_j^l) = \begin{cases} 1 & \text{for } \; k = j, \; l = k \\ \alpha_{N-1} & \text{for } \; k \neq j, \; l = k \\ \beta_{N-1} & \text{for } \; k \neq j, \; l \neq k \end{cases} \tag{7.22}$$

where

$$\begin{aligned} \beta_{N-1} &= \frac{1}{(N-1)^2}\left(\sqrt{N-3} - \frac{\sqrt{2}}{\sqrt{N-2}}\right)^2 , \\ \alpha_{N-1} &= 1 - (N-2)\beta_{N-1} \\ &= \frac{1}{(N-1)^2}\left(\sqrt{N-3} + \sqrt{2N-4}\right)^2 , \end{aligned} \tag{7.23}$$

with the subscript $N-1$ stating the number of different "no" outcomes. The first case ($k = j$) in Eq. (7.22) is the affirmative "yes, it is the jth oracle" answer that terminates the search. The second and third cases ($k \neq j$) both say "no, it is not the jth oracle."

"Yes" answer: After the SRM, if the outcome is $|T_j^j\rangle = |t_j^j\rangle$, then our choice, the index ket $|j\rangle$, was right. Consequently, we can say for sure that the black box executes the oracle $\mathcal{O}^j$. Afterwards, the search is over, and we can stop the computation. This fact is stated in the first condition ($k = j, l = k$) of Eq. (7.22).

"No" answer: If the measurement result turns out to be the ket $|T_j^{l\neq j}\rangle$, then certainly the black box does not execute the oracle $\mathcal{O}^j$. And, we will therefore

guess that the black box contains the lth oracle and choose $|t_l\rangle$ as the next test state; where α_{N-1} of Eqs. (7.23)—present in the second condition ($k \neq j, l = k$) of Eq. (7.22)—is the probability of guessing right. But it is possible that the test state $|t_l\rangle$ also turns out to be a wrong choice. In other words, the black box could execute one of the other $N-2$ oracles corresponding to the other $N-2$ "no" outcomes. So, β_{N-1} of Eqs. (7.23) is the probability of guessing wrong, which appears in the third condition ($k \neq j, l \neq k$) of Eq. (7.22).

The SRM maximizes the probability of guessing right. Thereby, the probability α_{N-1} of getting the lth outcome when the black box implements the lth oracle ($l = k$) is larger than the probability β_{N-1} for all other $N-2$ "no" outcomes ($l \neq k$).

After the first wrong guess $|j\rangle$, we exclude the index ket $|j\rangle$ from the list of candidates, and have the set

$$\mathcal{S}_Q^{N-1} = \big\{|0\rangle, \cdots, |j-1\rangle, |j+1\rangle, \cdots, |N-1\rangle\big\} \tag{7.24}$$

of the remaining $N-1$ index kets for the next round. Having found the SRM outcome Π_j^l, we repeat the iteration described in Sec. 7.1 on the set $\mathcal{S}_Q^{N-1}$ by taking the index ket $|l\rangle$ as the next educated guess, for which the "no" probabilities are α_{N-2} and β_{N-2}. If this guess is also wrong, then the lth index ket can be excluded as well, and we are left with $N-2$ candidates and a new educated guess for the next test state with "no" probabilities α_{N-3} and β_{N-3}, and so forth, until we either get the "yes" answer, or we are left with four candidates only, having excluded $N-4$ index kets successively. The common test state for $N = 4$ will then surely give us the "yes" answer; in the present context, this is confirmed by $\alpha_3 = 1$ and $\beta_3 = 0$ in Eqs. (7.22) and (7.23).

In each round of iteration in the test-state search, we are using the given black box once. Accordingly, the average number of oracle queries before a "yes" answer

is obtained, is given by

$$G_T(N) = p_1^{(N)} + 2\,p_2^{(N)} + 3\,p_3^{(N)} + \cdots + (N-3)\,p_{N-3}^{(N)} \tag{7.25}$$

where $p_m^{(N)}$ is the probability that the search terminates after the mth round with the test-state approach[3]. For $N > 4$, these probabilities are

$$\begin{aligned}
p_1^{(N)} &= \frac{1}{N}\,, \\
p_2^{(N)} &= \left(1 - p_1^{(N)}\right)\alpha_{N-1} \\
&= \frac{N-1}{N}\,\alpha_{N-1}\,, \\
p_3^{(N)} &= \left(1 - p_1^{(N)} - p_2^{(N)}\right)\alpha_{N-2} \\
&= \frac{N-1}{N}(1-\alpha_{N-1})\,\alpha_{N-2}\,, \\
&\vdots \\
p_{N-4}^{(N)} &= \left(1 - p_1^{(N)} - p_2^{(N)} - \cdots - p_{N-5}^{(N)}\right)\alpha_5 \\
&= \frac{N-1}{N}(1-\alpha_{N-1})\cdots(1-\alpha_6)\,\alpha_5\,, \\
p_{N-3}^{(N)} &= 1 - p_1^{(N)} - p_2^{(N)} - \cdots - p_{N-4}^{(N)} \\
&= \frac{N-1}{N}(1-\alpha_{N-1})\cdots(1-\alpha_6)(1-\alpha_5)\,.
\end{aligned} \tag{7.26}$$

Without the educated guesses provided by the SRM, one would have to resort to choosing the test state for the next iteration at random, just as one does in a purely classical search, which amounts to the replacement $\alpha_L \to 1/L$ and yields $p_1^{(N)} = p_2^{(N)} = \cdots = p_{N-4}^{(N)} = 1/N$, $p_{N-3}^{(N)} = 4/N$. But with the systematic educated guesses, we have

$$\alpha_L \approx \frac{3+\sqrt{8}}{L} \quad \text{for } L \gg 1\,, \tag{7.27}$$

and the probabilities for early termination are substantially larger than $1/N$.

[3]Note that this probability $p_m^{(N)}$ is different from the probability $p_k^{(N)}$ given for GA in Eq. (6.12).

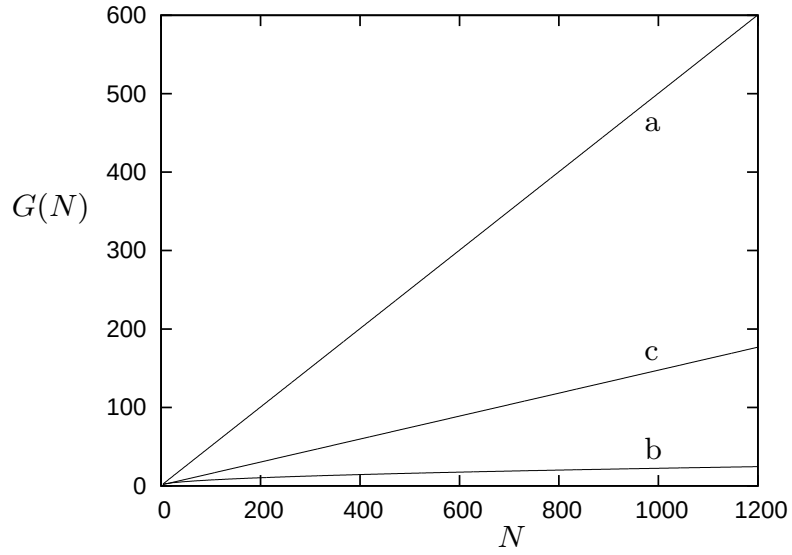


Figure 7.1: Average number $G(N)$ of oracle queries as a function of the total number N of index kets. Curve "a" shows $G_C(N)$ of Eq. (6.2) for the classical search strategy. Curve "b" shows $G_Q(N)$ of Eq. (7.34) for Grover's search algorithm, supplemented by test-state verification and optimized for least number of queries per search cycle. Curve "c" shows $G_T(N)$ of Eq. (7.28) for the test-state search.

Equations (7.25) and (7.26) yield the recurrence relation

$$G_T(N+1) = 1 + \frac{N}{N+1}(\alpha_N - \beta_N) + \frac{N^2\beta_N}{N+1}G_T(N)\,, \tag{7.28}$$

which commences with $G_T(4) = 1$ and reduces, as it should, to its $G_C(N)$ analog in Eq. (6.3) for $\alpha_N = \beta_N = 1/N$. With the aid of the large-L form of α_L in Eq. (7.27) and the corresponding statement for β_L, we then find that the average number of queries in the test-state search is given by

$$G_T(N) \approx \frac{N}{4+\sqrt{8}} = \frac{N}{6.83} \quad \text{for} \;\; N \gg 1\,. \tag{7.29}$$

The comparison with the classical search,

$$\frac{G_T(N)}{G_C(N)} \approx \frac{1}{2+\sqrt{2}} = \frac{1}{3.41} = 0.293 \quad \text{for} \;\; N \gg 1, \tag{7.30}$$

shows that the judicious choice of the next test state has a substantial pay-off: We need much fewer queries.

Since the test-state search and GA both determines the actual oracle inside the given quantum black box, the classical-type "yes/no" approach of the test-state search sets the benchmark for the quantum search with GA. It is true, that both $G_C(N)$ and $G_T(N)$ grow linearly with the number N of candidate items, whereas $G_Q(N)$ grows proportional to $\sqrt{N}$—and this quadratic speed-up is, of course, the striking advantage of the quantum search algorithm—but the reduction of the average number of queries by the factor of 0.293 is truly remarkable by itself. It, too, is a benefit of the superposition principle. Furthermore, if we apply the test-state approach to the quantum search problem of Sec. 6.2, although it takes more steps than GA, it definitely provides us the correct result. So, there is no need to run it again. The three search strategies are compared in Fig. 7.1, which shows $G_C(N)$, $G_T(N)$, and $G_Q(N)$ as functions of N.

7.3 Grover's search algorithm with test-state verification

As recalled in Sec. 6.2 above, a single GA cycle consists of the preparation of the initial state, k applications of $\mathcal{G} = \mathcal{D}\mathcal{O}$, followed by a measurement in the computational basis that is composed of the index kets. If the measurement outcome corresponds to the index ket $|j\rangle$, then we apply the oracle to test state $|t_j\rangle$, measure the resulting state with the SRM, and so decide whether the actual oracle is $\mathcal{O}^j$ or not. The search terminates when this test says "yes." But if the reply is "no," we execute another GA cycle.

The probability that a GA cycle finds the correct index state is $p_k^{(N)}$ of Eq. (6.12). It follows that the probability that the search terminates after the mth cycle is

$$\big(1 - p_k^{(N)}\big)^{m-1} p_k^{(N)} = \cos\big((2k+1)\theta_N\big)^{2(m-1)} \sin\big((2k+1)\theta_N\big)^2 \tag{7.31}$$

for $m = 1, 2, 3, \ldots$.

Each cycle queries the oracle k times, once for each application of $\mathcal{G}$, plus one more time during the test-state verification. The verification is only done, however, if the result of the GA cycle is not an index state to an oracle that is already known to be wrong from the verification step of an earlier cycle. If the search terminates after the mth cycle, the oracle has been queried as many as

$$mk + 1 + (N-1)\left[1 - \left(\frac{N-2}{N-1}\right)^{m-1}\right] \tag{7.32}$$

times on average, where the last summand is the average number of wrong test states that are tried out during the unsuccessful $m-1$ preceding cycles.

Accordingly, the average number that we need to query the oracle before we know which oracle is the actual one, is given by

$$G_Q(N;k) = \frac{k}{p_k^{(N)}} + \frac{N - p_k^{(N)}}{1 + (N-2)\, p_k^{(N)}}\,. \tag{7.33}$$

This expression ignores the very small correction of no consequence that results from the possibility that the search can terminate after trying out $N-1$ test states for wrong oracles and so learning that the one remaining oracle must be the actual one.

In $G_Q(N;k)$, k is the number of oracle queries per cycle, so that we can optimize GA by minimizing $G_Q(N;k)$ with respect to k,

$$G_Q(N) = \min_k G_Q(N;k)\,. \tag{7.34}$$

The asymptotic form of Eq. (6.13) is obtained from

$$\lim_{N\to\infty} \frac{G_Q(N)}{\sqrt{N}} = \frac{\phi/2}{(\sin\phi)^2} = 0.6900\,, \tag{7.35}$$

where $\phi = 1.1656$ is the smallest positive solution of $2\phi = \tan\phi$. For $N \gg 1$, one needs $(\sin\phi)^{-2} = 1.18$ cycles on average before GA concludes successfully, and the optimal k value is $k = \frac{1}{2}\phi\sqrt{N} = 0.58\sqrt{N}$, which is slightly less than 75% of $k = \frac{1}{4}\pi\sqrt{N}$, the value that maximizes the single-cycle success probability $p_k^{(N)}$.

7.4 Alternative test-state search strategies

The GA search of Sec. 7.3 is consistently carried out in the full space spanned by all index kets, as requested by the standard form of GA that we accept as its definition. By contrast, the successive iteration rounds of the test-state search [see Secs. 7.1 and 7.2] are conducted in the relevant subspace spanned by the remaining candidate index kets. As a consequence of this systematic shrinking of the searched space, the successive educated guesses get better from one iteration round to the next.

In actual implementations, however, it may not be practical to limit the search to the relevant subspace because it is usually much easier to realize the necessary operations in the full $N = 2^n$ dimensional space [Sec. 7.5]. If all iteration rounds of the test-state search are indeed performed in the full space, we have

$$G'_T(N) = \frac{2-d}{1-d} - \frac{1}{N}\frac{1-d^N}{(1-d)^2} - \frac{1}{N}d^{N-2} \quad \text{with } d = (N-1)\beta_{N-1} \tag{7.36}$$

instead of $G_T(N)$ of Eq. (7.28). The large-N form thereof is

$$G'_T(N) \approx \frac{e^{-\gamma} - 1 + \gamma}{\gamma^2} N = \frac{N}{6.08} \quad \text{with } \gamma = 2 + \sqrt{8}\,. \tag{7.37}$$

Compared with the classical search, the reduction is still by more than a factor of 3, but the full-space test-state search needs about 12% more queries than the relevant-space search.

One could wonder if there is a benefit in using the measurement for unambiguous discrimination (MUD) [62, 63] rather than the SRM, because the MUD gives a small

chance of identifying the actual oracle with a wrong test state. The probability of finding the right one of N oracles with a randomly chosen test state is then

$$\frac{1}{N} + \frac{N-1}{N}\frac{2}{N-2} = \frac{3N-4}{N(N-2)} \tag{7.38}$$

where $2/(N-2)$ is the success probability for the MUD to the $(N-1)$-edged pyramid of the $|t_j^k\rangle$ kets with $j \neq k$ [61].

The price for this increase of the bare $1/N$ probability is paid by getting an inconclusive result from the MUD if it fails to identify the right state, so that we have no information that would facilitate an educated guess for the next test state. The resulting average number of oracle queries is

$$G_T^{(\mathrm{MUD})}(N) = \frac{(N-1)(3N+4)}{12N} \tag{7.39}$$

if we successively search in the relevant subspace only, and

$$G_T^{(\mathrm{MUD})'}(N) = \frac{1}{1-d} - \frac{1}{N}\frac{d-d^{N+1}}{(1-d)^2} - \frac{1}{N}d^{N-1} \quad \text{with } d = \frac{N-4}{N-2} \tag{7.40}$$

if the search is consistently carried out in the full space. The large-N forms

$$\begin{aligned} G_T^{(\mathrm{MUD})}(N) &\approx \frac{N}{4}\,, \\ G_T^{(\mathrm{MUD})'}(N) &\approx \frac{N}{4/(1+e^{-2})} = \frac{N}{3.52} \end{aligned} \tag{7.41}$$

show clearly that this price is high: The test-state search with MUD needs substantially more oracle queries than the search with SRM. In addition, the MUD is a proper POM and more difficult to implement than the SRM.

One could also rely on the MUD rather than the SRM in the verification step of GA. There are then modifications in Eqs. (7.32) and (7.33), but the large-N statement of Eq. (7.35) remains the same.

7.5 Unitary operations for realizing the test-state approach

While the test state $|t_0\rangle$ could be realized for any value of N, we only discuss here the important case of $N = 2^{\mathsf{n}}$—when the oracles are unitary operators acting on n qubits. There, the test states $|t_j\rangle$ of Eq. (7.1) are locally equivalent to $|t_0\rangle$, in the sense that we can transform the test state $|t_0\rangle$ into any other test state by applying the X operations on the relevant qubits. In the following, a construction of $|t_0\rangle$ and a realization of the SRM of Eqs.'(7.11) and (7.18) are presented in sequence.

7.5.1 Construction of the test state

Let us first take the case of three qubits ($N = 8$) as an example; then $a = \sqrt{5/12}$, $b = \sqrt{1/12}$ in Eq. (7.6) with $a^2 + 7b^2 = 1$. For preparing the three-qubit test state $|t_0(8)\rangle$, the input register is initialized in the state $|0\rangle^{\otimes 3}$, and then the single-qubit gate

$$V^{(1)} := e^{-i\theta_1 Y} \quad \text{with} \quad \tan(\theta_1) := \frac{2b}{\sqrt{a^2 + 3b^2}} = \frac{1}{\sqrt{2}} \tag{7.42}$$

is performed on the first qubit. Thereafter, the controlled gate[4] $\Delta^1 V^{(2)}$ with

$$V^{(2)} := e^{-i\theta_2 Y} \quad \text{and} \quad \tan(\theta_2) := \frac{\sqrt{a^2 + b^2} - \sqrt{2}\, b}{\sqrt{a^2 + b^2} + \sqrt{2}\, b} = 2 - \sqrt{3} \tag{7.43}$$

is performed on the second qubit by taking the first qubit as control (with the control set to $|0\rangle$) followed by the Hadamard gate $H^{(2)}$ of Eq. (2.23). Subsequently, the doubly-controlled gate $\Delta^{12} V^{(3)}$ with

$$V^{(3)} := e^{-i\theta_3 Y} \quad \text{and} \quad \tan(\theta_3) := \frac{a - b}{a + b} = \frac{3 - \sqrt{5}}{2} \tag{7.44}$$

[4] In the case of controlled-unitary operation, the symbols Δ and Λ represent that (every) control is set to the kets $|0\rangle$ and $|1\rangle$, respectively.

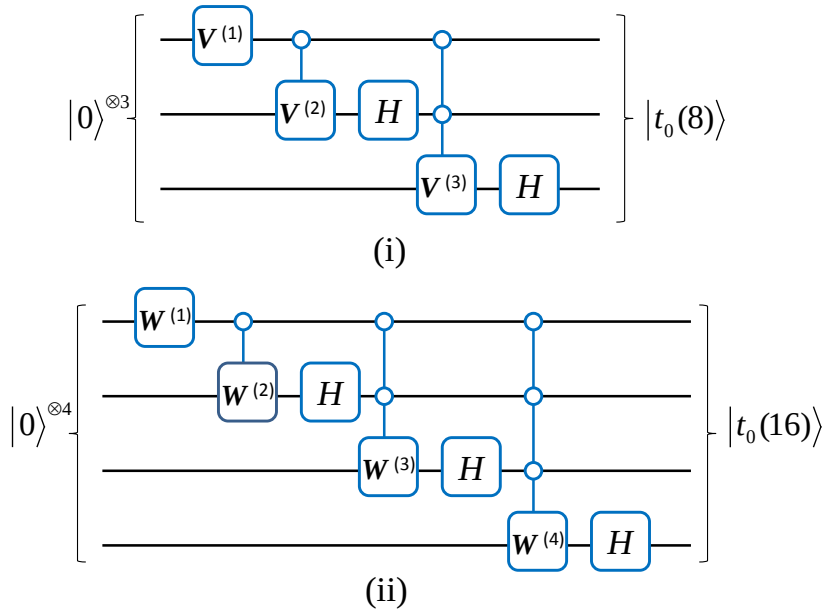


Figure 7.2: Quantum circuit (i) is for preparing the three-qubit test state $|t_0(8)\rangle$ and (ii) is for preparing the four-qubit test state $|t_0(16)\rangle$, respectively. Here, the input state is $|0\rangle^{\otimes \mathsf{n}}$ ($\mathsf{n} = 3, 4$), the Hadamard operations are depicted by H, and the explicit forms of the various controlled gates (the Vs and Ws) are given in the text, where all single-qubit operations are Y rotations.

is executed on the third qubit by taking the first and second qubits as controls (with both controls set to $|0\rangle$) followed by the Hadamard gate $H^{(3)}$. The over-all unitary operation $\mathbf{u}$ for the case of three qubits can be narrated as

$$\mathbf{u} := H^{(3)} \times \left[\Delta^{12} V^{(3)}\right] \times H^{(2)} \times \left[\Delta^{1} V^{(2)}\right] \times V^{(1)}\,, \tag{7.45}$$

and the corresponding quantum circuit is depicted in Fig. 7.2(i).

The quantum circuit displayed in Fig. 7.2(ii) is for the construction of the four-qubit test state $|t_0(16)\rangle$; where $a = \sqrt{13/28}$, $b = \sqrt{1/28}$, and $a^2 + 15b^2 = 1$. In this case,

$$W^{(1)} := e^{-i\vartheta_1 Y} \quad \text{with} \quad \tan(\vartheta_1) := \frac{\sqrt{8}\,b}{\sqrt{a^2 + 7b^2}} = \sqrt{\frac{2}{5}} \tag{7.46}$$

and

$$W^{(2)} := e^{-i\vartheta_2 Y} \quad \text{with} \quad \tan(\vartheta_2) := \frac{\sqrt{a^2 + 3b^2} - 2b}{\sqrt{a^2 + 3b^2} + 2b} = \frac{1}{3} \tag{7.47}$$

as well as $W^{(3)} = V^{(2)}$ and $W^{(4)} = V^{(3)}$.

The generalization to the n-qubit case is immediate. The method to efficiently implement a multiqubit controlled unitary operation $\Delta^{1\cdots(\mathsf{n}-1)}V^{(\mathsf{n})}$ (single-qubit gate V with $\mathsf{n}-1$ control qubits) in terms of universal gates [see Sec. 2.4.3] with $\mathsf{n}-2$ work qubits is given in Secs. 2.4.2, 4.3.3 and also in Refs. [23, 1]. Let us note that its circuit complexity is of the order of n. Consequently, the circuit complexity for constructing the n-qubit test state with quantum circuits of the kind shown in Fig. 7.2 is $O(\mathsf{n}^2)$. In Appendix C, an alternative method for constructing the test state $|t_0\rangle$ is given, where the amplitudes a and b are complex numbers.

7.5.2 Realization of the SRM

In order to perform the SRM of Sec. 7.1 after passing the test state $|t_0\rangle$ through the black box, one needs the corresponding unitary transformation

$$\mathbf{M}_0 := \sum_{l=0}^{N-1} |l\rangle\langle T_0^l| \tag{7.48}$$

that turns each basis ket $|T_0^l\rangle$ into the corresponding ket $|l\rangle$ of the computational basis. Since every test state $|t_j\rangle$ is locally equivalent to $|t_0\rangle$, one can easily get $\mathbf{M}_j$ from $\mathbf{M}_0$ with some local operations. With Eq. (7.18) we have

$$\begin{aligned}\mathbf{M}_0 &= -a|0\rangle\langle 0| - x\sum_{l=1}^{N-1}|l\rangle\langle l| + b\sum_{l=1}^{N-1}\big(|0\rangle\langle l| + |l\rangle\langle 0|\big) + y\sum_{\substack{k,l=1\\k>l}}^{N-1}\big(|k\rangle\langle l| + |l\rangle\langle k|\big)\\ &= -\mathbf{I} + (1-a)|0\rangle\langle 0| + b\big(|0\rangle\langle v| + |v\rangle\langle 0|\big) + y|v\rangle\langle v|\end{aligned} \tag{7.49}$$

with

$$|v\rangle := \sum_{k=1}^{N-1}|k\rangle\,, \tag{7.50}$$

which has one eigenvalue $+1$ and $N-1$ eigenvalues -1, so that the unitary operators $-\mathbf{M}_0$ and the n-qubit unitary operation $\Lambda^{12\cdots(\mathsf{n}-1)}Z^{(\mathsf{n})}$ of Eq. (6.14) have the same set of eigenvalues, that is: they are unitarily equivalent. The eigenkets of $\mathbf{M}_0$

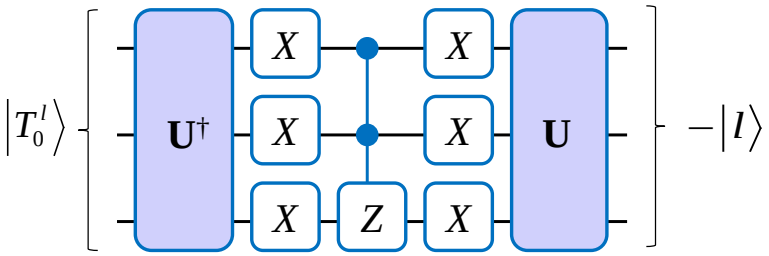


Figure 7.3: The quantum circuit for the implementation of $-\mathbf{M}_0$ in the case of three qubits. The operation $\mathbf{U}$ is implemented by the circuit shown in Fig. 7.2(i) after changing the parameters in accordance with Eq. (7.53). And, if we run the same circuit in the reverse order we can also implement $\mathbf{U}^\dagger$. The quantum gate shown in the center of circuit is the $\Lambda^{12}Z^{(3)}$ given by Eqs. (2.53) and (6.14).

are

$$
\begin{aligned}
|e_0\rangle &:= \sqrt{\frac{1-a}{2}}|0\rangle + \frac{b}{\sqrt{2(1-a)}}|v\rangle\,, \\
|e_1\rangle &:= -\sqrt{\frac{1+a}{2}}|0\rangle + \frac{b}{\sqrt{2(1+a)}}|v\rangle\,, \\
|e_j\rangle &:= \frac{1}{\sqrt{2}}\big[-|1\rangle + |j\rangle\big] \quad \text{for } \; j = 2, 3, \ldots, N-1\,,
\end{aligned}
\tag{7.51}
$$

with $\mathbf{M}_0|e_0\rangle = |e_0\rangle$ and $\mathbf{M}_0|e_{j\neq 0}\rangle = -|e_{j\neq 0}\rangle$. In view of the degeneracy of $\mathbf{M}_0$, the set of orthonormal eigenkets for eigenvalue -1 is not unique, but the choice of Eqs. (7.51) is particularly useful in the present context. For, the eigenket $|e_0\rangle$ has the same structure as the test state $|t_0\rangle$ of Eq. (7.1), and we know from Sec. 7.5.1 how to construct $|t_0\rangle$.

We relate $\mathbf{M}_0$ to $\Lambda^{12\cdots(\mathsf{n}-1)}Z^{(\mathsf{n})}$ through the unitary operator $\mathbf{U}\,X^{\otimes\mathsf{n}}$ that diagonalizes $\mathbf{M}_0$ in the computational basis,

$$
\mathbf{M}_0 = -\mathbf{U}\,X^{\otimes\mathsf{n}}\left[\Lambda^{12\cdots(\mathsf{n}-1)}Z^{(\mathsf{n})}\right]X^{\otimes\mathsf{n}}\,\mathbf{U}^\dagger\,. \tag{7.52}
$$

The operator $\mathbf{U}$ itself is such that $\mathbf{U}|0\rangle^{\otimes\mathsf{n}} := |e_0\rangle$ or $\mathbf{U}\,X^{\otimes\mathsf{n}}|1\rangle^{\otimes\mathsf{n}} := |e_0\rangle$, and we

realize it by the circuit for **u** [see Fig. 7.2 and Eq. (7.45)] with the replacements

$$\begin{aligned} a &\to \sqrt{\frac{1-a}{2}}\,, \\ b &\to \frac{b}{\sqrt{2(1-a)}}\,, \end{aligned} \tag{7.53}$$

while $\mathbf{U}^\dagger$ is implemented by the circuit of Fig. 7.2 with its gates in reverse order and all respective θ angle parameters replaced by $-\theta$. Accordingly, all unitary factors on the right-hand side of Eq. (7.52) have known realizations, as illustrated for $N = 2^3$ in Fig. 7.3.

With the SRM thus implemented and the corresponding test states of Sec. 7.5.1, we can verify the GA outcome and complete the quantum search as discussed in Sec. 7.3. Also, we can perform the full-space test-state search of Sec. 7.4, for which Eqs. (7.36) and (7.37) apply. Of course, there are implementations as well of the test states in successively smaller spaces and of the corresponding SRMs, but their economic implementations are not yet known. The restriction to the subspaces of yet-to-probe index states is rather awkward in practice.

Chapter 8

Conclusion and outlook

The two main results of this thesis are the introduction of the hybrid quantum computation model, as an improvement over the unitary-evolution-based model and the measurement-based model in terms of resources, and the test-state approach to the quantum search, as a procedure to verify and terminate faster the search.

In Part I, we have established the hybrid quantum computation model, which is a combined model of the measurement-based quantum computation model and the unitary-evolution-based quantum computation model. The hybrid model chooses different methods of implementation (either unitary evolution or measurements) for different elementary gates to optimize the consumption of resources such as qubits, entanglement, elementary operations and measurements. It is a universal computation model, which can simulate any quantum algorithm.

Similar to the case of unitary-evolution-based model, first, a big unitary gate (of an algorithm) under simulation is decomposed into a sequence of elementary gates. In the hybrid model, the set of elementary gates consists of single-qubit rotations, the CZ gate, and the multiqubit rotations around the z axis $U_{zz\cdots z}^{12\cdots n}(\theta)$. Every single-qubit rotation and the CZ gate are executed by unitary evolution. However, every multiqubit rotation is performed by preparing a respective star-graph state followed by a single measurement.

In addition, we have not only achieved the optimization, but also obtained a straightforward structure for the classical information processing in the hybrid model. It only requires a $2n$-component information flow vector and the propagation matrices for the elementary gates. Also, we have shown that the classical information processing only takes as many steps as the number of elementary gates in a quantum circuit under simulation. No preprocessing or additional computational step is required for the classical information processing in the hybrid model.

Furthermore, to justify the practical significance of our model, we have presented a number of examples such as the multiqubit controlled-unitary gate $\Lambda^{12\cdots(n-1)}Z^{(n)}$ This gate together with the single-qubit Hadamard H and the Pauli X gates is sufficient to realize Grover's algorithm in the framework of the hybrid model.

A real quantum computer needs to be universal, scalable and fault-tolerant. The hybrid model is completed by presenting the basic ideas for its fault-tolerant version. We have considered the Steane 7-qubit code and provided the corresponding encoded elementary gates and classical information processing. It turns out that the classical information-processing parts of hybrid model fit perfectly in its fault-tolerant version. Indeed, the classical information processing only requires the same $2n$-component information flow vector and $2n \times 2n$ propagation matrices.

One can carry on this investigation in the following directions. In addition to the multiqubit rotations, one could include a few other important gates—which can be executed in a single shot without adding further complications to the model—in the set of elementary gates. Furthermore, one could pursue the investigation of a fault-tolerant version of the hybrid model by considering a noise model.

In Part II, we have introduced the test states that enable one to verify whether the outcome of a quantum search with Grover's algorithm is the actual oracle or not. Grover's algorithm together with the test-state verification successfully terminates the search earlier than the algorithm itself. Indeed, the performance of the algorithm is improved about 25% in terms of speed.

We thereby regard the search problem as defined by the set of possible oracles, which are those considered by Grover. Automatically, other search problems, such as the one studied by Høyer [68], are not covered. The corresponding test states—if they exist—have to be found separately for each search problem. This is also the case for Grover-type searches with more than one matching item, that is, when the oracle is a product of two or more different unitary operators of the kind defined in Eq. (6.4).

It is possible that there are no test states for some of these other search problems. In such a scenario, one may not be able to verify the success of the search by test states of some sort, by a method like the one described in Appendix B or by any other procedure. Also, if it is not possible to verify the outcome then one may need to revise the problem itself. We leave this as a moot point.

With the test states at hand, we have the option of solving the quantum search problem with a classical search strategy. But there is a twist: while there is one "yes" answer, each "no" answer is slightly different. With the help of the square-root measurement, this difference can be exploited systematically for a judicious choice of the test state for the next round. This educated guessing is rewarded by much fewer queries of the oracle on average than that is needed for the simple "yes/no" search. A speed-up by a factor of 3.41 is achievable in principle, and a practical scheme still gains a factor of more than three. In our view, the classical-search benchmark is set by the search that exploits the differences between the "no"s fully.

The test-state approach is completed by giving explicit circuits for the preparation of the n-qubit test states. The circuit complexity is of order n^2, and a variant of the same circuit is the main ingredient in the realization of the square-root measurement. Also, one can employ the hybrid model to construct the n-qubit test states in a single shot.

The hybrid model and the test-state approach are two relatively simple solutions to a more efficient use of resources in quantum computation. We hope this work

will trigger further developments for resource-efficient quantum computation.

Appendices

Appendix A

The reversible classical circuit model

A computer is a machine to perform the desirable computational tasks, such as adding two numbers, with the help of algorithms. An algorithm is an effective method, expressed in terms of a finite set of well-defined instructions, for calculating a required function. Basically, a computer takes an input, calculates the necessary function and produces the output. In classical computation, the input is given and the output is read in terms of bits, and a logic gate is the function

$$f \ : \ \{0,1\}^{\mathsf{n}} \rightarrow \{0,1\}^{\mathsf{m}} \tag{A.1}$$

which provides a m-bit output for a given n-bit input.

In the $\mathsf{m} = 1$ case of Eq. (A.1), the functions—with n input bits and a single output bit—are called *Boolean functions*, and a general function with m output bits is equivalent to m Boolean functions. In the case of 2^{n} possible n-bit inputs and one-bit output, there are altogether $(2)^{2^{\mathsf{n}}}$ Boolean functions. A Boolean function can be decomposed further into a sequence of elementary functions (gates). A list of elementary gates for classical computation is presented in the following [1].

Table A.1: Truth tables of the AND, OR, XOR, NAND, and NOR gates

Inputs		Outputs of gates				
a	b	a AND b	a OR b	a XOR b	a NAND b	a NOR b
0	0	0	0	0	1	1
0	1	0	1	1	1	0
1	0	0	1	1	1	0
1	1	1	1	0	0	0

- The NOT gate: It interchanges the bit value 0 to 1 and 1 to 0. In other words, it takes a one-bit input a, computes the function $f(a) = 1 \oplus a$, and provides the one-bit output $1 \oplus a$, where $\oplus$ represents the modulo-2 addition. It is the only nontrivial reversible one-bit gate.

- The AND gate: It—takes a two-bit input to a one-bit output—returns the single-bit output 1, if and only if both of its input bits are 1. Interestingly, the function associated to the AND gate effectively finds the minimum between the values of the two input bits.

- The OR gate: It—has two-bit input and one-bit output—provides the single-bit output 1, if and only if at least one of its input bits is 1. In other words, it gives the output 0, if and only if both of its input bits are 0; otherwise it returns 1. The function associated to the OR gate effectively finds the maximum between the values of the two input bits.

- The XOR gate: It provides the modulo-2 addition of both the input bits as an output. In this case, the output 1 results, if one and only one of its input bit is 1. If both the input bits are 0 or 1, the output results in 0.

- The NAND gate: It is the opposite to the AND gate and is achieved by applying the NOT gate on the output of the AND gate.

- The NOR gate: It is the result of the negation of the OR gate. It gives the output 1 if and only if both of its input bits are 0.

One could also consider two additional operations—the FANOUT and CROSSOVER. In classical computation, it is allowed to replace a bit with two copies of itself. This operation is called the FANOUT[1]. Also, the operation which interchanges the values of two bits is known as the CROSSOVER (or SWAP).

In quantum computation, the action of a quantum gate can be represented by a unitary matrix in the computational basis. Similarly, in classical computation, the action of a logic gate can be understood by its *Truth table*. Each one of the AND, OR, XOR, NAND, and NOR gates has two-bit input and one-bit output, and their truth tables are combined in Table A.1. The NAND gate is universal for the classical computation—provided that the ancilla bits and the FANOUT gate are available—in a sense that any boolean function can be implemented by using a combination of NAND gates. There are other universal sets of gates such as {AND, NOT}, {OR, NOT}, and {NOR}.

The *reversibility* of a gate is a very important issue because it is deeply linked to energy consumption in computation[2]. A logic gate is reversible if it takes each input (n-bit string) to a unique output (n-bit string). Consequently, each one of the AND, OR, XOR, NAND, and NOR gates are irreversible. The Toffoli gate takes a three-bit input to a unique three-bit output according to its truth table given in Table A.2. Together with ancilla bits, it gives a universal set for reversible classical computation.

If one chooses the same input and output registers, then only $(2^n)!$ gates turns out reversible in the case of n bits. According to the *classical theory of reversible computation*, one can make any gate reversible by choosing separate input and output registers. For example, one can convert the XOR gate into the CNOT gate, $(a, b) \to (a, a \oplus b)$.

[1]The FANOUT operation cannot be performed in a straightforward way in quantum computation, due to the *no-cloning theorem*.

[2]Irreversibility can be thought in terms of information erasure which cost a certain amount of energy.

Table A.2: Truth tables of the Toffoli and Fredkin gates

Inputs			Outputs					
			Toffoli			Fredkin		
a	b	c	a'	b'	c'	a''	b''	c''
0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	1
0	1	0	0	1	0	0	1	0
0	1	1	0	1	1	0	1	1
1	0	0	1	0	0	1	0	0
1	0	1	1	0	1	1	1	0
1	1	0	1	1	1	1	0	1
1	1	1	1	1	0	1	1	1

Let us now see the connection between reversible classical computation and quantum computation. As every gate in the quantum regime is a unitary operator so, every quantum gate is reversible. Hence, there exist legitimate unitary operators X of Eqs. (2.19), $\Lambda^{\mathsf{a}}X^{(\mathsf{b})}$ of Eq. (2.39), and $\Lambda^{12}X^{(3)}$ of Eq. (2.52) for the NOT, CNOT, and Toffoli gates, respectively. However, the AND, OR, XOR, NAND, and NOR gates have no straightforward quantum analog. Since the classical Toffoli gate is universal and has a quantum analog, any CC can be simulated with a QC.

A combination of reversible gates is also reversible, e.g., the SWAP gate can be made of three CNOT gates [see Eq. (2.48)], and the Fredkin gate can be composed of three Toffoli gates [see Eq. (2.54)]. Working of the Fredkin gate[3] is explained in Table A.2. Like the Toffoli gate, the Fredkin gate is also universal for reversible classical computation.

Let us add here the following. Although, the classical CNOT and NOT gates are reversible, they cannot implement the Toffoli (or Fredkin) gate. Therefore, they do not constitute a universal set of gates for classical computation. While, in the quantum regime, it is possible to decompose the Toffoli gate into a sequence of single-qubit and the CNOT gates [see Fig. 2.4]. Thus, they form a universal set of gates for quantum computation [see Sec. 2.4.3]. As an additional remark, one

[3]Application of the Fredkin gate conserves the number of 1s (or 0s) between the input and output.

can use the above mentioned classical theory of reversible computation to perform indirect quantum measurements and to bring all those measurements which appear in the intermediate steps of the quantum computation to the end.

In summary, the task of a CC is to compute a required function similar to Eq. (A.1) which can be further decomposed in terms of Boolean functions. Each Boolean function is further decomposed into a sequence of universal logic gates. A sequence of gates is represented by a circuit diagram, where connecting wires depict bits, and the ancilla bits are provided in a standard state. After initializing the input register in a n-bit state, the logic gates of the circuit are performed in the required order, and, in the end, the output is readout.

Appendix B

An alternative confirmation step for Grover's search algorithm

Here we describe an alternative procedure for verifying the result obtained by GA. This method does not rely on the construction of test states of Chapter 7. Rather it employs a simple circuit that distinguishes between two selected *target oracles* and the other $N-2$ oracles. The verification is achieved by having the GA-outcome oracle in two different target pairs, and thus requires two queries of the oracle.

Suppose GA has had oracle $\mathcal{O}^j$ [see Eq. (6.4)] as the outcome. The corresponding index ket $|j\rangle := |x_1\chi\rangle$ has value x_1 for the first qubit and the values of qubits $2, 3, \cdots, \mathsf{n}$ are summarized by the string χ. We pair $|j\rangle$ with $|\widehat{j}\rangle := |\widehat{x_1}\chi\rangle$ where

$$\widehat{x_1} := x_1 + 1 \pmod 2 = \begin{cases} 1 \text{ if } x_1 = 0\,, \\ \\ 0 \text{ if } x_1 = 1\,, \end{cases} \tag{B.1}$$

so that j and $\widehat{j}$ differ in the first bit value only.

As indicated in Fig. B.1, qubit 1 is prepared in the state with ket $|0\rangle$, and the χ part of the index state is encoded in qubits 2 through n. So, the ket of the n-qubit

Appendix B. An alternative confirmation step for GA

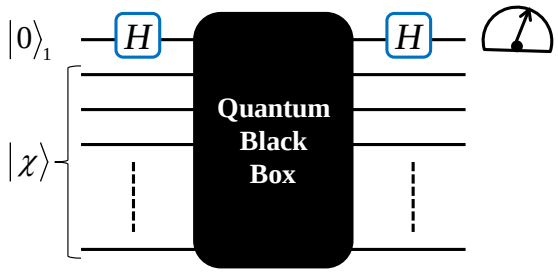


Figure B.1: A single iteration of the alternative confirmation is exhibited in terms of quantum circuit diagram. The input state with ket $|\phi_{\text{in}}\rangle = |0\chi\rangle$ of Eq. (B.2) is passed through the sequence of the Hadamard gate H of Eq. (2.23), the quantum black box, and another Hadamard gate. Finally, the 1st qubit of the output state $|\phi_{\text{out}}\rangle$ is measured in the computational basis.

input state is

$$|\phi_{\text{in}}\rangle := |0\chi\rangle = \begin{cases} |j\rangle \text{ if } x_1 = 0\,, \\ |\widehat{j}\rangle \text{ if } x_1 = 1\,. \end{cases} \tag{B.2}$$

We pass it through the quantum circuit of Fig. B.1, where the given black box is used only once. If the black box is implementing either oracle $\mathcal{O}^j$ or oracle $\mathcal{O}^{\widehat{j}}$, then the output state will have ket

$$|\phi_{\text{out}}^{(\text{yes})}\rangle = |1\chi\rangle\,. \tag{B.3}$$

If, however, the black box is implementing one of the other $N-2$ oracles, the output state will have ket

$$|\phi_{\text{out}}^{(\text{no})}\rangle = |0\chi\rangle\,. \tag{B.4}$$

Finally, qubit 1 is measured in the computational basis. If we find 0, the "no" output is the case, and we can be sure that the actual oracle is neither $\mathcal{O}^j$ nor $\mathcal{O}^{\widehat{j}}$. But when we find 1, we know that one of these oracles is inside the black box. We determine which one by pairing $|j\rangle$ with a third index ket that also differs only by the value of one qubit, which then plays the role of the privileged qubit in the corresponding circuit of the kind depicted in Fig. B.1, where qubit 1 is singled out.

So, we either get a definite "no" answer to the question "Is the jth or the $\widehat{j}$th oracle the case?" or we are told "yes, it is one of these two." In the latter situation,

we know for sure which one it is after a second round.

Appendix C

An alternative construction of the test states

In Sec. 7.5.1, a construction of the test states of Eq. (7.1) is given for the case of $N = 2^{\mathsf{n}}$ with the real coefficients a and b of Eq. (7.6). Here, we provide an alternative method by which one produces the alternative test states with complex a and b amplitudes, as exemplified by

$$\begin{aligned} |t_0\rangle &= a\,|0\rangle + b\sum_{l=1}^{N-1}|l\rangle \\ &= (a-b)|0\rangle^{\otimes \mathsf{n}} + b\sqrt{N}\,|+\rangle^{\otimes \mathsf{n}}\,, \end{aligned} \tag{C.1}$$

where $|+\rangle$ is given by Eq. (2.28), and the absolute values $|a|$ and $|b|$ are, of course, still those of Eq. (7.6). As before, it is enough to show how $|t_0\rangle$ is made, the other test states are then available by applying some single-qubit X gates.

One can obtain a ket of this kind by applying the multi-Hadamard unitary operator

$$U_{HH\cdots H}^{12\cdots \mathsf{n}}(\theta) := \exp\left(-i\frac{\theta}{2}H^{\otimes \mathsf{n}}\right)\,, \tag{C.2}$$

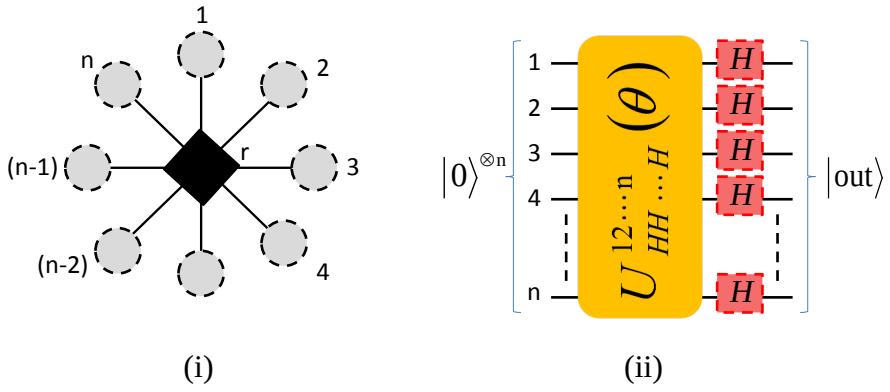


Figure C.1: This figure is similar to Fig. 3.5. In here, the star graph (i) correspond to the graph state with the ket $|\Phi(1+\mathsf{n})\rangle$ of Eq. (C.7). In the star graph, the dotted gray circles represent the input register of n qubits, the bonds represent the controlled-Hadamard gates $\textsc{ch}(\mathsf{n})$ of Eq. (C.6), and the black diamond represents the ancilla qubit r. Circuit (ii) represents the net effect on the input ket $|0\rangle^{\otimes \mathsf{n}}$, when the ancilla qubit is measured in an appropriately chosen basis.

to $|j=0\rangle = |0\rangle^{\otimes \mathsf{n}}$,

$$U^{12\cdots \mathsf{n}}_{HH\cdots H}(\theta)|0\rangle^{\otimes \mathsf{n}} = \cos\left(\tfrac{1}{2}\theta\right)|0\rangle^{\otimes \mathsf{n}} - i\sin\left(\tfrac{1}{2}\theta\right)|+\rangle^{\otimes \mathsf{n}}\,. \tag{C.3}$$

Now, for $b\sqrt{N} = -i\sin\left(\frac{1}{2}\theta\right) = -i\sqrt{N/(2N-4)}$ we need to set the angle parameter θ to the value determined by

$$\tan\left(\tfrac{1}{2}\theta\right) = \sqrt{\frac{N}{N-4}}\,, \tag{C.4}$$

and one verifies that

$$a = \cos\left(\tfrac{1}{2}\theta\right) - \frac{i}{\sqrt{N}}\sin\left(\tfrac{1}{2}\theta\right) = \frac{\sqrt{N-4}-i}{\sqrt{2N-4}} \tag{C.5}$$

also has the absolute value required by Eq. (7.6). So, if we set θ in accordance with Eq. (C.4), then the output state of Eq. (C.3) is the test state $|t_0\rangle$ of Eq. (C.1). We note that $\theta = \pi$ for $N = 4$, and $\theta = \pi/2 + 2/N$ for $N \gg 1$.

One can execute the unitary operation $U^{12\cdots \mathsf{n}}_{HH\cdots H}(\theta)$ on the n-qubit input state $|0\rangle^{\otimes \mathsf{n}}$ by a similar method as the one given for the unitary operation $U^{12\cdots \mathsf{n}}_{zz\cdots z}(\theta)$ in Sec. 3.1.5. Here, the input quantum register of n qubits [circles in Fig. C.1(i)] and

the ancilla qubit[1] r [diamond in Fig. C.1(i)] are initialized in the n-qubit input state with ket $|0\rangle^{\otimes \mathsf{n}}$ and the state with ket $|+\rangle_{\mathsf{r}}$, respectively. Then, similar to the n CZ operations in Sec. 3.1.5, here the n controlled-Hadamard operations

$$\mathrm{CH}(\mathsf{n}) := \big(|0\rangle\langle 0|\big)_{\mathsf{r}} \otimes I^{\otimes \mathsf{n}} + \big(|1\rangle\langle 1|\big)_{\mathsf{r}} \otimes H^{\otimes \mathsf{n}} \tag{C.6}$$

are performed between the ancilla qubit and each one of the n qubits. All the controlled-Hadamard operations represented by the bonds in Fig. C.1(i) can be carried out at the same time, because they all commute with each other. This leads to the resultant star-graph state with the ket

$$|\Phi(1+\mathsf{n})\rangle = \frac{1}{\sqrt{2}}\big[|0\rangle_{\mathsf{r}} \otimes |0\rangle^{\otimes \mathsf{n}} + |1\rangle_{\mathsf{r}} \otimes |+\rangle^{\otimes \mathsf{n}}\big]\,. \tag{C.7}$$

The label $1+\mathsf{n}$ reveals the number of qubits of the final graph state.

A single-qubit projective measurement on the ancilla qubit r in the basis

$$\mathcal{B}_{\theta,\frac{\pi}{2}} = \big\{|\uparrow,\downarrow(\theta,\tfrac{1}{2}\pi)\rangle_{\mathsf{r}}\big\}\,, \tag{C.8}$$

[same as Eq. (3.10)] transforms the input ket of the n qubits into the ket

$$|\mathrm{out}\rangle = \big(H^{\otimes \mathsf{n}}\big)^{m_{\mathsf{r}}} U^{12\cdots \mathsf{n}}_{HH\cdots H}(\theta)|0\rangle^{\otimes \mathsf{n}}\,. \tag{C.9}$$

Here, $m_{\mathsf{r}} \in \{0,1\}$ is the measurement result, and $\big(H^{\otimes \mathsf{n}}\big)^{m_{\mathsf{r}}}$ is the by-product operator [37, 38], which is represented by the red boxes on all the n qubits in Fig. C.1(ii).

After undoing the effect of the by-product operator in Eq. (C.9), one has the test state of Eq. (C.1), and can then apply the necessary single-qubit X gates to get the test state that one needs. Alternatively and more efficiently, one can combine these X gates with the by-product operator and execute the resulting single-qubit

[1]Note that r is just the label of the ancilla qubit. Like n, it does not represent any number.

gates in one go.

Bibliography

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2011).

[2] N. D. Mermin, *Quantum Computer Science: An Introduction* (Cambridge University Press, Cambridge, 2007).

[3] D. E. Knuth, *The Art of Computer Programming, Vol. 3: Sorting and Searching* (Addison-Wesley, 2rd edition, Boston, 1998).

[4] C. E. Shannon, *A mathematical theory of communication*, Bell System Tech. J. **27**, 379 (1948) and Bell System Tech. J. **27**, 623 (1948).

[5] B. Schumacher, *Quantum coding*, Phys. Rev. A **51**, 2738 (1995).

[6] P. W. Shor, *Scheme for reducing decoherence in quantum computer memory*, Phys. Rev. A **52**, 2493(R) (1995).

[7] A. M. Steane, *Error correcting codes in quantum theory*, Phys. Rev. Lett. **77**, 793 (1996).

[8] A. R. Calderbank and P. W. Shor, *Good quantum error-correcting codes exist*, Phys. Rev. A **54**, 1098 (1996).

[9] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, *Mixed-state entanglement and quantum error correction*, Phys. Rev. A **54**, 3824 (1996).

[10] D. Gottesman, *Stabilizer codes and quantum error correction*, e-print arXiv:quant-ph/9705052.

[11] C. H. Bennett and G. Brassard, *Quantum cryptography: Public key distribution and coin tossing*, In Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing, pages 175-179, Bangalore, India, December 1984.

[12] A. K. Ekert, *Quantum cryptography based on Bells theorem*, Phys. Rev. Lett. **67**, 661 (1991).

[13] C. H. Bennett and S. J. Wiesner, *Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states*, Phys. Rev. Lett. **69**, 2881 (1992).

[14] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels*, Phys. Rev. Lett. **70**, 1895 (1993).

[15] C. H. Bennett and P. W. Shor, *Quantum Information Theory*, IEEE Trans. Inf. Theory **44**, 2724 (1998).

[16] A. M. Turing, *On Computable Numbers, with an application to the Entscheidungsproblem*, Proc. Lond. Math. Soc. **s2-42 (1)**, 230 (1937).

[17] R. P. Feynman, *Simulating physics with computers*, Int. J. Theor. Phys. **21**, 467 (1982).

[18] P. Benioff, *Quantum mechanical models of Turing machines that dissipate no energy*, Phys. Rev. Lett. **48**, 1581 (1982).

[19] D. Deutsch, *Quantum theory, the Church-Turing Principle and the universal quantum computer*, Proc. R. Soc. Lond. A **400**, 97 (1985).

[20] D. Deutsch, *Quantum computational networks*, Proc. R. Soc. Lond. A **425**, 73 (1989).

[21] D. P. DiVincenzo, *Quantum computation*, Science **270**, 255 (1995).

[22] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, *Experimental realization of any discrete unitary operator*, Phys. Rev. Lett. **73**, 58 (1994).

[23] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, *Elementary gates for quantum computation*, Phys. Rev. A **52**, 3457 (1995).

[24] A. Galindo and M. A. Martin-Delgado, *Information and computation: Classical and quantum aspects*, Rev. Mod. Phys. **74**, 347 (2002).

[25] P. W. Shor, *Fault-tolerant quantum computation*, e-print arXiv:quant-ph/9605011.

[26] J. Preskill, *Fault-tolerant quantum computation*, e-print arXiv:quant-ph/9712048.

[27] D. Gottesman, *A theory of fault-tolerant quantum computation*, e-print arXiv:quant-ph/9702029.

[28] P. O. Boykin, T. Mor, M. Pulver, V. Roychowdhury, and F. Vatan, *On universal and fault-tolerant quantum computing*, e-print arXiv:quant-ph/9906054.

[29] E. Knill, R. Laflamme, and W. Zurek, *Threshold Accuracy for Quantum Computation*, e-print arXiv:quant-ph/9610011.

[30] E. Knill, R. Laflamme, and W. H. Zurek, *Resilient Quantum Computation: Error Models and Thresholds*, Proc. R. Soc. Lond. A **454**, 365 (1998).

[31] D. Deutsch and R. Jozsa, *Rapid solution of problems by quantum computation*, Proc. R. Soc. Lond. A **439**, 553 (1992).

[32] D. R. Simon, *On the power of quantum computation*, In Proceedings, 35th Annual IEEE Symposium on Foundations of Computer Science, pages 116-123, (1994); SIAM J. Comp. **26**, 1474 (1997).

[33] P. W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, In Proceedings, 35th Annual IEEE Symposium on the Foundations of Computer Science, pages 124-134, (1994); SIAM J. Comp. **26**, 1484 (1997); e-print arXiv:quant-ph/9508027.

[34] L. K. Grover, *Quantum mechanics helps in searching for a needle in a haystack*, Phys. Rev. Lett. **79**, 325 (1997).

[35] H. J. Briegel and R. Raussendorf, *Persistent entanglement in arrays of interacting particles*, Phys. Rev. Lett. **86**, 910 (2001).

[36] M. Hein, J. Eisert, and H. J. Briegel, *Multiparty entanglement in graph states*, Phys. Rev. A **69**, 062311 (2004).

[37] R. Raussendorf and H. J. Briegel, *A one-way quantum computer*, Phys. Rev. Lett. **86**, 5188 (2001).

[38] R. Raussendorf, D. E. Browne, and H. J. Briegel, *Measurement-based quantum computation on cluster states*, Phys. Rev. A **68**, 022312 (2003).

[39] R. Raussendorf and H. Briegel, *Computational model underlying the one-way quantum computer*, Quant. Inf. Comp. **6**, 443 (2002); e-print arXiv:quant-ph/0108067.

[40] D. E. Browne and H. J. Briegel, *One-way quantum computation – a tutorial introduction*, e-print arXiv:quant-ph/0603226.

[41] A. Sehrawat, D. Zemann, and B.-G. Englert, *Hybrid quantum computation*, Phys. Rev. A **83**, 022317 (2011).

[42] A. Sehrawat, L. H. Nguyen, and B.-G. Englert, *Test-state approach to the quantum search problem*, Phys. Rev. A **83**, 052311 (2011).

[43] D. Jaksch, H.-J. Briegel, J. I. Cirac, C. W. Gardiner, and P. Zoller, *Entanglement of atoms via cold controlled collisions*, Phys. Rev. Lett. **82**, 1975 (1999).

[44] M. A. Nielsen, *Optical quantum computation using cluster states*, Phys. Rev. Lett. **93**, 040503 (2004).

[45] D. E. Browne and T. Rudolph, *Resource-efficient linear optical quantum computation*, Phys. Rev. Lett. **95**, 010501 (2005).

[46] T. P. Bodiya and L.-M. Duan, *Scalable generation of graph-state entanglement through realistic linear optics*, Phys. Rev. Lett. **97**, 143601 (2006).

[47] C. Y. Lu, X. Q. Zhou, O. Gühne, W. B. Gao, J. Zhang, Z. S. Yuan, A. Goebel, T. Yang, and J. W. Pan, *Experimental entanglement of six photons in graph states*, Nature Physics **3**, 91 (2007).

[48] J. A. Jones, M. Mosca, and R. H. Hansen, *Implementation of a quantum search algorithm on a quantum computer*, Nature **393**, 344 (1998).

[49] I. L. Chuang, N. Gershenfeld, and M. Kubinec, *Experimental implementation of fast quantum searching*, Phys. Rev. Lett. **80**, 3408 (1998).

[50] M. S. Anwar, D. Blazina, H. A. Carteret, S. B. Duckett, and J. A. Jones, *Implementing Grover's quantum search on a para-hydrogen based pure state NMR quantum computer*, Chem. Phys. Lett. **400**, 94 (2004).

[51] F. Yamaguchi, P. Milman, M. Brune, J. M. Raimond, and S. Haroche, *Quantum search with two-atom collisions in cavity QED*, Phys. Rev. A **66**, 010302(R) (2002).

[52] Z. J. Deng, M. Feng, and K. L. Gao, *Simple scheme for two-qubit Grover search in cavity QED*, Phys. Rev. A **72**, 034306 (2005).

[53] W. L. Yang, C. Y. Chen, and M. Feng, *Implementation of three-qubit Grover search in cavity QED*, Phys. Rev. A **76**, 054301 (2007).

[54] P. G. Kwiat, J. R. Mitchell, P. D. D. Schwindt, and A. G. White, *Grover's search algorithm: An optical approach*, J. Mod. Opt. **47**, 257 (2000).

[55] N. Bhattacharya, H. B. van Linden van den Heuvell, and R. J. C. Spreeuw, *Implementation of quantum search algorithm using classical Fourier optics*, Phys. Rev. Lett. **88**, 137901 (2002).

[56] P. Walther, K. J. Resch, T. Rudolph, E. Schenck, H. Weinfurter, V. Vedral, M. Aspelmeyer, and A. Zeilinger, *Experimental one-way quantum computing*, Nature **434**, 169 (2005).

[57] K. Chen, C. M. Li, Q. Zhang, Y. A. Chen, A. Goebel, S. Chen, A. Mair, and J. W. Pan, *Experimental realization of one-way quantum computing with two-photon four-qubit cluster states*, Phys. Rev. Lett. **99**, 120503 (2007).

[58] A. Peres, *Neumark's theorem and quantum inseparability*, Found. Phys. **20**, 1441 (1990).

[59] A. Peres and W. K.Wootters, *Optimal detection of quantum information*, Phys. Rev. Lett. **66**, 1119 (1991).

[60] M. Ban, K. Kurukowa, R. Momose, and O. Hirota, *Optimum measurements for discrimination among symmetric quantum states and parameter estimation*, Int. J. Theor. Phys. **36**, 1269 (1997).

[61] B.-G. Englert and J. Řeháček, *How well can you know the edge of a quantum pyramid?*, J. Mod. Opt. **57**, 218 (2010).

[62] I. D. Ivanovic, *How to differentiate between non-orthogonal states*, Phys. Lett. A **123**, 257 (1987).

[63] R. B. M. Clarke, A. Chefles, S. M. Barnett, and E. Riis, *Experimental demonstration of optimal unambiguous state discrimination*, Phys. Rev. A **63**, 040305(R) (2001).

[64] C. Zalka, *Grovers quantum searching algorithm is optimal*, Phys. Rev. A **60**, 2746 (1999).

[65] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, *Tight bounds on quantum searching*, Fortschr. Phys. **46**, 493 (1998); e-print arXiv:quant-ph/9605034.

[66] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, *Quantum amplitude amplification and estimation*, e-print arXiv:quant-ph/0005055.

[67] G. Brassard, P. Høyer, and A. Tapp, *Quantum counting*, e-print arXiv:quant-ph/9805082.

[68] P. Høyer, *Arbitrary phases in quantum amplitude amplification*, Phys. Rev. A **62**, 052304 (2000).