Quantum computation and experiments on IBM quantum systems

Pro Gradu
University of Turku
Theoretical physics
2021
BSc Juho Nykänen
Examiners:
Jyrki Piilo
Matteo Rossi

Turun yliopiston laatujärjestelmän mukaisesti tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -järjestelmällä.

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin OriginalityCheck service.

UNIVERSITY OF TURKU
Fysiikan ja tähtitieteen laitos

Nykänen, Juho Quantum computation and experiments on IBM quantum systems

Pro Gradu-tutkielma, 82 sivua
Fysiikka
Kesäkuu 2021

Kvantti-informaatioteoriaa käytetään kuvaamaan informaatiota kvanttimekaanisissa systeemeissä ja sen laskentaan keskittyvää osaa sanotaan kvanttilaskennaksi. Kvanttilaskennassa ongelmanratkaisu mallinnetaan piirillä, jolla kuvataan koko laskentaprosessi algoritmeineen. Kvantti-algoritmeissa voidaan käyttää hyväksi niin kutsuttua kvantti-rinnakkaisuutta, ilmiötä, jonka avulla tietyntyyppiset ongelmat voidaan ratkaista nopeammin kuin vastaavilla klassisilla algoritmeilla.

Tässä työssä verrataan IBM:n kvanttitietokoneilla tehtyjen piirien mittauksia ideaalisten tietokonesimulaatioiden tuloksiin. Ensimmäisessä kokeessa näytetään, kuinka kvanttiteleportaatio tapahtuu ja kuinka hyvin se onnistuu IBM:n kvanttitietokoneella virheettömään simulaatioon verrattuna. Seuraavissa kokeissa käsitellään samaan malliin Deutsch–Jozsa-, Bernstein–Vazirani- ja Simonin algoritmejä sekä Groverin hakualgoritmia. Viimeisessä kokeessa tutkitaan Lüderin instrumenttia käyttävän usean peräkkäisen sumean mittauksen vaikutusta binäärisen mittauksen onnistumistodennäköisyyteen. Kokeessa verrataan ennustetun mallin arvoja sekä ideaalisen simulaation että yksinkertaisen virhemallin sisältävän simulaation tuloksiin.

Avainsanat: kvantti-informaatioteoria, kvanttilaskenta, kvanttiteleportaatio, Deutsch–Jozsa-algoritmi, Bernstein–Vazirani-algoritmi, Simonin algoritmi, Groverin hakualgoritmi, peräkkäiset mittaukset

UNIVERSITY OF TURKU
Department of Physics and Astronomy

Nykänen, Juho Quantum computation and experiments on IBM quantum systems

Master's thesis, 82 pages
Physics
June 2021

Quantum computation is a part of quantum information theory. By using quantum mechanical phenomena it is possible to speed up the solving of some problems with quantum algorithms over classical probabilistic algorithms. In some cases quantum algorithms can utilise quantum parallelism in order to speed up problem solving.

In this thesis experiments on IBM quantum computers are compared with ideal error free simulations. In the first part quantum teleportation is shown in practice, similarly the performance of Deutsch–Jozsa, Bernstein–Vazirani, Simon's and Grover's search algorithms are shown in action on IBM quantum computers and each is compared against an ideal simulation. The last experiment shows how the binary success probability of an unsharp measurement using Lüders instrument can be improved by adding sequential measurements by using an ideal simulation and a simulation using simple noise model.

Keywords: quantum information theory, quantum computation, quantum teleportation, Deutsch–Jozsa algorithm, Bernstein–Vazirani algorithm, Simon's algorithm, Grover's search algorithm, repeatable measurements

Contents

Preface

As the invention of transistor and the progress of computation theory have lead to the modern age of computing, the recent advances on creating and maintaining physical quantum systems have shown that it is possible to utilise quantum phenomena such as superposition and entanglement in computation. Quantum computation is a part of quantum information theory which focuses in manipulating and processing information and has its roots in the early 1980s. Quantum information theory differs from the classical information theory by integrating quantum mechanics in its core.

Quantum bits or qubits, the elementary bits of information in quantum information theory, can be entangled with other qubits, or they can exist in a superposition. In this thesis there are a number of experiments that utilise entanglement, superposition and quantum parallelism in order to perform computations faster than it would be possible with traditional computers.

The IBM has quantum systems to which it provides open access. The IBM quantum computers have from 1 to 15 qubits with different layouts connecting the qubits to each other. Quantum states are fragile and their manipulation is technically challenging, as a result quantum computers are prone to errors. Each algorithm used in this thesis was designed to be scalable and each experiment was performed as large as feasible within the limitations of the quantum systems available. The most significant constraints can be identified as the physical layout of the qubits in a given system and the error rate that grows with the number of operations carried out in a quantum circuit performing the algorithm in question. Other limitations such as the decoherence of quantum states in physical qubits have limited effect in comparison when using simple circuits.

In this thesis, in most experiments the results from a real quantum computer are compared with the results from an ideal simulation using the largest number of qubits in a circuit while maintaining an acceptable error rate that confirms the

functionality of the algorithm in question. Out of the two different experiments that do not test common quantum algorithms, one shows an example of a quantum phenomenon and the other is a practical simulation that compares measurement using an error free simulation and a simulation using a simple noise model.

The experiments begin with quantum teleportation, a technique that is used to transfer the state of one qubit onto another. In this technique an entangled qubit pair is used to perform nearly instantaneous transmission of information once over distance in one direction.

The next experiments benchmark the performance of quantum algorithms. The first algorithm, the Deutsch–Jozsa algorithm, is one of the earliest developed to show how quantum parallelism speeds up finding the solution to a specific problem. The Deutsch–Jozsa algorithm is a generalisation of the Deutsch algorithm. The algorithm finds a solution for a guessing game between two recipients, Alice and Bob, in which Alice tries to find if the function Bob has is either balanced or constant. The quantum algorithm allows Alice to query all possible inputs in a single query, not one input at a time like the classical algorithm would have to.

The Bernstein–Vazirani algorithm on the other hand has an unknown function that has two possible outputs that is used to find out a hidden bitstring b. As with Deutsch–Jozsa problem, the quantum parallelism allows to test all the possible inputs to the function with one query.

Simon's algorithm uses quantum parallelism to try multiple inputs with one query to see if an unknown function is one-to-one or two-to-one mapping. If it is two-to-one mapping it solves the unknown bitstring b that represents the mapping.

The preceding algorithms are artificial in a sense that they have been developed as examples to show the power of quantum computing. The Grover's search algorithm, however, takes on solving a practical problem in computing. Grover's search algorithm shows that searching specific items from an unstructured dataset is a

faster process when using qubits instead of classical bits and a classical probabilistic search algorithm testing one query at a time.

The last experiment was limited to simulations due to the physical limitations of the open access IBM quantum systems. No available computer has the required layout that would have natively supported the desired circuit. The initial testing with the largest available computer resulted in too many operations in order to overcome the physical limitations of its layout. As quantum systems are prone to errors in every operation performed on qubits the results were unusable. The last part is a study with simulations in which it is shown that the probability of attaining a correct measurement of a binary observable is increased with sequential unsharp measurements on a qubit, when using an observable whose state transformations are of a Lüders type.

1 Quantum information theory

Classical information theory is a well-established theory that forms the basis for other theories, and it is a fundamental part of computer sciences, where bits are stored, transferred and manipulated according to well-defined rules and logic. However, it was found out that classical information theory fails to describe the unexpected behaviour of quantum systems, as a bit is not enough to fully describe a quantum state and there are limitations on how the information can be manipulated. In his 1995 paper, Benjamin Schumacher [1] coined the term quantum bit, or *qubit*, for the smallest measure of information within the developing quantum information theory. Quantum systems behave in a manner described by quantum mechanics, the most experimentally successful theory for describing the natural world. In the heart of quantum information theory lies fundamental restrictions such as the no-teleportation theorem, no-cloning theorem, no-deleting theorem and no-broadcast theorem. None of these theorems exist in the classical information theory and have led to further incompatibility between the two theories. However, quantum information theory does provide tools and tricks that allow solving some classically difficult tasks more efficiently and provides advantages in other aspects, for example breaking classical encryption can be an easy task for a quantum computer yet quantum information theory allows to create unbreakable encryption for classical and quantum computers.

The fundamental difference between a classical bit and a quantum bit is that a qubit can exist in a *superposition* of 'on' and 'off' states. Quantum states are hard to sustain and difficult to manipulate. The quantum systems and the delicate quantum states are prone to noise from environment that leads to constant coherence loss of states over time, a phenomenon known as decoherence. Outside ideal quantum simulations there are no perfectly isolated systems, therefore all quantum systems are open. Even if there are no ideal physical systems they can still be mathematically

treated as such. As will be seen in Chapter 2, for an experiment with a quantum algorithm the differences between the result of the calculation and the result of an experiment on a real quantum computer diverge enough for the difference to be shown easily, but the plausibility of the quantum algorithm in question remains undiminished.

Two or more qubits can also exist in an intertwined state, phenomenon what Einstein famously called spooky action at a distance, the intertwined qubits are said to be *entangled*. Finally, the measurement of a quantum system consisting of qubits breaks all superposition and entanglement of qubits and forces them in one definite state, converting the system to a form that can be represented with classical bits.

1.1 Quantum mechanics

In this section the basics of quantum mechanics, related to finite discrete quantum computation, are covered. Proofs of results are omitted here and can be found in the literature [2–4].

1.1.1 Hilbert space

A qubit is a quantum mechanical object, meaning it can be described mathematically as an element in a complete (two-dimensional) inner product space known as a Hilbert space $\mathcal{H}$. Specifically, a Hilbert space is a complex vector space with an inner product, denoted by $\langle\cdot|\cdot\rangle$ in so-called Dirac notation. The inner product is a mapping $\mathcal{H} \times \mathcal{H} \to \mathbb{C}$ that fulfils the following conditions

(i) $\langle\varphi|\alpha\psi + \beta\phi\rangle = \alpha\langle\varphi|\psi\rangle + \beta\langle\varphi|\phi\rangle$,

(ii) $\langle\varphi|\psi\rangle = \overline{\langle\psi|\varphi\rangle}$,

(iii) $\langle\varphi|\varphi\rangle \geq 0, \quad \forall\, \varphi \in \mathcal{H}$,

(iv) $\quad \langle\varphi|\varphi\rangle = 0 \iff |\varphi\rangle = 0,$

where $\alpha, \beta \in \mathbb{C}$ and $\varphi, \phi, \psi \in \mathcal{H}$. Any space with an inner product satisfying conditions (i-iv) is referred to as an inner product space. Moreover, a Hilbert space is a complete inner product space, where completeness ensures that techniques from calculus can be used.

Also every inner product space $\mathcal{H}$ is a normed space and for every $\varphi \in \mathcal{H}$ its norm is a real-valued function

$$||\varphi|| \coloneqq \sqrt{\langle\varphi|\varphi\rangle}.$$

A common definition from linear algebra, arising as a consequence of the inner product, is that two vectors $\varphi, \psi \in \mathcal{H}$ are orthogonal if their inner product satisfies

$$\langle\varphi|\psi\rangle = 0.$$

Orthogonal vectors may be denoted by $\varphi \perp \psi$. A pair of orthogonal vectors $\psi_1 \perp \psi_2$ are *orthonormal* if they are norm-one, $||\psi_i|| = \sqrt{\langle\psi_i|\psi_i\rangle} = 1$. It is known that there exists sets of orthonormal vectors for which linear combinations of their elements span the whole of $\mathcal{H}$, and such sets are maximal sets. These sets lead to the concept of a (orthonormal) basis:

Definition 1. *A basis is a maximal (non-extendible) set $\{e_i\}$ of mutually orthonormal vectors in $\mathcal{H}$ such that any vector $\psi \in \mathcal{H}$ can be expressed as a linear combination of its elements:*

$$\psi = \sum_{i=1}^{d} \psi_i e_i.$$

The order d of the basis, i.e. the largest possible number of mutually orthogonal vectors in $\mathcal{H}$, is the dimension of $\mathcal{H}$. If d mutually orthogonal vectors can be found for any $d \in \mathbb{N}$, then the Hilbert state is infinite-dimensional, otherwise it is finite.

A finite-dimensional Hilbert space is called separable since it has a countable orthonormal basis. Infinite-dimensional Hilbert spaces will not be used here and so

all further Hilbert spaces are finite. There are multiple elementary and important results for inner product spaces such as the Cauchy-Schwarz inequality, triangle inequality, Pythagorean formula, Bessel's inequality and the parallelogram law, all of which are covered in the literature [3, 4].

The interactions between quantum systems are described with the use of a tensor product. The tensor product creates a new Hilbert space out of two or more Hilbert spaces. For two finite-dimensional Hilbert spaces $\mathcal{H}$ and $\mathcal{K}$ a new product space can be defined, referred to as the tensor product of $\mathcal{H}$ and $\mathcal{K}$ that is denoted by $\mathcal{H} \otimes \mathcal{K}$, as follows: consider the bases $\{\psi_i\}_i$ and $\{\varphi_j\}_j$ for $\mathcal{H}$ and $\mathcal{K}$, respectively. Let $\mathcal{B} = \{\psi_i \otimes \varphi_j\}_{i,j}$ denote the set of all pairs of basis vectors for $\mathcal{H}$ and $\mathcal{K}$, where $\otimes$ is assumed to be bilinear[1], and let $\mathcal{H} \otimes \mathcal{K} = \left\{ \sum_{i,j} c_{i,j} \psi_i \otimes \varphi_j \mid \psi_i \otimes \varphi_j \in \mathcal{B}, c_{i,j} \in \mathbb{C} \right\}$ by linear extension. In addition, an inner product can be defined on $\mathcal{H} \otimes \mathcal{K}$ via $\langle \xi \otimes \eta | \, \psi \otimes \varphi \rangle = \langle \xi | \, \psi \rangle \langle \eta | \, \varphi \rangle$ for any $\xi, \psi \in \mathcal{H}$ and $\eta, \varphi \in \mathcal{K}$ making $\mathcal{H} \otimes \mathcal{K}$ an inner product space with a basis $\mathcal{B}$ (as its elements are orthonormal). Since $\mathcal{B}$ is also finite, with $\dim(\mathcal{H}) \dim(\mathcal{K})$ elements, $\mathcal{H} \otimes \mathcal{K}$ is also finite Hilbert space.

1.1.2 Operators

Operations on quantum states are achieved with linear transformations, operators, where a bounded linear operator is a linear mapping $T : \mathcal{H} \mapsto \mathcal{H}$, such that there is a constant $c \in [0, \infty)$ satisfying $||T\varphi|| = c||\varphi||$ for all $\varphi \in \mathcal{H}$. The collection of all bounded linear operators forms a normed vector space $\mathcal{L}(\mathcal{H})$, with the norm $|| \cdot ||$ given by

$$||T|| = \sup_{||\Psi||=1} ||T\Psi||.$$

A bounded linear operator transforms a quantum state in one direction, the opposite transformation is called its adjoint, every finite dimensional $T \in \mathcal{L}(\mathcal{H})$ operator has an adjoint.

[1] **$\otimes$ is bilinear if,** $\forall \psi, \psi_1, \psi_2 \in \mathcal{H}$, $\varphi, \varphi_1, \varphi_2 \in \mathcal{K}$ **and** $c \in \mathbb{C}$, $(\psi_1 + \varphi_1) \otimes (\psi_2 + \varphi_2) = \psi_1 \otimes \varphi_1 + \psi_1 \otimes \varphi_2 + \psi_2 \otimes \varphi_1 + \psi_2 \otimes \varphi_2$ **and** $c(\psi \otimes \varphi) = (c\psi) \otimes \varphi = \psi \otimes (c\varphi)$.

Definition 2. *For bounded linear operator $T \in \mathcal{L}(\mathcal{H}) \exists\ T^* \in \mathcal{L}(\mathcal{H}^*)$ such that*

$$\langle T^*\Psi|\varphi\rangle = \langle \Psi|T\varphi\rangle\,,$$

where T^* is called an adjoint of T.

The adjoint has the following properties for $S, T \in \mathcal{L}(\mathcal{H})$ and $\alpha \in \mathbb{C}$

(i) $T^{**} = T$,

(ii) $(S+T)^* = S^* + T^*$,

(iii) $(\alpha T)^* = \overline{\alpha} T^*$,

(iv) $(ST)^* = T^* S^*$,

(v) $||T|| = ||T^*||$,

(vi) $||T^*T|| = ||T||^2$

There are special cases of bounded linear operators such as normal operators, self-adjoint operators, unitary operators, and positive operators.

Definition 3. *A collection of definitions for bounded linear operators.*

(i) *A bounded linear operator $T \in \mathcal{L}(\mathcal{H})$ is a normal operator if it satisfies $T^*T = TT^*$.*

(ii) *A bounded linear operator $T \in \mathcal{L}(\mathcal{H})$ is self-adjoint if $T^* = T$, the set of bounded self-adjoint operators on $\mathcal{H}$ is denoted as $\mathcal{L}_s(\mathcal{H})$*

(iii) *A bounded linear operator $U \in \mathcal{L}(\mathcal{H})$ is unitary when $U^*U = UU^* = I$, where I is the identity map of $\mathcal{H}$. The set of unitary operators on $\mathcal{H}$ is denoted by $\mathcal{U}(\mathcal{H})$.*

(iv) *A bounded operator $T \in \mathcal{L}(\mathcal{H})$ is positive if $\langle \psi|T\psi\rangle \geq 0$ for every $\psi \in \mathcal{H}$. A positive operator is denoted by $T \geq 0$.*

(v) *A vector $\psi \in \mathcal{H}$ is an eigenvector of a bounded operator T if there exists a number $\lambda \in \mathbb{C}$ such that $T\psi = \lambda\psi$, where λ is referred to as the eigenvalue associated with ψ.*

In the case of finite-dimensional Hilbert spaces the eigenvalues of T are found by solving the characterised equation $\det(T - \lambda I) = 0$, and every finite-dimensional operator has a complete set of eigenvalues. The solutions to $\det(T-\lambda I) = 0$ form the spectrum of T, which in the finite-dimensional case coincide with the eigenvalues of T. If each eigenvalue of T has a single corresponding eigenvector, then its spectrum is said to be nondegenarate, otherwise it is degenerate [3, p. 16].

The notion of positivity induces a partial ordering on $\mathcal{L}_s(\mathcal{H})$, whereby $T \geq S$ if and only if $T - S \geq 0$ for any $T, S \in \mathcal{L}_s(\mathcal{H})$. The product of every $T \in \mathcal{L}(\mathcal{H})$ with its adjoint T^* is positive, as for any $\psi \in \mathcal{H}$

$$\langle \psi | T^*T\psi \rangle = \langle T\psi | T\psi \rangle = ||T\psi||^2 \geq 0. \tag{1}$$

It is also true that any positive operator can be given in the form $S = T^*T$. This is a result of the square root lemma.

Lemma 1. (Square root lemma)
Let $T \in \mathcal{L}_s(\mathcal{H})$ be a positive operator. There is a unique positive operator $S \in \mathcal{L}_s(\mathcal{H})$ satisfying $(S)^2 = T$. The operator S is called the square root of T, or $S = T^{1/2} = \sqrt{T}$. The square root operator has the following properties.

(i) *If $M \in \mathcal{L}(\mathcal{H})$ and $MT = TM$, then $MT^{1/2} = T^{1/2}M$.*

(ii) *if T is invertible then $T^{1/2}$ is also invertible and $(T^{1/2})^{-1} = (T^{-1})^{1/2}$.*

In addition, the absolute value of an operator is defined as follows.

Definition 4. *Let $T \in \mathcal{L}(\mathcal{H})$, its absolute value is $|T| \coloneqq (T^*T)^{1/2}$.*

A self-adjoint operator P can be further classified as a projection if $P^2 = P$; from this it is clear that projections are positive operators (see Eq. (1)). The set

of projections is denoted by $\mathcal{P}(\mathcal{H})$, and they are used to represent pure states in quantum mechanics.

A projection $O \neq P \neq I$ has norm $||P|| = 1$, its eigenvalues are 1 and 0 and for every vector $\psi \in \mathcal{H}$ there are orthogonal vectors $\psi_i \in \mathcal{H}$ such that for some fixed i

$$P\psi_i = 1, \quad \text{then } P\psi_j = 0,$$

where $i \neq j$ and $\psi = \psi_i + \sum_j \psi_j$.

Along with the complement of a projection $P^\perp \coloneqq (I - P)$, the set of projections $\mathcal{P}(\mathcal{H})$ forms an orthocomplemented lattice, but $\mathcal{P}(\mathcal{H})$ is not a vector space [3, p. 23]. As a subset of self-adjoint operators, projections inherit the partial order structure.

Proposition 1. *For projections $P, Q \in \mathcal{P}(\mathcal{H})$ $P - Q$ is a projection if and only if $P \geq Q$.*

A similar results to Proposition 1 can be found if the projection Q is replaced with a positive operator $T \in \mathcal{L}(\mathcal{H})$, namely if $T \leq P$ then $TP = PT = T$ [3, p. 25]. In addition to the subtraction of projections, their sum is also a projection if the following condition is fulfilled:

Proposition 2. *For projections $P, Q \in \mathcal{P}(\mathcal{H})$ $P + Q$ is a projection if and only if they are orthogonal, i.e. $PQ = QP = O$.*

A projection P_η is one dimensional if it is of the form

$$P_\eta \Psi = \langle \eta | \Psi \rangle \, \eta,$$

where $\eta \in \mathcal{H}$ is a unit vector. In Dirac notation it is common to use the form $P_\eta = |\eta\rangle \langle \eta|$. Multi-dimensional projections can be written as a finite sum of orthogonal one-dimensional projections [3, p. 33].

The group of unitary operators, (iii) in Definition 3, represent quantum logical gates in quantum computing. A unitary operator is also a mapping from one orthonormal basis to another [3, p. 27]. To see this, consider a basis $\mathcal{B} = \{\varphi_i\}_{i=1}^d$ for

a d-dimensional $\mathcal{H}$ and so $\langle\varphi_i|\varphi_j\rangle = \delta_{ij}$. Let $\mathcal{B}' = \{\psi_i = U\varphi_i|\varphi_i \in \mathcal{B}\}$ and notice that for any pair of vectors $\psi_i, \psi_j \in \mathcal{B}'$, $\langle\psi_i|\psi_j\rangle = \langle U\varphi_i|U\varphi_j\rangle = \langle\varphi_i|(U^*U)\varphi_j\rangle = \langle\varphi_i|\varphi_j\rangle = \delta_{ij}$. In other words, the orthogonality of the basis is preserved under the action of U and so $\mathcal{B}'$ is itself a new orthonormal set of vectors in $\mathcal{H}$ with the same size as $\mathcal{B}$, i.e. $\mathcal{B}'$ is a new basis for $\mathcal{H}$.

From the definition of unitary operators a number of observations can be made: Firstly, the identity operator is itself a unitary operator as $I^* = I$ and $I^2 = I$; Secondly, $U^* = U^{-1}$ and so the inverse operator U^{-1} of U is also unitary; Thirdly, for a pair of unitary operators, and from (iv) of Definition 2,

$$(UV)(UV)^* = UVV^*U^* = I = V^*U^*UV = (UV)^*UV,$$

and so any pair of unitary operators is also unitary. These three properties show that $\mathcal{U}(\mathcal{H})$ has a group structure under the operation of operator multiplication.

A connection between unitary and self-adjoint operators can be made by noting that for any $T \in \mathcal{L}_s(\mathcal{H})$, e^{iT} is unitary, where the exponential map is given by

$$e^T := \sum_{n=0}^{\infty} \frac{T^n}{n!}. \tag{2}$$

The trace of an operator $T \in \mathcal{L}(\mathcal{H})$ is given by

$$\mathrm{tr}[T] = \sum_{i=1}^{d} \langle\varphi_i|T\varphi_i\rangle,$$

for any orthonormal basis $\{\varphi_i\}_{i=1}^d$ of $\mathcal{H}$. The trace of an operator is the sum of its eigenvalues, counting multiplicity [3, p. 30].

Normal operators ((i) in Definition 3) can be deconstructed with a spectral decomposition into a sequence of its eigenvalues and an orthonormal basis according to the following lemma.

Lemma 2. (Spectral decomposition) *For a normal operator T with an eigenbasis $\{\varphi_i\}$ and corresponding eigenvalues $\{\lambda_i\}$ it can be rewritten as*

$$T = \sum_i \lambda_i |\varphi_i\rangle\langle\varphi_i|.$$

1.1.3 States and effects

A state in quantum mechanics is a description of an ensemble of similarly prepared systems [3, p. 48], when measured it produces an effect on the measuring apparatus. An effect is a property of the measurement device, and depending on the properties of the device the outcome might be simply 'yes' or 'no'. With the knowledge of the state of the system and the measurement effects, the probability distribution for an experiment can be known.

All quantum states ρ belong to a state space $\mathcal{S}(\mathcal{H})$ which is defined as

$$\mathcal{S}(\mathcal{H}) = \{\rho \in \mathcal{L}_s(\mathcal{H}) |\ \rho \geq 0, \mathrm{tr}[\rho] = 1\}.$$

Pure states are one-dimensional projections with $\mathrm{tr}[\rho^2] = 1$. Any state ρ can be given as a spectral decomposition of pure states P_i and its eigenvalues λ_i

$$\rho = \sum_i \lambda_i P_i,$$

where for all i the eigenvalues λ_i sum to one $\sum_i \lambda_i = 1$. Furthermore, if a state ρ is a pure state, there exists i such that $\lambda_i = 1$ and $\lambda_j = 0$ for all $i \neq j$ leading to the purity $\mathcal{P}(\rho)$ of a state defined as

$$\mathcal{P}(\rho) \coloneqq \mathrm{tr}[\rho^2] = \sum_i \lambda_i^2.$$

If a state is pure $\mathcal{P}(\rho) = 1$, otherwise $\mathcal{P}(\rho) \leq 1$; any $\rho \in \mathcal{S}(\mathcal{H})$ that is not pure is mixed.

Definition 5. *An effect E is an operator satisfying $O \leq E \leq I$. The set of effects is*

$$\mathcal{E}(\mathcal{H}) = \{E \in \mathcal{L}_s(\mathcal{H}) | O \leq E \leq I\}.$$

The set of effects is a convex subspace of $\mathcal{L}_s(\mathcal{H})$ and the set of projections are a subset of effects leading to the inclusion relation [3, p. 70]

$$\mathcal{P} \subset \mathcal{E}(\mathcal{H}) \subset \mathcal{L}_s(\mathcal{H}).$$

1.1.4 Measurement

For states $\rho \in \mathcal{S}(\mathcal{H})$ and effects $E \in \mathcal{E}(\mathcal{H})$ the value $\mathrm{tr}[\rho E]$ is a number between zero and one corresponding to the probability that the measurement event represented by E occurs when the system is prepared in the state ρ, showing that an effect is a mapping from $\mathcal{S}(\mathcal{H}) \to [0,1]$ i.e. $E(\rho) \coloneqq \mathrm{tr}[\rho E] \in [0,1]$.

For a composite system $\mathcal{H}_A \otimes \mathcal{H}_B = \mathcal{H}_{AB}$ with independent measurements and preparations for each, the joint probability of E_A and E_B is

$$\mathrm{tr}[\overline{\gamma}(\rho_A, \rho_B)\gamma(E_A, E_B)] = \mathrm{tr}[\rho_A E_A]\mathrm{tr}[\rho_B E_B],$$

where $\gamma : \mathcal{E}(\mathcal{H}_A) \otimes \mathcal{E}(\mathcal{H}_A) \to \mathcal{E}(\mathcal{H}_{AB})$ and $\overline{\gamma} : \mathcal{S}(\mathcal{H}_A) \otimes \mathcal{S}(\mathcal{H}_A) \to \mathcal{S}(\mathcal{H}_{AB})$. An operator T (state ρ) on $\mathcal{H}_A \otimes \mathcal{H}_B$ is a product state if $T = T_A \otimes T_B$ ($\rho = \rho_A \otimes \rho_B$), separable if a linear combination of product operators (states) and intertwined if neither. Compound operators (states) are called entangled [3, p. 99]. To understand the subsystems of a compound state we can use the partial trace.

Definition 6. *Consider the tensor product $\mathcal{H}_A \otimes \mathcal{H}_B$ of systems A and B. The partial trace over system A is the linear mapping*

$$\mathrm{tr}_A : \mathcal{L}(\mathcal{H}_A \otimes \mathcal{H}_B) \to \mathcal{L}(\mathcal{H}_B)$$

satisfying

$$\mathrm{tr}[\mathrm{tr}_A[T]E] = \mathrm{tr}[T(I \otimes E)]$$

for all $T \in \mathcal{L}(\mathcal{H}_A \otimes \mathcal{H}_B)$ and $E \in \mathcal{L}(\mathcal{H}_B)$. The partial trace tr_B over the subsystem $\mathcal{H}_B$ is defined similarily.

As a result of the following Proposition, the partial trace of a state remains a state [3, p. 101].

Proposition 3. *Let $T \in \mathcal{L}(\mathcal{H}_A \otimes \mathcal{H}_B)$. Then:*

(i) $\mathrm{tr}[T] = \mathrm{tr}[\mathrm{tr}_A[T]] = \mathrm{tr}[\mathrm{tr}_B[T]]$;

(ii) $T \geq O$ implies that $\mathrm{tr}_A[T] \geq O$ and $\mathrm{tr}_B[T] \geq O$.

For a state $\rho \in \mathcal{S}(\mathcal{H}_A \otimes \mathcal{H}_B)$ of a composite system $A + B$ the reduced states $\rho_A = \mathrm{tr}_B[\rho]$ and $\rho_B = \mathrm{tr}_B[\rho]$ describe the states of the subsystems and the state $\rho_{AB} \equiv \rho$ is their joint state. If the reduced states are pure, then the joint state is of the form $\rho_{AB} = \rho_A \otimes \rho_B$, this would mean that the states are uncorrelated [3, p. 102].

When performing a measurement one of a number of possible outcomes will occur, each of which must be desrcibed mathematically; these descriptions are referred to as *effects*, and the mathematical construct describing the measurement overall is called an *observable*. In particular, an observable is given as a positive operator-valued measure. Consider a set Ω and its power set 2^Ω. A subset $\mathcal{F} \subset 2^\Omega$ is a σ-algebra if

(i) $\emptyset, \Omega \in \mathcal{F}$;

(ii) if $X \in \mathcal{F}$ then $X^\mathrm{c} = \Omega \setminus X \in \mathcal{F}$;

(iii) if $X_i \in \mathcal{F}$ for a sequence $\{X_i\}$, then $\cup_i X_i \in \mathcal{F}$.

The pair $(\Omega, \mathcal{F})$ is called a measurable space, a set $X \in (F)$ is called an event and a σ-algebra is a collection of all events. A map $\mathcal{F} \to \mathcal{E}(\mathcal{H})$ is a positive operator-valued measure (POVM) if and only if the mapping $X \mapsto \mathrm{tr}[\rho \mathsf{A}(X)]$ is a probability measure for every state $\rho \in \mathcal{S}(\mathcal{H})$ [3, p. 110].

Definition 7. *An observable is a positive operator-valued measure (POVM) and thus it is a mapping* $\mathsf{A} : \mathcal{F} \to \mathcal{E}(\mathcal{H})$ *such that*

(i) $\mathsf{A}(\emptyset) = 0$;

(ii) $\mathsf{A}(\Omega) = I$;

(iii) $\mathsf{A}(\cup_i X_i) = \sum_i \mathsf{A}(X_i)$ for any sequence $\{X_i\}$ of disjoint sets in $\mathcal{F}$.

As projections are subset of effects, a POVM can be a projection-valued, which leads to the next definition.

Definition 8. *A POVM is a projection-valued or sharp measure (PVM) if* $\mathsf{A}(X)$ *is a projection for every* $X \in \mathcal{F}$.

A POVM is a generalisation of PVMs and by using POVM we allow for the use of unsharp observables. The measurement process uses observables that are either positive operator-valued or projection-valued measures in order to get the probability of a given effect.

1.1.5 Lüders instrument

Up until now, only the probability distributions resulting from measuring an observable have been considered, but the post-measurement state must also be taken into account. Depending on the implementation of the observable, there are countless number of different ways the state may be transformed, but we restrict ourselves to a form introduced by Lüders, further reading on instruments can be found in the literature [3, 4].

Definition 9. *The Lüders instrument* $\mathcal{I}^L$ *associated with a discrete observable* A *with outcome space* $(\Omega, \mathcal{F})$ *is a map* $X \mapsto \mathcal{I}^L_X(\cdot)$, *where*

$$\mathcal{I}^L_X(\rho) = \mathsf{A}(X)^{1/2}\rho\mathsf{A}(X)^{1/2},$$

for all $X \in \mathcal{F}$ *and* $\rho \in \mathcal{S}(\mathcal{H})$.

For a sharp observable A with effects $\mathsf{A}(i) = |\varphi_i\rangle\langle\varphi_i|$, where $\{\varphi_i\}$ is an orthonormal basis for the considered Hilbert space, the above definition can be written as

$$\mathcal{I}^L_i(\rho) = \mathsf{A}(i)\rho\mathsf{A}(i),$$

since $\mathsf{A}(i)^{\frac{1}{2}} = \mathsf{A}(i)$ for one-dimensional projections.

The object $\mathcal{I}_X^L(\rho)$, which satisfies

$$\mathrm{tr}[\mathcal{I}_X^L(\rho)] = \mathrm{tr}[\mathsf{A}(X)\rho],$$

is the (non-normalised) state of the system after obtaining measurement outcome X for A when the system is in state ρ. We refer to a measurement of A whose state transformations are of a Lüders type as a *Lüders measurement*.

A measurement causes unavoidable disturbance in the system, with the post-measurement state of the system differing from the input. This disturbance can be expressed with the concept of a conditional output state. By first measuring the observable A and obtaining outcome X, followed by a measurement of observable B, on a system in initial state ρ, the conditional measurement $p_\rho(\mathsf{B} \in Y | \mathsf{A} \in X)$ of obtaining outcome Y is given by

$$\begin{aligned} p_\rho(\mathsf{B} \in Y | \mathsf{A} \in X) &= \frac{p_\rho(\mathsf{B} \in Y \& A \in \mathsf{X})}{p_\rho(\mathsf{A} \in X)} = \frac{\mathrm{tr}[\mathcal{I}_X^L(\rho)\mathsf{B}(Y)]}{\mathrm{tr}[\mathcal{I}_X^L(\rho)]} \\ &=: \mathrm{tr}[\tilde{\rho}_X \mathsf{B}(Y)]. \end{aligned}$$

The state

$$\tilde{\rho}_X = \frac{1}{\mathrm{tr}[\mathcal{I}_X^L(\rho)]}\mathcal{I}_X^L(\rho) \tag{3}$$

is called the conditional output state.

Consider a sequential measurement of two discrete observables A and B, with A being measured by its Lüders measurement. If the measurement outcome probabilities of B do not depend on whether A has been measured first, then the measurement of observable B is not disturbed by the measurement of A. For the Lüders instrument $\mathcal{I}^L$ the nondisturbance condition is

$$\mathrm{tr}[\mathcal{I}_\Omega^L(\rho)\mathsf{B}(y)] = \mathrm{tr}[\rho \mathsf{B}(y)],$$

which is required to hold for all outcomes y and states ρ and $\mathcal{I}_\Omega^L(\rho)$ is the post-measurement state of the system after measuring A but by not recording the specific measurement outcome. This leads into the following theorem.

Theorem 1. (Lüders theorem)

Let A *and* B *be discrete observables and further suppose that* A *is sharp. A Lüders measurement of* A *does not disturb* B *if* A *and* B *commute,* $\mathsf{A}(x)\mathsf{B}(y) = \mathsf{B}(y)\mathsf{A}(x)$ *for all* x, y.

Take a Lüders instrument $\mathcal{I}$ describing the measurement of a discrete observable A on a system in state ρ, the conditional output state corresponding to the measurement of a nonzero outcome x is

$$\tilde{\rho}_x = \frac{1}{\mathrm{tr}[\mathcal{I}_x(\rho)]}\mathcal{I}_x(\rho),$$

which is the exact same form as in Equation (3). This means that one use of a measurement is in a preparation of the system.

1.2 Quantum computing

A qubit is a representation of two state object $|0\rangle \cong (1,0)^\mathsf{T}, |1\rangle \cong (0,1)^\mathsf{T} \in \mathcal{H} \cong \mathbb{C}^2$. A qubit is similar to a classical bit: when measured on the basis $|0\rangle$, $|1\rangle$, it falls in either basis state $|0\rangle$ or $|1\rangle$, however before it is measured it can exist in a superposition of the states which can be given using Dirac notation as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{4}$$

where $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$. The Equation (4) defines fully a pure qubit state. A qubit has a probability $|\alpha|^2$ of being found in state $|0\rangle$ and $|\beta|^2$ of being in state $|1\rangle$. Because of these superpositions it takes $2^n - 1$ complex numbers to completely describe a quantum system containing n qubits, instead of just n as in classical systems. Because the square of the amplitudes α and β sum to one the Equation (4) can be written as

$$|\psi\rangle = e^{i\gamma}\left(\cos\frac{\theta}{2}\,|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}\,|1\rangle\right), \tag{5}$$

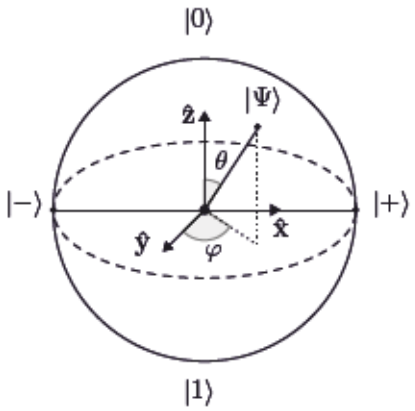


Figure 1: The Bloch sphere representation of a qubit with two orthogonal states $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ and an arbitrary state $|\Psi\rangle$.

where θ, φ and γ are real numbers. The global phase, the first exponent $e^{i\gamma}$, does not have any observable effect and so it can be ignored [2, p. 93] leaving us with the state

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle .$$

The numbers θ and φ define a point on a three-dimensional unit sphere, thereby allowing for a useful means of visualising the state of a single qubit, known as the Bloch sphere (Figure 1). In contrast to the global phase $e^{i\gamma}$, the relative phase $e^{i\varphi}$ does have an observable effect. For example, setting $\theta = \pi/2$ and $\varphi = 0$ or π leads to the orthogonal states

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

A third set of states on the Bloch spehere can be found by setting $\theta = \pi/2$ and $\varphi = \pi/2$ or $-\pi/2$, resulting in the orthogonal states

$$|R\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle),$$
$$|L\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle).$$

These three pairs of states $|0/1\rangle$ and $|+/-\rangle$ and $|R/L\rangle$ form the three computational Z, X and Y bases respectively. The tensor product $|\psi\rangle = |\phi\rangle \otimes |\varphi\rangle$ (denoted also as $|\phi\rangle\,|\varphi\rangle$ or $|\phi,\varphi\rangle$), for two states $|\phi\rangle = \alpha_0\,|0\rangle + \alpha_1\,|1\rangle$ and $|\varphi\rangle = \beta_0\,|0\rangle + \beta_1\,|1\rangle$ is written as

$$\begin{aligned} |\psi\rangle &= (\alpha_0\,|0\rangle + \alpha_1\,|1\rangle)\,(\beta_0\,|0\rangle + \beta_1\,|1\rangle) \\ &= \gamma_{00}\,|00\rangle + \gamma_{01}\,|01\rangle + \gamma_{10}\,|10\rangle + \gamma_{11}\,|11\rangle\,, \end{aligned}$$

where the amplitudes $\gamma_{j,k}$ are the products of amplitudes α_j and β_k. The state $|\psi\rangle$ exists in a superposition of four states, and the states $|00\rangle\,, |01\rangle\,, |10\rangle$ and $|11\rangle$ form a new computational basis. One particular collection of superpositions of the previously mentioned states are the so-called Bell states

$$\begin{aligned} |\Phi^+\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \\ |\Phi^-\rangle &= \frac{|00\rangle - |11\rangle}{\sqrt{2}}, \\ |\Psi^+\rangle &= \frac{|01\rangle + |10\rangle}{\sqrt{2}}, \\ |\Psi^-\rangle &= \frac{|01\rangle - |10\rangle}{\sqrt{2}}. \end{aligned} \tag{6}$$

It is known that the Bell states form a maximally entangled basis for four-dimensional Hilbert space for two qubits.

1.2.1 Qubit operations

Qubits can be operated on in the defined Z-, X- and Y-bases with quantum logical gates. There are two types of quantum logical gates: single qubit gates and multi-qubit gates. Unlike classical logic gates all quantum gates are reversible and described mathematically with self-adjoint unitary operators. There are many single qubit logical gates, but for the experiments in this thesis the most important ones are defined as follows. The first set of gates generate rotations about their respective

axes

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

which are commonly referred to as the Pauli operators. They operate on states $|0\rangle$ and $|1\rangle$ as follows

$$\begin{aligned} X|0\rangle &= |1\rangle, \quad X|1\rangle = |0\rangle, \\ Y|0\rangle &= i|1\rangle, \quad Y|1\rangle = -i|0\rangle, \\ Z|0\rangle &= |0\rangle, \quad Z|1\rangle = -|1\rangle. \end{aligned}$$

In addition to these three operators, the Hadamard operation

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

creates a superposition for a single qubit in the Z-basis by rotating it to the X-basis $H|x>=\sum_{y=0}^{1}(-1)^{xy}|y>/\sqrt{2}$ or

$$\begin{aligned} H|0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \\ H|1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \end{aligned}$$

Phase gates change the phase of a qubit in state $|1\rangle$ while the probability of measuring $|0\rangle$ or $|1\rangle$ remains unchanged. The square root of Pauli Z operator $\sqrt{Z} = S$, is called the phase gate and the square root of phase gate is T

$$\sqrt{Z} = S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \qquad \sqrt[4]{Z} = T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}.$$

All of the single qubit gates defined above are special cases of a general unitary operation

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -e^{-i\lambda}\sin(\frac{\theta}{2}) \\ e^{i\theta}\sin(\frac{\theta}{2}) & e^{i\lambda+i\theta}\cos(\frac{\theta}{2}) \end{pmatrix},$$

where θ, λ are real numbers.

Multi-qubit gates work similarly to classical gates by forming a truth table out of the control qubits. The basic multi qubit gates are the SWAP- and controlled-NOT (CNOT) gates

$$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

The SWAP gate swaps two input qubits and the controlled-NOT gate uses one qubit as control and switches the state of the other if the input (control) is $|1\rangle$. They are denoted here as CNOT(control qubit | target qubit)$|\psi\rangle$

$$CNOT(0|2)\,|1_0, 0_1, 0_2\rangle = |1_0, 0_1, 1_2\rangle\,.$$

All further notation will use a operation $X(target)$ for single qubit operations and $(control|target)$ for multi qubit gates. There are other controlled gates such as the controlled Pauli operations (CX, CY, CZ) that rotate the target qubit, with CX gate being the CNOT gate. Additionally the CNOT gate can be used to extend a superposition to other qubits in order to form, for example, Bell states.

The last type of gates covered here are the multi-control gates such as the Toffoli gate that takes two control qubits and operate on one qubit. The Toffoli gate has a similar structure to the CNOT gate with the target qubit switching from $|0\rangle$ to $|1\rangle$, and vice versa, and with no effect on the control qubits; this can be scaled up to n target qubits. Explicitily the Toffoli gate operates as follows:

$$TOFF(0, 1, |2)\,|i, j, x\rangle = |i, j, z\rangle\,,$$

where $z = x \oplus ij \pmod 2$. In general if all the control qubits in a multi-control gate are in state $|1\rangle$ the operation in question is performed on the target.

1.2.2 Quantum circuit

Quantum circuits are a means of visualising a quantum computation, that have one or multiple registers consisting of qubits that are transformed with operations in order to perform calculations of other tasks. These tasks are often referred to as algorithms. As described earlier, operations on states are unitary (in closed systems), which leads to the broad definition of a quantum algorithm being a unitary transformation that can be decomposed into a product of simpler unitary transformations [5]. On a physical quantum computer not all qubit operations are natively supported. The smallest set of gates from which all possible operations can be derived from are informally called the universal gates. For example, the IBM quantum systems support three unitary single gate operations and a controlled-NOT between two qubits, from which all other supported gates are constructed.

Some of the key elements in quantum algorithms are the oracle and quantum parallelism. Without quantum parallelisation an algorithm in quantum logic circuits would not perform faster than a classical algorithm, for example searching an unstructured database using Grover's algorithm provides an edge over a classical algorithm as will be seen in Chapter 2.

Quantum parallelism rises from the fact that qubits can exist in a superposition of two states, this enables running both states at the same time, in parallel. For example operating with a Hadamard gate on each qubit in a registry of three qubits in the state $|\psi_0\rangle = |000\rangle$ transforms into

$$\begin{aligned} H\,|\psi_0\rangle = |\psi\rangle_1 &= |+++\rangle = \frac{1}{\sqrt{2^3}}\big(\,|0\rangle + |1\rangle\,\big)^{\otimes 3} \\ &= \frac{1}{\sqrt{2^3}}\big(\,|000\rangle + |001\rangle + \cdots + |111\rangle\,\big). \end{aligned} \tag{7}$$

The three qubit register in Equation (7) has eight distinct states. This register (input) can be operated on by an oracle function, which transforms the register according to the algorithm in question (output). The term oracle comes from com-

puter science, and simply refers to a procedure or a function that takes an input and provides an output. An oracle is built specifically for each algorithm, and various different yet similar oracles are realised in the experiments in Chapter 2. This output is then usually measured in the last part of the circuit.

1.2.3 Measurement

For a qubit system in a state Ψ, the probability of getting an outcome X when measuring observable A is given by

$$p(x) = \langle\psi|\, \mathsf{A}(x)\, |\psi\rangle\,, \tag{8}$$

where $\mathsf{A}(x)$ are the measure outcomes. In the case of a two qubit system in the state $|\psi\rangle = \alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle$ the process of finding the probability of measuring the first qubit in state $|0\rangle$ or $|1\rangle$ can be reduced to just collecting the terms of the state

$$|\psi\rangle = |0\rangle\,(\alpha\,|0\rangle + \beta\,|1\rangle) + |1\rangle\,(\gamma\,|0\rangle + \delta\,|1\rangle)\,,$$

in other words finding the effects $E_0 = |0\rangle\langle 0|$ and $E_1 = |1\rangle\langle 1|$ for the first qubit results in probabilities

$$p(0) = \langle\psi|\, E_0\, |\psi\rangle = |\alpha|^2 + |\beta|^2,$$
$$p(1) = \langle\psi|\, E_1\, |\psi\rangle = |\gamma|^2 + |\delta|^2.$$

The corresponding post-measurement states on the second qubit are

$$|\psi_0\rangle = \frac{\alpha\,|0\rangle + \beta\,|1\rangle}{\sqrt{|\alpha|^2 + |\beta|^2}}, \quad |\psi_1\rangle = \frac{\gamma\,|0\rangle + \delta\,|1\rangle}{\sqrt{|\gamma|^2 + |\delta|^2}}.$$

Since there are two other computational bases beside the Z-base, the partial measurement of the qubit system in state $|\psi\rangle = \alpha'\,|e_0 0\rangle + \beta'\,|e_0 1\rangle + \gamma'\,|e_1 0\rangle + \delta'\,|e_1 1\rangle$ of finding the first qubit in either state $|e_0\rangle$ or $|e_1\rangle$ is done in the same manner. A

qubit can be measured in another basis by representing the qubits in question in this other basis, for example Z-basis states in the X-basis are

$$|0\rangle = \frac{|+\rangle + |-\rangle}{\sqrt{2}}, \quad |1\rangle = \frac{|+\rangle - |-\rangle}{\sqrt{2}}.$$

In general terms the measurement of qubits in state $|\psi_0\rangle$ of an effect E_i can be expressed as

$$p(i) = \langle\psi_0| E_i |\psi_0\rangle,$$

and the subsequent post measurement state is given by

$$|\psi\rangle_1 = \frac{E_i |\psi_0\rangle}{\sqrt{\langle\psi_0| E_i |\psi_0\rangle}}.$$

The order and number of the qubits does not matter in the measurement process.

Scaling up quantum systems by introducing increasing number of qubits can permit quantum algorithms to gain an edge over their classical counterparts. Outside of ideal cases increasing the number of qubits in a system results in the system becoming more prone to errors, though there exists error correcting methods that will not be used or covered in theoretical detail in this thesis.

2 Experiments on IBM quantum systems

In order to utilise the effects present within quantum mechanics, a quantum computer is necessary. The core principles of a quantum computer were laid out in DiVicenzo's criteria [6], which give general requirements for quantum information processing, qubit manipulation, quantum gates and other necessary aspects that define a quantum computer.

Physical qubits can manifest as any systems which have two distinct quantum states, hence the number of different physical implementations that are being pursued [7–11]. The current front runner is the superconducting qubit for creating quantum computers that are more powerful than current classical supercomputers [12], with first commercial implementations by D-wave [13] and IBM [14].

As the technical problems behind quantum computing are solved we near the point of quantum supremacy, a milestone idealised by John Preskill, where a quantum computer exceeds supercomputers in solving problems that would classically require time frames of millions of years, if not more. The limit where quantum supremacy is achieved depends on the algorithm and the problem at hand. Quantum supremacy is easy to understand but hard to define. There are several front runners in becoming the first task approaching the point of quantum supremacy, one such candidate is solving factoring problems [15, 16]. In October 2019 Google announced that it achieved quantum supremacy with their 53 qubit computer, the machine having checked outputs from a quantum random number generator [17]. Google's competitor IBM was quick to criticise and question the claim [17, 18].

The editor-in-chief of MIT Technology review Gideon Lichfield interviewed both companies over the disagreement on Google achieving quantum supremacy and explains the views on the matter in an interview [19]. Lichfield starts by explaining that IBM sees that solving the problem used to show supremacy wouldn't take the 10,000 years to calculate on a classical supercomputer like Google claimed, but only

a couple of days instead. Engineers at Google implore IBM to prove this claim by performing this calculation in a couple of days, instead IBM criticises the idea that quantum supremacy can be achieved by just showing that quantum computers outperform classical computers in one specific problem and challenge the idea and the importance of quantum supremacy in general. Lichfield shares his cautious opinion in the interview that he believes that Google has achieved quantum supremacy in accordance to Preskill's challenge. No consensus on the matter has been achieved by the scientific community as of the time of writing of this thesis.

At the heart of a superconducting quantum computer is the macroscopic electric circuit, which when cooled enough starts to present quantum properties: quantised energy levels, superposition of states and entanglement. In superconductors electrons form Cooper pairs, thereby becoming new charge carriers; these pairs are condensed into a single macroscopic state described by a wave function allowing macroscopic circuits to exhibit atomic-scale phenomena [20]. Quantum states in superconducting machines are manipulated by using electromagnetic pulses to control the magnetic flux, the electric charge or the phase difference across a Jospehson junction [21]. IBM uses a so-called transmon variant [22] which is a capacitive variant of the Cooper pair-box that is also known as a charge qubit [12].

Classical computers are able to simulate quantum information processes and test quantum algorithms, in other words simulate how a quantum computer works at a cost. IBM provides access to a such cloud-based simulator, and has further developed an open source software development kit called Qiskit for working with their freely available IBM Q Experience cloud platform [23, 24]. The cloud platform provides tools and access to their quantum computers and a quantum simulator. The OpenQASM simulator can be set up to simulate noise that happens in operations in real quantum computers, which makes it really useful in testing algorithms with up to 32 qubits with arbitrary internal structures, before running them on real

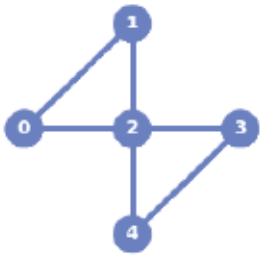
machines. Qiskit also allows local simulations. By simulating the circuits locally it is possible to quickly prototype algorithms and find the idealised solutions before running the tests on a computer. The computers are occasionally heavily utilised, which might result in impractical waiting times when running experiments.

In this thesis multiple experiments were performed as a local simulation, on IBM 5 qubit quantum computers Ourense and Yorktown and on 15 qubit Melbourne. The algorithms were chosen to show quantum parallelism, quantum phenomena and the scalability of selected experiments.

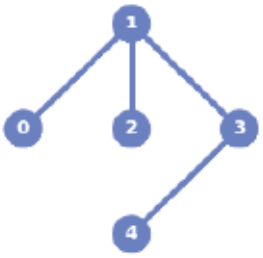
IBM provides access to multiple computers with various qubit layouts, number of qubits and error rates. The machines are regularly calibrated and maintained, which results in varying error rates over time for single qubit operations and CNOT gates between qubits on same circuits performed at different times. This leads to increased variance between experimental measurements.

The layouts of the machines can be seen in Figure 2. Controlling IBM quantum systems and OpenQASM is possible with an open source software development kit (SDK) called Qiskit and its built-in application programming interfaces (APIs), with further information on Qiskit available on their online handbook [25]. Collectively Qiskit and its APIs provide tools for manipulating, optimising, measuring and visualising experiments and measurement data. For example by utilising the transpiler it is possible to map qubits defined in software to specified physical qubits in such a way to minimise error rates.

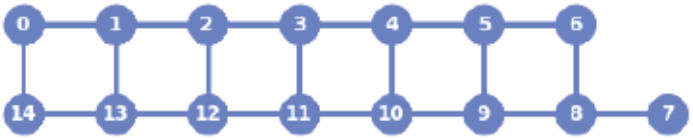
Noise in experiments can be managed in multiple ways: Minimising the amount of operations done, avoiding CNOT operations between non-directly connected qubits and choosing optimal qubits for operations. This can either be done manually or by using the transpiler component of Qiskit's Terra API. The transpiler provides multiple automated optimisation levels from no optimisation at level 0 to maximum optimisation at level 3 (more on transpiler can be found on online resources [25]).



(a) Yorktown



(b) Ourense



(c) Melbourne

Figure 2: Layouts of IBM quantum computers used in the experiments done on this thesis.

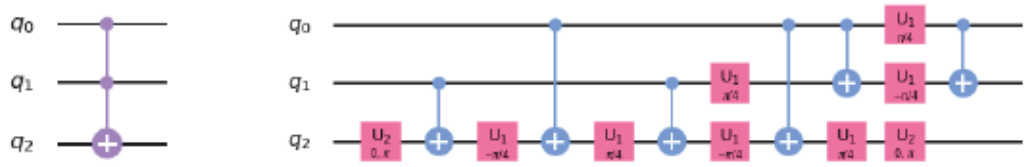
A transpiler with optimisation level 3 was chosen for experiments in this thesis, as by doing so the code could be focused on generating more modular and general algorithms instead of highly optimised algorithms with case by case optimisation.

The supported operations on qubits are generated by combining three single qubit unitary operations with CNOT gates between qubits. The three unitary single qubit operations are

$$U3(\theta, \phi, \lambda) = U(\theta, \phi, \lambda),$$

$$U2(\theta, \phi, \lambda) = U3(\pi/2, \phi, \lambda),$$

$$U1(\theta, \phi, \lambda) = U3(0, 0, \lambda)$$



(a) Toffoli gate (b) Full decomposition of a Toffoli gate.

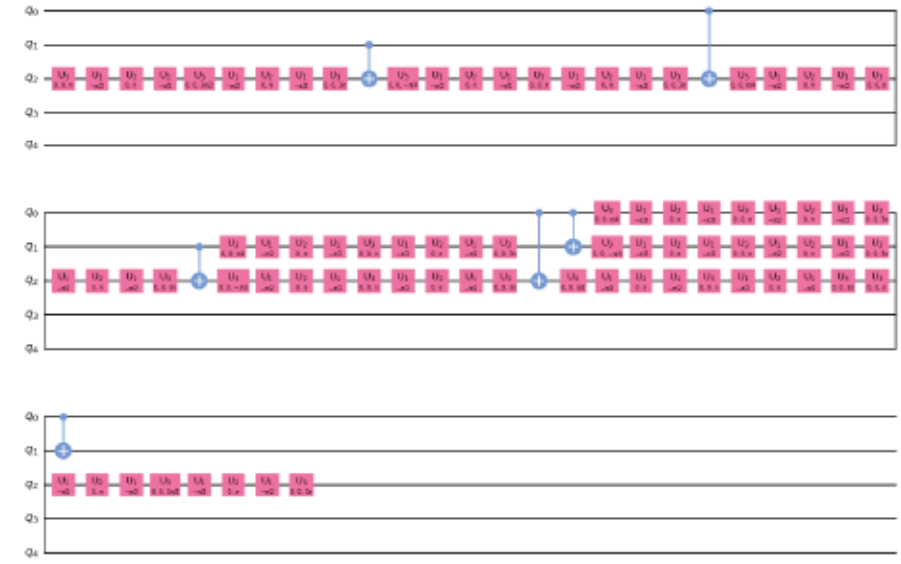
Figure 3: Example on how circuits can be decomposed into elementary operations.

and they are defined by a general unitary operation

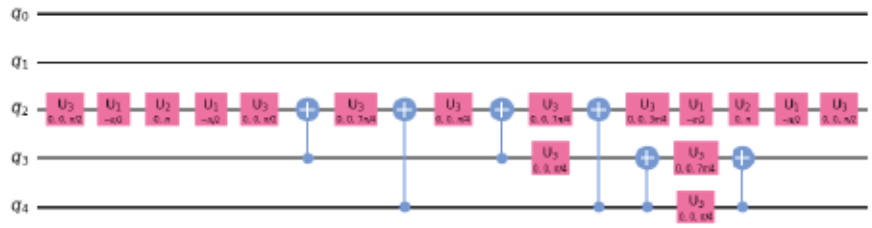
$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -e^{-i\lambda}\sin(\frac{\theta}{2}) \\ e^{i\theta}\sin(\frac{\theta}{2}) & e^{i\lambda+i\theta}\cos(\frac{\theta}{2}) \end{pmatrix}. \tag{9}$$

The only native multi-qubit operation is the two qubit CNOT gate. In Figure 3 a simple example circuit shows how a Toffoli gate is decomposed into the basic single qubit and CNOT operations. Taking the Toffoli gate from Figure 3 and transpiling it on the quantum computer Yorktown shows with optimisation levels 0 and 3 shows how the optimisation works. In Figure 4 it can be seen that with no optimisation the number of single qubit operations is greatly increased on the physical computer, but the number of CNOT operations remains the same. The optimised circuit shows a greatly reduced number of qubit operations, same number of CNOT gates and qubits q_3 and q_4 are used instead of q_0 and q_1.

In detail information on the operations performed by of the transpiler with optimisation level 3 can be found on online resources [25]. But two effects can be identified: first it tries to identify and choose the optimal qubits for a given circuit, in practice this could mean taking the qubits with smallest error rates and identifying qubits that take part in the largest number of CNOT gates and finding the solution that produces least cumulative error. Secondly single qubit operations not separated with barriers or CNOT gates are grouped and changed to operations that achieve the same state transformation for the qubit with least single qubit gates.



(a) No optimisation



(b) Optimisation level 3.

Figure 4: Example on how optimisation level affects the decomposition of the Toffoli gate after transpiling on the Yorkotown quantum computer.

The testing performed with different optimisation levels before the final experiments provided in this thesis lead to the choice of optimisation level 3 as the best choice here.

Finally the IBM quantum systems are limited on when the measurement can occur, the measurement process can happen only once and always at the very end of the circuit, which limits the design options for circuits and the ability to use measurements in the middle of circuit for any controlled actions later on.

Qiskit also provides tools for visualising measurement data via the open source

Python library Matplotlib. The Aer API in Qiskit provides tools for a local quantum simulator and an option for either ideal simulation or a simulation with a noise model.

2.1 Quantum teleportation

In 1993 Bennet *et al* [26] proposed a method, known as *quantum teleportation*, for transferring a quantum state via a combination of classical and Einstein-Podolsky-Rosen channels. Quantum teleportation is a technique for moving quantum states from one particle onto another without a direct quantum communications channel between the source and the destination after the initialisation [2]. However, even if the state is moved instantaneously, actual information is not transferred faster than light.

Following Nielsen's book [2], a quantum teleportation protocol transferring a single qubit state needs three qubits, as well as two classical bits to be transferred between Alice and Bob. Alice begins with two qubits, with one being the (arbitrary) transferred state

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{10}$$

and a second qubit that forms an entangled Bell pair with Bob's qubit. The pair can be in any of the Bell states

$$\begin{aligned} |\Phi^{\pm}\rangle &= \frac{|00\rangle \pm |11\rangle}{\sqrt{2}}, \\ |\Psi^{\pm}\rangle &= \frac{|01\rangle \pm |10\rangle}{\sqrt{2}}. \end{aligned} \tag{11}$$

Suppose Alice has the state $|\phi\rangle$ and shares the Bell state $|\Phi^{+}\rangle$ with Bob leading to the combined system being prepared in the state

$$\begin{aligned} |\psi_0\rangle &= \frac{1}{\sqrt{2}}\Big((\alpha|0\rangle + \beta|1\rangle)(|00\rangle + |11\rangle)\Big) \\ &= \frac{1}{\sqrt{2}}\Big[\alpha|0\rangle\big(|00\rangle + |11\rangle\big) + \beta|1\rangle\big(|00\rangle + |11\rangle\big)\Big] \end{aligned} \tag{12}$$

The teleportation protocol starts with Alice performing a CNOT operation between her qubits

$$|\psi_1\rangle = \text{CNOT}(0|1)\,|\psi_0\rangle = \frac{1}{\sqrt{2}}\Big[\alpha\,|0\rangle\,\big(\,|00\rangle + |11\rangle\,\big) + \beta\,|1\rangle\,\big(\,|10\rangle + |01\rangle\,\big)\Big] \tag{13}$$

followed by a Hadamard gate on the qubit she wants to teleport to Bob

$$\begin{aligned} |\psi_2\rangle &= H(0)\,|\psi_1\rangle \\ &= \frac{1}{2}\Big[\alpha(|0\rangle + |1\rangle)\big(\,|00\rangle + |11\rangle\,\big) + \beta(|0\rangle - |1\rangle)\big(\,|10\rangle + |01\rangle\,\big)\Big] \\ &= \frac{1}{2}\Big[\,|00\rangle\,\Big(\alpha\,|0\rangle + \beta\,|1\rangle\,\Big) + |01\rangle\,\Big(\alpha\,|1\rangle + \beta\,|0\rangle\,\Big) \\ &\qquad + |10\rangle\,\Big(\alpha\,|0\rangle - \beta\,|1\rangle\,\Big) + |11\rangle\,\Big(\alpha\,|1\rangle - \beta\,|0\rangle\,\Big)\Big]. \end{aligned} \tag{14}$$

Alice next performs a measurement on her qubits and sends the measurement results to Bob; by learning this information Bob can transform his qubit with a combination of X and Z-rotations in order to retrieve the state $|\phi\rangle$. This protocol requires that Alice and Bob have a classical form of information transfer and thus there is no faster than light transfer of information. Furthermore, this protocol does not contradict the no cloning principle [27] as instead of cloning, the state of the original qubit is altered and then destroyed in the measurement process. From state $|\psi_2\rangle$ it can be seen that Bob has four different possible outcomes and can therefore deduce their corresponding rotations to gain Alice's initial state based on her measurement result

$$\begin{aligned} 00 &: \alpha\,|0\rangle + \beta\,|1\rangle \rightarrow \quad \text{No rotations}, \\ 01 &: \alpha\,|1\rangle + \beta\,|0\rangle \rightarrow \quad X\text{-rotation}, \\ 10 &: \alpha\,|0\rangle - \beta\,|1\rangle \rightarrow \quad Z\text{-rotation}, \\ 11 &: \alpha\,|1\rangle - \beta\,|0\rangle \rightarrow \quad ZX\text{-rotation}. \end{aligned}$$

On IBM quantum systems this protocol does not work as is because the measurement of any qubits is only allowed at the end of a circuit. This means that Alice can't measure her qubits and send the information necessary for Bob to know

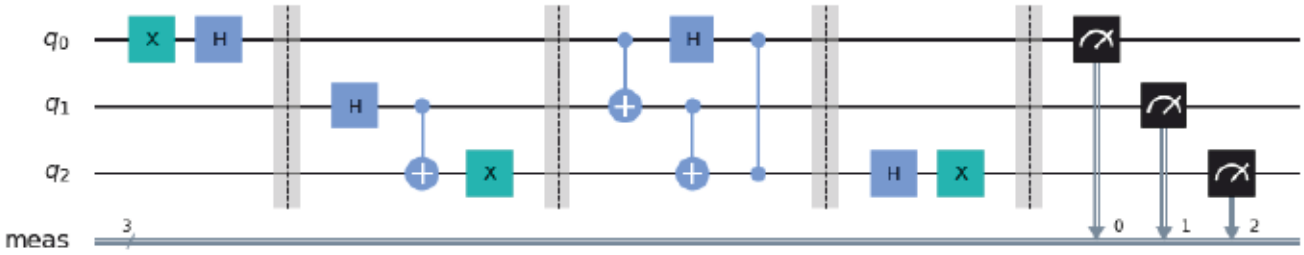


Figure 5: Quantum teleportation circuit drawn with Qiskit. Qubit q_0 is prepared before the first barrier, whilst qubits q_1 and q_2 form the Bell basis. The teleportation protocol is shown between second and third barrier and reversal of qubit q_0's preparation will return q_2 to state $|0\rangle$, followed by measurements after the last gate.

which actions are required to perform on his qubit in order for the protocol to work. This can be circumvented by controlled rotations which mimic the actions Bob takes based on Alice's measurement, after which the initialisation of the teleported state is reversed to $|0\rangle$ showing that the protocol worked in the final measurement.

The IBM quantum system Ourense was chosen for this test due to acceptable error rates with single qubit gates and qubit to qubit CNOT gates. Lower utilisation by other users also was a factor that resulted in significantly less queueing time for the circuits at the time the experiments were ran.

The circuit seen in Figure 5 starts with every qubit in state $|0\rangle$, with Alice in possession of qubits q_0 and q_1 and Bob the remaining qubit q_2. Here a secret state $|\phi\rangle = 1/\sqrt{2}(|0\rangle - |1\rangle)$, now $\alpha = \beta = 1/\sqrt{2}$, is prepared with a single X-rotation followed by a Hadamard gate on qubit q_0. The Bell state $|\Psi^+\rangle$ is chosen for qubits q_1 and q_2 and is achieved by Hadamard transformation on qubit q_1, a CNOT between qubits q_1 and q_2 and a X-rotation gate on qubit q_2. The system is prepared in the state

$$
\begin{aligned}
|\psi_0\rangle = |\phi, \Psi^+\rangle &= \frac{1}{\sqrt{2}}\Big((\alpha\,|0\rangle + \beta\,|1\rangle)(|00\rangle + |11\rangle)\Big) \\
&= \frac{1}{\sqrt{2}}\Big[\alpha\,|0\rangle\,\big(\,|01\rangle + |10\rangle\,\big) + \beta\,|1\rangle\,\big(\,|00\rangle + |11\rangle\,\big)\Big],
\end{aligned} \tag{15}
$$

where $\alpha = -\beta = 1/\sqrt{2}$. The teleportation of state q_0 begins as before with Alice performing a CNOT on q_0 and q_1 with q_0 as control, and performing a Hadamard gate on qubit q_0, resulting in the state

$$
\begin{aligned}
|\psi_{1a}\rangle &= H(q_0)(\mathrm{CNOT}(q_0|q_1)\,|\psi_0\rangle) \\
&= \frac{1}{2}\Big[\alpha(|0\rangle + |1\rangle)\big(\,|01\rangle + |10\rangle\,\big) + \beta(|0\rangle - |1\rangle)\big(\,|11\rangle + |00\rangle\,\big)\Big] \\
&= \frac{1}{2}\Big[\,|00\rangle\,\Big(\alpha\,|1\rangle + \beta\,|0\rangle\Big) + |01\rangle\,\Big(\alpha\,|0\rangle + \beta\,|1\rangle\Big) \\
&\qquad + |10\rangle\,\Big(\alpha\,|1\rangle - \beta\,|0\rangle\Big) + |11\rangle\,\Big(\alpha\,|0\rangle - \beta\,|1\rangle\Big)\Big].
\end{aligned} \tag{16}
$$

A CNOT and a Z- rotation is applied on qubit q_2, depending on the qubits q_1 and q_0 respectively

$$
\begin{aligned}
|\psi_{1b}\rangle &= \mathrm{CNOT}(q_1|q_2)\,|\psi_{1a}\rangle \\
&= \frac{1}{2}\Big[\,|00\rangle\,\Big(\alpha\,|1\rangle + \beta\,|0\rangle\Big) + |01\rangle\,\Big(\alpha\,|1\rangle + \beta\,|0\rangle\Big) \\
&\qquad + |10\rangle\,\Big(\alpha\,|1\rangle - \beta\,|0\rangle\Big) + |11\rangle\,\Big(\alpha\,|1\rangle - \beta\,|0\rangle\Big)\Big].
\end{aligned} \tag{17}
$$

And a controlled Z-rotation will change the sign of state $|1\rangle$ if the control q_0 is $|1\rangle$

$$
\begin{aligned}
|\psi_{1c}\rangle &= CZ(q_0|q_2)\,|\psi_{1b}\rangle \\
&= \frac{1}{2}\Big[\,|00\rangle\,\Big(\alpha\,|1\rangle + \beta\,|0\rangle\Big) + |01\rangle\,\Big(\alpha\,|1\rangle + \beta\,|0\rangle\Big) \\
&\qquad + |10\rangle\,\Big(-\alpha\,|1\rangle - \beta\,|0\rangle\Big) + |11\rangle\,\Big(-\alpha\,|1\rangle - \beta\,|0\rangle\Big)\Big].
\end{aligned} \tag{18}
$$

Finally the preparation done on q_0 is reversed. First a Hadamard gate is applied to

qubit q_2

$$
\begin{aligned}
|\psi_{2a}\rangle = \;& H(q_2)\,|\psi_{1b}\rangle \\
= \;& \tfrac{1}{2\sqrt{2}}\Big[\quad |00\rangle\Big(\alpha\big[\,|0\rangle - |1\rangle\,\big] + \beta\big[\,|0\rangle + |1\rangle\,\big]\Big) \\
& + \;|01\rangle\Big(\alpha\big[\,|0\rangle - |1\rangle\,\big] + \beta\big[\,|0\rangle + |1\rangle\,\big]\Big) \\
& + \;|10\rangle\Big(-\alpha\big[\,|0\rangle - |1\rangle\,\big] - \beta\big[\,|0\rangle + |1\rangle\,\big]\Big) \\
& + \;|11\rangle\Big(-\alpha\big[\,|0\rangle - |1\rangle\,\big] - \beta\big[\,|0\rangle + |1\rangle\,\big]\Big)\Big].
\end{aligned} \tag{19}
$$

Now in order to clean up, the values of α and β can be inserted

$$
|\psi_{2a}\rangle = \frac{1}{2}\Big[\Big(-|00\rangle - |01\rangle + |10\rangle + |11\rangle\Big)\,|1\rangle\Big]. \tag{20}
$$

The final X-rotation on qubit q_2 turns it to state $|0\rangle$

$$
|\psi_{2b}\rangle = X(q_2)\,|\psi_{2a}\rangle = \frac{1}{2}\Big[\Big(-|00\rangle - |01\rangle + |10\rangle + |11\rangle\Big)\,|0\rangle\Big]. \tag{21}
$$

The state above is the one measured with the measurement results given in Figure 6. Note that the measurements show the qubits in the opposite order with qubit q_2 being first and q_0 last. The measured states show qubit q_2 being in the zero state in all instances in the simulator and the real quantum computer to differ 8.9% at most, specifically on the state $|110\rangle$ in Figure 6. The results from the real quantum system also show a notably higher probability for first two qubits being in state $|00\rangle$ compared to them either being in state $|01\rangle$ or $|10\rangle$, and the case of state $|11\rangle$ being the least likely measurement outcome. Other quantum teleportation experiments on superconducting qubits have similar fidelity values [28], as do experiments with photons over a distance [29].

The calibration data of Ourense at the time the experiment was performed and the measurement counts can be found in Appendices A and B. The code for this experiment can be found on GitHub [30].

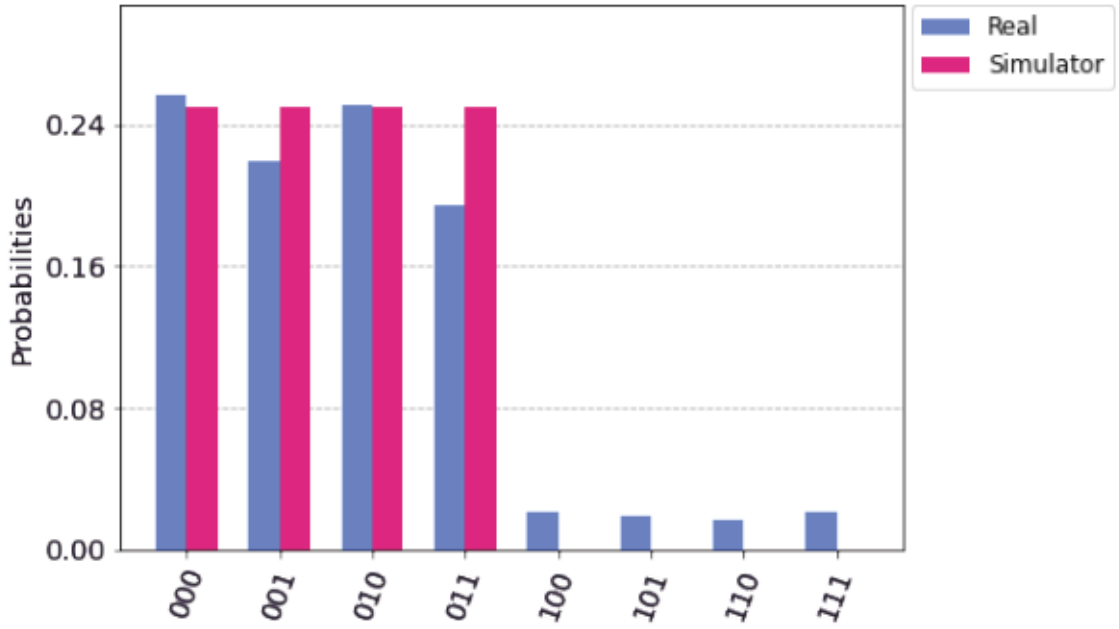


Figure 6: The results of the quantum teleportation simulation compared to the results from the real computer, with 8192 shots on each.

2.2 Deutsch–Jozsa algorithm

The Deutsch–Jozsa algorithm provides an answer to Deutsch's problem, which was constructed as an example to show how quantum algorithms can utilise quantum parallelism for better performance over classical algorithms in solving probability-based problems [31]. In Deutsch's problem Alice guesses whether the function f, which Bob has, is either a constant or balanced function by using as few messages as possible. Each message Alice sends Bob has n bits of information. The function $f : \{0,1\}^n \to \{0,1\}$ is constant if f is a fixed value for all possible choices of x and balanced when $f(x)$ is equal to one for half of all values of values of x. There is no known application for the Deutsch–Jozsa algorithm except to provide an (artificial) example of a situation in which quantum parallelism proves superior to classical information processing [2].

In the classical case, Alice can send only one query of x in each message leading to Alice having to send at least $2^n/2 + 1$ queries, since she could receive up to $2^n/2$

identical results before receiving a different one (in case that f is balanced). This is in fact the best possible rate Alice can classically obtain [2].

By utilising quantum parallelism the Deutsch–Jozsa algorithm can deduce whether $f(x)$ is constant or balanced with a single query from Alice to Bob. The function $f(x)$ is calculated with a unitary transformation U_f, Alice has a query qubit register of n qubits and an answer register of one qubit is shared between Alice and Bob. The system is initially in the state

$$|\psi_0\rangle = |0\rangle^{\otimes n} |1\rangle , \tag{22}$$

where state $|1\rangle$ is the answer register and all preceding states represent the query register of n qubits. Alice then prepares the query and answer qubit registers with Hadamard transformations leaving them in the equal superposition

$$|\psi_1\rangle = H^{\otimes n} |\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]. \tag{23}$$

Now the query register is in a superposition of all possible input values, and the answer register is in an evenly weighed superposition of 0 and 1. At this point it is Bob's turn to evaluate $f(x)$ using the unitary transformation $U_f : |x, y\rangle \mapsto |x, y \oplus f(x)\rangle$, where the operator $\oplus$ denotes addition modulo 2, giving

$$|\psi_2\rangle = U_f |\psi_1\rangle = \sum_x \frac{(-1)^{f(x)} |x\rangle}{\sqrt{2^n}} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]. \tag{24}$$

Bob's evaluated function is now stored in the amplitudes of Alice's superposition state. Alice now applies a Hadamard transformation upon her query register. By checking the effect of the Hadarmard transformation on the state $|x\rangle$ where $x = 0$ and $x = 1$ separately, it can be seen that for a single qubit the transform can be expressed as $H|x\rangle = \sum_z (-1)^{xz} |z\rangle / \sqrt{2}$, which leads to the following expression for multiple qubits

$$H^{\otimes n} |x_1, \ldots, x_n\rangle = \frac{\sum_{z_1,\ldots,z_n} (-1)^{x_1 z_1 + \cdots + x_n z_n} |z_1, \ldots, z_n\rangle}{\sqrt{2^n}}. \tag{25}$$

This can be written more concisely [2, page 36] as

$$H^{\otimes n} |x\rangle = \frac{\sum_z (-1)^{x \cdot z} |z\rangle}{\sqrt{2^n}}, \tag{26}$$

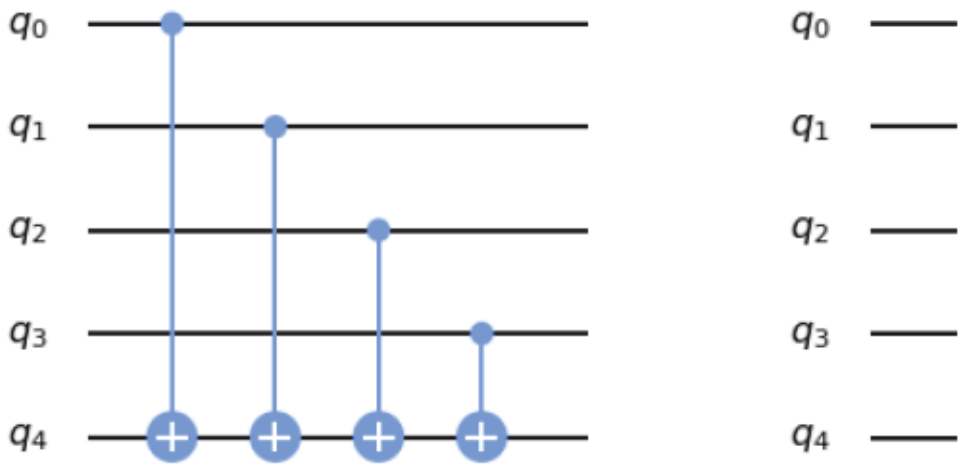
where $x \cdot z$ is the bitwise inner product of x and z modulo 2. By using equations (24) and (26) it is possible to evaluate the Hadamard transformation Alice performs on the query register

$$|\psi_3\rangle = H^{\otimes n} |\psi_2\rangle = \sum_z \sum_x \frac{(-1)^{x \cdot z + f(x)} |z\rangle}{2^n} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right]. \tag{27}$$

By now observing the query register Alice can determine if $f(x)$ is constant or balanced. The amplitude for the state $|0\rangle^{\otimes n}$ is $\sum_x (-1)^{f(x)}/2^n$, and if $f(x)$ is constant the amplitude for $|0\rangle^{\otimes n}$ is either $+1$ or -1. Because $|\psi_3\rangle$ is normalised, all other components of the superposition must be equal to zero in this case. This results in observation yielding zeros for all qubits in the query register. By contrast, if the function $f(x)$ is balanced, the positive and negative contributions to the amplitude for $|0\rangle^{\otimes n}$ cancel, leading to an amplitude of zero, thereby requiring at least one of the query qubits to be other than zero. In conclusion, if Alice measures all query qubits as $|0\rangle$ the function is constant, otherwise it is balanced.

The constant oracle unitary transform U_f Bob performs on the query register has no limitations since there are no CNOT gates between qubits and poses no scaling limitations. The balanced oracle on the other hand limits the scalability on IBM quantum systems, as the answer register qubit has to be connected to every qubit on query register. By choosing the IBM Yorktown the algorithm works up to a 4-qubit query register, with the center qubit being physically connected to all other qubits. The oracles used in the experiment can be seen in Figure 7. Qubits q_0, q_1, q_2 and q_3 are chosen as the query register and qubit q_4 as the answer register. The first experiment, seen in Figure 8, is done on circuit with a balanced oracle.

Qubits in the query register are prepared with a Hadamard gate on each, whilst the answer register is subject to an X-rotation gate before a Hadamard.The system



(a) Balanced oracle.

(b) Constant oracle.

Figure 7: Example of circuits to represent balanced and constant oracles with qubits $q_0 - q_3$ belonging to the query register and last qubit in the answer register. Note that no operations are performed in the constant oracle.

is therefore prepared in the state

$$|\psi_0\rangle = |++++-\rangle\,, \tag{28}$$

In the $|\pm\rangle$ basis the target and control are switched via a CNOT operation, thus the oracle will transform the state

$$|\psi_1\rangle = \mathrm{CNOT}(q_4|q_0, q_1, q_2, q_3)\,|\psi_0\rangle = |-----\rangle\,. \tag{29}$$

The transformation that the balanced oracle performs on the query register adds negative phase to exactly half of the states in the query register, with the state of the register after query orthogonal to the quantum state of the register before querying the oracle. The final Hadamard transformation the qubits in the query register will transform the state into

$$|\psi_2\rangle = H(q_0, q_1, q_2, q_3)\,|\psi_1\rangle = |1111-\rangle\,. \tag{30}$$

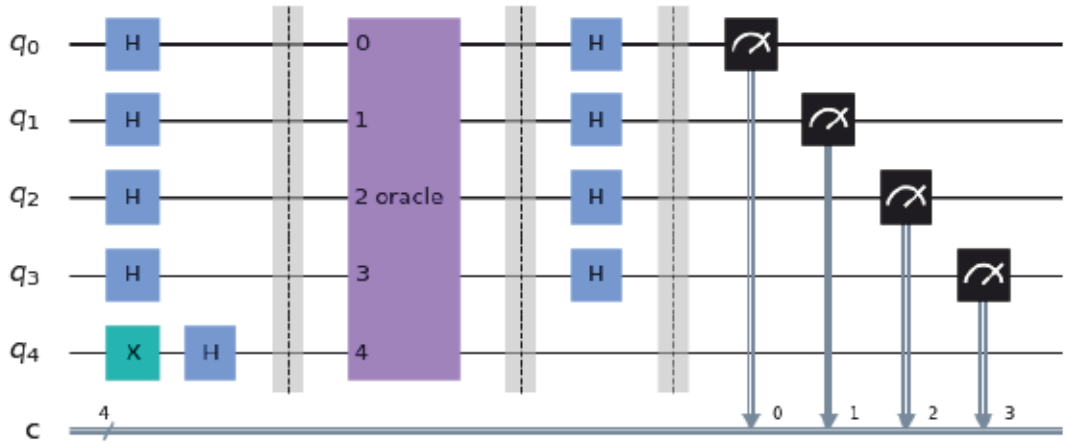


Figure 8: Deutsch–Jozsa experiment circuit.

The query register can now be measured and is in the state $|1111\rangle$ which is orthogonal to the initial state $|0000\rangle$.

By contrast, an experiment with a constant oracle using the same circuit as in Figure 8 and is prepared in exactly the same way as the first experiment, differing only in the chosen oracle (see Figure 7). The prepared state of the system

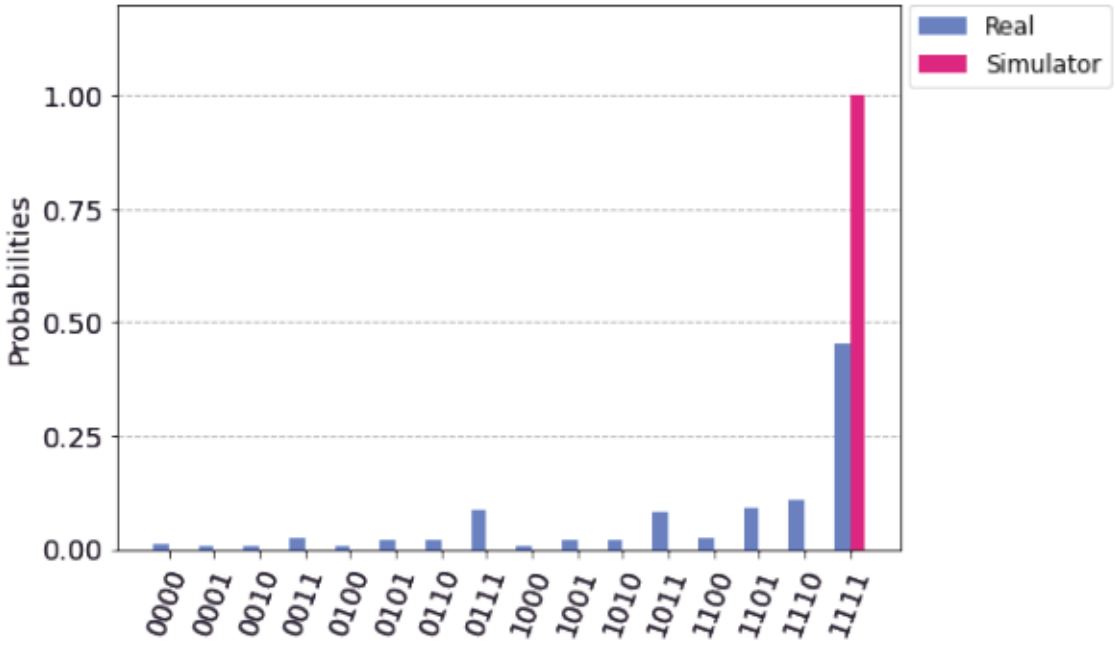
$$|\psi_0\rangle = |++++-\rangle \tag{31}$$

experiences no transformation on the query register and the Hadamard transformation the query register changes the state into

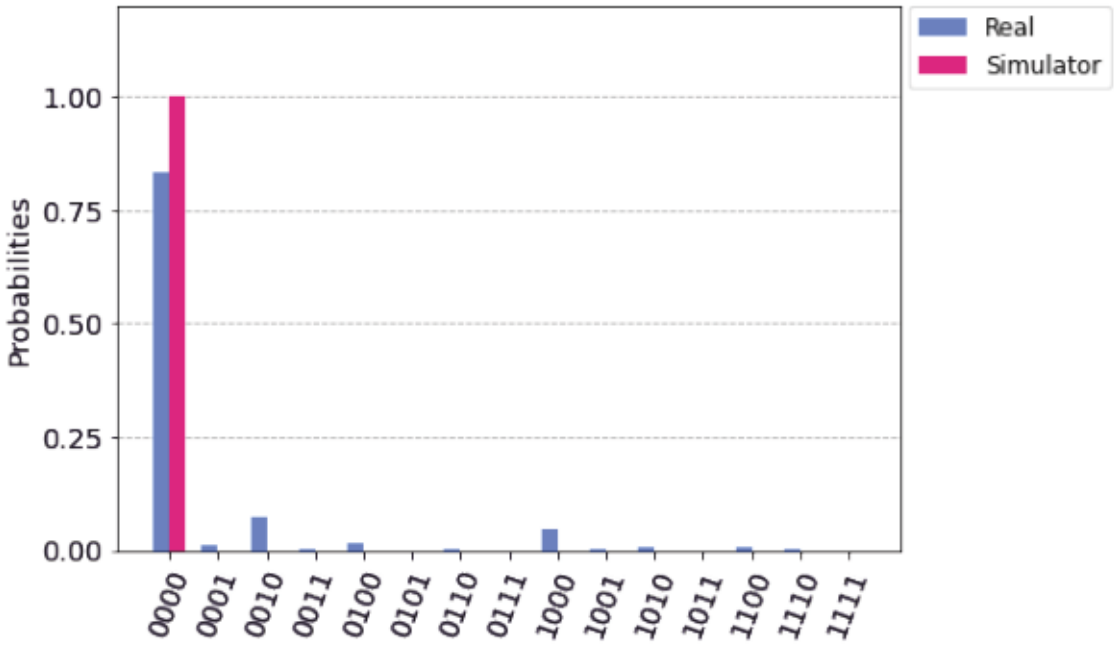
$$|\psi_1\rangle = H(q_0, q_1, q_2, q_3)\,|\psi_0\rangle = |0000-\rangle\,, \tag{32}$$

which should result in measuring state $|0000\rangle$.

The results of the Deutsch–Jozsa experiment, shown in Figure 9, show that the constant oracle has a smaller absolute error rate when measuring the desired state compared to the balanced oracle. The constant-oracle measurement gives the correct result with 83.3% probability, with one notable error for the state $|0100\rangle$ (reversed in Figure 9), appearing with 7.4% probability. This spike might be due to a physical



(a) Balanced oracle.



(b) Constant oracle.

Figure 9: The results of Deutsch–Jozsa experiment comparing the simulation to the results from the real quantum computer, with 8192 shots on each.

qubit having a high error rate for single qubit operations. The balanced oracle experiment gives the expected result of $|1111\rangle$ as the most likely, but with more net noise and much smaller absolute probability. However, as mentioned earlier, since Alice needs to measure $|0\rangle$ for each qubit in order to conclude a constant oracle, and this is found for 1.2% of instances, in conclusion the experiments with a balanced oracle we correctly guess the right oracle with probability nearly 99% probability.

Other measurements done with the scalable algorithm here show that with a smaller query register the probability of the correct measurement rises. Further error mitigation with a query register of 4 qubits on Yorktown might not be possible, due to the already simple structure of the circuit and minimal operations. An experiment by Gulde *et al* in 2003 done on ion-trap quantum computers with two qubits showed experimental success of over 90% [32], which is considerably higher compared to the constant oracle experiments with 4 superconducting qubits here. The calibration data of Yorktown at the time the experiment was performed and the measurement counts can be found in Appendices A and B. The code for this experiment can be found on GitHub [30].

2.3 Bernstein–Vazirani algorithm

Another algorithm to show the superiority of quantum parallelism is the Bernstein–Vazirani algorithm. Bernstein and Vazirani studied quantum computation from a complexity theory standpoint in their 1997 article [5] and found a problem a quantum machine is capable of solving exponentially faster than a classical computer. The Bernstein–Vazirani problem has an unknown function $f(x)$ which takes as a input a string of bits (x) and returns either zero or one as output

$$f : \{0,1\}^n \to \{0,1\}. \tag{33}$$

The Bernstein–Vazirani problem has similarities with the Deutsch–Jozsa problem and can be thought as an extension of it. Instead of having a balanced or constant function $f(x)$, the function is now guaranteed to return the bitwise product of the input with an unknown string $b \in \{0,1\}^n$. Given an input $x \in \{0,1\}^n$ the function outputs $f(x) = b \cdot x$ (modulo 2). The aim of the problem is to find what the unknown bitstring b is.

The classical solution can have only one input x_i

$$\begin{aligned} x_0 &= 10\ldots0 \\ x_1 &= 01\ldots0 \\ &\vdots \\ x_n &= 00\ldots1, \end{aligned}$$

with each query revealing a different bit b_i of b. As a result of this is that the classical queries need to be ran n times.

The quantum solution can solve the problem with confidence after only one call of the function $f(x)$. The input register consists of n qubits and one output qubit initialised in state $|-\rangle$, with the whole system being in state

$$|\psi_0\rangle = |0\rangle^{\otimes n} |-\rangle . \tag{34}$$

A Hadamard transformation is applied on the input register

$$|\psi_1\rangle = H^{\otimes n} |\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |-\rangle . \tag{35}$$

Next the oracle is applied to the input register

$$|\psi_2\rangle = f(x) |\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{b \cdot x} |x\rangle |-\rangle . \tag{36}$$

Finally, b can be obtained by running the Hadamard transformation on the qubits in the input register and transforming the output register back to state $|0\rangle$ with a

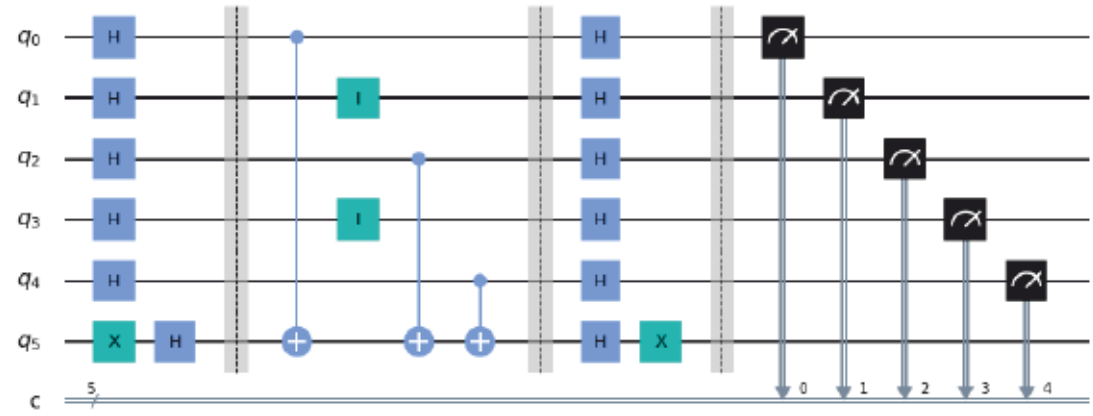


Figure 10: Bernstein–Vazirani algorithm circuit with bitstring $b = 10101$.

X rotation after applying a Hadamard on it

$$|\psi_3\rangle = X(q_{n+1})(H^{\otimes n+1}\,|\psi_2\rangle) = |b\rangle\,|0\rangle\,. \tag{37}$$

Now by measuring the input register Alice has the unknown bitstring b with just a single query to Bob.

When applying this algorithm as a circuit on the IBM quantum computer it is possible to create a big scalable circuit with low error rates if the unknown bitstring has some limitations. By utilising the 15 qubit Melbourne system it is possible to create circuits with query registers up to 14 qubits by limiting the bitstring to have up to 3 ones in order to limit the total error rate. In Figure 10 a bitstring $b = 10101$ is selected as a result on a register of $n = 5$ qubits $q_0, \ldots, q_4$ and the target register as q_5. Initialisation starts by X-rotation gate on qubit q_5 followed by Hadamard transformation all qubits

$$|\psi_0\rangle = |+++++\rangle\,|-\rangle\,, \tag{38}$$

where the target register is kept separate for clarity. The oracle does nothing on qubits q_1 and q_3 and applies multiple CNOT gates using qubits q_0, q_2 and q_4 as controls and q_5 as targets

$$|\psi_1\rangle = \mathrm{CNOT}(q_0, q_2, q_4|q_5)\,|\psi_0\rangle = |-+-+-\rangle\,|-\rangle\,. \tag{39}$$

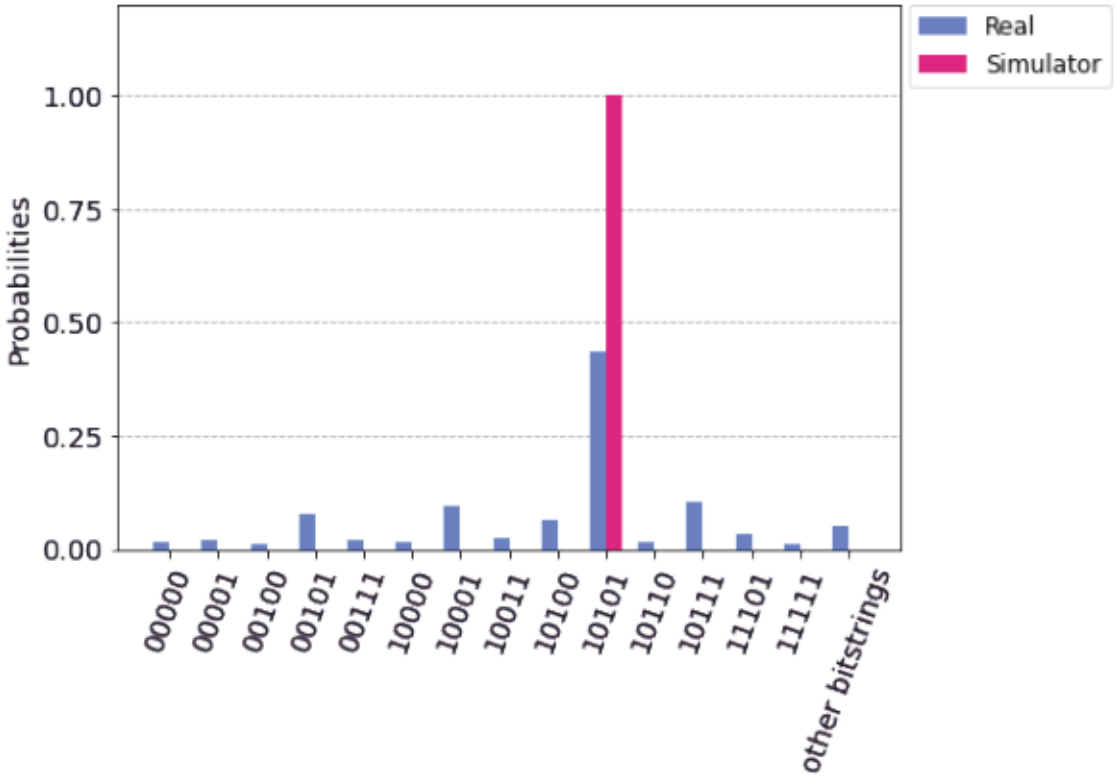


Figure 11: The results of Bernstein–Vazirani experiment comparing the simulated results to those from the real quantum computer, with 8192 shots on each. The results of query register and events with over 1% probability are visible, whereas events below the 1% threshold are collected under 'other bitstrings'.

The initial preparation is done in inverse on all qubits followed by a final measurement on all qubits

$$|\psi_2\rangle = X(q_5)(H(q_0, q_1, q_2, q_3, q_4, q_5)\,|\psi_1\rangle) = |10101\rangle\,|0\rangle = |b\rangle\,|0\rangle\,. \tag{40}$$

Now the query register is in the state $|b\rangle$, which is reflected by the results in Figure 11, where there is a spike corresponding to the measurement outcome $|10101\rangle$. IBM Melbourne has a relatively large error rate on single qubit gates and CNOT gates between qubits resulting in noticeable noise in the results, with the rare (those with less than 1% probability) events collected in 'other bitstrings' summed up to a total probability of 5%. The results show higher probabilities for events that partially

correspond to the bitstring, with the second most likely event being $|11101\rangle$ with 10.6% and third being $|10001\rangle$ with 9.5%. The calibration data of Melbourne at the time the experiment was performed and the measurement counts can be found in Appendices A and B. The code for this experiment can be found on GitHub [30].

2.4 Simon's algorithm

Simon in his 1997 article [33] presented a problem and an algorithm to show how a function supplied as an oracle can outperform a classical computer solving the same problem. Simon's algorithm solves Simon's problem, in which an unknown function $f(x)$ is guaranteed to be either a one-to-one mapping or two-to-one mapping. The function $f : \{0,1\}^n \to \{0,1\}^n$ maps one bitstring $x \in \{0,1\}^n$ to another $y \in \{0,1\}^n$ according to a hidden bitstring $b \in \{0,1\}^n$

$$f(x) = f(y) \iff y = x \oplus b, \tag{41}$$

where $\oplus$ is binary addition modulo 2.

In case of a one-to-one mapping a function maps one unique input bitstring x_i to one unique output bitstring $f(x_i) = y_i$ and the hidden bitstring is all zeros, $b = 0$. Two-to-one mapping on the other hand maps two unique inputs x_i, x_j to a single unique output $f(x_i) = f(x_j) = y_k$, where $i \neq j$, and the unknown bitstring

$$b = x_i \oplus x_j. \tag{42}$$

The goal of Simon's problem is to answer how quickly it is possible to determine the unknown bitstring b.

In the classical solution in order to fully determine b for a given $f(x)$ it is necessary to check up to $2^{n-1} + 1$ inputs, the number n is the number of bits in the input. There exists deterministic classical algorithms that can find the answer with a lower bound of $\Omega(2^{n/2})$ queries [34], yet in general it can be agreed upon that the complexity grows exponentially with n.

In the quantum solution to Simon's problem there are two n qubit registers. Both registers are initialised in the zero state so the whole system is initially in state

$$|\psi_0\rangle = |0\rangle^{\otimes n} |0\rangle^{\otimes n}. \tag{43}$$

During the second preparation step a Hadamard transformation is applied on the first register

$$|\psi_1\rangle = H^{\otimes n} |\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle |0\rangle^{\otimes n}. \tag{44}$$

Now it is time to query the oracle U_f, which transforms the second register using the first register as input

$$|\psi_2\rangle = U_f |\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle |f(x)\rangle. \tag{45}$$

After querying the oracle the second register is measured. A certain value of $f(x)$ will be observed, resulting in value of $f(x)$ corresponding to two possible inputs: x and $y = x \oplus b$, therefore the first register becomes

$$|\psi_3\rangle = \mathrm{M}(f(x)) |\psi_2\rangle = \frac{1}{\sqrt{2}}(|x\rangle + |y\rangle). \tag{46}$$

A Hadamard is applied on the first register

$$|\psi_4\rangle = H^{\otimes n} |\psi_3\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{z\in\{0,1\}^n} [(-1)^{x\cdot z} + (-1)^{y\cdot z}] |z\rangle, \tag{47}$$

followed by a measurement of the first register. The first register will give an output only if the sum is nonzero

$$(-1)^{x\cdot z} = (-1)^{y\cdot z}, \tag{48}$$

which means for the exponents that

$$\begin{aligned} x\cdot z &= y\cdot z \\ &= (x\oplus b)\cdot z \\ &= x\cdot z \oplus b\cdot z \\ \Rightarrow b\cdot z &= 0 \pmod 2. \end{aligned} \tag{49}$$

The measurements lead to obtaining string z. Because the inner product modulo 2 between z and the unknown bitstring b will produce zero, it can be used to calculate the unknown bitstring. By repeating Simon's algorithm n times, the algorithm will produce n different values of z resulting in a system of equations

$$\begin{cases} b \cdot z_1 & = 0 \\ \vdots & \\ b \cdot z_n & = 0 \end{cases}, \tag{50}$$

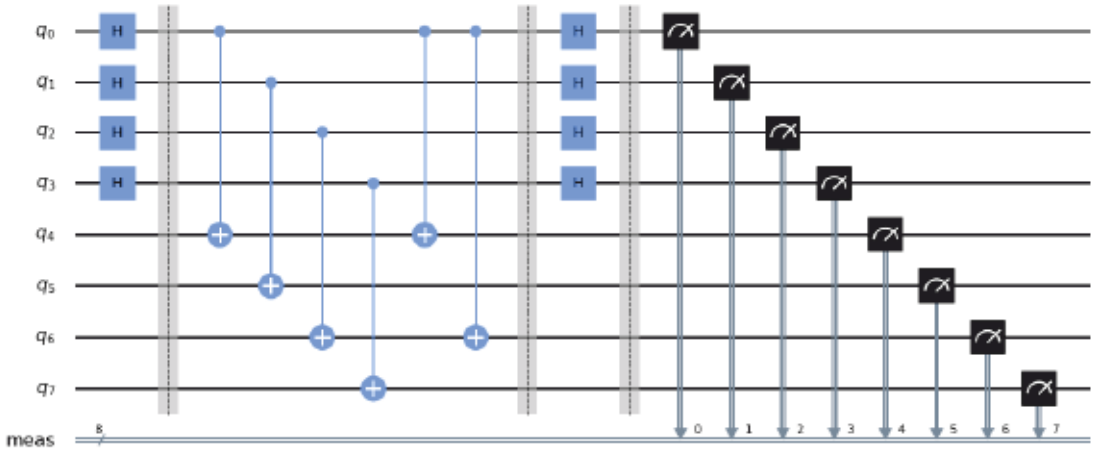
from which the unknown bitstring b can be determined. This shows that the classical algorithm requires exponentially more steps compared to Simon's quantum algorithm.

In the Qiskit implementation used here, the oracle U_f in equation (45) depends on the properties of $f(x)$, and starts identically in all possible cases by copying the first register to the second register and then the oracle performs a CNOT operation using bitstring b

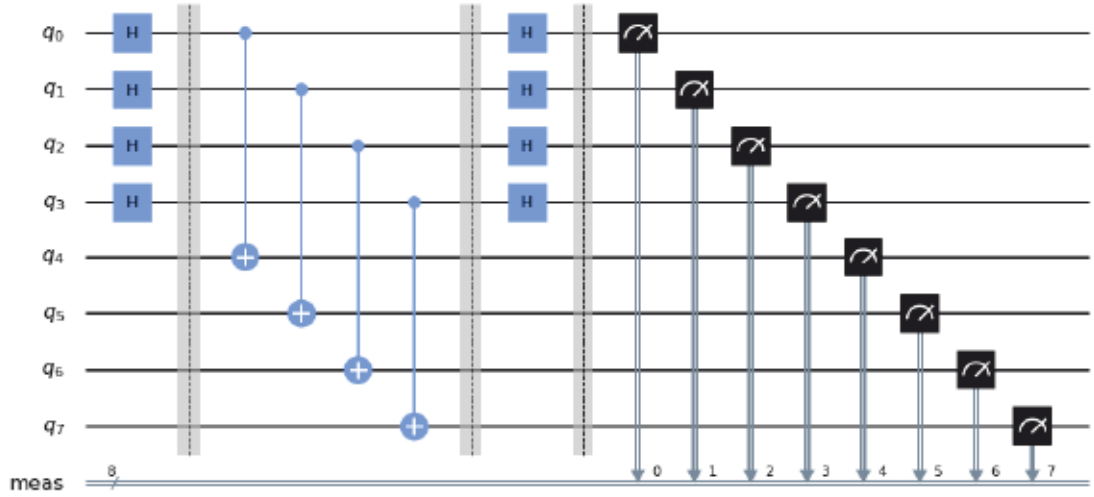
$$|x\rangle |0\rangle \rightarrow |x\rangle |x \oplus b\rangle . \tag{51}$$

The index i of the first non-zero bit in bitstring b is used to choose a corresponding qubit x_i in the first register to be used as the control for all the non-zero indices j in b for operating on the corresponding qubits second register. If the bitstring b is all zeros no actions are performed after the copying.

The demand for two n qubit registers, coupled with the easy scalability of the circuit design, leads to the opportunity to use larger quantum computers, such as the 16-qubit Melbourne. Since Bob's 'unknown' bitstring must be predetermined and only the first nonzero bitstring in the Oracle is used as the control for the nonzero targets in the target register, the only error reducing factor becomes the physical layout of the CNOT gates between physical qubits. This allows us to create a random bitstring with up to three nonzero bits. Two cases considered here have two-to-one bitstring $b = 1010$ and one-to-one bitstring $b = 0000$, thus two quantum



(a) Oracle with bitstring $b = 1010$.



(b) Oracle with bistring $b = 0000$.

Figure 12: Circuits for Simon's algorithm.

registers of $n = 4$ qubits, with qubits $q_0, ..., q_3$ in the first and qubits $q_4, ..., q_7$ in the second. The circuits are shown in Figure 12. Starting with the two-to-one experiment, the qubits in the first register are prepared with Hadamard gates

$$\begin{aligned} |\psi_0\rangle &= |++++\rangle\,|0000\rangle \\ &= \frac{1}{\sqrt{2^4}}\Big(|0000\rangle + |0001\rangle + \cdots |1111\rangle\Big)\,|0000\rangle \end{aligned} \tag{52}$$

The oracle performs a CNOT operation using the first register as control and the second register as the target

$$|\psi_{1a}\rangle = \mathrm{CNOT}(q_i|q_j)\,|\psi_0\rangle = \frac{1}{\sqrt{2^4}}\Big(\quad |0000\rangle\,|0000\rangle + \quad |0001\rangle\,|0001\rangle \quad \vdots \quad + \quad |1111\rangle\,|1111\rangle\Big), \tag{53}$$

where $i = 0, 1, 2, 3$ and $j = 4, 5, 6, 7$. The oracle uses the qubit q_0 representing the first nonzero bit $b = 1010$ as control and operates on the second register

$$\begin{aligned} |\psi_{1b}\rangle &= U_f\,|\psi_{1a}\rangle \\ &= \frac{1}{\sqrt{2^4}}\Big(\big(|0000\rangle + |1010\rangle\big)\,|0000\rangle + \big(|0001\rangle + |1011\rangle\big)\,|0001\rangle \\ &\quad + \big(|0010\rangle + |1000\rangle\big)\,|0010\rangle + \big(|0011\rangle + |1001\rangle\big)\,|0011\rangle \\ &\quad + \big(|0100\rangle + |1110\rangle\big)\,|0100\rangle + \big(|0101\rangle + |1111\rangle\big)\,|0101\rangle \\ &\quad + \big(|0110\rangle + |1100\rangle\big)\,|0110\rangle + \big(|0111\rangle + |1101\rangle\big)\,|0111\rangle\Big), \end{aligned} \tag{54}$$

where the possible outcomes for the first registers are grouped with the corresponded outcomes on the second register. Finally the last Hadamard transformation each qubit in the first register is performed, and the resulting state has 8 different measurement outcomes for the first register for each possible outcome on the second register that only differ in the phase

$$\frac{1}{8}\quad\Big(\quad |0000\rangle \pm |0001\rangle \pm |0100\rangle \pm |0101\rangle \quad \pm \quad |1010\rangle \pm |1011\rangle \pm |1110\rangle \pm |1111\rangle\Big)\,|x\rangle\,, \tag{55}$$

where $|x\rangle$ are the possible outcomes for the second register (the second register is

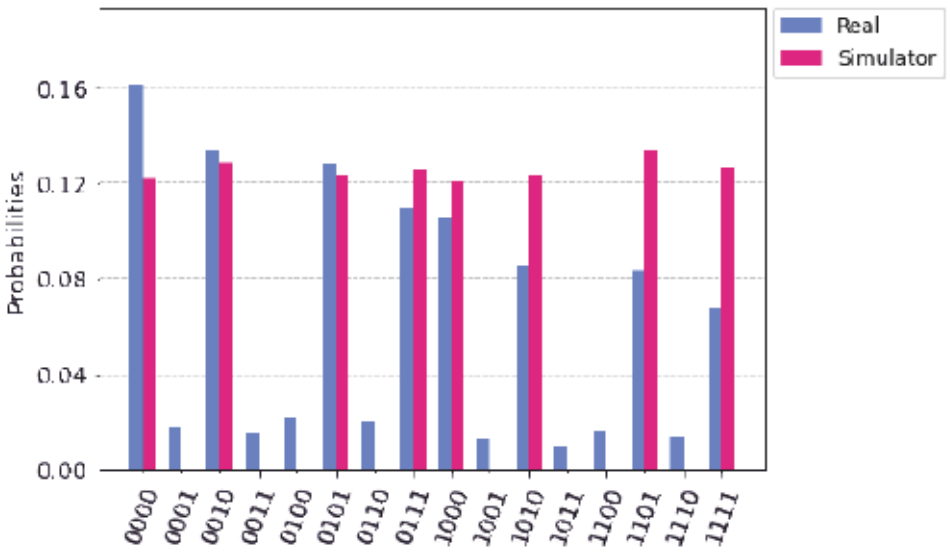


Figure 13: The results of two-to-one experiment comparing the simulation to the results from the real quantum computer, with 8192 shots on each.

omitted from the results seen in Figure 13). The possible measurement outcomes form a group of equations similarly to (50)

$$\begin{cases} b \oplus 0000 & = b_0 \cdot 0 + b_1 \cdot 0 + b_2 \cdot 0 + b_3 \cdot 0 = 0 \pmod 2 \\ b \oplus 0001 & = b_0 \cdot 0 + b_1 \cdot 0 + b_2 \cdot 0 + b_3 \cdot 1 = 0 \pmod 2 \\ b \oplus 0100 & = b_0 \cdot 0 + b_1 \cdot 1 + b_2 \cdot 0 + b_3 \cdot 0 = 0 \pmod 2 \\ b \oplus 0101 & = b_0 \cdot 0 + b_1 \cdot 1 + b_2 \cdot 0 + b_3 \cdot 1 = 0 \pmod 2 \\ b \oplus 1010 & = b_0 \cdot 1 + b_1 \cdot 0 + b_2 \cdot 1 + b_3 \cdot 0 = 0 \pmod 2 \\ b \oplus 1011 & = b_0 \cdot 1 + b_1 \cdot 0 + b_2 \cdot 1 + b_3 \cdot 1 = 0 \pmod 2 \\ b \oplus 1110 & = b_0 \cdot 1 + b_1 \cdot 1 + b_2 \cdot 1 + b_3 \cdot 0 = 0 \pmod 2 \\ b \oplus 1111 & = b_0 \cdot 1 + b_1 \cdot 1 + b_2 \cdot 1 + b_3 \cdot 1 = 0 \pmod 2 \end{cases}$$

Solving this collection of simultaneous equations above gives the bitstring $b = b_0b_1b_2b_3 = 1010$ as the only possibility, thereby showing that the algorithm works.

The second experiment with the one-to-one oracle is made by using bitstring

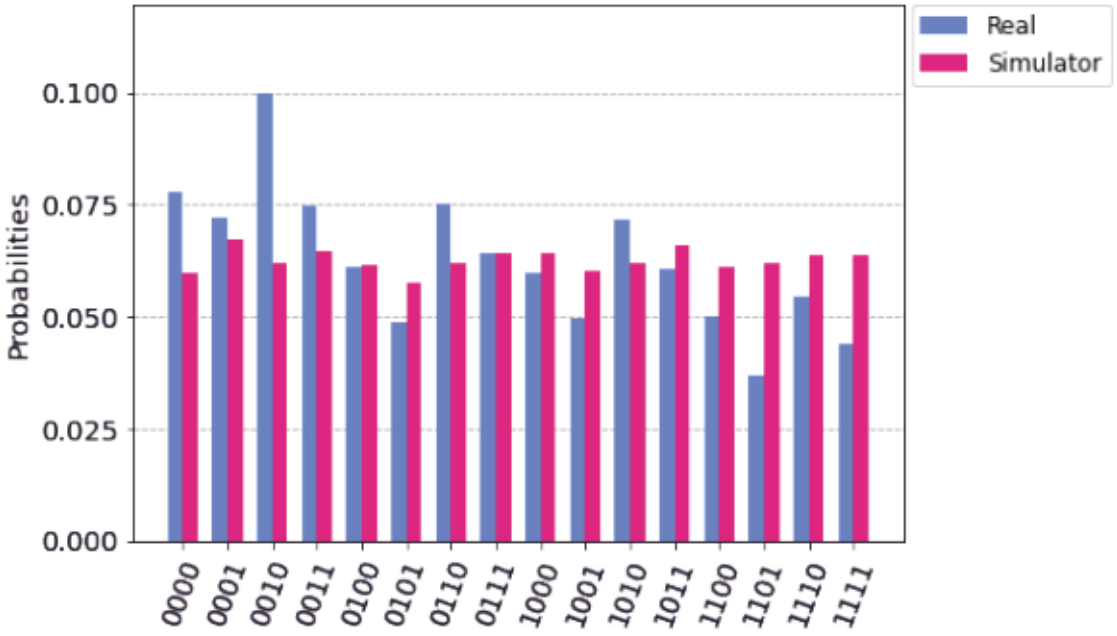


Figure 14: Results of one-to-one experiment comparing the simulation to the results from the real quantum computer, with 8192 shots on each.

$b = 0000$, but preparations are otherwise the same, leading to state $|\psi_1\rangle$ identical to state $|\psi_{1a}\rangle$ in equation (53).

Since all bits in the hidden bitstring are zero, no CNOT operations are performed on $|\psi_1\rangle$. The final Hadamard on the qubits in first register leads to the final state

$$\begin{aligned} \frac{1}{16} \Big(& |0000\rangle \pm |0001\rangle \pm |0010\rangle \pm |0011\rangle \\ \pm & |0100\rangle \pm |0101\rangle \pm |0110\rangle \pm |0111\rangle \\ \pm & |1000\rangle \pm |1001\rangle \pm |1010\rangle \pm |1011\rangle \\ \pm & |1100\rangle \pm |1101\rangle \pm |1110\rangle \pm |1111\rangle \Big) |x\rangle \,, \end{aligned} \tag{56}$$

where $|x\rangle$ is any of the possible outcomes for the second register. The measurement outcomes for the one-to-one experiment can be seen in Figure 14.

The results of the experiment seen in Figure 14 show a universal bias towards any qubit being found in state $|0\rangle$, this is phenomenon is found in both experiments. In the two-to-one this leads to event $|0000\rangle$ being close to twice as likely as $|1111\rangle$, with

the simulator providing nearly uniform probabilities between events. As expected the simulator provided no false results, unlike the real experiment which provided false results with 12.6% probability. The one-to-one experiment has reducing probability from $|0000\rangle$ to $|1111\rangle$ with the most likely being more than twice as likely as the rarest event. The calibration data of Melbourne at the time the experiment was performed and the measurement counts can be found in Appendices A and B. The code for this experiment can be found on GitHub [30].

2.5 Grover's search algorithm

Searching is a fundamental problem in data processing, and for classical search algorithms the efficiency of searching depends on the order of elements and size of the search database. Searching unstructured datasets is a probabilistic process, the search process can be made faster by utilising quantum parallelism. In an imaginary case with a large list of N items, there is one item of interest. By using classical computation and checking items at random, there is a $1/N$ possibility to hit the right item but it could take checking all N items to find the right one, and on average it takes $N/2$ tries to find the right item [35]. Grover created in his 1996 article [35] an algorithm to show that with quantum parallelism the average number of attempts can be reduced to $\sqrt{N}$. Furthermore, Grover's search algorithm is generic: the speed of the quantum algorithm does not depend on the internal structure of the search database.

Grover's algorithm utilises a unitary transform, oracle, to distinguish the desired outcome state. Let a three qubit database be comprised of all the possible computational basis states qubits can be in, $|000\rangle, |001\rangle, \ldots, |111\rangle$. The oracle in Grover's algorithm adds a negative phase to the solution states. The oracle tests if for a given input state x_i $f(x_i) = 1$ then $x_i = \omega$, otherwise $f(x_i) = 0$. In other words when ω

is the desired state and x a random state then a unitary transform to find ω is

$$U_\omega |x\rangle = \begin{cases} |x\rangle \text{ if } x \neq \omega \\ -|x\rangle \text{ if } x = \omega \end{cases}. \tag{57}$$

The unitary transform U_ω representing oracle is a diagonal matrix, where the entry that corresponds to the desired item will have a negative phase. If $\omega = 110$, the oracle matrix will be

$$U_\omega = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{58}$$

The aim of Grover's algorithm is to convert all problems to this diagonal form. Again, all the possible solutions x are used as input and tested with a function $f(x)$, that returns $f(x) = 0$ if $x \neq \omega$ and $f(x) = 1$ when $x = \omega$. With this function the oracle can be used as a unitary operation on an arbitrary input state $|x\rangle$

$$U_\omega |x\rangle = (-1)^{f(x)} |x\rangle \tag{59}$$

and the oracle itself has a diagonal form

$$U_\omega = \begin{bmatrix} (-1)^{f(0)} & 0 & \cdots & 0 \\ 0 & (-1)^{f(1)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (-1)^{f(2^n)} \end{bmatrix}. \tag{60}$$

Without knowing where the desired item is, a guess can be represented with a uniform superposition

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \,. \tag{61}$$

In a standard basis $\{|x\rangle\}$ this superposition would collapse to any one of the basis states with the same probability of $1/N = 1/2^n$, therefore chances of guessing ω is $1/2^n$. Amplitude amplification can increase the amplitude of the desired outcome and thus increase its likelihood.

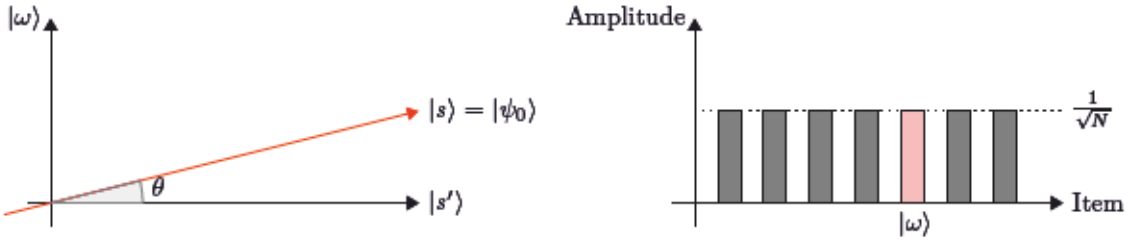


Figure 15: The system is initialised in a uniform probability state between all possible states. Average amplitude is shown with the dashed line.

The algorithm can be visualised with two reflections, which generate a rotation in a two-dimensional plane. Initially the system is in a uniform probability state

$$|s\rangle \ = |\psi_0\rangle = H^{\otimes n} |0\rangle^n \,. \tag{62}$$

as seen in Figure 15. Using a two-dimensional plane spanned by perpendicular vectors $|\omega\rangle$ and $|s'\rangle$ allows the initial state to be expressed as $|s\rangle = |\omega\rangle \sin\theta + |s'\rangle \cos\theta$, where the angle $\theta = \arcsin \langle s | \omega \rangle = \arcsin \frac{1}{\sqrt{N}}$. The amplitude of each element in the database can be represented with a bar graph as seen in Figure 15.

There are two interesting states: the correct solution $|\omega\rangle$ and the uniform superposition $|s\rangle$. These vectors span the aforementioned two-dimensional plane in the $\mathbb{C}^N$ vector space. The states are not exactly perpendicular, for $|\omega\rangle$ occurs in the superposition with amplitude $N^{-1/2}$ as well. This can be fixed by introducing an additional state $|s'\rangle$ that is in the span of these two vectors, which is perpendicular to $|\omega\rangle$ and is obtained from $|s\rangle$ by removing $|\omega\rangle$ and rescaling.

The oracle that performs a reflection U_f is applied to the state $|s\rangle$, as shown in Figure 16. Geometrically this corresponds to a reflection of the state $|s\rangle$ about $|s'\rangle$, transforming the amplitude of state $|\omega\rangle$ to negative and reducing the average amplitude.

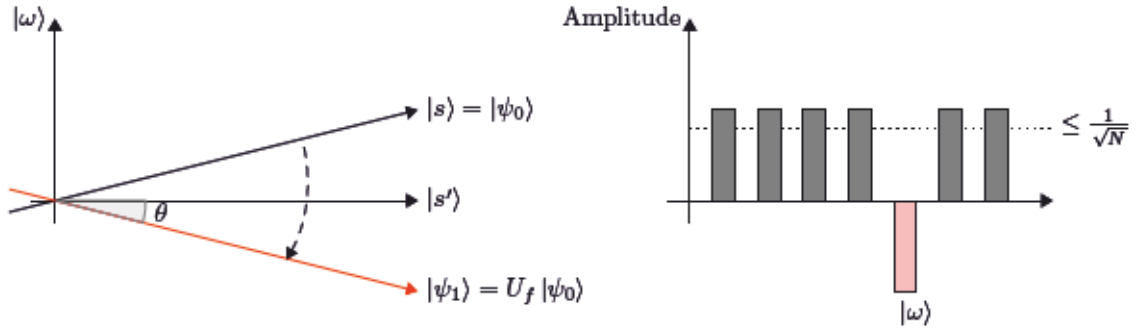


Figure 16: The system resulting after the reflection U_f , over the vector $|s'\rangle$ and changing the phase of event ω. The transformed state $|\psi_1\rangle$ is shown by the red arrow.

Next the oracle performs an additional reflection $U_s = 2|s\rangle\langle s| - I$ about the state $|s\rangle$, resulting in the state $U_s U_f |s\rangle$ which can be seen in Figure 17 to show the amplitude ω is greatly increased and other amplitudes decreased. The action of the reflection U_s can be thought of as a reflection about the average amplitude. Since the first reflection lowered the average amplitude, the second transform boosts the negative amplitude $|\omega\rangle$ to roughly three times its original value while decreasing the other amplitudes. The reflections can be repeated multiple times to increase the probability amplitude even further. After t steps the system will be in state

$$|\psi_t\rangle = (U_s U_f)^t |s\rangle . \tag{63}$$

The relevant question is how many times the rotation has to be applied, and in the literature the answer has been found out to be roughly $\sqrt{N}$ times. The amplitude of $|\omega\rangle$ should grow linearly with the number of applications $\sim tN^{-1/2}$ [35]. In the case of multiple solutions M the amount of rotations is $\sqrt{(N/M)}$ [2].

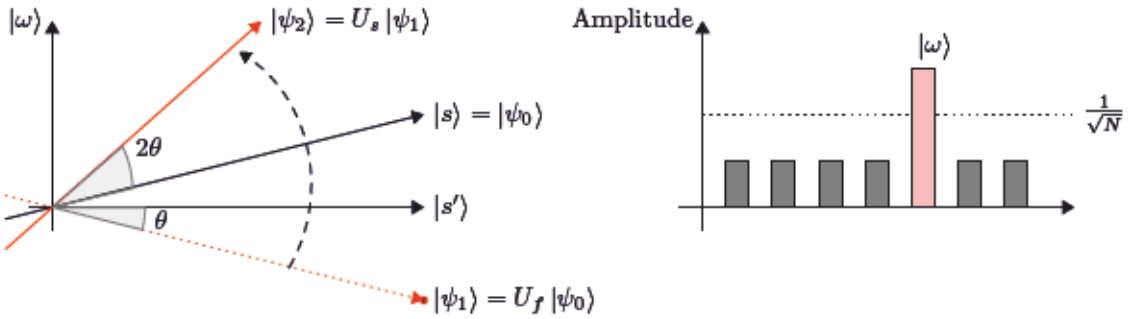


Figure 17: The system after the second reflection U_s. The amplitude of $|\omega\rangle$ is increased as other items have their amplitudes reduced in response.

Due to the fact that multi-control Z-gates are not implemented as native operations but instead as a series of CNOT operations in physical quantum computers, the real experiments deteriorate aggressively with the number of qubits taking part in multi-control Z-gates. The physical implementation of Grover's algorithm on IBM's Quantum systems are limited to at most 3 qubit states due to growing error rate in circuits with more qubits. However, the algorithm can be scaled up to 32 qubits on the QASM-simulator with little error. The implementation of a phase oracle for single and two solution cases can be found in an article by C. Figgatt et al [36], where an alternative style of oracle is also covered.

In the experiment with 3 qubits q_0, q_1 and q_2, a random state $|\omega\rangle = |110\rangle$ is chosen as the secret state to be found. The oracle in the circuit shown in Figure 18 is split into two parts between the first and third barriers.

The qubits are initialised in an uniform superposition state by applying a Hadamard transformation to each qubit

$$\begin{aligned} |\psi_0\rangle &= |+++\rangle \\ &= \frac{1}{\sqrt{2^3}}(|000\rangle + |001\rangle + \cdots + |111\rangle). \end{aligned} \tag{64}$$

The oracle starts with the first rotation U_f in order to switch the phase of state $|\omega\rangle = |110\rangle$ to negative, which is done via a combination of X-rotation gates and a

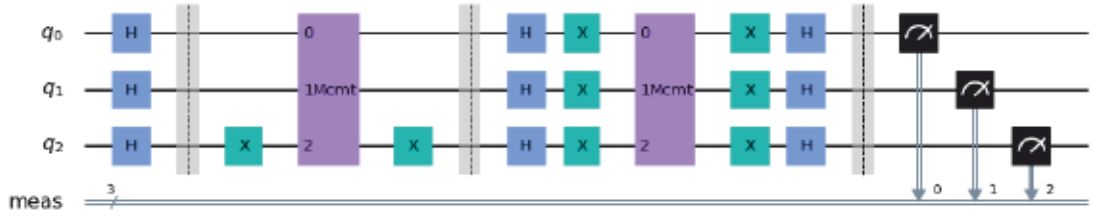


Figure 18: Grover's algorithm circuit. The rotation U_f for state $|\omega\rangle$ is between the first two barriers. The second rotation U_s is between the second and third gate.

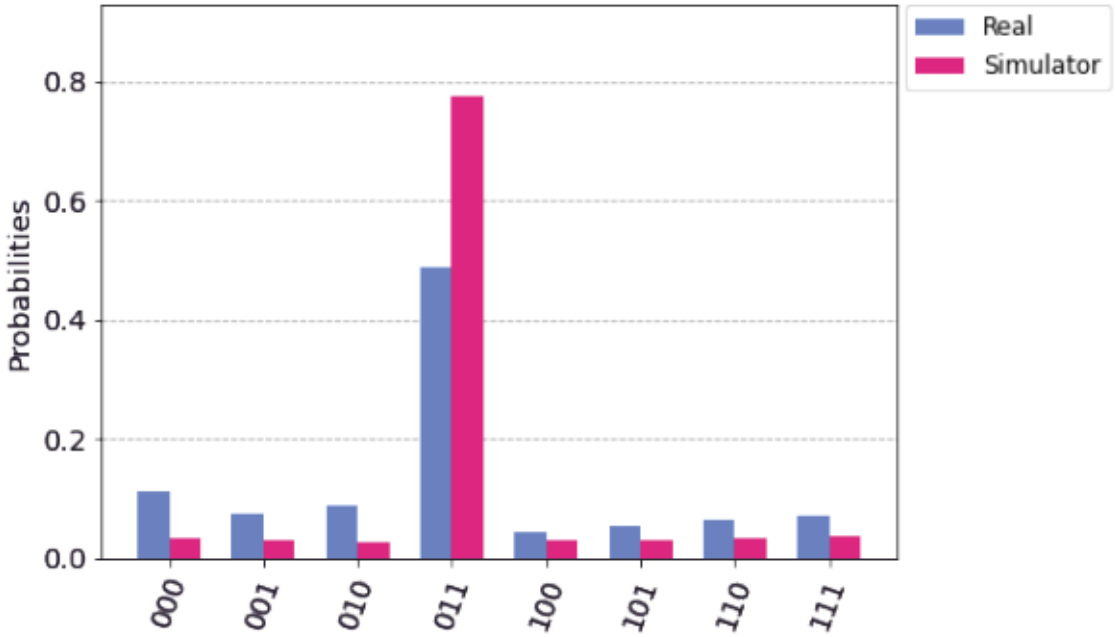


Figure 19: The results of Grover's algorithm comparing the simulation to the results from the real quantum computer, with 8192 shots on each.

multi-controlled Z-rotation on qubit q_2 gate using qubits q_0 and q_1 as controls for the MCZ

$$\begin{aligned} |\psi_1\rangle &= X(q_2)(\mathrm{MC}Z(q_0, q_1|q_2)(X(q_2)\,|\psi_0\rangle)) \\ &= \frac{1}{\sqrt{2^3}}(|000\rangle + |001\rangle + \cdots - |110\rangle + |111\rangle). \end{aligned} \tag{65}$$

The second rotation consists of identical multi-controlled Z-operation sandwiched

between Hadamard and X-rotation gates on each qubit

$$\begin{aligned} |\psi_2\rangle &= H(q_i)X(q_i)(\mathrm{MC}Z(q_0, q_1|q_2)(X(q_i)H(q_i)(|\psi_1\rangle))) \\ &= \frac{1}{16\sqrt{2}}(-4\,|000\rangle - 4\,|001\rangle + \cdots - 20\,|110\rangle - 4\,|111\rangle), \end{aligned} \tag{66}$$

where all states except state $|\omega\rangle = |110\rangle$ have amplitude of $-4/(16\sqrt{2})$ and $i = 0, 1, 2$. This agrees with the measurements seen in Figure 19, where the amplified state has a theoretical probability of $[-20/(16\sqrt{2})]^2 = 78.125\%$ and the rest occur with probability $[-4/(16\sqrt{2})]^2 \approx 3\%$. The real experiment fails to achieve state $|\omega\rangle$ on nearly on half the measurement outcomes. Further testing with more amplitude amplifications failed to increase the probability of the result, instead the circuit was more prone to noise. Noise was most likely due to how the multi-control Z-gate is implemented as a series of qubit gates.

The measurements show a bias towards the last qubit q_2 being in state $|0\rangle$. This phenomenon has occurred in previous experiments, showing nonuniformity between results. The calibration data of the IBM quantum computer Ourense and detailed measurement counts can be seen in the Appendices A and B. The code for this experiment can be found on GitHub [30].

2.6 Repeatable measurements with Lüders instrument

In their paper [37] Bullock and Heinosaari studied the use of repeating of an unsharp measurement on a quantum system and were able to increase the probability of successfully distinguishing states, whilst showing that this improvement does not occur with every increase in the number of measurements. The reason for this type of setup is that projective measurements, which distinguish orthogonal states, are physically impossible to implement on quantum systems, but this can be overcome with repeated non-invasive measurements [38]. However, unsharp measurements can allow sequential measurements without complete loss of the initial state.

This repeated unsharp measurement scheme can be achieved using a measurement model with its associated instrument being a Lüders instrument. Consider a binary observable $\mathsf{A} : \pm \mapsto \mathsf{A}(\pm)$ with corresponding Lüders instruments $\mathcal{I}_X^L(\rho) = \mathsf{A}(X)^{1/2}\rho\mathsf{A}(X)^{1/2}$. Suppose that after measuring A on ρ and obtaining outcome x_1 a second measurement of A is performed on the same system and an outcome x_2 is registered. The probability of obtaining outcomes x_1 and x_2 over the two measurements is given by

$$\begin{aligned} p(x_1, x_2) &= \mathrm{tr}[\mathsf{A}(x_2)\mathcal{I}_{x_1}^L(\rho)] \\ &= \mathrm{tr}[\mathsf{A}(x_2)\mathsf{A}(x_1)^{1/2}\ \rho\ \mathsf{A}(x_1)^{1/2}] \\ &= \mathrm{tr}[\mathsf{A}(x_1)^{1/2}\mathsf{A}(x_2)\mathsf{A}(x_1)^{1/2}\ \rho] \\ &=: \mathrm{tr}[\mathsf{A}^{(2)}(x_1, x_2)\ \rho], \end{aligned}$$

where $\mathsf{A}^{(2)}$ is the effective two-step observable measured. By using induction an n-step observable $\mathsf{A}(n) : \{\pm\}^n \to \mathcal{E}(\mathcal{H})$ can be constructed as

$$\mathsf{A}^{(n)}(x_1, \ldots, x_n) = \sqrt{\mathsf{A}(x_1)} \ldots \sqrt{\mathsf{A}(x_{n-1})}\ \mathsf{A}(x_n)\ \sqrt{\mathsf{A}(x_{n-1})} \ldots \sqrt{\mathsf{A}(x_1)}. \tag{67}$$

Since $\mathsf{A}(-) = I - \mathsf{A}(+)$ it follows that these effects commute, as do their square root operators $\sqrt{\mathsf{A}(+)}$ and $\sqrt{\mathsf{A}(-)}$, and so (67) can be rewritten as

$$\mathsf{A}^{(n)}(x_1, \ldots, x_n) = \mathsf{A}(x_1) \ldots \mathsf{A}(x_n). \tag{68}$$

The success probability $P_{succ}^{(1)}$ of distinguishing states with observable A is given by

$$P_{succ}^{(1)} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{tr}\left[\rho_i \mathsf{A}(i)\right]. \tag{69}$$

In their paper Bullock and Heinosaari provided a general formula for the success probability of distinguishing between two states ρ_+ and ρ_- satisfying

$$\mathsf{A}(\pm)\rho_\pm = \lambda\rho_\pm, \quad \mathsf{A}(\pm)\rho_\mp = (1-\lambda)\rho_\mp, \tag{70}$$

for a binary observable A in the case of an odd number n of repetitions of sequential Lüders measurements the success probability is given by

$$P_{succ}^{(n)} = \frac{1}{2}\left(1 + \sum_{i=\frac{n+1}{2}}^{n} \binom{n}{i} \left(\lambda^i(1-\lambda)^{n-i} - \lambda^{n-i}(1-\lambda)^i\right)\right), \tag{71}$$

and in the case of an even number of repetitions the sum index would start from $i = n/2+1$. This result is generalised as a "rule of three", where every odd measurement after the third measurement provides an increase in the success probability in distinguishing between two orthogonal states with a binary observable whilst the probability does not change for even numbers of measurements.

By using the general solution for the success probability a practical progression of the probability can be predicted with a theoretical baseline which can be calculated for the range of the experiments, in this case from 1 to 21 measurements or a set of circuits ranging in size from 2 to 22 qubits in a simulation. The upper limit of the range of simulations is determined only by the allowed calculation time provided by IBM on their QASM system, and in this case it is 22-qubit circuits for simulations using a simple bit flip noise model.

The simulation was chosen over a real quantum computer in order to overcome the physical limitations present in the free access machines. The experiment scheme used requires one qubit to be connected to another via a CNOT-gate for every Lüders measurement performed, with Melbourne being the only machine providing access to more than 4 measurements but not having the ideal layout. In trial runs with Melbourne the SWAP actions between qubits needed to overcome the physical layout were, even for 4 sequential measurements, too error prone to be usable.

The measurement can be implemented by coupling the system to an additional one via a unitary measurement. Out of the possible computational bases, the operator $e^{i\lambda X\otimes Y}$ was chosen. The operator is calculated by using equation (2) which

leads to the form

$$e^{i\lambda X\otimes Y} = \begin{pmatrix} \cos\lambda & 0 & 0 & \sin\lambda \\ 0 & \cos\lambda & -\sin\lambda & 0 \\ 0 & \sin\lambda & \cos\lambda & 0 \\ -\sin\lambda & 0 & 0 & \cos\lambda \end{pmatrix}. \tag{72}$$

Qiskit provides a function for creating arbitrary unitary operations. The amount of universal operations on the circuit needed in order to realise the operator varies depending on the variable λ: for example with $\lambda = \pi/4$ the two-qubit operator transforms a state $|{+}{+}\rangle$ as follows

$$\begin{aligned} e^{i\lambda X\otimes Y}\,|{+}{+}\rangle &= (\cos\lambda I + i\sin\lambda X\otimes Y)\,|{+}{+}\rangle \\ &= \cos\lambda\,|{+}{+}\rangle + i\sin\lambda X\,|{+}\rangle\otimes Y\,|{+}\rangle \\ &= \cos\lambda\,|{+}{+}\rangle + i\sin\lambda\,|{+}\rangle\otimes(-i\,|{-}\rangle) \\ &= \cos\lambda\,|{+}{+}\rangle + \sin\lambda\,|{+}{-}\rangle \\ &= |{+}\rangle\otimes(\cos\lambda\,|{+}\rangle + \sin\lambda\,|{-}\rangle) \\ &= |{+}\rangle\otimes\left(\frac{1}{\sqrt{2}}(\cos\lambda+\sin\lambda)\,|0\rangle + \frac{1}{\sqrt{2}}(\cos\lambda-\sin\lambda)\,|1\rangle\right) \\ &= |{+}\rangle\otimes\left(\sqrt{\frac{1}{2}(1+\sin 2\lambda)}\,|0\rangle + \sqrt{\frac{1}{2}(1+\sin 2\lambda)}\,|1\rangle\right) \\ &= |{+}\rangle\otimes(\alpha_0\,|0\rangle + \alpha_1\,|1\rangle)\,, \end{aligned} \tag{73}$$

whereas letting $\lambda = \pi/4$ results in

$$e^{i\frac{\pi}{4}X\otimes Y}\,|{+}{+}\rangle = |{+}\rangle\otimes\left(\sqrt{\frac{1}{2}\left(1+\sin\frac{\pi}{2}\right)}\,|0\rangle + \sqrt{\frac{1}{2}\left(1-\sin\frac{\pi}{2}\right)}\,|1\rangle\right), \tag{74}$$

in other words $e^{i\frac{\pi}{4}X\otimes Y}\,|{+}{+}\rangle = |{+}\rangle\otimes|0\rangle$.

In the circuit there are two qubit registers; the first register has one qubit that is prepared with a Hadamard gate, and a second register consisting of n qubits prepared with Hadamard gates. The operator goes through pairing the first register

one at a time with every qubit in the second register performing n measurements, and the resulting success probability is calculated from the final measurement performed on the second register in the circuit. The two other operations $e^{i\lambda X\otimes X}$ and $e^{i\lambda X\otimes Z}$ were discarded for this experiment due to the need for single gate rotations after the operation and before measuring the second register.

As an example, a four-qubit circuit with one qubit q_0 in the first register and the other three q_1, q_2, q_3 in second register will perform three Lüders measurements. The system in state $|\psi_0\rangle = |q_0, q_1, q_2, q_3\rangle = |0000\rangle$ is initialised with Hadamard gates on each qubit

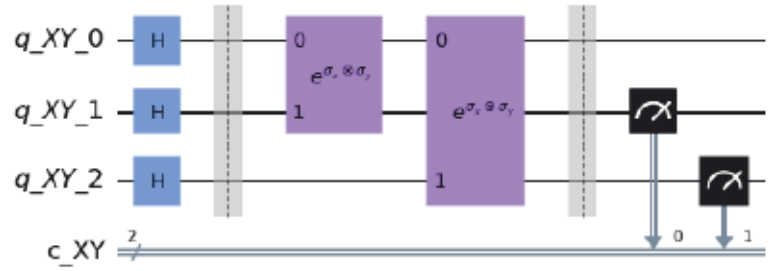
$$|\psi_1\rangle = H^{\otimes 4}\,|0000\rangle = |++++\rangle\,. \tag{75}$$

Next the Lüders instrument $e^{i\lambda X\otimes Y}$ is applied on the qubits in the second register using the first register as control

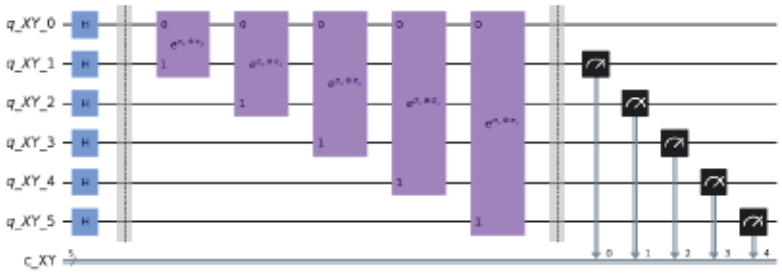
$$\begin{aligned} |\psi_2\rangle &= e^{i\lambda X_0\otimes Y_3}e^{i\lambda X_0\otimes Y_2}e^{i\lambda X_0\otimes Y_1}\,|++++\rangle \\ &= |+\rangle\otimes(\alpha_0\,|0\rangle + \alpha_1\,|1\rangle)\otimes(\alpha_0\,|0\rangle + \alpha_1\,|1\rangle)\otimes(\alpha_0\,|0\rangle + \alpha_1\,|1\rangle) \\ &= |+\rangle\otimes\Big[\alpha_0^3\,|000\rangle + \alpha_0^2\alpha_1(|001\rangle + |010\rangle + |100\rangle) \\ &\qquad +\alpha_0\alpha_1^2(|011\rangle + |101\rangle + |110\rangle) + \alpha_1^3\,|111\rangle\Big], \end{aligned} \tag{76}$$

where the notation from the last line in (73) is used for probability amplitudes α_0 and α_1. By adding more qubits to the second register there will be more unique coefficients representing the states, but they will behave in the exact same manner with one coefficient α_0 for a $|0\rangle$ in the state and a coefficient α_1 for each $|1\rangle$. The success probability of measuring 0 with the post-processed binary observable is the sum of the squares of probability amplitudes of states with more zeros, with the cases containing an equal number of ones and zeros being divided equally between both binary cases.

As shown previously there exist special cases such as $\lambda = \pi/4$ in which the operator used will always produce the result zero, which allows us to limit $\lambda \in$



(a) Two sequential measurements.



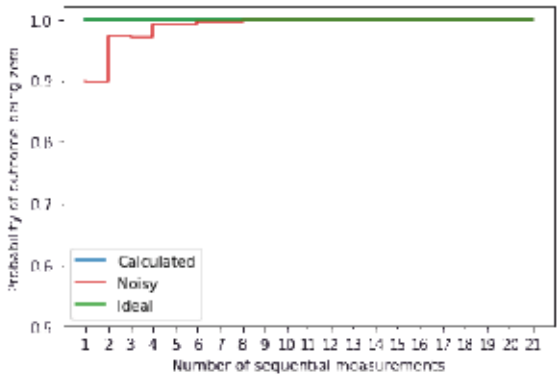
(b) Five sequential measurements.

Figure 20: Two example circuits used in the repeated measurements experiment.

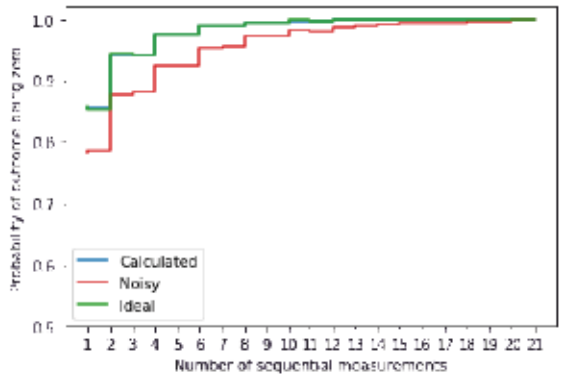
$[0, \pi/4]$ for the range of values to be tested.

The circuits produced have a simple scalable structure as can be seen in Figure 20, and our interest lies in how the success probability acts for different values of λ in circuits with a different number of Lüders measurements performed. Two sets of simulations will be performed for each variable: an ideal simulation without noise modelisation and a simulation with a simple bit flip noise model. The graphs produced by these two simulations are compared with a graph calculated with equation (71).

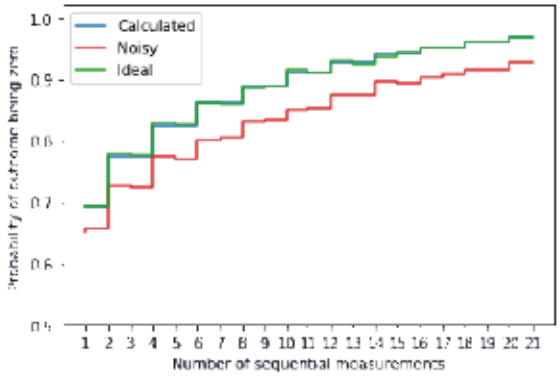
The bit flip error model used here flips the state of a qubit from $|0\rangle$ to $|1\rangle$ and vice versa. This flip is applied with a 5% error to all single qubit operations, and on two-qubit operations it applies the single qubit error on both qubits. In addition, on the final measurement performed on qubits there is a 10 % possibility of a bit flip on each qubit measured.



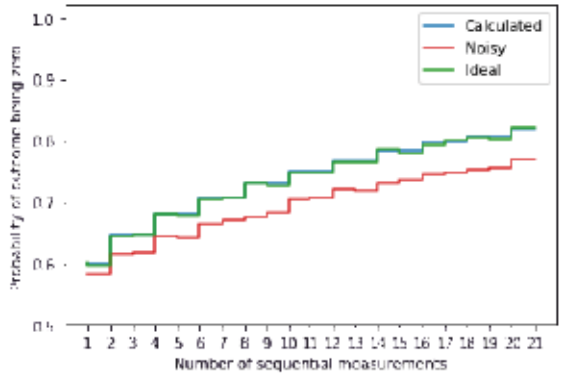
(a) $\lambda = \pi/4$.



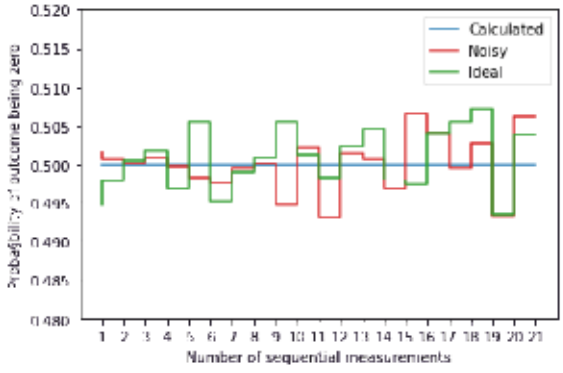
(b) $\lambda = \pi/8$.



(c) $\lambda = \pi/16$.



(d) $\lambda = \pi/32$.



(e) $\lambda = 0$.

Figure 21: The results from Lüders instrument simulations compared with a graph produced with the prediction (Equation (71)) for different values of λ. For each sequential measurement, the corresponding circuit was ran twice resulting in total of 16384 shots from which the success probability was calculated in post processing.

Experimental data from 1 to 21 repeated measurements for different λ can be seen in Figure 21, each circuit representing a number of sequential measurements was ran twice resulting in 16384 shots per circuit, from which the success probability for each number of sequential measurements was counted in post processing. The success probability does increase with repeated measurements in both simulated cases as was predicted by Bullock and Heinosaari in their paper.

The simulations show similar behaviour with the theoretical prediction: every odd-numbered sequential measurement increases the success probability. As the value of λ decreases from $\pi/4$ to 0 the ideal simulation and the theoretical model start to deviate from each other, this can best be seen with $\lambda = \pi/32$. At $\lambda = 0$ both simulations maintain the success probability of 50% within 7% accuracy. Simulation with a simple bit flip noise model produces exactly the same behaviour as the model with lower values for success probability when $\lambda > 0$, with the highest deviation being 10% but mostly staying within 5%. Note that for the fluctuations from 50% success probability when $\lambda = 0$ in the ideal case could be evened out with more shots.

The resulting counts from the simulations are not included as an appendix. The number of possible events in a circuit with 21 qubits measured is 2^{21}, and the collection of all possible events for all the circuits in the range of 2 to 22 qubits is $\sum_{i=1}^{21} 2^i = 2^{22} - 2$. But the simulations can be reproduced with the information in this chapter or by using the code found on GitHub [30].

Even if no available quantum computer had the necessary layout for this experiment, these simulated results give an encouraging view of how sequential unsharp measurements can be used to increase the probability of making a successful measurement when the measurement process is prone to errors, even if it does impose the need for more qubits as a consequence. The function for calculating the prediction of the success probability for sequential measurements can be found in Appendix C.

3 Conclusions

In this thesis a number of quantum algorithms were investigated using IBM's quantum systems. There are several quantum and classical–quantum hybrid algorithms that were not included, such as Shor's factoring algorithm, the quantum phase estimation algorithm and triangle finding algorithm. The algorithms here were chosen for various reasons, with significant factors being the ease of implementation and historical relevance. The investigation into repeatable measurements was guided by Bullock and was a learning experience and an opportunity for the author to implement something new and that provided a proof of concept for Bullock on his work with Heinosaari [37].

In the first experiment a technique, impossible for non-quantum systems, gave insight into how quantum computation gives access to one new method, although not without its drawbacks. The state teleported via quantum teleportation is altered in accordance with the no cloning principle. Quantum parallelisation, used in other experiments, allows multiple inputs to be run in parallel, but the rate of getting the correct outcome remained statistically low in experiments. In order to overcome this the circuits are run numerous times (here each circuit was ran 8192 times) in order to get statistics that show how successful an algorithm really is on a IBM quantum computer. And finally in the repeatable measurements with Lüders instrument experiment the available quantum computers were inadequate for showing progress in how sequential Lüders measurements with a binary observable can improve the success probability over a single iteration.

In Deutsch–Jozsa, Bernstein–Vazirani and Grover's search algorithms, where a singular state is the desired outcome, only the constant oracle in Deutsch–Jozsa experiment occurred with over 50% probability, while the probability for the correct answer in Grover's experiment was 49% and Bernstein-Vazirani was 44%. In quantum teleportation and Simon's algorithm experiments with multiple projected

outcomes a general bias towards a qubit being in state $|0\rangle$ and a decreasing probability towards the measured qubits all being in state $|1\rangle$ was shown. This could be the result of uneven probabilities within the Hadamard transformation or a quirk in the final measurement.

The repeated measurements with Lüders operator show that every odd number of unsharp measurement increases the probability of measuring successfully a binary observable. This is a promising result if it can be applied in circuits in a practical manner, in order to decrease the error that occurs in measurement.

As seen in the experiments, quantum computing provides an advantage over classical methods in certain situations. In situations where quantum phenomena cannot be utilised, quantum computers provide no advantage. This gives room for two different approaches in the future of computation: first in some cases it might be possible to create hybrid computing machines that utilise classical and quantum computation methods; secondly it might be possible to find other ways to solve some classical problems in a way that quantum phenomena can be used to accelerate problem solving.

An interesting further study would be to research error mitigation techniques that could help to reduce the error produced by every operation present on the circuit. This could work in conjunction with engineering advances in reducing the noise affecting the systems.

References

[1] B. Schumacher, *Quantum coding*, Physical Review A **51**, 2738 (1995).

[2] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2011).

[3] T. Heinosaari and M. Ziman, *The Mathematical Language of Quantum Theory: From Uncertainty to Entanglement* (Cambridge University Press, 2011).

[4] P. Busch, P. Lahti, J.-P. Pellonpää and K. Ylinen, *Quantum Measurement* (Springer, 2016).

[5] E. Bernstein and U. Vazirani, *Quantum Complexity Theory*, SIAM journal on computing **26**, 1411 (1997).

[6] D. P. DiVincenzo, *The Physical Implementation of Quantum Computation*, Fortschritte der Physik **48**, 771 (2000).

[7] T. P. Orlando *et al.*, *Superconducting persistent-current qubit*, Physical review. B, Condensed matter **60**, 15398 (1999).

[8] E. Knill, R. Laflamme and G. J. Milburn, *A scheme for efficient quantum computation with linear optics*, Nature (London) **409**, 46 (2001).

[9] A. J. Ferguson, P. A. Cain, D. A. Williams and G. A. D. Briggs, *Ammonia-based quantum computer*, Physical Review A **65**, 034303 (2002).

[10] B. B. Blinov, D. Leibfried, C. Monroe and D. J. Wineland, *Quantum Computing with Trapped Ion Hyperfine Qubits*, Quantum Information Processing **3**, 45 (2004).

[11] J. Clarke and F. K. Wilhelm, *Superconducting quantum bits*, Nature **453**, 1031 (2008).

[12] M. Kjaergaard *et al.*, *Superconducting Qubits: Current State of Play*, arXiv:1905.13641 (2019).

[13] T. O'Brien, D-Wave One claims mantle of first commercial quantum computer, 2011.

[14] A. C. Santos, *The IBM Quantum Computer and the IBM Quantum Experience*, arXiv:1610.06980 (2016).

[15] A. W. Harrow and A. Montanaro, *Quantum computational supremacy*, Nature (London) **549**, 203 (2017).

[16] S. Boixo *et al.*, *Characterizing quantum supremacy in near-term devices*, Nature physics **14**, 595 (2018).

[17] E. Gibney, *Hello quantum world! Google publishes landmark quantum supremacy claim*, Nature (London) **574**, 461 (2019).

[18] R. Waters, *IBM researchers dismiss Google 'quantum supremacy' claim*, "IBM researchers dismiss Google 'quantum supremacy' claim". `https://www.ft.com/content/852188c6-f43e-11e9-b018-3ef8794b17c6` accessed September 20th 2020 (2019).

[19] W. Roush, *The Google-IBM "quantum supremacy" feud*, MIT Technology Review. `https://www.technologyreview.com/2020/02/26/905777/google-ibm-quantum-supremacy-computing-feud/` accessed February 17th 2021 (2020).

[20] J. Bardeen, L. N. Cooper and J. R. Schrieffer, *Theory of Superconductivity*, Phys. Rev. **108**, 1175 (1957).

[21] J. Clarke and F. K. Wilhelm, *Superconducting quantum bits*, Nature **453**, 1031 (2008).

[22] J. M. Gambetta, J. M. Chow and M. Steffen, *Building logical qubits in a superconducting quantum computing system*, npj Quantum Information **3**, 2 (2017).

[23] D. Castelvecchi, *IBM's quantum cloud computer goes commercial*, Nature (London) **543**, 159 (2017).

[24] G. García-Pérez, M. A. C. Rossi and S. Maniscalco, *IBM Q Experience as a versatile experimental testbed for simulating open quantum systems*, npj quantum information **6**, 1 (2020).

[25] Qiskit Development Team, *Qiskit 0.21.0 documentation*, `https://qiskit.org/documentation/index.html`. Accessed September 9th 2020 (2020).

[26] C. H. Bennett *et al.*, *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels*, Physical review letters **70**, 1895 (1993).

[27] W. K. Wootters and W. H. Zurek, *A single quantum cannot be cloned*, Nature (London) **299**, 802 (1982).

[28] M. Baur *et al.*, *Benchmarking a Quantum Teleportation Protocol in Superconducting Circuits Using Tomography and an Entanglement Witness*, Phys. Rev. Lett. **108**, 40502 (2012).

[29] R. Ursin *et al.*, *Communications Quantum teleportation across the Danube*, Nature **430**, 849 (2004).

[30] J. Nykänen, *Thesis experiments code*, `https://github.com/juilny/Thesis-experiments` (2021).

[31] D. Deutsch and R. Jozsa, *Rapid solution of problems by quantum computation*, Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences **439**, 553 (1992).

[32] S. Gulde *et al.*, *Implementation of the Deutsch–Jozsa algorithm on an ion-trap quantum computer*, Nature (London) **421**, 48 (2003).

[33] D. R. Simon, *On the Power of Quantum Computation*, SIAM journal on computing **26**, 1474 (1997).

[34] G. Cai and D. Qiu, *Optimal separation in exact query complexities for Simon's problem*, Journal of computer and system sciences **97**, 83 (2018).

[35] L. K. Grover, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, STOC '96* (Association for Computing MachineryNew York, NY, USA, 1996), pp. 212–219.

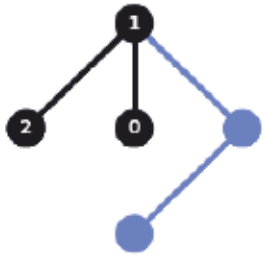
[36] C. Figgatt *et al.*, *Complete 3-Qubit Grover search on a programmable quantum computer*, Nature communications **8**, 1918 (2017).

[37] T. Bullock and T. Heinosaari, *Quantum state discrimination via repeated measurements and the rule of three*, Quantum Studies: Mathematics and Foundations **8**, 137 (2021).

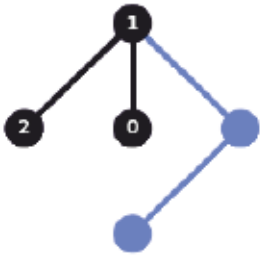
[38] Y. Guryanova, N. Friis and M. Huber, *Ideal Projective Measurements Have Infinite Resource Costs*, Quantum **4**, 222 (2020).

Appendices

A Optimisation mappings

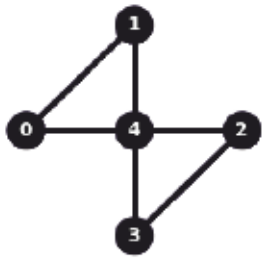


(a) Quantum teleportation

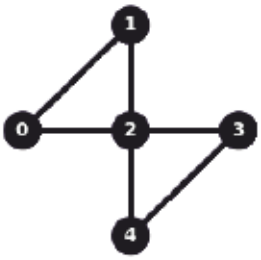


(b) Grover's search algorithm

Figure 22: Optimization layouts for experiments on Ourense.

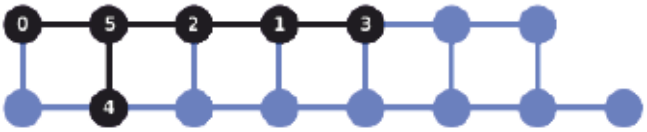


(a) Deutsch–Jozsa algorithm with balanced oracle.

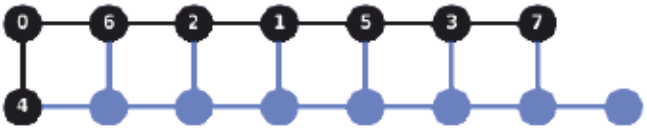


(b) Deutsch–Jozsa algorithm with constant oracle.

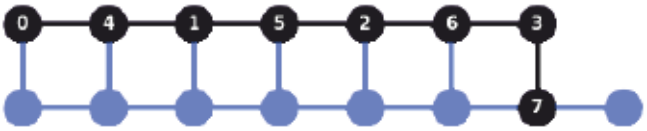
Figure 23: Optimization layouts for experiments on Yorktown.



(a) Bernstein–Vazirani algorithm.



(b) Simon's algorithm for b=1010.



(c) Simon's algorithm for b=0000.

Figure 24: Optimization layouts for experiments on Melbourne.

B Tables for measurement counts and calibration data

Table I: Measurement counts of events in quantum teleportation experiment.

'000': 2074	'001': 1806	'010': 1838	'011': 1745
'100': 195	'101': 229	'110': 142	'111': 163

Table II: Measurement counts of events in Grover's search algorithm experiment.

'000': 924	'001': 627	'010': 738	'011': 4000
'100': 355	'101': 437	'110': 530	'111': 581

Table III: Measurement counts of events in Deutsch–Jozsa experiment with balanced oracle.

'0000' : 103	'0001' : 68	'0010' : 74	'0011' :193
'0100': 71	'0101': 159	'0110': 169	'0111':702
'1000': 69	'1001': 172	'1010': 171	'1011' :691
'1100': 209	'1101': 751	'1110': 879	'1111' : 3711

Table IV: Measurement counts of events in Deutsch–Jozsa experiment with constant oracle.

'0000' : 6827	'0001' : 84	'0010' : 604	'0011' :16
'0100': 124	'0101': 4	'0110': 30	'0111':1
'1000': 396	'1001': 8	'1010': 42	'1011' :1
'1100': 48	'1101': 0	'1110': 6	'1111' : 1

Table V: Measurement counts of events in Bernstein–Vazirani experiment.

'00000': 116	'00001': 175	'00010': 26	'00011': 51
'00100': 107	'00101': 646	'00110': 28	'00111': 158
'01000': 6	'01001': 13	'01010': 2	'01011': 6
'01100': 8	'01101': 54	'01110': 5	'01111': 14
'10000': 133	'10001': 775	'10010': 38	'10011': 216
'10100': 535	'10101': 3560	'10110': 132	'10111': 866
'11000': 12	'11001': 59	'11010': 6	'11011': 31
'11100': 40	'11101': 265	'11110': 7	'11111': 102

Table VI: Measurement counts of events in Simon's algorithm experiment for two-to-one circuit.

'00000000':203	'00000001':18	'00000010':161	'00000011':17	'00000100':22	'00000101':166
'00000110':31	'00000111':127	'00001000':110	'00001001':13	'00001010':103	'00001011':7
'00001100':25	'00001101':81	'00001110':21	'00001111':74	'00010000':15	'00010010':9
'00010011':2	'00010100':1	'00010101':11	'00010111':9	'00011000':4	'00011001':1
'00011010':4	'00011011':2	'00011100':2	'00011101':3	'00011111':5	'00100000':141
'00100001':20	'00100010':118	'00100011':26	'00100100':22	'00100101':97	'00100110':17
'00100111':107	'00101000':99	'00101001':14	'00101010':75	'00101011':6	'00101100':17
'00101101':74	'00101110':13	'00101111':61	'00110000':5	'00110001':2	'00110010':3
'00110011':1	'00110101':4	'00110111':4	'00111000':7	'00111010':2	'00111101':5
'00111111':3	'01000000':180	'01000001':20	'01000010':140	'01000011':17	'01000100':26
'01000101':131	'01000110':22	'01000111':125	'01001000':119	'01001001':14	'01001010':101
'01001011':11	'01001100':24	'01001101':97	'01001110':12	'01001111':79	'01010000':8
'01010001':2	'01010010':9	'01010100':2	'01010101':7	'01010110':1	'01010111':9
'01011000':8	'01011010':3	'01011101':2	'01011111':2	'01100000':143	'01100001':11
'01100010':111	'01100011':13	'01100100':10	'01100101':108	'01100110':16	'01100111':90
'01101000':76	'01101001':15	'01101010':58	'01101011':7	'01101100':15	'01101101':68
'01101110':7	'01101111':43	'01110000':6	'01110010':3	'01110011':1	'01110100':1
'01110101':4	'01110111':1	'01111000':3	'01111001':1	'01111010':3	'01111101':3
'01111111':1	'10000000':199	'10000001':26	'10000010':153	'10000011':13	'10000100':29
'10000101':164	'10000110':25	'10000111':116	'10001000':125	'10001001':14	'10001010':123
'10001011':14	'10001100':18	'10001101':98	'10001110':20	'10001111':82	'10010000':8
'10010010':5	'10010011':1	'10010100':2	'10010101':11	'10010110':2	'10010111':11
'10011000':5	'10011001':2	'10011010':3	'10011011':1	'10011100':2	'10011101':4
'10011110':1	'10011111':2	'10100000':135	'10100001':11	'10100010':129	'10100011':15
'10100100':18	'10100101':91	'10100110':15	'10100111':89	'10101000':96	'10101001':8
'10101010':71	'10101011':7	'10101100':12	'10101101':62	'10101110':17	'10101111':51
'10110000':6	'10110010':2	'10110101':9	'10110110':2	'10110111':4	'10111000':9
'10111001':1	'10111010':4	'10111100':1	'10111101':2	'10111110':1	'10111111':4
'11000000':145	'11000001':21	'11000010':129	'11000011':12	'11000100':27	'11000101':123
'11000110':20	'11000111':111	'11001000':123	'11001001':6	'11001010':94	'11001011':9
'11001100':9	'11001101':87	'11001110':13	'11001111':81	'11010000':7	'11010001':2
'11010010':6	'11010100':3	'11010101':12	'11010111':7	'11011000':4	'11011010':2
'11011101':10	'11011111':4	'11100000':119	'11100001':9	'11100010':106	'11100011':10
'11100100':16	'11100101':99	'11100110':15	'11100111':79	'11101000':75	'11101001':12
'11101010':49	'11101011':13	'11101100':8	'11101101':79	'11101110':9	'11101111':57
'11110000':3	'11110010':6	'11110011':1	'11110101':4	'11110111':7	'11111000':2
'11111101':5	'11111111':6				

Table VII: Measurement counts of events in Simon's algorithm experiment for one-to-one circuit.

'00000000':54	'00000001':55	'00000010':76	'00000011':53	'00000100':40	'00000101':38	'00000110':67
'00000111':53	'00001000':41	'00001001':29	'00001010':42	'00001011':35	'00001100':31	'00001101':30
'00001110':52	'00001111':33	'00010000':43	'00010001':40	'00010010':59	'00010011':50	'00010100':42
'00010101':29	'00010110':49	'00010111':45	'00011000':30	'00011001':32	'00011010':38	'00011011':45
'00011100':28	'00011101':29	'00011110':31	'00011111':26	'00100000':57	'00100001':47	'00100010':65
'00100011':54	'00100100':50	'00100101':31	'00100110':41	'00100111':46	'00101000':40	'00101001':34
'00101010':36	'00101011':35	'00101100':44	'00101101':21	'00101110':36	'00101111':33	'00110000':49
'00110001':41	'00110010':42	'00110011':61	'00110100':36	'00110101':34	'00110110':35	'00110111':42
'00111000':34	'00111001':37	'00111010':40	'00111011':37	'00111100':25	'00111101':24	'00111110':31
'00111111':22	'01000000':41	'01000001':38	'01000010':67	'01000011':33	'01000100':27	'01000101':19
'01000110':35	'01000111':34	'01001000':27	'01001001':22	'01001010':38	'01001011':28	'01001100':27
'01001101':17	'01001110':21	'01001111':12	'01010000':35	'01010001':37	'01010010':38	'01010011':30
'01010100':24	'01010101':18	'01010110':22	'01010111':24	'01011000':27	'01011001':18	'01011010':40
'01011011':23	'01011100':19	'01011101':15	'01011110':21	'01011111':27	'01100000':31	'01100001':29
'01100010':35	'01100011':24	'01100100':20	'01100101':23	'01100110':27	'01100111':22	'01101000':35
'01101001':13	'01101010':35	'01101011':23	'01101100':25	'01101101':16	'01101110':21	'01101111':13
'01110000':29	'01110001':34	'01110010':28	'01110011':21	'01110100':25	'01110101':23	'01110110':28
'01110111':16	'01111000':14	'01111001':19	'01111010':26	'01111011':16	'01111100':13	'01111101':12
'01111110':21	'01111111':22	'10000000':52	'10000001':34	'10000010':70	'10000011':48	'10000100':46
'10000101':27	'10000110':57	'10000111':34	'10001000':38	'10001001':21	'10001010':51	'10001011':42
'10001100':36	'10001101':26	'10001110':32	'10001111':19	'10010000':40	'10010001':40	'10010010':58
'10010011':42	'10010100':34	'10010101':22	'10010110':44	'10010111':40	'10011000':38	'10011001':28
'10011010':39	'10011011':43	'10011100':27	'10011101':24	'10011110':34	'10011111':36	'10100000':45
'10100001':42	'10100010':55	'10100011':36	'10100100':47	'10100101':29	'10100110':42	'10100111':35
'10101000':41	'10101001':33	'10101010':48	'10101011':31	'10101100':37	'10101101':29	'10101110':30
'10101111':25	'10110000':40	'10110001':43	'10110010':45	'10110011':34	'10110100':31	'10110101':28
'10110110':39	'10110111':38	'10111000':29	'10111001':32	'10111010':42	'10111011':39	'10111100':26
'10111101':18	'10111110':26	'10111111':17	'11000000':37	'11000001':22	'11000010':48	'11000011':50
'11000100':25	'11000101':23	'11000110':39	'11000111':29	'11001000':29	'11001001':15	'11001010':30
'11001011':29	'11001100':21	'11001101':9	'11001110':33	'11001111':22	'11010000':29	'11010001':19
'11010010':49	'11010011':27	'11010100':14	'11010101':15	'11010110':35	'11010111':20	'11011000':21
'11011001':32	'11011010':29	'11011011':27	'11011100':16	'11011101':12	'11011110':25	'11011111':16
'11100000':30	'11100001':34	'11100010':39	'11100011':28	'11100100':25	'11100101':17	'11100110':31
'11100111':23	'11101000':24	'11101001':20	'11101010':30	'11101011':23	'11101100':21	'11101101':11
'11101110':21	'11101111':17	'11110000':26	'11110001':36	'11110010':43	'11110011':20	'11110100':15
'11110101':23	'11110110':23	'11110111':26	'11111000':22	'11111001':22	'11111010':23	'11111011':19
'11111101':8	'11111110':10	'11111111':19				

Table VIII: Yorktown Calibration data Sun Nov 15 2020 08:05:24 GMT+0200 (EET)

Qubit	T1 (μs)	T2 (μs)	Frequency (GHz)	Readout error	Single-qubit U2 error rate	CNOT error rate
Q0	64.80097906876216	26.192727956867945	5.282919951313925	6.0799999999999965e-2	1.3400103595488319e-3	cx0_1: 2.222e-2, cx0_2: 2.043e-2
Q1	53.56604623515277	25.023175407667644	5.2476217295538765	1.0339999999999994e-1	1.1462717795467229e-3	cx1_0: 2.222e-2, cx1_2: 2.351e-2)
				2		cx2_0: 2.043e-2,
Q2	55.552761434052336	82.92899129593724	5.033636211697712	2.7700000000000058e-	8.576727866211151e-4	cx2_1: 2.351e-2, cx2_3: 1.752e-2, cx2_4: 1.727e-2
Q3	61.45885030821752	37.54054139202137	5.292252591685098	1.7900000000000027e-2	4.138009204792204e-4	cx3_2: 1.752e-2, cx3_4: 1.719e-2
Q4	67.43680228697596	33.552092824582324	5.0784543319605	2.6899999999999924e-2	8.212272540665525e-4	cx4_2: 1.727e-2, cx4_3: 1.719e-2

Table IX: Ourense Calibration data Sun Nov 15 2020 10:40:36 GMT+0200 (EET)

Qubit	T1 (μs)	T2 (μs)	Frequency (GHz)	Readout error	Single-qubit U2 error rate	CNOT error rate
Q0	79.17617976932256	59.08480036801689	4.8203215973142 28	1.8000000000000016e-2	4.874267047358945e-4	cx0_1: 8.282e-3
Q1	92.43976357075755	28.605031221592753	4.890165387144746	2.62e-2	3.874368923640628e-4	cx1_0: 8.282e-3, cx1_2: 7.438e-3, cx1_3: 1.194e-2
Q2	140.60969797741268	124.22126482616791	4.716564708316591	1.429999999999998e-2	1.9875310767462887e-4	cx2_1: 7.438e-3
Q3	147.82211558994035	31.170956028718695	4.789068251734187	3.7900000000000045e-2	4.04727016957678e-4	cx3_1: 1.194e-2, cx3_4: 8.179e-3
Q4	74.48148473401162	24.315174637966724	5.02379644537416	4.190000000000005e-2	3.4445961154887803e-4	cx4_3: 8.179e-3

Table X: Melbourne Calibration data Sun Nov 15 2020 11:50:13 GMT+0200 (EET)

Qubit	T1 (μs)	T2 (μs)	Frequency (GHz)	Readout error	Single-qubit U2 error rate	CNOT error rate
Q0	68.27523377702911	108.19381517522997	5.11486542683711	2.9000000000000026e-2	3.8758509065723413e-4	cx0_1: 1.382e-2, cx0_14: 2.137e-2
Q1	60.79888617714073	64.13752157438199	5.235705486790535	3.610000000000002e-2	7.656847396780038e-4	cx1_0: 1.382e-2, cx1_2: 1.077e-2, cx1_13: 4.813e-2
Q2	58.579146250363344	71.21684101956009	5.038467762238707	4.7700000000000076e-2	7.990158223759236e-4	cx2_1: 1.077e-2, cx2_3: 3.962e-2, cx2_12: 4.685e-2
Q3	51.64125012805223	15.829341889856636	4.894580594210236	1.4570000000000005e-1	5.045318020054128e-4	cx3_2: 3.962e-2, cx3_4: 2.084e-2, cx3_11: 3.564e-2
Q4	52.95195188158424	61.54079009009009	5.02158070958438	7.850000000000001e-2	9.067403611800831e-4	cx4_3: 2.084e-2, cx4_5: 2.739e-2, cx4_10: 3.042e-2
Q5	19.862924994761737	35.43501640563119	5.073083586083812	5.479999999999996e-2	2.2521332860055555e-3	cx5_4: 2.739e-2, cx5_6: 5.419e-2, cx5_9: 3.393e-2
Q6	78.6808535799579	32.70838524808492	4.929435469113168	3.920000000000001e-2	1.462513163271206e-3	cx6_5: 5.419e-2, cx6_8: 3.034e-2
Q7	39.700775591844426	14.241881701060104	4.982610871865756	4.8899999999999944e-2	1.8128957457700869e-3	cx7_8: 4.245e-2
Q8	93.92963644927887	120.3043456480796	4.751547322307862	5.4300000000000015e-2	6.019398423756427e-4	cx8_6: 3.034e-2, cx8_7: 4.245e-2, cx8_9: 4.737e-2
Q9	41.503837013378416	58.86743131102095	4.973450400477016	6.730000000000003e-2	1.6531600793574896e-3	cx9_5: 3.393e-2, cx9_8: 4.737e-2, cx9_10: 3.306e-2
Q10	51.614592657105796	61.38721568118689	4.944784488344414	3.3600000000000074e-2	1.6633667682413767e-3	cx10_4: 3.042e-2, cx10_9: 3.306e-2, cx10_11: 2.369e-2
Q11	58.1653534679787	77.84591153709329	4.997446810443168	1.733e-1	4.6183630966750234e-4	cx11_3: 3.564e-2, cx11_10: 2.369e-2, cx11_12: 1.599e-2
Q12	65.79778878695761	54.957645070494316	4.763413076843809	8.299999999999996e-2	8.813064425955343e-4	cx12_2: 4.685e-2, cx12_11: 1.599e-2, cx12_13: 2.620e-2
Q13	29.41533119773304	27.722289644490505	4.974274979120487	8.830000000000005e-2	3.8256971238350623e-3	cx13_1: 4.813e-2, cx13_12: 2.620e-2, cx13_14: 3.942e-2
Q14	44.3831995654476	49.073937313402304	5.006779112604153	5.699999999999994e-2	8.221729519392484e-4	cx14_0: 2.137e-2, cx14_13: 3.942e-2

C Function for calculating the success probability

```
# Returns the success probability of n sequential measurements for a given
    Lambda
def successProbabilityPrediction(n,L):
    i = int((n+1)/2)
    Lambda = (1 + np.sin(2*L))/2
    summa = 0
    while(i <= n):
        bin1 = comb(n,i,exact=True)
        power1 = Lambda**i*(1-Lambda)**(n-i)
        power2 = Lambda**(n-i)*(1-Lambda)**i

        tmp = (bin1*(power1-power2))
        summa = summa + tmp

        i += 1
    p = 1/2*(1+summa)
    return p
```