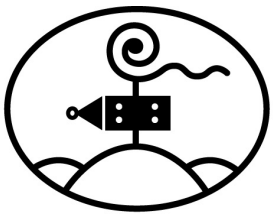


SCHOTTENGYMNASIUM

VORWISSENSCHAFTLICHE ARBEIT

Working Principles and Current Status of Quantum Computing

Arbeitsweise und aktueller Stand der Technik von Quantencomputern

Maximilian Franz Schwärzler

Klasse: 8B

Betreuer: Mag. Markus Kiesenhofer

Schuljahr: 2022/23

Wien, Februar 2023

Abstract

This paper addresses the fundamental and most pressing questions that anyone new to quantum computers will have.

In the beginning it introduces the reader to the fascinating world of quantum physics and the quantum phenomena on which quantum computers are built. The next chapter deals with the physical effects underlying common types of quantum computing hardware. The mathematical knowledge needed to understand quantum operations inside a quantum computer is conveyed in a separate chapter. Based on this mathematical toolbox, quantum gates are introduced. These gates are combined into algorithms and some famous examples are dealt with, going into detail with the Deutsch algorithm. The paper explains why quantum computers are superior to their classical counterparts. The final chapter lists the most influential companies and projects on the field of quantum computing and mentions current challenges.

This work aims at people new to the field of quantum computing who are keen to get an overview of this fast changing technology. After reading this work, a general understanding of quantum computers is achieved and can be used as a basis for further research into the topic.

Preface

I have always been interested in science. Since I was a child I have been active in taking things apart, analyzing them, reading operation manuals of cars and technical equipment, and discovering how things work. Over time, I have developed a deep understanding of the inner workings of all that I explore.

A few years ago, I had the idea to build my own PC. While researching this project, I stumbled upon the new technology of quantum computers. I voraciously consumed every bit of information about this technology. Even after a year of deep dive into the science of quantum computers, I remain as curious as ever about this complex topic.

In this paper I aim to provide a view into the world of quantum computing. Moreover, I want to help the reader better understand the quantum computer.

I would like to express my sincere gratitude to all those who have contributed to the development of this paper, and especially to my advisors, Mag. Markus Kiesenhofer and Mag.a Sonja Schafler. Mag. Markus Kiesenhofer agreed to the topic and graciously took over the supervision. He always supported me patiently, encouraged me to continue writing, and provided productive feedback throughout the entire project. I am truly grateful to Mag.a Schafler for her expertise and support with proofreading.

Lastly I would like to wholeheartedly thank my family and especially my father, with whom I sometimes discussed complex topics long into the night.

Typeset using LaTeX

Vienna, February 6, 2023 Maximilian Schwärzler

Contents

Chapter 1

Introduction

Quantum computers make use of the rather strange effects of quantum physics.[1] This field explores the behavior of matter at its most fundamental state which is entirely different from the ones in everyday life. Phenomena, that would be just ridiculous to think about in the world as we know it, are happening all the time when the objects are microscopically small.

Over the last few years, quantum computers have been in the headlines of newspaper articles more and more often. In 2019, Google announced that they allegedly achieved quantum supremacy.[2] Some articles claim that quantum computers are a threat to our internet, because they can crack the encryption. Other articles however state that quantum computers are still just a thought experiment and will never exist in reality and will forever be a part of science fiction.

This paper will try to answer a couple of questions: First of all what components a quantum computer is made of and why it is different from a classical computer. Mathematical representations of quantum gates and algorithms will also be explained and analyzed. A final chapter will give an overview over the current status of quantum computer research.

[1] See Zeilinger 2003.
[2] See Gibney 2019.

Chapter 2

Essentials of Quantum Mechanics

2.1 The Photoelectric Effect

In 1886 Heinrich Hertz made the first observations in an experiment known as the photoelectric effect. He observed that light directed at a metal sheet was able to eject electrons from its surface. Classical physics stated that the energy, that the electrons have after being ejected, should correlate to the intensity of the light source. In contrast to this, the experimental results showed that the energy did not correlate to the light intensity but the light's frequency (which corresponds to its color). High frequency light had a different impact on the electrons than lower frequencies. Changing the brightness, however, did not make a change to the energy.[3]

2.2 Double-Slit Experiment

Up until 1804 scientists thought that light is a wave — like the waves in water, when a stone is thrown into it or acoustic waves — but nobody was able to find proof for that. Then British scientist Thomas Young came up with the idea for an experiment widely known as the double-slit experiment. The experiment shows that light is not only a wave and does not only consist of particles, but that it is both at the same time, a phenomenon known as wave-particle dualism.

The experiment depicted in Figure 2.1 is made up of a light source that emits visible light, an opaque board with two slits placed at the right positions and a screen that visualizes the light that hits it.[4] The light source is pointed at the sheet, so light only passes through the openings. Behind the slits there is the screen, onto which the light falls.

[3] See Homeister 2018, p. 279.

[4] When Young first conducted the experiments, he used the sun as a light source and a piece of paper as a screen. Nowadays, a monochromatic lamp or an electron beam gun and a photosensitive screen are being used instead.

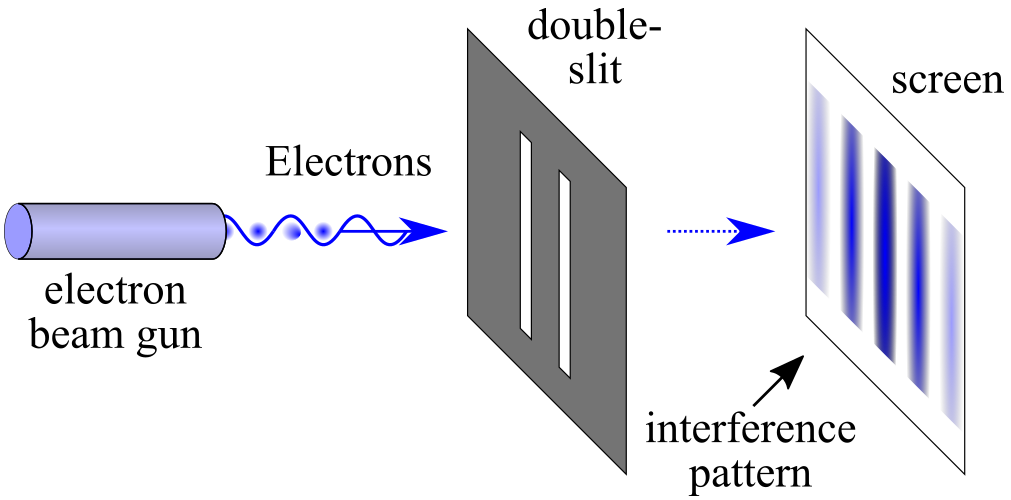


Figure 2.1: *Double slit experiment*

When the light source is switched on, the screen shows an interference pattern, similar to that with water and sound waves. This phenomenon would indicate that light is a wave. Assuming that light is made up of particles, the intensity of the light source could be turned down so low, that only one particle at a time is being emitted. In classical physics, the particles cannot interact with each other anymore and therefore no interference pattern should emerge. However, the experiment shows that over time an interference pattern builds up on the photographic plate, independently of the number of photons emitted at a certain point in time. The question, which slit the quantized light actually passes through, cannot be answered because no measurement has been made in order to determine that.

The next step is to add a sensor that determines which of the two slots the particle passed through. As soon as such a measurement device that might beep every time a particle passes through a slit is fitted, for example, the interference pattern disappears, and only two lines appear behind the slits.[5] However, this behavior supports the thesis that light is made up of particles. The takeaway from the experiment is that light is both, wave and particle, at the same time and behaves differently if certain kinds of measurements are performed on it or not.[6]

2.3 De Broglie hypothesis

Combining Hertz and Young's experimental results Albert Einstein proposed in a paper dated 1905 that light could be quantized and called the associated particles "light quantum". (As Einstein spoke German, the word "Lichtquant" was translated word-for-word.)[7] Only in 1926 American scientist Gilbert N. Lewis proposed the term "photon" instead of "light quantum", which is still in use today.[8] The energy of the photon is dependent on its

[5]See Marianne 2020.
[6]See Maxwell 1864.
[7]See Homeister 2018, p. 255.
[8]See APS News 2012.

wavelength (which is inversely proportional to its frequency f), which, when multiplied with the Planck constant h, gives the energy the electron has when exiting the structure, E_{e^-}.[9] This equation is known as the Planck relation.[10]

$$E_{e^-} = f \cdot h \tag{2.1}$$

In 1923 French scientist Louis De Broglie released a paper for his doctoral dissertation. Based on Einstein's photon theory he claimed that not only photons but all material objects oscillate with their own frequency. To describe this mathematically, he suggested the following equation

$$\lambda = \frac{h}{p} \tag{2.2}$$

where λ is the wavelength of that object, h is Plank's constant and p is the object's momentum. This means that objects with a small mass and therefore a small momentum have a rather long wavelength while bigger objects have a wavelength so short that it is hardly observable. Because the momentum is the product of the mass times the velocity, $p = m \cdot v$, the De Broglie wavelength extends to:[11]

$$\lambda = \frac{h}{p} = \frac{h}{mv} \tag{2.3}$$

De Broglie's conclusion was that all objects oscillate with a frequency inversely proportional to their mass. In the case of the photons which have a rather high frequency, it caused them to interfere with each other and to seemingly go through both slits at the same time.

2.4 Stern-Gerlach Experiment

The Stern-Gerlach experiment was proposed by Otto Stern in 1921 and was conducted by him together with Walther Gerlach in 1922. At this time the atomic model by Niels Bohr was the most popular one, stating that the positively charged core of an atom is being orbited by a certain number of electrons, depending on the element. The electrons move on defined orbits together with a number of other electrons. The orbits fill up starting from the innermost so that the electrons' magnetic fields cancel each other out. Only when an orbit is not completely filled, the whole atom can produce a positive and a negative magnetic pole and behave like a magnet.[12] This property is key for the Stern-Gerlach experiment where a beam of atoms with magnetic poles passes through an inhomogeneous magnetic field.

[9] $h = 6.626\,070\,15 \cdot 10^{-34}\,\mathrm{J} \cdot \mathrm{s}$.
[10] See Homeister 2018, p. 279.
[11] See Jones 2020.
[12] See Bernhardt 2019, pp. 1 sq.

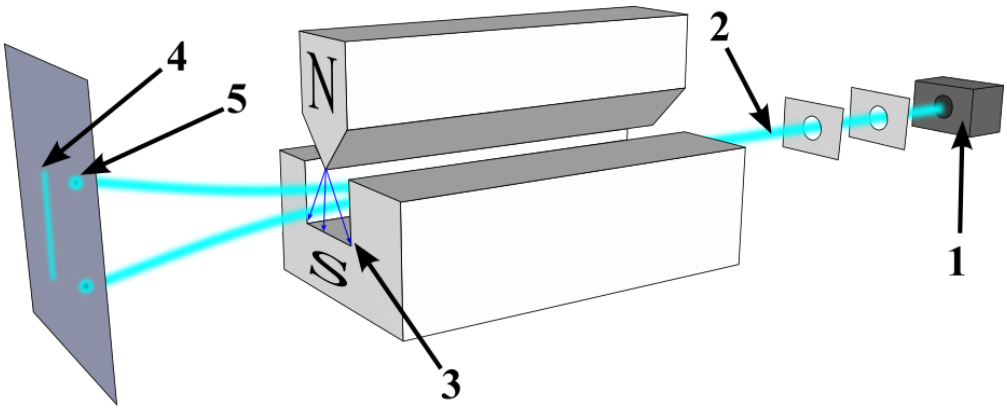


Figure 2.2: *Stern-Gerlach Experiment*

In their first experiment Otto Stern and Walther Gerlach used silver atoms with 47 electrons. Accordingly, a single electron is always alone in an orbit and the atom therefore has magnetic poles.

The experiment uses a magnet with both poles facing the middle fixed in place with a gap in between (Figure 2.2, No. 3). Behind the magnet a screen visualizes the atoms as they hit it (Figure 2.2, No. 4). A beam of silver atoms then gets directed at the screen, passing between the magnets (Figure 2.2, No. 1). The north pole on top is convex and the south pole at the bottom is concave, so that the top magnetic field is stronger than the bottom one. The atoms now could have the south pole on top and the north pole at the bottom. In this case, because the field from the north-facing magnet is stronger and therefore attracts the atom more, the atom gets deflected in the upwards direction. The same applies to the other case where the atom is flipped and the top magnet repels the atom stronger than the bottom, so the atom is deflected downwards.[13]

Common sense would say that the magnetic poles of the atoms are distributed randomly in every orientation. Accordingly, the presumed outcome of the experiment would be a line going from top to bottom (Figure 2.2, No. 4). But Stern and Gerlach observed a different behavior: The screen only showed two dots, one at the top and one at the bottom (Figure 2.2, No. 5). This led to the conclusion that the atoms had one of two vertical orientations only.[14]

2.5 Important quantum phenomena

These early experiments pointed to specific quantum effects which are also key for quantum computing:

Quantization: Quantum systems are in discrete states. This was discovered through the Stern-Gerlach experiment.

[13]See Bernhardt 2019, pp. 1 sq.
[14]See Bernhardt 2019, p. 3.

Superposition: A quantum particle can be in multiple quantum states at the same time. When the particle is put into superposition, it is in multiple classical states at a time. Each of the states has a certain probability of occurring when the superposition is destroyed (by measuring the particle or by decoherence[15]). This is the effect that was discovered with the Double-Slit experiment.

Entanglement: Two quantum particles can be put into superposition, so that their quantum states are "entangled", meaning that the measurement of one of the particles limits or defines the other's state. It is important to note that the physical distance of the particles does not matter once they are entangled. The measurement of one particle leads to the instant definition of the second particle. (However, no *information* can be transmitted, which would break the rules of the special theory of relativity.)[16]

[15]Decoherence time is the time, until a quantum particle looses its superposition and "collapses" into a classical state. Decoherence happens unintentionally, usually through interactions with other atoms or particles.

[16]See Cornwall 2015.

Chapter 3

Basics of Quantum Computer Hardware

3.1 The Qubit

In classical computers a bit, a word combining "binary", meaning two, and "digit", is used to calculate and to store data. A bit can only be in one of two states at a time, either 0 or 1. In quantum computers the fundamental piece of information is called a qubit (derived from "quantum" and "bit"). Throughout most of the quantum operations, such a qubit can be 0 and 1 at the same time, only to collapse into one state when measured.[17]

In a classical computer, a bit is represented by electricity flowing or not flowing. When saving bits, the bit is saved by putting a physical object into a state which it can hold for some time until it is read or changed, e.g. a USB-Drive, a hard drive or a CD.

Qubits also have to be realized in a physical form. For this matter any object that behaves according to the rules of quantum mechanics, has two distinct classical states and can be put in superposition, can act as a qubit. The following chapter will present some ways of how qubits are realized today.

3.2 Physical realizations of qubits

Using photons

Light from common sources like the sun or from lightbulbs usually is not polarized: The waves oscillate in all directions. Waves can be filtered according to their orientation, so only light oscillating in a certain direction can pass through the filter, thus creating polarized light waves.[18] Examples of such polarized light can be seen in Figure 3.1. Such a filter represents a

[17] See Bernhardt 2019, pp. 49 sq.
[18] See Homeister 2018, p. 259.

measurement to the original superposition of quantum states into which the system collapses into one state according to its probabilities. Using special optics like semitransparent mirrors, it is also possible to create entangled photons.

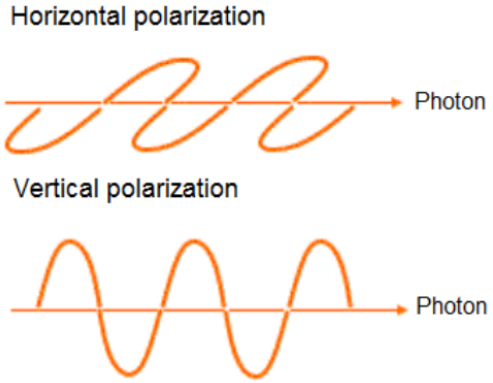


Figure 3.1: *States of a qubit realized with photons*

Using Ions

Ions are electrically charged atoms and therefore can be held in place with an electromagnetic field. This feature is utilized in a "Paul Trap" with temperatures close to absolute zero. The Paul Trap was developed by Wolfgang Paul who, together with Hans Dehmelt, was awarded the Nobel Prize for their research.[19] Figure 3.2 shows a schematic drawing of a Paul trap. The Paul trap is at the heart of an ion quantum computer. To achieve such low temperatures, the ions are cooled by a laser.[20] Even multiple ions can be trapped and therefore there needs to be a way to manipulate a single ion. Directing a very focused laser beam with a specific frequency at a single ion can manipulate its quantum state to make the ion act as a qubit.[21]

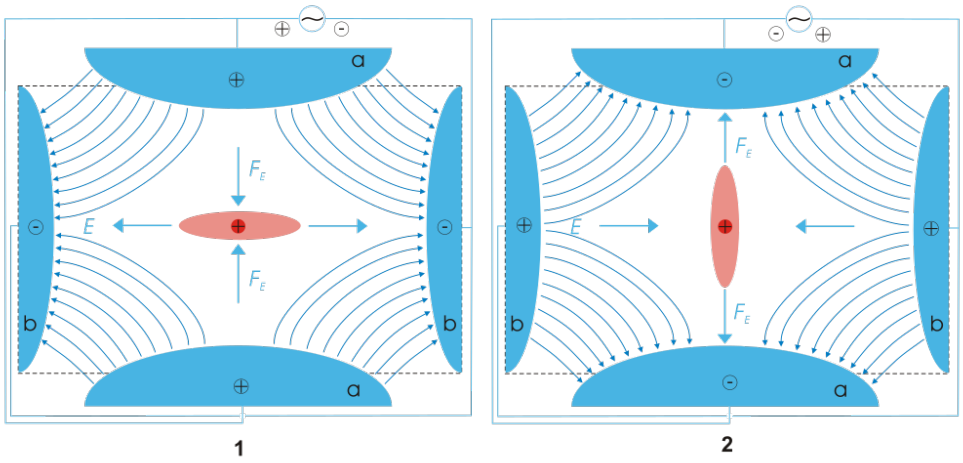


Figure 3.2: *A schematic drawing of the Paul Trap*

[19] See Podlesnic 2022, p. 8.
[20] See Podlesnic 2022, pp. 13–16.
[21] See Blatt et al. 2004, p. 64.

Using superconductors

In an annular superconductor currents can flow without resistance.[22] For this to happen the superconductors have to be cooled down close to absolute zero. When a current is induced inside the superconductor a so-called flux, a never-ending ring current, can be created. The qubit states are encoded with the direction of the flowing current. A superposition of both directions is also possible.[23]

As described by the BCS-Theory, short for Bardeen-Cooper-Schrieffer, electrons in superconductors pair up to so-called Cooper-Pairs and then have a common wave function. When two superconductors are placed very close to each other, electrons in superposition have a possibility to "tunnel" from one ring to the other when measured. This setup is called a Josephson-Contact and can be seen in Figure 3.3.[24]

These superconducting qubits are more likely to be manipulated by noise like fluctuations of the magnetic fields than other types of qubits. However, since these qubits are on a macroscopic scale when using Josephson junctions it is easier to observe quantum effects like entanglement and coherence on them.[25]

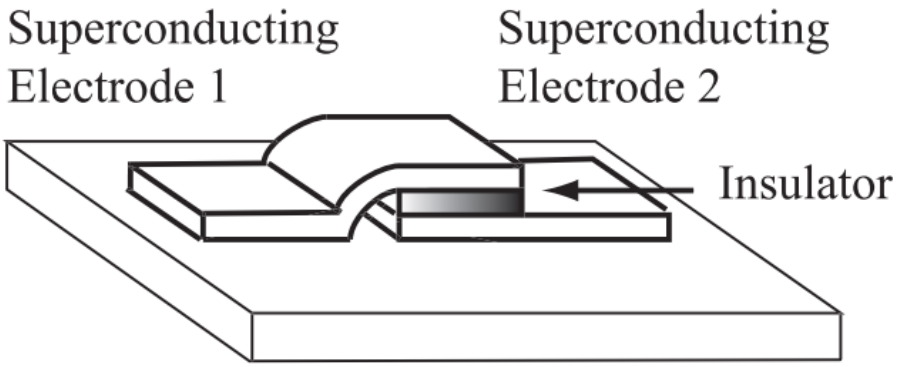


Figure 3.3: *A schematic drawing of a Josephson Contact*

Other types

There are also some other methods of realizing qubits such as Nuclear Magnetic Resonance (NMR), Majorana fermions and Diamond Nitrogen Vacancy-Centers. Multiple neutral atoms can also be trapped and cooled in an array similar to ions.[26]

[22] See Woody, George, and Velasco 2005, p. 1.
[23] See Homeister 2018, p. 270.
[24] See Homeister 2018, pp. 269 sq.
[25] See Dong et al. 2015, p. 1.
[26] See Medium 2021.

3.3 DiVincenzo Criteria

In 1997 David DiVincenzo, an American physicist, proposed the DiVincenzo criteria in collaboration with Daniel Loss. According to them a quantum computer can only be considered viable if it fulfills the following set of rules.[27]

1. **Scalable and well-defined data storage through qubits**
   Every quantum computer needs to have a way to store information. Just like a classical computer that has memory to store information, a quantum computer needs a way to represent and store qubits. The qubits can be represented in many ways, for example with electrons, a spin 1/2 nucleus or two orthogonal polarization states of a photon. Another important criterion is that every qubit has to be addressable separately and the number of qubits has to be scalable easily. A way to make quantum computers more viable is using mixed types of qubits.

2. **Resetting into an initial state**
   A quantum computer needs to have a way to reset its qubits to an initial and well-defined state like $|00\ldots0\rangle$. In many physical realizations of quantum computers this is done by cooling the system down to its ground state.

3. **Decoherence times $\gg$ gate operation times**
   Another problem with building reliable quantum computers is decoherence. While classical computers usually store information for about a decade, quantum computers are very sensitive to external noise that could interfere with the computed state. The time until a certain quantum state is disturbed and hence unreliable to continue working is called decoherence time. This timespan is usually close to a few microseconds. Nevertheless, the decoherence time itself is not as important as the ratio between it and the gate operation time for that quantum computer. When the time needed to apply a gate to a state is much shorter than the decoherence time, a quantum computer can still execute many operations before the state is destroyed. The decoherence time as well as the gate operation time vary between the different technologies. Nonetheless, an increase of the decoherence time can, for example, be achieved with quantum error correcting codes.

4. **Universal quantum gates**
   A quantum computer needs universal quantum gates. A universal gate in the general sense is a gate, which can be combined with itself to emulate all other possible gate operations on a classical computer. An example of a classical universal gate will be illustrated in section 5.1 with the NAND Gate.

[27] See Nakahara and Ohmi 2008, pp. 234 sqq.

5. **A way to measure the qubits**
   In classical computers reading the result of a computation is very trivial in contrast to quantum computing. While a classical computer only has to read the bits and then print them out on paper or on the screen, the physical realizations of the qubits on a quantum computer vary enormously.

6. **A way to convert the qubits between the different realizations**
   Whereas some types of qubits might be better for performing operations on them others might be better for transporting quantum information within the computer. Therefore, a quantum computer could involve several types of qubits and needs a way of converting from one to the other. This is similar to a classical computer. In this case the CPU uses semiconductors to save information and the hard drive uses a magnetized disk.

7. **A reliable way to transmit qubits to other systems**
   This criterion becomes important in connection with distributed quantum computing and quantum communication such as quantum key distribution.

Chapter 4

Mathematical Principles

4.1 Introduction

Quantum mechanical effects and the operations of quantum computers can be mathematically described using the concepts of linear algebra.[28] However, some specific features are best captured using the bra-ket notation, originally introduced by Paul Dirac in 1939 and accordingly also known as the Dirac notation.

The following chapters very briefly summarize the most important notations and concepts which are widely used in quantum mechanics and quantum computing.

4.2 Bra-Ket Notation

The bra-ket notation uses bras and kets, which are written like in Equation 4.1 and Equation 4.2. The name is derived from the word **bra**-(c)**ket**. A bra is a vector with the entries arranged horizontally, while a ket has the entries arranged vertically.

$$\langle a| = \begin{bmatrix} 1 & 0 & -\pi & 23 \end{bmatrix} \tag{4.1}$$

$$|a\rangle = \begin{bmatrix} 2 \\ 0.5 \\ -3 \end{bmatrix} \tag{4.2}$$

Bras and kets are used to describe a certain quantum state. Most of the time it does not matter if kets or bras are used. In the remainder of this work kets will be used. However, when performing operations on qubits, special rules for the bra-ket notation apply, as described in the next section.[29]

[28] See Bernhardt 2019, p. 17.
[29] See Bernhardt 2019, p. 19.

4.3 Bra-Ket Operations

Bra-ket operations are performed on vectors, but with a clear differentiation between row and column vectors. The bra $\langle a|$ stands for a row vector $\vec{a}$ and the ket $|a\rangle$ for an otherwise identical vector, but in column notation. For both the general rules for vector mathematics apply. However, the bra-ket notation has some extra ones.

Addition Only bras and kets of the same type and dimension, e.g. two bras or two kets with both n dimensions can be added up, as seen below:

$$|a\rangle + |b\rangle = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} a_0 + b_0 \\ a_1 + b_1 \\ \vdots \\ a_n + b_n \end{bmatrix} \tag{4.3}$$

Multiplication A bra and a ket can be multiplied with each other, or vice versa. Because a bra looks like $\langle a|$ and a ket looks like $|b\rangle$, the vertical line is usually merged, thus written as $\langle a|b\rangle$. Mathematically, this operation is the dot product of two vectors:

$$\langle a|b\rangle = \begin{bmatrix} a_0 & a_1 & \cdots & a_n \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} = a_0 b_0 + a_1 b_1 + \cdots + a_n b_n \tag{4.4}$$

4.4 Orthonormal Bases

As seen in chapter 3, a measurement of the qubits has to be made in order to determine their final state. With photons, for example, a measurement of their polarization is possible, e.g. horizontally and vertically. This setup defines the basis against which the photons are measured, in this case a two-dimensional which shall be represented by two-dimensional kets.

Generally, an orthonormal basis for R^n always consists of n unit vectors $|b_1\rangle, |b_2\rangle, \cdots, |b_n\rangle$ that are orthogonal with each other.[30] For a single qubit the basis vectors are in R^2 whereas for n qubits basis vectors in R^{2^n} are needed. Using said orthonormal basis, the state of a qubit can be expressed as a linear combination of the basis vectors. In the case of a single qubit these quantum states are written as

$$\alpha \cdot |0\rangle + \beta \cdot |1\rangle \tag{4.5}$$

[30] See Bernhardt 2019, pp. 21 sqq.

where α and β are complex numbers representing the amplitudes of the states, with:

$$|\alpha|^2 + |\beta|^2 = 1 \tag{4.6}$$

$|0\rangle$ and $|1\rangle$ stand for $|b_0\rangle$ and $|b_1\rangle$, following the convention of bits in a classical computer. These rules allow a qubit to be in two classical states at the same time, a phenomenon called superposition.[31]

While in superposition a qubit has the probability amplitudes of α and β. But as soon as a qubit is measured, it falls into either $|0\rangle$ or $|1\rangle$, with the probabilities of $|\alpha|^2$ and $|\beta|^2$, respectively.[32]

The so-called *standard basis* is where $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Both of the vectors are unit vectors ($\langle b_1|b_1\rangle = \langle b_2|b_2\rangle = 1$) and are perpendicular with each other ($\langle b_1|b_2\rangle = 0$). This can be interpreted as measuring vertical and horizontal polarization on photons or "spin up" and "spin down" on electrons with regard to a 0° rotation. Rotating the measurement apparatus can give an indefinitely big amount of bases, for example the following:[33]

$$\left\{ \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{bmatrix}, \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \right\} \tag{4.7}$$

$$\left\{ \begin{bmatrix} \frac{1}{2} \\ \frac{-\sqrt{3}}{2} \end{bmatrix}, \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{bmatrix} \right\} \tag{4.8}$$

The notation $\{|b_1\rangle, |b_2\rangle\}$, defines an unordered basis where the order of the vectors does not matter. An ordered basis however is written as $(|b_1\rangle, |b_2\rangle)$ and the order does matter. So, $\{|b_1\rangle, |b_2\rangle\} = \{|b_2\rangle, |b_1\rangle\}$, but $(|b_1\rangle, |b_2\rangle) \neq (|b_2\rangle, |b_1\rangle)$.[34]

4.5 Matrices

A matrix is an array of numbers with m rows and n columns, and is therefore called an $m \times n$ matrix, e.g. $A = \begin{bmatrix} 1 & -4 & 2 \\ 2 & 3 & 0 \end{bmatrix}$ and $B = \begin{bmatrix} 1 & 2 \\ 7 & 5 \\ 6 & 1 \end{bmatrix}$.

When *transposing* a matrix M, written as M^T, all the rows are interchanged to columns and all the columns to rows. A and B in this case would be $A^T = \begin{bmatrix} 1 & 2 \\ -4 & 3 \\ 2 & 0 \end{bmatrix}$ and $B^T = \begin{bmatrix} 1 & 7 & 6 \\ 2 & 5 & 1 \end{bmatrix}$.

[31] See Homeister 2018, p. 20.
[32] See Homeister 2018, pp. 21 sq.
[33] See Bernhardt 2019, p. 26.
[34] See Bernhardt 2019, p. 29.

Kets are matrices with one column, and bras are matrices with one row. This means that $\langle a| = |a\rangle^T$ and $|a\rangle = \langle a|^T$.

The product of two matrices A and B, written as AB, uses the idea of the bra-ket product and the first matrix can be thought of as bras stacked on one another, and the second matrix as kets standing next to each other. (Bras always come before kets.) It is important to remember that the dimension of the bras has to equal the dimension of the kets, otherwise a multiplication of the two is not possible. Below the matrix product AB with the bra-ket product denoted by $\langle a_i|b_j\rangle$ in the ith row and the jth column is illustrated:[35]

$$AB = \begin{bmatrix} \langle a_1|b_1\rangle & \langle a_1|b_2\rangle & \cdots & \langle a_1|b_j\rangle & \cdots & \langle a_1|b_n\rangle \\ \langle a_2|b_1\rangle & \langle a_2|b_2\rangle & \cdots & \langle a_2|b_j\rangle & \cdots & \langle a_2|b_n\rangle \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \langle a_i|b_1\rangle & \langle a_i|b_2\rangle & \cdots & \langle a_i|b_j\rangle & \cdots & \langle a_i|b_n\rangle \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \langle a_m|b_1\rangle & \langle a_m|b_2\rangle & \cdots & \langle a_m|b_j\rangle & \cdots & \langle a_m|b_n\rangle \end{bmatrix} \tag{4.9}$$

A matrix with the same number of rows and columns ($m = n$) is called a *square* matrix. When a square matrix has all entries on the main diagonal set to 1 and all other ones to 0, it is called an *identity* matrix. An identity matrix with $n \times n$ dimensions is denoted as I_n. Multiplying a matrix with the identity matrix of the right dimension is equivalent to multiplying by 1.[36]

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ \ldots \tag{4.10}$$

As mentioned earlier, it often needs to be checked if a list of n-dimensional kets $\{|b_1\rangle, |b_2\rangle, \cdots, |b_n\rangle\}$ describe an orthonormal basis. However, instead of checking every vector for unity and then building the dot product of every possible combination, a matrix can be used instead, assuming that the $n \times n$ matrix $A = \begin{bmatrix} |b_1\rangle & |b_2\rangle & \cdots & |b_n\rangle \end{bmatrix}$. First this matrix is transposed, $A^T = \begin{bmatrix} \langle b_1| \\ \langle b_2| \\ \vdots \\ \langle b_n| \end{bmatrix}$, then the product is calculated $A^T A$ and it is checked if $A^T A = I_n$.

$$A^T A = \begin{bmatrix} \langle b_1| \\ \langle b_2| \\ \vdots \\ \langle b_n| \end{bmatrix} \begin{bmatrix} |b_1\rangle & |b_2\rangle & \cdots & |b_n\rangle \end{bmatrix} = \begin{bmatrix} \langle b_1|b_1\rangle & \langle b_1|b_2\rangle & \cdots & \langle b_1|b_n\rangle \\ \langle b_2|b_1\rangle & \langle b_2|b_2\rangle & \cdots & \langle b_2|b_n\rangle \\ \vdots & \vdots & \vdots & \vdots \\ \langle b_n|b_1\rangle & \langle b_n|b_2\rangle & \cdots & \langle b_n|b_n\rangle \end{bmatrix} \tag{4.11}$$

[35] See Bernhardt 2019, pp. 30 sqq.
[36] See Bernhardt 2019, p. 32.

The main diagonal is the multiplication of every vector with itself, checking if it is a unit vector, equalling 1 if true. All the other entries are the multiplications of all the vectors with each other, checking if they are orthogonal, equalling 0 if this is the case. Fortunately, this pattern is the identity matrix of n dimensions and therefore if $A^T A = I_n$ is true, then this specific set of vectors describes an orthonormal basis and can be used to measure qubits.[37]

In connection with qubits, matrices either represent orthonormal bases or operations on qubits, changing their state. For a single qubit these matrices have to be of the dimension 2×2. Furthermore, matrices describing operations have to be unitary, meaning that they have to fulfill the following condition: $A^\dagger = A^{-1}$.[38] Multiplying the state vector of a qubit with such a matrix is equivalent to executing the operation. Such operations will be explained in chapter 5.

4.6 Tensors and Entanglement

The state of a single qubit measured in the basis $(|a_0\rangle\,,\ |a_1\rangle)$ can be described by $|v\rangle = c_0\,|a_0\rangle + c_1\,|a_1\rangle$. Another qubit is measured in the basis $(|b_0\rangle\,,\ |b_1\rangle)$ and is given by $|w\rangle = d_0\,|b_0\rangle + d_1\,|b_1\rangle$. Using the so-called *tensor* product or Kronecker product, the qubits are combined into a single linear combination. The tensor product is denoted by $\otimes$, so the tensor product of the qubits $|v\rangle$ and $|w\rangle$ would be $|v\rangle \otimes |w\rangle$. In expanded form this is written as $|v\rangle \otimes |w\rangle = (c_0\,|a_0\rangle + c_1\,|a_1\rangle) \otimes (d_0\,|b_0\rangle + d_1\,|b_1\rangle)$. The tensor product itself is multiplied just like $(a+b)(c+d)$.[39]

$$\begin{aligned}
&(c_0\,|a_0\rangle + c_1\,|a_1\rangle) \otimes (d_0\,|b_0\rangle + d_1\,|b_1\rangle) \\
&= c_0 d_0\,|a_0\rangle \otimes |b_0\rangle + \quad c_0 d_1\,|a_0\rangle \otimes |b_1\rangle + \quad c_1 d_0\,|a_1\rangle \otimes |b_0\rangle + \quad c_1 d_1\,|a_1\rangle \otimes |b_1\rangle \\
&= \underbrace{c_0 d_0}_{\text{r}}\,|a_0\rangle\,|b_0\rangle + \quad \underbrace{c_0 d_1}_{\text{s}}\,|a_0\rangle\,|b_1\rangle + \quad \underbrace{c_1 d_0}_{\text{t}}\,|a_1\rangle\,|b_0\rangle + \quad \underbrace{c_1 d_1}_{\text{u}}\,|a_1\rangle\,|b_1\rangle
\end{aligned} \tag{4.12}$$

where r, s, t and u are the probability amplitudes with regard to the new basis $|a_0\rangle\,|b_0\rangle\,,\ |a_0\rangle\,|b_1\rangle\,,\ |a_1\rangle\,|b_0\rangle$ and $|a_1\rangle\,|b_1\rangle$. Accordingly, $r^2 + s^2 + t^2 + u^2 = 1$. If $ru = st$, meaning that ru as well as st equal $c_0 c_1 d_0 d_1$, the qubits $|v\rangle$ and $|w\rangle$ are not *entangled*. However, if $ru \neq st$, the qubits are *entangled*.[40] Next to superposition, entanglement is the second important distinction of quantum computing from classical computing. When the qubits in a quantum system are entangled, then measurement of one leads to the instant definition of all the others. Additionally, the probability amplitudes are linked, so that modifications on one qubit lead to a change in state of the other ones. When expressing the state in terms of tensors, when $ru \neq st$, no separation of equal terms per qubit is possible and therefore a dependency exists.[41]

[37] See Bernhardt 2019, p. 33.
[38] $A^\dagger$ is the complex conjugated and transposed matrix A, also called the *adjugate* matrix.
[39] See Bernhardt 2019, pp. 57 sq.
[40] For details and examples, see Bernhardt 2019, pp. 59 sqq.
[41] See Bernhardt 2019, pp. 58 sq.

Assuming that both of the qubits mentioned above are measured in the standard basis $|a_0\rangle = \begin{bmatrix}1\\0\end{bmatrix}$ and $|a_1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$, the basis vectors can be combined with the help of the tensor product.

$$\begin{aligned} & r\begin{bmatrix}1\\0\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix} + s\begin{bmatrix}1\\0\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix} + t\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix} + u\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix} \\ \text{with the basis } & \Bigg(\underbrace{\begin{bmatrix}1\\0\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix}}, \underbrace{\begin{bmatrix}1\\0\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}}, \underbrace{\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix}}, \underbrace{\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}}\Bigg) \\ \text{written as } & \left(\begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right) \end{aligned} \tag{4.13}$$

4.7 Linear Algebra Toolbox

There are three essential operations which have to be performed very often, so a brief overview is provided below:[42]

1. With a number of kets, check if they describe an orthonormal basis. First set $A = \Big[|b_1\rangle \quad |b_2\rangle \quad \cdots \quad |b_n\rangle\Big]$. Then compute $A^T A$. Finally, check if $A^T A$ is an identity matrix, and if it is, the kets describe an orthonormal basis, otherwise they do not.

2. With an orthonormal basis and a ket $|v\rangle$, write the ket as a linear combination of the basis vectors. First solve the following equation: $|v\rangle = x_1\,|b_1\rangle + \cdots x_i\,|b_i\rangle \cdots + x_n\,|b_n\rangle$. Then, combines all basis vectors to a matrix like $A = \Big[|b_1\rangle \quad |b_2\rangle \quad \cdots \quad |b_n\rangle\Big]$. Then $\begin{bmatrix}x_1\\x_2\\\vdots\\x_n\end{bmatrix} = A^T\,|v\rangle = \begin{bmatrix}\langle b_1|v\rangle\\\langle b_2|v\rangle\\\vdots\\\langle b_n|v\rangle\end{bmatrix}$

3. With an orthonormal basis and a ket as the linear combination of the basis vectors, find the length of the ket $|v\rangle$ with $|\,|v\rangle\,|^2 = c_1^2 + c_2^2 + \cdots + c_i^2 \cdots + \cdots c_n^2$.

[42] See Bernhardt 2019, pp. 35 sq.

Chapter 5

Quantum Operators

This chapter will explain the ways a quantum computer can interact with qubits with the help of operators. These operators are the logic building blocks of quantum algorithms. Quantum operations exist independently of quantum hardware.

5.1 Classical logic gates

A classical computer works with Boolean algebra which was named after George Bool, a British mathematician, who came up with a way to define logic mathematically. Boolean algebra can be performed with three basic operations: AND, OR and NOT. These basic building blocks, also called gates or logic gates, can be combined to make more complex gates. Every gate has inputs and outputs. Boolean algebra uses only two states: true and false. In classical computers they are often represented as electricity flowing or not flowing. In computer programming the notation 0 for false and 1 for true is also very common.[43]

NOT, OR and AND: The simplest gate is the NOT gate. The input is a statement and the output is the inverse of the statement. If the statement on the input is true, then the output is false. If the input statement is false, the output is true. Considering the statement x, the negation can be written as $\neg x$. The truth table for NOT is shown in Table 5.1.[44]

The OR gate has two inputs and one output. The output is true if any of the input statements or both are true. It is denoted by $\vee$. If x or y or both are true, then $x \vee y$ is also true, otherwise it is false.[45]

[43] See Petzold 2014, p. 87.
[44] See Bernhardt 2019, p. 90.
[45] See Bernhardt 2019, p. 91.

The AND gate also has two inputs and one output. The output of the AND gate is only true if both statements are true and is denoted by $\wedge$. For example, considering the statements x and y, $x \wedge y$ is only true if x and y are true.[46]

NOT gate

x	$\neg x$
0	1
1	0

OR gate

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

AND gate

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

Table 5.1: *The logic tables for the NOT, OR and AND gates*

XOR and NAND: The *exclusive* OR (XOR) is another important gate and is denoted by $\oplus$. It is similar to OR, with the difference that it is not true when both statements are true. It is most often used to check if two statements are different.

The NAND is a gate that combines an AND gate and a NOT gate. NAND is said to be "functionally complete" or "universal" in classical computing, meaning that any truth table (like the ones above) can be constructed with NAND gates only. In quantum computing, however, the NAND gate is not universal.[47]

XOR gate

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

NAND gate

x	y	$x \uparrow y$
0	0	1
0	1	1
1	0	1
1	1	0

Table 5.2: *The logic tables for the XOR and NAND gates*

CNOT and Toffoli: Until now gates with one or two inputs have been discussed, but they always had only one output. The CNOT and the Toffoli gate have two inputs and two outputs. The word CNOT is a portmanteau word combining controlled and not. Here the first bit passes through so that the first output is always equal to the first input and the second input is flipped (NOT) only if the first bit is true. If the first bit is 0, then the output is equal to the input. If the first bit is 1, then the first bit stays the same, but the second bit is flipped.

The Toffoli gate is similar in functionality but has three inputs and three outputs. The last bit is flipped if the first and the second bit are true (AND). The first two bits pass through like in the CNOT gate. A specialty of the CNOT and the Toffoli gate is that they are their own inverse. This means that if either two CNOT or two Toffoli

[46] See Bernhardt 2019, pp. 90 sq.
[47] See Bernhardt 2019, pp. 96–101.

gates are put in series, the second gate reverses all changes done by the first gate, thus returning the initial input state.[48]

CNOT gate

Input		Output	
x	y	x	$x \oplus y$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Toffoli gate

Input			Output		
x	y	z	x	y	$(x \wedge y) \oplus z$
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Table 5.3: *Truth tables for the CNOT and the Toffoli gate*

In a classical computer these gates are combined to form more complex gates and logic building blocks like half and full adders. Together with just a few other features like storage capability using flip-flops an adding machine is at the heart of any classical computer.

5.2 Quantum logic gates

Now lets turn to the quantum computing equivalent of classical logic gates, the quantum logic gates.

As discussed in chapter 4 the definition of a basis in which to measure a qubit is needed. For the sake of simplicity the standard basis $\left(\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}\right)$ will be used to measure the qubits. The first ket $\begin{bmatrix}1\\0\end{bmatrix}$ represents the first possible measurement outcome denoted by $|0\rangle$ and is equivalent to 0 or false in classical computers. The second ket $\begin{bmatrix}0\\1\end{bmatrix}$ is the other possible outcome and is denoted by $|1\rangle$. As the basis will not be changed during computation, the state of the qubits will be changed by applying operators to them.

5.2.1 Single-Qubit gates

Single-Qubit gates act on one qubit alone and therefore cannot create or destroy entanglement. However, superposition can be achieved and modified by these gates.

Hadamard gate: The Hadamard gate is one of the most essential gates and immediately shows the advantage of quantum algorithms over classical algorithms. Through apply-

[48] See Bernhardt 2019, pp. 105–108.

ing the Hadamard gate on a basis state like $|0\rangle$ or $|1\rangle$, an equal superposition of the basis vectors is created.[49]

Figure 5.1 shows the matrix representation of the Hadamard gate on the left and its symbolic representation on the right side.

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad -\boxed{H}-$$

Figure 5.1: *Hadamard gate, matrix and symbol*

When the Hadamard gate is applied on a qubit in its basis state it puts the qubit in superposition as shown in Figure 5.2.

$$H|0\rangle = H \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \tag{5.1}$$

$$H|1\rangle = H \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \tag{5.2}$$

Figure 5.2: *The effect of the Hadamard gate on basis states*

In both cases a measurement on the qubit in superposition will result with a probability of 50 percent in $|0\rangle$ and also 50 percent probability in $|1\rangle$. This is because the probability of being measured in a specific quantum basis state is the square of its probability amplitude, in these cases the square of $\frac{1}{\sqrt{2}}$. However, the two superposition states before measurement are not identical as the probability amplitude of $|1\rangle$ is negative in Equation 5.2. This fact is regularly exploited in quantum algorithms.

Pauli gates: There are four more gates[50] operating on individual qubits which are named after Wolfgang Pauli[51], the I, Z, X and Y gates.

I gate

The I gate, also known as the identity gate, is the quantum equivalent of multiplying by 1 and therefore has no effect on the qubit. It does not have an obvious use case in quantum algorithms, but is used as a fill-in for a qubit if there is no operation on it in a specific step, as will be explained later.

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad -\boxed{I}-$$

Figure 5.3: *I gate, it leaves qubits unchanged*

[49] See Portugal 2022, pp. 7 sq.
[50] See Bernhardt 2019, pp. 121 sq.
[51] See Von Meyenn and Schucking 2001.

Z gate

The Z gate does not affect the amplitude of $|0\rangle$ but changes the sign of the amplitude of $|1\rangle$.

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad -\boxed{Z}-$$

Figure 5.4: *Z gate, only changes the sign of the second probability amplitude*

X and Y gate

X and Y are the quantum equivalents to the classical NOT gate and interchange the probability amplitudes of $|0\rangle$ and $|1\rangle$. The X gate just flips, while Y flips and changes the sign of the probability amplitude for the second basis.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad -\boxed{X}-$$

$$Y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad -\boxed{Y}-$$

Figure 5.5: *X and Y gates, the quantum equivalents to NOT*

5.2.2 Multi-Qubit gates

With gates that act on two or more qubits at the same time, one qubit's state after the operation often depends on the other qubit's state before the operation. These gates are able to entangle the input qubits.

CNOT gate: The CNOT gate was already introduced in the chapter about classical gates. However, because a qubit can now be in a superposition of the two base states, there are endless states possible after this gate. When porting the truth table for the CNOT gate to the quantum world, the classical states of 0 and 1 are now described by the quantum basis vectors $|0\rangle$ and $|1\rangle$. The CNOT gate is also an example of a universal quantum gate.

Input		Output	
x	y	x	$x \oplus y$
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$

Input	Output
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|01\rangle$
$\|10\rangle$	$\|11\rangle$
$\|11\rangle$	$\|10\rangle$

Table 5.4: *Quantum CNOT gate*

Using the tensor notation introduced earlier, a combination of the input qubits and the output qubits into a quantum system is possible. This has a number of advantages as will be explained later. To better understand what effect the CNOT gate has on the qubits, the state can be written as a linear combination of the basis vectors:

$$\text{CNOT}\left(r\left|00\right\rangle + s\left|01\right\rangle + t\left|10\right\rangle + u\left|11\right\rangle\right) = r\left|00\right\rangle + s\left|01\right\rangle + u\left|10\right\rangle + t\left|11\right\rangle \tag{5.3}$$

Note that the CNOT gate swaps the probability amplitudes t and u. The following is the matrix representation and the effect of the CNOT gate:

$$\text{CNOT} = \begin{bmatrix} I_2 & \\ & X \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Figure 5.6: *Matrix representation of the CNOT gate*

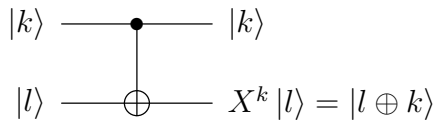


Figure 5.7: *The effect of the CNOT gate on a set of qubits*

According to the truth table and the CNOT's representation, the first qubit stays unchanged, whereas the second one is flipped if and only if the first qubit is in the state $|1\rangle$. However, when applied to a qubit in superposition like

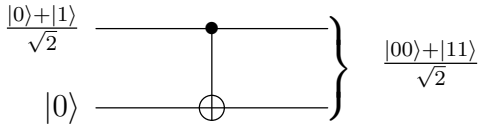


Figure 5.8: *CNOT gate applied to a qubit in superposition*

the output is an entangled state of the two qubits. To understand the state in which the qubits are after applying the CNOT gate, the procedure will be analyzed with linear algebra.

As seen earlier in section 4.6, the Kronecker product can be used to combine two qubit states into a tensor describing the common input state.

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |0\rangle = \frac{1}{\sqrt{2}} |0\rangle \otimes |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \otimes |0\rangle = \frac{|00\rangle + |10\rangle}{\sqrt{2}}$$

Applying the CNOT gate to the input gives:

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \tag{5.4}$$

Since $ru \neq st$ this final state is entangled, so that the qubits cannot be separated into their own expressions anymore.

5.3 Quantum algorithms as matrix multiplications

The quantum operators from section 5.2 together with some more will be combined to form a quantum algorithm. Every quantum operator can be described using a unitary matrix similar to Figure 5.1 or Figure 5.3 to Figure 5.6. The product of two unitary matrices is always another unitary matrix. Therefore, multiple gates can be combined to a single matrix to describe an algorithm.

To combine the matrices, the following rules apply: Two gates which act on separate qubits at the same time are combined using the Kronecker product. Gates in series are combined using standard matrix multiplication, from back to front. If no gate is applied to a qubit at a certain time, the place is filled with an I gate of the appropriate size.

The following algorithm is considered:

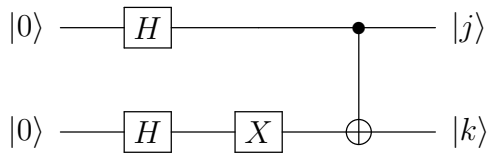


Figure 5.9: *Sample quantum algorithm*

First, the input state is expressed as a matrix:

$$|0\rangle |0\rangle = |00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.5}$$

The first set of Hadamard gates is combined using the Kronecker product:

$$H \otimes H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{5.6}$$

In the second step of Figure 5.9 there is no operation on the first qubit, so an I_2 gate is substituted as the first factor of the Kronecker product:

$$I_2 \otimes X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.7}$$

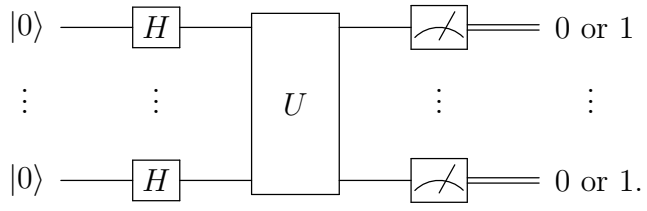
For the last step the Kronecker product is not needed anymore, because the CNOT acts on both qubits. These 4×4 matrices are now multiplied to form a matrix describing all operations.

$$\frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{5.8}$$

The last matrix represents the complete quantum algorithm shown in Figure 5.9.

5.4 Quantum parallelism

In its most general form, a quantum algorithm is described by the following gates, with U being a unitary operator:



In the circuit above all qubits are initially in the state $|0\rangle$. Then they are put into superposition by the H gates and the unitary operator U is applied. Finally, the qubit states are measured to yield a result.[52]

[52] See Portugal 2022, p. 24.

Using a quantum computer for an algorithm with more than one input all the possible solutions can be executed simultaneously. The same algorithm would take exponential time on a classical computer. If the input x is an n long number of bits, the gate to operate on these bits has to be a $2^n \times 2^n$ matrix. After the measurement the output y is n bits long again. The Hadamard gate at the beginning is always applied to all the qubits at the same time and produces a superposition of all possible outcomes. Similarly, the U gate is applied to all the qubits *simultaneously*. This means that 2^n calculations on all the possible inputs are calculated at once. A classical computer would have to calculate all the 2^n possibilities one after the other and would therefore need exponential time. Thus, quantum parallelism is one of the advantages of a quantum computer over a classical one.[53]

5.5 No Cloning Theorem

In classical computation a bit can be copied by using a fan-out operation. A fan-out is an output to which two connections are made. The two connections then always have the same state. In quantum computers, however, it is not possible to simply copy or "clone" a general qubit, as is proposed in Figure 5.10.

An exception are qubits in the states $|0\rangle$ or $|1\rangle$. They can be copied with a CNOT gate and an extra ancilla bit that is always 0.

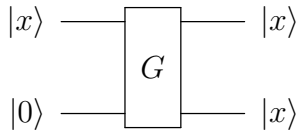


Figure 5.10: *Can this circuit exist?*

Using the proposed gate G, the two resulting qubits would not be entangled as the second qubit would be a copy of the first one. This gate could exist only when using basis vectors ($CNOT(|0\rangle|0\rangle) = |0\rangle|0\rangle$ and $CNOT(|1\rangle|0\rangle) = |1\rangle|1\rangle$). However, for an arbitrary qubit $|x\rangle$, this is not true. For example, trying to clone the qubit $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ does not yield the anticipated result. As can be seen in Figure 5.8, this input to a CNOT gate entangles the qubits and does not clone the first bit.[54] There is also a formal proof that cloning of qubits is not possible:

[53] See Portugal 2022, pp. 24 sq.
[54] See Bernhardt 2019, p. 125.

Proof that gate G cannot exist. Assuming that G exists, its cloning properties can be described using the following rules:

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

These three statements can be restated as follows:

1. $G(|00\rangle) = |00\rangle$.
2. $G(|10\rangle) = |11\rangle$.
3. $G(\frac{1}{\sqrt{2}}\,|00\rangle + \frac{1}{\sqrt{2}}\,|10\rangle) = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$.

The gate G, like all matrix operations, must be linear meaning that

$G(\frac{1}{\sqrt{2}}\,|00\rangle + \frac{1}{\sqrt{2}}\,|10\rangle) = \frac{1}{\sqrt{2}}G(|00\rangle) + \frac{1}{\sqrt{2}}G(|10\rangle)$.

Replacing $G(|00\rangle)$ and $G(|10\rangle)$ using statements (1) and (2) gives

$G(\frac{1}{\sqrt{2}}\,|00\rangle + \frac{1}{\sqrt{2}}\,|10\rangle) = \frac{1}{\sqrt{2}}\,|00\rangle + \frac{1}{\sqrt{2}}\,|10\rangle$.

But statement (3) says that

$G(\frac{1}{\sqrt{2}}\,|00\rangle + \frac{1}{\sqrt{2}}\,|10\rangle) = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$.

However,

$\frac{1}{\sqrt{2}}\,|00\rangle + \frac{1}{\sqrt{2}}\,|11\rangle \neq \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$. *End of proof*

The inability to clone a general qubit may seem like a significant disadvantage to computations on quantum computers. However, when it comes to encryption and safe data transfer, this property is used to an advantage.[55]

[55] See Bernhardt 2019, pp. 124 sqq.

Chapter 6

Quantum Algorithms

The quantum gates from the last chapter form the "building blocks" for creating quantum algorithms. At the end of the last chapter it was noted that a quantum algorithm almost always has a Hadamard gate at the beginning acting on all qubits in order to use the advantages of superposition and quantum parallelism. Afterwards, probability amplification, quantum Fourier transformation or quantum annealing are used to solve specific problems. This chapter deals with some of them.

6.1 P and NP Problems

In computer science a problem can have no way (at the current time), one way or even multiple ways to be solved using algorithms. Most problems have some form of a variable input size. For example, when the problem is multiplying two prime numbers, the input size is the number of digits the factors have.[56]

The opposite of the problem above is as follows: Given the product of two unknown prime numbers, the two numbers themselves have to be found, a problem known as prime factorization. In this case adding a single digit to the original prime numbers already has a notable impact on the time needed to find the original factors. Prime factorization is used among other algorithms in cryptography. The most notable example is internet encryption, which relies on the fact that it would take an attacker too long to factor the number in order to extract any information in a timely manner.

The problems mentioned above both have a variable input size. Multiplying two (prime) numbers is a problem in which the time needed to solve increases *linearly* when the input size is increased, i.e. bigger numbers with more digits are used. Linear increase is also called polynomial, so multiplying two numbers is part of the complexity class **P**, which is short for polynomial.[57] When $T(n)$ is a function that indicates the time needed for a complexity of n,

[56] See Bernhardt 2019, p. 142.
[57] See Bernhardt 2019, pp. 142 sq.

then polynomial time is defined as $T(n) \leq k \cdot n^p$ with k and p being positive numbers. If, however, $T(n) > k \cdot c^n$ with k a positive number and $c > 1$, the problem is called as being solvable in exponential time only.

Regarding the prime factorization problem, if the product as well as the prime numbers are known, it is easy to check if the prime numbers' product is equivalent to the product that was given. If a problem itself is not solvable in polynomial time (e.g. prime factorization), but checking the answer is, the problem is part of the **NP** complexity class, where NP stands for nondeterministic polynomial.[58]

When a problem itself is solvable in polynomial time, checking the answer is also always possible in polynomial time. In the worst case scenario, checking if the answer is correct is a case of repeating the actions taken to solve the problem. Mathematically, this means that $P \in NP$.[59]

All of this is important for quantum computing because there are problems that are NP for a classical computer, whereas for a quantum computer the same problem is in P.[60] In other words, there are problems that can only be solved by quantum computers in P but the results can be checked on classical computers in time.

6.2 Deutsch's Algorithm

The Deutsch algorithm was first proposed by David Deutsch in 1985, who is a professor at the University of Oxford.[61]

6.2.1 The Problem

The problem proposed by Deutsch will be explained using a coin. Real coins have a side with a number (the value of the coin) and a side with a symbol, whereas fake coins in this context have two identical sides. The task is to find out whether a coin is real or fake by flipping the coin as few times as possible.[62]

The Deutsch algorithm makes use of a so-called "oracle". The oracle is a function that takes an input and provides an output, in this case which side and what is printed on it. Checking both sides of the coin is synonymous to asking the oracle twice for an answer, once for each side. Every input has a predetermined output, but the test subject does not know how the function works. This is also called a "black box". This example assumes that the coin acts as the oracle. The oracle is asked "What is on the first side of the coin?" and an answer is

[58] See Bernhardt 2019, pp. 142 sq.
[59] In fact most people believe that P is not equal to NP, but this is unproven and one of the "Millennium Prize Problems".
[60] See Bernhardt 2019, p. 144.
[61] See Homeister 2018, p. 33.
[62] See Homeister 2018, p. 33.

provided. However, a human or a classical computer has to ask the oracle about the second side too to determine if the coin is fake or not.[63]

Generally, oracle-functions can be put into two categories: balanced and constant. If the function is balanced, then $f(0) \neq f(1)$ and the coin is considered real. If it is constant, then $f(0) = f(1)$ and the coin would have two identical sides and would be considered fake.[64]

For $f : \{0,1\} \to \{0,1\}$ there are four possible functions. Functions f_0 and f_3 are constant

x	$f_0(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$
0	0	0	1	1
1	0	1	0	1

and the other ones are balanced. As mentioned above a classical computer would have to evaluate f twice to find out if it is balanced or not whereas a quantum computer running the Deutsch algorithm only needs to do this once.

6.2.2 The Algorithm

The Deutsch algorithm tackles the above problem by exploiting quantum parallelism[65] and is depicted in Figure 6.1.

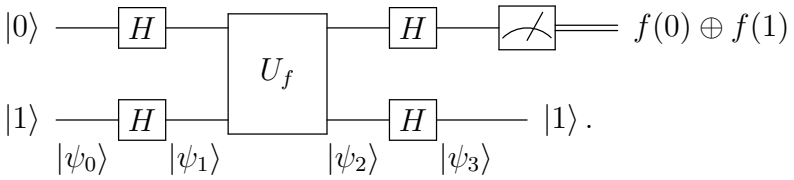


Figure 6.1: *Deutsch Algorithm*

In Figure 6.1, U_f represents the oracle which is applied to qubits in superposition.

6.2.3 Analyzing the Algorithm

In Figure 6.1 the intermediate states are labeled $|\psi_0\rangle$ to $|\psi_3\rangle$ and will be referenced in the following section. They represent the quantum systems' state at a certain time while executing the algorithm. U_f is one of the four possible functions f_0, f_1, f_2 and f_3. As the executor of the algorithm himself does not know which of the four functions is implemented in U_f, the objective is to find out if U_f is balanced or not. The function U_f has two inputs and two outputs but will treat them identically.[66]

[63] See Homeister 2018, p. 33.
[64] See Homeister 2018, p. 33.
[65] See section 5.4.
[66] See Portugal 2022, pp. 28 sq. for the analysis

The system is initially set to the state $|01\rangle$[67].

$$|\psi_0\rangle = |0\rangle \otimes |1\rangle \tag{6.1}$$

In the next step both qubits are put into superposition by the Hadamard transformation.[68]

$$|\psi_1\rangle = (H\,|0\rangle) \otimes (H\,|1\rangle) \tag{6.2}$$

$$= \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{6.3}$$

Now, $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ is simplified to $|+\rangle$ and $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ to $|-\rangle$. Only substituting $|-\rangle$ yields

$$|\psi_1\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |-\rangle \tag{6.4}$$

After applying the function U_f to the qubits, the system is in state $|\psi_2\rangle$.

$$|\psi_2\rangle = U_f\,|\psi_1\rangle \tag{6.5}$$

$$= \frac{U_f\,|0\rangle\,|-\rangle + U_f\,|1\rangle\,|-\rangle}{\sqrt{2}} \tag{6.6}$$

The oracle U_f is a unitary operator acting on two qubits as follows:

$$U_f\,|x\rangle\,|j\rangle = |x\rangle\,|j \oplus f(x)\rangle \tag{6.7}$$

where $\oplus$ is the Boolean XOR operator. One can then infer from this definition that

$$U_f(|x\rangle \otimes |-\rangle) \tag{6.8}$$

equals

$$(-1)^{f(x)}\,|x\rangle \otimes |-\rangle \tag{6.9}$$

Proof: Using the definition of $|-\rangle$, one obtains

$$U_f(|x\rangle \otimes |-\rangle) = \frac{U_f\,|x\rangle\,|0\rangle - U_f\,|x\rangle\,|1\rangle}{\sqrt{2}}$$

[67] These two qubits, which at this point still behave like classical bits, essentially represent the two possible inputs to the oracle as if they were classical.

[68] If the qubits were to be measured after this step, there would be no measurable difference between the first and the second qubit. However, while in superposition, the qubits are in two distinguishable states, a fact that will be exploited for the algorithm.

Using the definition of U_f as in Equation 6.7, one obtains

$$\begin{aligned} U_f(|x\rangle \otimes |-\rangle) &= \frac{|x\rangle\,|f(x)\rangle - |x\rangle\,|1 \oplus f(x)\rangle}{\sqrt{2}} \\ &= \begin{cases} \quad |x\rangle\,|-\rangle & \text{if } f(x) = 0 \\ - \quad |x\rangle\,|-\rangle & \text{if } f(x) = 1 \end{cases} \\ &= (-1)^{f(x)}\,|x\rangle \otimes |-\rangle \end{aligned}$$

End of proof

The definition of U_f from above can be used to simplify $|\psi_2\rangle$ to

$$|\psi_2\rangle = \frac{(-1)^{f(0)}\,|0\rangle + (-1)^{f(1)}\,|1\rangle}{\sqrt{2}} \otimes |-\rangle \tag{6.10}$$

$$= \begin{cases} \pm\,|+\rangle \otimes |-\rangle & \text{if } f(0) = f(1) \\ \pm\,|-\rangle \otimes |-\rangle & \text{if } f(0) \neq f(1) \end{cases} \tag{6.11}$$

The first ket can be prefixed with $\pm$, as it will make no difference as soon as the Hadamard gates in the next step are applied.

$$|\psi_3\rangle = (H \otimes H)\,|\psi_2\rangle \tag{6.12}$$

$$= \begin{cases} \pm\,|0\rangle \otimes |1\rangle & \text{if } f(0) = f(1) \\ \pm\,|1\rangle \otimes |1\rangle & \text{if } f(0) \neq f(1) \end{cases} \tag{6.13}$$

The fact that $H\,|+\rangle = |0\rangle$ and $H\,|-\rangle = |1\rangle$ has been exploited above. The second qubit will always be in the state $|1\rangle$ after completing the algorithm. However, after the measurement the first qubit will be 0 if $f(0) = f(1)$ and 1 if $f(0) \neq f(1)$.[69]

Therefore, a measurement of 0 on the first qubit means that the function U_f is constant and a measurement of 1 means that the function U_f is balanced.[70]

It is important to note that the Deutsch algorithm provides an answer after consulting the oracle, the function U_f, only once, while a classical computer or human would always have to execute the function twice to be sure. Thus, the quantum computer is more efficient.

6.2.4 Implementing the Deutsch Algorithm

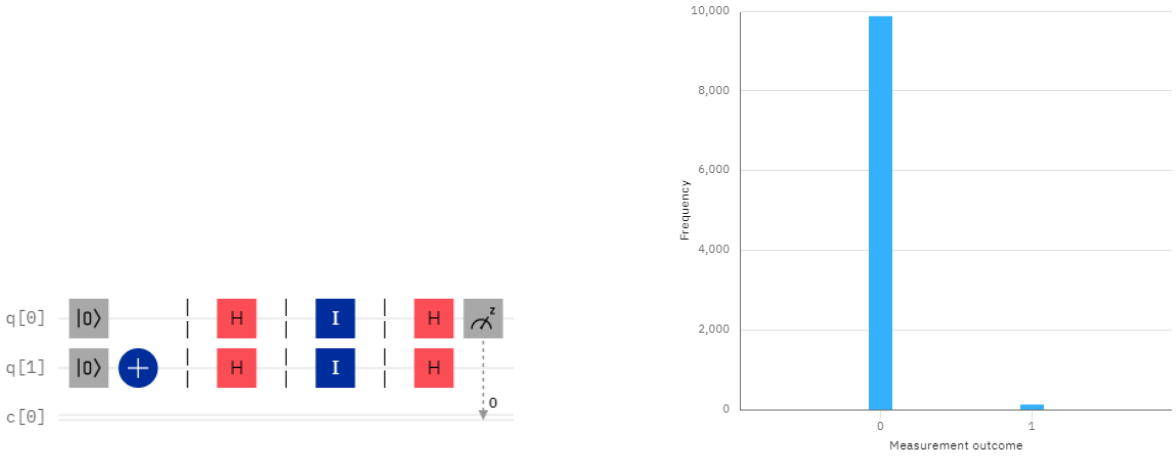
At the time of writing there are quantum computers commercially available and IBM even grants limited online access free of charge[71]. Using a graphical toolbox – the IBM Quantum Composer – individual quantum gates can be combined to form algorithms. After completing

[69] See Portugal 2022, p. 29.
[70] See Portugal 2022, p. 29.
[71] https://quantum-computing.ibm.com/.

the algorithm, it is executed on a quantum computer through the IBM Cloud and results can be downloaded. The IBM Quantum Composer was used to build the Deutsch algorithm for f_0 and f_1 and it ran on a real quantum computer. Figures 6.2 and 6.3 show the respective algorithms and results.

In both Figures `q[0]` and `q[1]` are the two qubits on which the algorithm is run and `c[0]` is a classical bit for storing the output. $|0\rangle$ initializes the respective qubits and $\oplus$ inverts it, effectively producing $|1\rangle$. The other gates are the ones explained in chapter 5: Hadamard, Identity and CNOT. The last symbol on the right represents the measurement on a qubit.

The f_0 oracle function can be implemented using $I \otimes I$ and the f_1 function using a CNOT, as can be seen in Figure 6.2a and Figure 6.3a. In Figure 6.2b the results after running the algorithm 10000 times[72] on a quantum computer. According to the Deutsch algorithm, with the f_0 function ("fake coin") implemented, this theoretically should always result in a measurement of 0.[73] But quantum computers are prone to measurement errors even more than classical computers leading to a measurement of 1 sometimes.

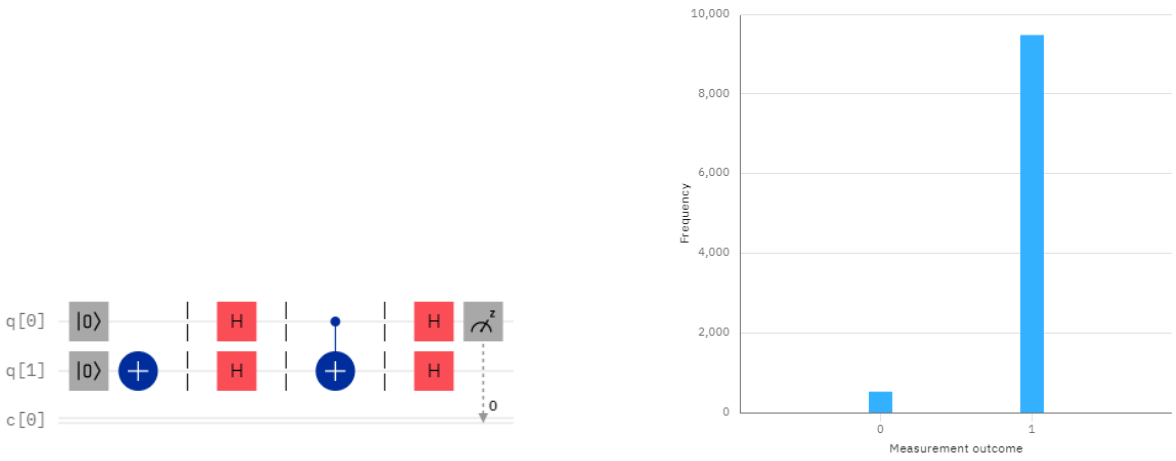


(a) *The algorithm* **(b)** *The result*

Figure 6.2: *The Deutsch f_0 algorithm*

The Deutsch algorithm with the f_1 function ("real coin") implemented in Figure 6.3 should always have a measurement of 1. Once again the measurement errors are visible.

[72] One run on the quantum computer only yields one output, 0 or 1. Therefore, the algorithm is executed multiple times in succession to get statistical significance.
[73] see subsection 6.2.1.



(a) *The algorithm* **(b)** *The result*

Figure 6.3: *The Deutsch f_1 algorithm*

6.3 Deutsch-Jozsa Algorithm

The Deutsch-Jozsa algorithm is the generalized form of the Deutsch algorithm. It was first published in 1992 by David Deutsch and Richard Jozsa.[74]

This algorithm also has the function $f : \{0,1\}^n \longrightarrow \{0,1\}$ at its core. Again this function is either balanced or constant and the aim is to find out the function type by asking the oracle as few times as possible. The algorithm in Figure 6.4 works similar to the Deutsch algorithm. It is the general form of the Deutsch algorithm and the first quantum algorithm that is exponentially faster than its equivalent classical deterministic counterpart.[75]

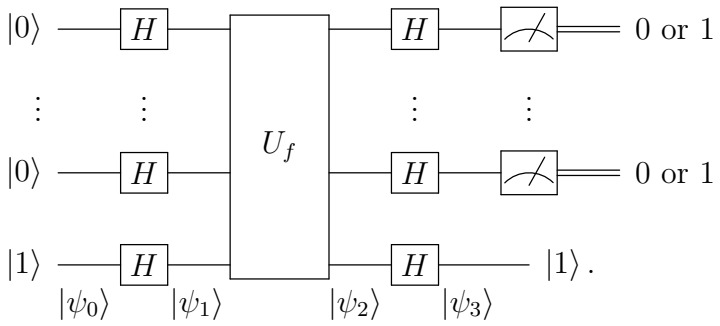


Figure 6.4: *Deutsch-Jozsa algorithm*

6.4 Shor's Algorithm

Shor's algorithm was first published in 1994 by Peter Shor. The algorithm can factor integers, meaning that it can find the two prime numbers whose product is the input number.

[74] See Homeister 2018, p. 63; Portugal 2022, p. 33.
[75] See Portugal 2022, p. 33.

This algorithm can achieve prime factorization in polynomial time while the best classical algorithm can only do it in sub-exponential time.[76] Prime factorization is very important in cryptography and especially in internet encryption because of its usage in RSA for encrypting internet traffic, which relies on prime factorization. Therefore, Shor's algorithm could pose a threat to the security of the internet.[77] The algorithm exploits entanglement as well as quantum parallelism.

6.5 Grover's Algorithm

Grover's algorithm was first published by Lov Grover in 1996. It is a search algorithm for unstructured data, denoting that it can find data inside a big collection of data very fast. The algorithm also revolves around an oracle, which can be evaluated multiple times and returns 1 only for the element that is searched for. It is important to note that this algorithm is already optimal for its purpose and no better algorithm can be found.[78]

6.6 GHZ State

Strictly speaking, this is not an algorithm but a series of operators which lead to a unique quantum state.

In 1935 Albert Einstein, Boris Podolsky and Nathan Rosen (usually shortened to EPR) published a paper[79] that tried to disprove the bold claims put up by quantum theory, especially the theory of entanglement. They did not believe that the measurement of an entangled photon could instantly determine the other photon's state. EPR thought that the information of a quantum object's state upon measurement was already determined when the photons were being entangled ("hidden variable"). In fact, EPR did not believe in what they called "spooky action at a distance". Back then there was no way to clarify whether EPR or Bohr's quantum theory's entanglement is correct.

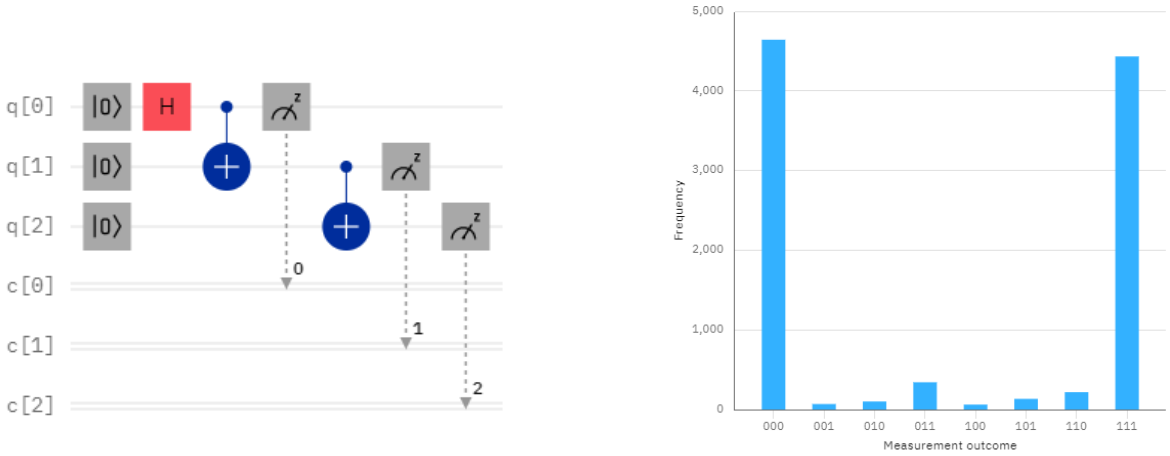
In 1964, John Stewart Bell published a theorem[80] that proposed a way to determine the right theory. A few years later scientists were able to prove that the theory of entanglement was indeed the correct one. Usually, the outcome of a measurement is random, with the probabilities determined by the angle between the particles when measured.[81]

However, Bell's theorem did not include an explanation for the case where the measurement probabilities are at an extreme, e.g. when the angle when creating the superposition is the same or the opposite as when measuring the qubit. Common sense says that this is

[76] See Portugal 2022, p. 56.
[77] See Buchanan and Woodward 2017, pp. 1 sq.
[78] See Portugal 2022, p. 81.
[79] See Einstein, Podolsky, and Rosen 1935.
[80] See Bell 1964.
[81] See Greenberger, Horne, and Zeilinger 1989.

equivalent to a 100% probability of measuring a specific outcome and therefore could be predicted with a classical computer. In 1989 Daniel Greenberger, Michael Horne and Anton Zeilinger managed to prove that in the context of quantum mechanics even this seemingly predictable state is unexplainable using classical physics.[82]

In this context they referred to a specific quantum state, the GHZ state, which represents the most extreme form of entanglement. As shown in Figure 6.5 measuring any of the entangled qubits leads to all of them being in the same state.



(a) *The algorithm* **(b)** *The result*

Figure 6.5: *Circuit to create the GHZ state with 3 qubits*

[82]See Greenberger, Horne, and Zeilinger 1989.

Chapter 7

Current status of Quantum Computers

At the current time a lot of big tech companies are investing money and resources into developing quantum computers and the surrounding technology. In the following chapter a short description of the achievements and goals of some of these pioneers will be shown.[83] As progress in this field is fast, the following content is subject to rapid change and does not claim completeness.

7.1 IBM

Of all the major companies that do research into quantum computers, IBM is probably the one with the longest history when it comes to developing computers. They launched IBM Quantum[84] on the IBM cloud in 2019 through which researchers and the public can experiment with quantum algorithms and run them on IBM's quantum computers[85].

Over the years IBM has developed several quantum processors based on Josephson junctions.[86] Their first quantum processor, IBM Canary, only had 5 qubits. At the time of writing IBM has developed a quantum processor with 433 qubits called Osprey.[87]

[83] See Dargan 2022a.
[84] https://quantum-computing.ibm.com/.
[85] See Dargan 2022a; Dargan 2022b.
[86] See section 3.2.
[87] See IBM Quantum 2022.

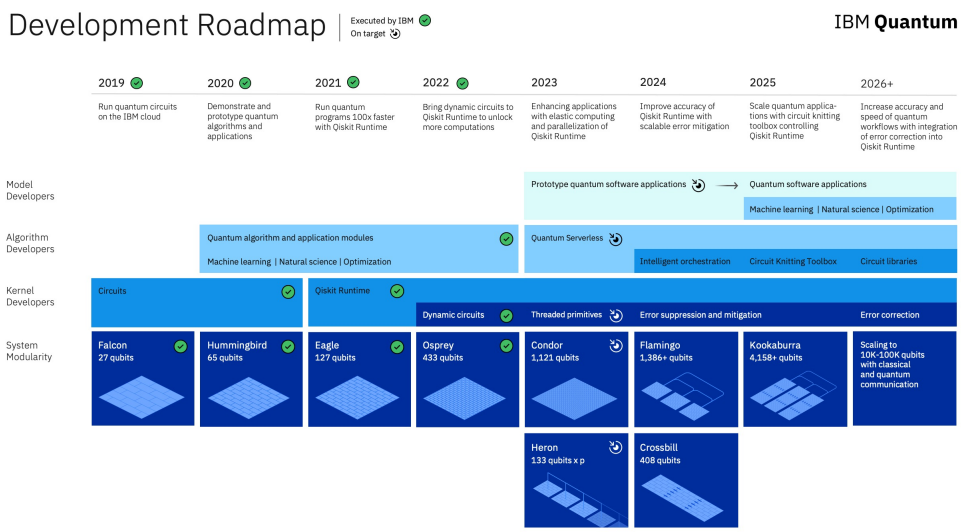


Figure 7.1: *IBM's Quantum Development Roadmap*

7.2 Intel

Intel, also a company that has a long history in developing computer hardware, has just released a quantum SDK[88] and a quantum simulator.[89]

James Clarke, Intel's director of quantum hardware assumes that usable quantum computers will be available in about a decade. He sees a correlation between the development of classical electronics and quantum computers. The first integrated circuits (IC) were built in 1958, and in line with Moore's law, it took about 43 years to get a million transistors onto an IC. As the first development into quantum computing started in the late 90s, Clarke comes to the conclusion that quantum computers will have a commercial market in 10 to 15 years.

Intel will have its first quantum processor in 2023, containing 12 qubits. James Clarke admits that 12 qubits is quite small, but argues that their hardware realization, spin qubits, will scale much better than others.[90]

7.3 Microsoft

Microsoft established the Azure Quantum platform to provide a quantum computing platform to researchers, similar to IBM's approach. Microsoft uses quantum computers with topological qubits.[91]

[88] **S**oftware **D**evelopment **K**it: a package of software written for a specific purpose.
[89] See Russell 2022.
[90] See Russell 2022.
[91] See Microsoft 2023.

7.4 Google

In 2019 Google announced that they reached quantum supremacy, the point at which a quantum computer performs better than a classical computer. Scientists at Google claimed that their quantum computer finished a task which would take a classical computer about 10000 years to complete.[92]

7.5 D-Wave

D-Wave Systems, a Canadian company, is focused on computing large datasets with the help of quantum computers.[93] In their computers they use quantum annealing to do optimization by letting the qubits "fall" into the "valley" with the lowest energy.[94] However, D-Wave also had to take backlash as they claimed to have an unreasonable amount of qubits in their quantum computer and at the time did not provide any proof that they had quantum technology running on their system.[95]

7.6 PsiQuantum

PsiQuantum is a company from California, USA that specializes in photonic quantum computers. They claim that photon qubits are the only way to achieve a reliable quantum computing system that is scalable. Their goal is to achieve a large-scale, general purpose quantum computer with 1 000 000 qubits using error correction.[96]

7.7 Alpine Quantum Technologies

Alpine Quantum Technologies (AQT) is a company from Innsbruck in Austria focusing research on trapped ion qubits. It is a spin-off of the University of Innsbruck, founded by quantum physicists Rainer Blatt, Thomas Monz and Peter Zoller. The company sells the worlds first general-purpose ion-trap quantum computer with 20 qubits and managed to perform a 14-qubit entanglement.[97]

[92] See Gibney 2019.
[93] See D-Wave Systems 2023a.
[94] See D-Wave Systems 2023b.
[95] See Bourne 2014.
[96] See PsiQuantum 2023.
[97] See Alpine Quantum Technologies GmbH 2023.

7.8 Current Challenges

Two of the most pressing challenges are controlling qubits and error correction. The challenge with controlling qubits is that it has to be very responsive and fast in order to execute even long algorithms before decoherence occurs.[98] Due to decoherence and the no-cloning property of qubits, quantum computers need sophisticated error correction. Unfortunately the associated effort grows significantly with the number of qubits.

[98] See Clarke 2019.

Chapter 8

Conclusion

In conclusion, this paper has provided an overview of the field of quantum computing, including the basic principles and the current state of the art. Discussed were the fundamental concepts of qubits, quantum gates, and quantum algorithms, as well as some challenges faced in building practical quantum computers.

In recent years, there has been significant progress in the development of quantum computers, and they are now being used in a variety of fields, including cryptography, finance, chemistry, and machine learning. This is a testament to the tremendous potential of quantum computers, which are capable of solving complex problems faster than classical computers.

However, there is still much work to be done before quantum computers can be widely adopted. Future research will focus on improving the scalability, stability, and error correction of quantum computers, as well as developing new algorithms and applications.

Despite these challenges, the future of quantum computing looks bright. With continued investment and development, quantum computers have the potential to revolutionize many fields and bring about unprecedented advances in science and technology. This paper shall serve as a starting point for exploring the exciting world of quantum computing and highlights the important role that it will play in shaping our future.

List of Figures

Bibliography

Alpine Quantum Technologies GmbH (Feb. 9, 2023). *Home - AQT — ALPINE QUANTUM TECHNOLOGIES*. URL: https://www.aqt.eu/ (visited on 02/09/2023).

APS News (Dec. 2012). "This Month in Physics History: December 18, 1926: Gilbert Lewis coins "photon" in letter to Nature". In: *APS News* 21.11. Ed. by Alan Chodos. URL: https://www.aps.org/publications/apsnews/201212/physicshistory.cfm.

Bell, J. S. (Nov. 1964). "On the Einstein Podolsky Rosen paradox". In: *Physics Physique Fizika* 1 (3), pp. 195–200. DOI: 10.1103/PhysicsPhysiqueFizika.1.195. URL: https://link.aps.org/doi/10.1103/PhysicsPhysiqueFizika.1.195.

Bernhardt, Chris (2019). *Quantum computing for everyone*. 4th ed. Cambridge: The MIT Press. ISBN: 9780262539531.

Blatt, R. et al. (Apr. 22, 2004). "Ion Trap Quantum Computing with Ca+ Ions". In: *Quantum Information Processing* 3.1-5, pp. 61–73. URL: https://www.quantumoptics.at/images/publications/papers/qip04_schmidtkaler.pdf (visited on 10/07/2022).

Bourne, Will (Jan. 9, 2014). "D-Wave's Dream Machine". In: *Inc.* URL: https://www.inc.com/will-bourne/d-waves-dream-machine.html (visited on 02/09/2023).

Buchanan, William and Alan Woodward (2017). "Will quantum computers be the end of public key encryption?" In: *Journal of Cyber Security Technology* 1.1, pp. 1–22. DOI: 10.1080/23742917.2016.1226650. eprint: https://doi.org/10.1080/23742917.2016.1226650. URL: https://doi.org/10.1080/23742917.2016.1226650.

Clarke, James S. (Mar. 22, 2019). "An Opimist's View of the 4 Challanges to Quantum Computing". In: *IEEE Spectrum*. URL: https://spectrum.ieee.org/an-optimists-view-of-the-4-challenges-to-quantum-computing (visited on 02/09/2023).

Cornwall, Remi (2015). "Disproof of the No-communication Theorem by Decoherence Theory". In: URL: https://vixra.org/pdf/1506.0068v1.pdf (visited on 02/09/2023).

D-Wave Systems (Feb. 8, 2023a). *D-Wave Systems — The Practical Quantum Computing Company*. URL: https://www.dwavesys.com/ (visited on 02/08/2023).

— (Feb. 9, 2023b). *What is Quantum Annealing?* URL: https://docs.dwavesys.com/docs/latest/c_gs_2.html (visited on 02/09/2023).

Dargan, James (Sept. 5, 2022a). "81 Quantum Computing Companies: The Ultimate List for 2023". In: *The Quantum Insider*. URL: https://thequantuminsider.com/2022/09/05/quantum-computing-companies-ultimate-list-for-2022/ (visited on 12/30/2022).

Dargan, James (June 30, 2022b). "A Century in The Making: IBM Quantum's Development Roadmap, Building The Future of a Nascent Technology". In: *The Quantum Insider*. URL: https://thequantuminsider.com/2022/06/30/a-century-in-the-making-ibm-quantums-development-roadmap-building-the-future-of-a-nascent-technology/ (visited on 12/31/2022).

Dong, Daoyi et al. (2015). "Robust manipulation of superconducting qubits in the presence of fluctuations". In: *Scientific Reports* 5.1, p. 7873. ISSN: 2045-2322. DOI: 10.1038/srep07873.

Einstein, A., B. Podolsky, and N. Rosen (May 1935). "Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?" In: *Phys. Rev.* 47.10 (10), pp. 777–780. DOI: 10.1103/PhysRev.47.777.

Gibney, Elizabeth (Oct. 24, 2019). "Hello quantum world! Google publishes landmark quantum supremacy claim". In: *Nature* 574, pp. 461–462.

Greenberger, Daniel M., Michael A. Horne, and Anton Zeilinger (1989). "Going Beyond Bell's Theorem". In: *Bell's Theorem, Quantum Theory and Conceptions of the Universe*. Ed. by Menas Kafatos. Dordrecht: Springer Netherlands, pp. 69–72. ISBN: 978-94-017-0849-4. DOI: 10.1007/978-94-017-0849-4_10. URL: https://doi.org/10.1007/978-94-017-0849-4_10.

Homeister, Matthias (2018). *Quantum Computing verstehen. Grundlagen - Anwendungen - Perspektiven*. 5th ed. Wiesbaden: Springer Vieweg. ISBN: 9783658228835.

IBM Quantum (2022). *2022 Development Roadmap*. URL: https://www.ibm.com/quantum/roadmap (visited on 12/31/2022).

Jones, Andrew Zimmermann (2020). "De Broglie Hypothesis". In: *ThoughtCo*. URL: https://www.thoughtco.com/de-broglie-hypothesis-2699351.

Marianne (Nov. 19, 2020). "Physics in a minute: The double slit experiment". In: *Plus Magazine*.

Maxwell, James Clerk (Dec. 1864). "II. A dynamical theory of the electromagnetic field". In: *Proceedings of the Royal Society of London* 13.155, pp. 459–512. DOI: 10.1098/rstl.1865.0008.

Medium (Feb. 17, 2021). "Quantum Hardware in a Nutshell". In: *Medium*. URL: https://medium.com/quantum-untangled/quantum-hardware-in-a-nutshell-50cc70c1ffd4 (visited on 02/06/2023).

Microsoft (Feb. 10, 2023). *Azure Quantum*. URL: https://azure.microsoft.com/en-us/products/quantum/ (visited on 02/10/2023).

Nakahara, Mikio and Tetsuo Ohmi (2008). *Quantum Computing. From Linear Algebra to Physical Realization*. CRC Press. ISBN: 978-0-7503-0983-7.

Petzold, Charles (Mar. 28, 2014). *Code. The Hidden Language of Computer Hardware and Software*. Ed. by Ben Ryan. Ed. by Kathleen Atkins. Ed. by Jim Fuchs. Microsoft Press. ISBN: 978-0-7356-1131-3.

Podlesnic, Verena (2022). “Ein robuster, kompakter Ionenfallen - Quantencomputer”. MA thesis. Fakultat fur Mathematik, Informatik und Physik der Leopold-Franzens Universitat in Innsbruck.

Portugal, Renato (2022). *Basic Quantum Algorithms*. DOI: 10.48550/ARXIV.2201.10574. URL: https://arxiv.org/abs/2201.10574.

PsiQuantum (Feb. 9, 2023). *PsiQuantum — Fault Tolerant Quantum Computing — Photonics*. URL: https://psiquantum.com/ (visited on 02/09/2023).

Russell, John (Dec. 13, 2022). “Intel Quantum Wisdom: Think Quantum is Powerful? You’re Right. Think it will Happen Soon. You’re Mistaken!” In: *HPCwire*. URL: https://www.hpcwire.com/2022/12/13/intel-quantum-wisdom-think-quantum-is-powerful-youre-right-think-it-will-happen-soon-youre-mistaken/ (visited on 12/28/2022).

Von Meyenn, Karl and Engelbert Schucking (Feb. 2001). “Wolfgang Pauli”. In: *Physics Today* 54.2, pp. 43–48. DOI: 10.1063/1.1359709.

Woody, Scott, Rhea George, and Matt Velasco (Oct. 1, 2005). “Qubit Implementation with Josephson Junctions”. In: URL: https://inst.eecs.berkeley.edu/~cs191/fa05/projectreports/Superconductors.doc (visited on 10/11/2022).

Zeilinger, Anton (2003). *Einsteins Schleier. Die neue Welt der Quantenphysik*. 5th ed. München: CH Beck. ISBN: 3406502814.