Introduction to Quantum Computing

Cornelius Schaetz

Contents

1 Quantum mechanical preparation

1.1 Physical systems and the uncertainty Principle

1.1.1 The classical state of a physical system

In this section we will learn what a *physical system* is and how the *classical state* of such a system is defined. Then we will look at the difference between *macroscopic* and *microscopic* physical systems , which will lead us directly into the world of quantum physics.

Before we can dive into the quantum mechanical part of this course, we have to take a look at the concepts of physical systems and the states they can be found in. The easiest way to start is by imagining a ball flying through the air, as you can see in Figure 1.1.

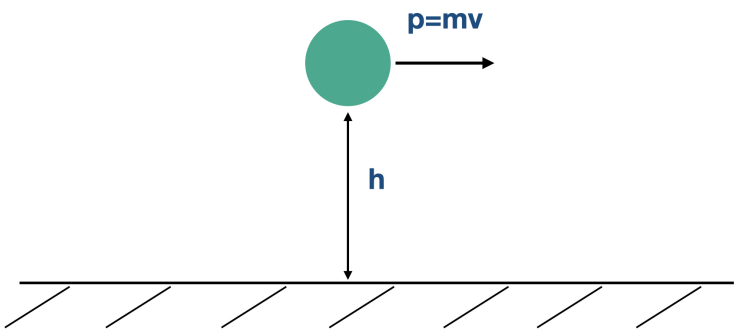


Figure 1.1: Example for a classical physical system. A ball flies through the air in a height h over the earth's surface with a momentum p in x-direction.

We are seeing the ball right now at this moment, so we observe the system at time $t = 0$. The ball has a height h over the ground and is moving with momentum p in x-direction. As a short reminder: the momentum of a body is defined via its mass and its velocity

$$\mathbf{p} = m\mathbf{v} \tag{1.1}$$

where the bold letters shall indicate that we are talking about 3-dimensional vectors. Every moving body has a momentum in x-,y- and z- direction, therefore the three dimensions. In the case of the ball flying through Figure 1.1, the momentum in y- and z-direction would be zero. Everything that we can see in Figure 1.1 can be understood as a *physical system*. A physical system is just a small part of the universe that we chose to observe and analyse while we are pretty much ignoring the environment.
When it comes to a physical system, we are not only interested in analysing it, but also in predicting its future evolution. The information that we need to do so is contained in the so-called *classical state* of the system. In our case, the information, that we need, would be the height of the ball above the earth's surface and its momentum in x-direction.
If we know h and p at time $t = 0$, we are able to determine the evolution of the physical system for $t > 0$. So the state of the system is represented by the pair of numbers (p, h). Mathematically, the classical state of a physical system would be a point in so-called *phase space*. In case of the example above, the state is represented by the pair of numbers (p, h), which would be a point in a 2-dimensional plane with the x-axis denoting the momentum in x-direction and the y-axis denoting the height above the ground - all at time $t = 0$. At another time (earlier or later) the point would be somewhere else in this plane.

Physical systems and classical states

A *physical system* is a small part of the universe, that we chose to observe and analyse, just like a ball flying over the ground. The *classical state* of a physical system is determined by its momentum and its spatial position, which can be represented as a dot in *phase space*. This dot is enough, to determine the entire future of the physical system.

For a better visualization of the phase space concept, we can assume that the ball has a mass $m = 1kg$ and is currently - at time $t = 0$ - $2m$ above the ground. Its velocity in x-direction shall be $v = 3m/s$. This would result in a momentum of:

$$p = mv = 1kg \cdot 3\frac{m}{s} = 3\frac{kgm}{s} \tag{1.2}$$

Now that we know the exact values for the height and for the momentum, we can represent the state of the system by the point $(3, 2)$ in 2-dimensional phase space (see Figure 1.2

Normally we would need three coordinates to fully describe the momentum vector and we would also need three coordinates to describe the spatial position. This is the reason, why the phase space of a physical system is in general a 6-dimensional space and a classical state would be a point in this space with the coordinates (x, y, z, p_x, p_y, p_z).
The concept of classical states as being a point in 6-dimensional phase space works pretty well to describe and determine the future of many different types of physical systems: Flying balls, moving cars, starting rocket,...
Physical systems like theese are called *macroscopic systems*: they are big enough to be

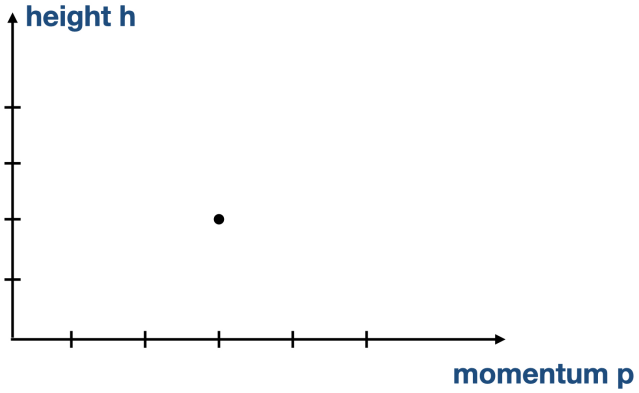


Figure 1.2: This is how the 2-dimensional phase space of the physical system from Figure 1 would look like. It's just a 2-dimensional plane with the x-axis denoting the momentum in x-direction and the y-axis denoting the height over the ground. The state of such a system is a point in this plane. In the case of the ball flying through the air, the state at time t=0 would be the point (3,2).

seen with our bare eyes - simply spoken.
The opposite of a macroscopic system is a *microscopic system*, which is to small to be seen with our eyes.
Molecules, atoms or subatomic particles would fall into the category of microscopic systems. And for microscopic systems, the concept of classical states does not work anymore.
As physicist Werner Heisenberg discovered in the 1920's, it is impossible to determine the exact classical state of a microscopic system. This is widely known as the uncertainty principle, which we now will take a closer look on.

1.1.2 The uncertainty principle

In this section we will learn what the *uncertainty principle* is and how it is related to quantum physics. The uncertainty principle will show us that the concept of classical states does not work when we want to describe the evolution of a quantum system.

So what is the uncertainty principle exactly?
The uncertainty principle states that it is impossible to measure the momentum and the spatial position of a microscopic particle at the same time *and* get a precise value for both of them.
The momentum and the position of a microscopic particle seem to be somehow connected to each other. If we for example measure the momentum of the particle and get a precise

value, it gets impossible to know exactly *where* the particle actually is at the same time. And this has nothing to do with our measuring devices being crap.
No, it is a fundamental law of nature.
The more precise you measure one value (it does not matter if the momentum or the position), the less exact your measurement outcome for the other one will be.
We could also express the uncertainty principle in terms of - well - *uncertainties*.
We shall call the uncertainty of the spatial position Δx and the uncertainty for the momentum Δp.
If we measure a box standing $x = 2m$ faway rom us and the uncertainty is $\Delta x = 1m$, then the box could have any distance between $1m$ and $3m$ away from us.
So the uncertainty principle states that a low uncertainty for one of the two values will result in a high uncertainty for the other value. If we for example measure the momentum of a particle exactly, that would make the uncertainty pretty small. But this small uncertainty will have an impact on the measurement outcome for the position and its uncertainty will grow to almost infinity.
Mathematically, the uncertainty principle is written in the following form:

$$\Delta x \Delta p \geq \frac{\hbar}{2} \tag{1.3}$$

In this equation, $\hbar = \frac{h}{2\pi}$ stands for the Planck constant (which has an extremely small value).
To better understand the meaning of the equation, we rewrite it:

$$\Delta x \geq \frac{\hbar}{2\Delta p} \tag{1.4}$$

Let's assume, we have an electron and we are able to precisely measure the momentum of it. This would make the uncertainty for the momentum pretty small, so the fraction $\frac{\hbar}{2\Delta p} = \Delta x$ gets extremely big. This means that the uncertainty of the location - where to find the electron - grows extremely big. We may just have found out the exact momentum of the electron, but we do not know, where it is located. It could be anywhere.
Since it is impossible to measure the momentum and the position equally precise at the same time, the concept of classical states no longer make sense. It is not possible anymore to represent the state of a microscopic system by a single point in phase space. Physical states, in which the uncertainty principle rules, are called *quantum systems*.

The uncertainty principle

The *uncertainty principle* (discovered by Werner Heisenberg) states that it is impossible to measure the momentum and the position of a microscopic particle simultaneously *and* get exact values for both. The more precise you measure the momentum, the less precise your position measurement will be and vice versa.
Microscopic physical systems that are small enough to fall into the reign of the uncertainty principle (just like molecules, atoms and subatomic particles) are also called *quantum systems*.
The uncertainty principle makes it impossible to get precise values for the momentum and the position of a quantum particle, so the concept of classical states no longer works to describe the evolution of a quantum system.

Since nature does not want us to determine the classical state of a quantum system, we need another way to get the information we need in order to predict the future evolution of the system. We need a new concept to describe the state of a quantum system, we need a *quantum state*. And the best way to understand the concept of quantum states is by taking a look at the famous double slit experiment.

1.2 The double slit experiment

1.2.1 Lightwaves

In this section we will learn about the *double slit experiment* with light and electrons. We will see that not only light behaves like a wave, but electrons as well. This will eventually lead us to the definition of a *quantum state* and one of the most fundamental laws of quantum physics: the *superposition principle*.

As we remember from our time back in school, light is a wave and its wavelength determines the color of the light. Shorter wavelengths correspond to blue and more energetic light, longer wavelengths correspond to red and less energetic light (see Figure 1.3).

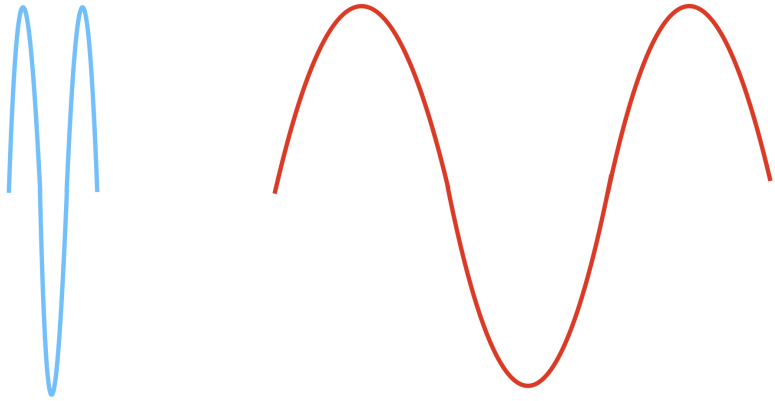


Figure 1.3: Illustration of waves with different wavelengths. The shorter the wavelength, the more energetic the light is. This makes red light (long wavelength) less energetic than blue light (short wavelength).

Now imagine we have a light source, which we assume to emit so-called *monochromatic* light. Monochromatic means that all the waves leaving the lightsource have exactly the same wavelength. The sunlight for example is **not** monochromatic - it contains all the wavelengths of the electromagnetic spectrum. A laser would be such a monochromatic light source.
We put this light source in front of a wall with two small slits in it and behind the wall we hang up a screen to observe the path of the light. The whole experimental setup is visualized in Figure 1.4.
We activate the light source and the emitted monochromatic light propagates through space and towards the double slit wall. The lightwaves pass through the slits, interfer with each other on the other side and produce a pattern of dark and bright bands on the screen behind the wall. This pattern is called *interference pattern*.

The only reason why it is possible to see this interference pattern on the screen is because light propagates as a wave through space and therefore can pass through both slits at the same time.

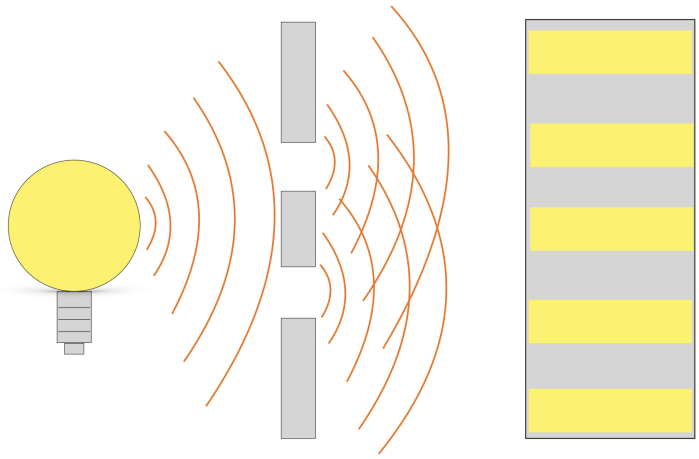


Figure 1.4: Monochromatic light leaves the light source, propagates through space towards the double slit and passes through both slits at the same time to interfer with itself on the other side and produce an interference pattern on the screen behind the double slit wall.

The double slit experiment

Light is wave and when you let it propagate through space towards a wall with two thin slits in it, the light will pass through both slits at the same time and will *interfer* with itself on the other side. We are able to see a sequence of bright and dark bands on the screen, which is called an *interference pattern*. The double slit experiment serves as a proof for the wave-like nature of light.

What would happen, if we used electrons instead of light waves? How would they behave, when we shoot them towards the double-slit wall?
Classically, we would imagine them as tiny balls flying around. But the experiment shows something else.

1.2.2 Double slit with electrons

From what we learned back in school, we can imagine microscopic particles like atoms or electrons as tiny little balls flying around. This is also how Niels Bohr imagined an atom to look like in the beginning of the quantum era. The electrons were just like planets orbiting the nucleus, which contains almost all of the mass of the atom.
But when we do the double slit experiment with electrons or other subatomic particles, they do not behave like classical little balls.

To do the double slit experiment we need to imagine having an electron gun, that is able to produce and shoot out a bunch of electrons. We point the gun in the direction of the double slit wall and start to shoot the electrons. The electrons pass through the slits and hit the screen (which we assume to light up everytime it gets hit by an electron). The whole experimental setup is demonstrated in Figure 1.5.

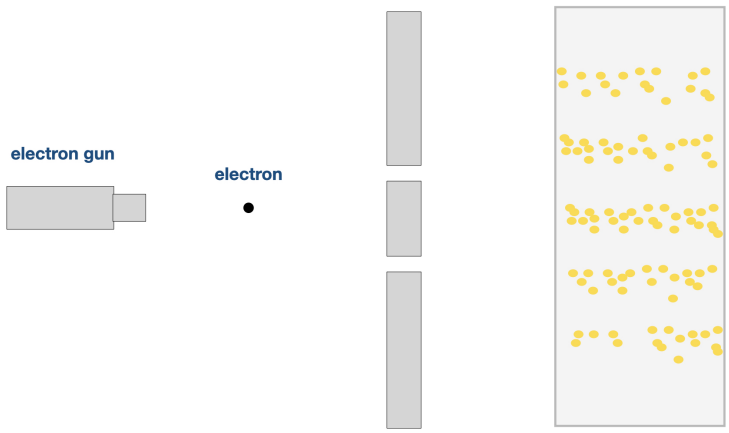


Figure 1.5: Double slit experiment, but this time with electrons. An electron gun shoots electrons towards a double slit wall. The electrons pass through the slits and produce an interference pattern on the screen on the other side. Although we would classically imagine electrons as tiny little balls flying around, they seem to behave like waves and pass through both slits at the same time.

After some time shooting electrons through the slits we will observe an interference pattern on the screen behind the double slit wall. Seeing an interference pattern means that the electrons must have behaved like waves. How is that possible? Aren't they supposed to be actual *particles*, tiny little balls flying around, not a wave vibing through space?
It just seems like microscopic particles do not behave anymore like we would classically expect them to. Electrons can behave like waves, the can interfer with themselves and pass through both slits at the same time.
But maybe this happens because the electrons "communicate" with each other? Maybe the interference pattern is just because a couple of thousand electrons produce that after passing as a group through a double slit wall? Maybe a group of electrons behaves like wave but a single electron is still behaves like a tiny little ball?
The crazy part is that we can repeat the experiment and shoot just single electrons out of the gun.
Let's assume, that the gun shoots out an electron, it passes through the slits and when it hits the screen, the gun gets a signal to shoot another one.
I have visualized that experimental setup in Figure 1.6.
At first, it may seem like a coincidence, where the electrons hit the screen. But after some time shooting particles through the double slit wall, we will start to observe an

interference pattern again.
So it seems that indeed a single electron propagates through space like a wave. We could do this experiment with any microscopic *quatum particle*. Physicists have done it with atoms and even with molecules.

Figure 1.6: Double slit experiment with electrons again, but this time the electron shoots only one electron at a time. As soon as the electron hits the screen behind the wall, the gun shoots a new one through the slits. In the beginning it seems like a coincidence, where on the screen the electrons would land. But after some time one would recognize an interference pattern.

Double slit with electrons

When you shoot electrons towards a double slit wall, they will pass through both of the slits at same time, showing their wave-like nature. They produce an interference pattern on the screen behind the wall, which happens to be there, even if you send the electrons one by one through the slits.
This experiment has been repeated with atoms and molecules and serves as a proof, that quantum particles like electrons and atoms behave like waves sometimes.

So quantum particles like electrons or atoms don't act as classical particles, they behave like waves. When we shoot quantum particles through a double slit wall, it seems like they are bassing through both slits at the same time.
One can say, the electron is two classical states at the same time. The whole experimental setup of the double slit experiment can be understood as a quantum system. Since the electrons are in two classical states at the same time, the quantum system is in two classical states.
The quantum state is a so-called *superposition* of classical states. And this is how we can finally define the state of a quantum system.
In general, a quantum state is defined as a superposition of a certain amount of classical states.
This is known as the *superposition principle* and it is one of the most fundamental prin-

ciples of quantum physics.

Quantum states and the superposition principle

We learned from the double slit experiment that the *quantum state* of a quantum system can be understood as an overlap of two classical states at the same time. This is known as the *superposition principle*. The quantum system is in a superposition of classical states.

Mathematically, a quantum state would be defined as a vector in a Hilbert space (which is a special type of vector space). The Hilbert space has a basis of vectors, which correspond to classical states and any state vector can be written as a linear combination (superposition) of the basis vectors.

The mathematical definition of a quantum state sounds pretty complex and in order to understand it, we will have to put in a little maths session right now. This will help us later to understand important concepts like qubits, quantum gates or entanglement. So let's get into the mathematics part, to understand those two sentences.

1.3 Vector spaces and quantum states

In this section we are going to learn how a *vector space* is defined and what we can imagine when we talk about the *basis* of a vector space. We will apply this knowledge to understand what a *Hilbert space* is and finally mathematically define a *quantum state* using the *superposition principle*.

1.3.1 Definition of a vector space

Let's jump right into the definition of a vector space. A vector space over a field K is a set of elements, equipped two operations - vector addition and scalar multiplication - that needs to fullfill a certain number of axioms.
Okay, we need to break that a little down.
First, what is a field?
A field is a set of elements, on which addition, substraction, multiplication and division are defined. This sounds pretty odd, but in mathematics, everything is defined rigourously. You cannot just take a set of numbers and add its elements. You first have to define the operation of addition of this specific set of numbers, before you can actually add the numbers up.
An example for a field would be the set of real numbers $\mathbb{R}$, which includes pretty much all the numbers we know from our daily life: 1,2,3,-1,-245,π, e, $\sqrt{2}$,... Let's return to the definition of the vector space.
A vector space is a set of elements (which are called vectors), equipped with two operations:

- vector addition $+ : V \times V \to V$: adding two vectors $\mathbf{v} + \mathbf{w}$ with $\mathbf{v}, \mathbf{w} \in V$ results in a third vector, that is also an element of the vector space V $\mathbf{v} + \mathbf{w} \in V$
- scalar multiplication $\cdot : V \times V \to V$: an element from the field K is called a scalar. Let $a \in K$ be a scalar and $\mathbf{v} \in V$ a vector in the vector space V. Then $a \cdot \mathbf{v} \in V$, the scalar multiplied with the vector still results in a vector of the vectorspace V.

To better understand those abstract formulations, we will take a look at the 2-dimensional vector space $\mathbb{R}^2$ over the field of real numbers $\mathbb{R}$.
A vector in $V = \mathbb{R}^2$ can be written in the form:

$$\mathbf{v} \in V \quad \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \tag{1.5}$$

Here, $v_1 \in \mathbb{R}$ and $v_2 \in \mathbb{R}$ are the components of the vector. Every vector in $\mathbb{R}^2$ can be written in this form, a vector is an element of the vector space $\mathbb{R}^2$, as long as the components are elements of the field $\mathbb{R}$.
A vector in the vector space $\mathbb{R}^2$ can be graphically imagined like in Figure 7, written in components it would be $\mathbf{v} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$.

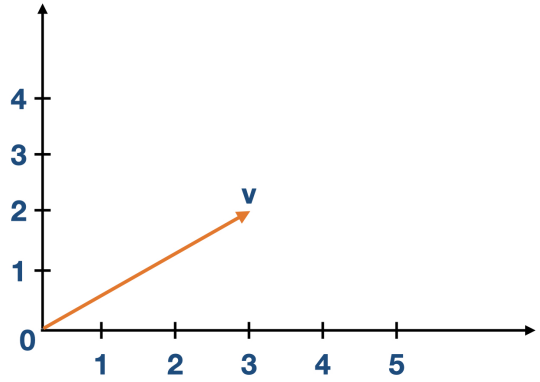


Figure 1.7: Graphical Visualization of the 2-dimensional vector $\mathbf{v} = (3, 2)$.

If we take two vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^2$

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \quad \mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \quad v_i, w_i \in \mathbb{R}, i = 1, 2 \tag{1.6}$$

and add them together

$$\mathbf{v} + \mathbf{w} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} v_1 + w_1 \\ v_2 + w_2 \end{pmatrix} \in \mathbb{R}^2 \tag{1.7}$$

then the result is still a vector in $\mathbb{R}^2$, since the components $v_1 + w_1 \in \mathbb{R}$ and $v_2 + w_2 \in \mathbb{R}$ are still elements of the field $\mathbb{R}$. Let's now take a look at the scalar multiplication on the vector space $\mathbb{R}^2$. Let $a \in \mathbb{R}$ be an element of the field $\mathbb{R}$ (a is a scalar), and $\mathbf{v} \in \mathbb{R}^2$ a vector. Performing a scalar multiplication would mean, to multiply the scalar with each component of the vector:

$$a \cdot \mathbf{v} = a \cdot \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} av_1 \\ av_2 \end{pmatrix} \in \mathbb{R}^2 \quad \textit{since} \quad a \cdot v_i \in \mathbb{R}, i = 1, 2 \tag{1.8}$$

In the beginning I said, that the set of vectors has to fulfill a certain set of axioms. We will work through those now.

- Associativity of addition: $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$
  in the example of $\mathbb{R}^2$ this would mean

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \left(\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \right) = \begin{pmatrix} u_1 + (v_1 + w_1) \\ u_2 + (v_2 + w_2) \end{pmatrix} \tag{1.9}$$

  and since the components $u_i, v_i, w_i \in \mathbb{R}, i = 1, 2$ are real numbers, the law of associativity holds for vectors on $\mathbb{R}^2$

- Commutativity of addition: $\mathbf{v} + \mathbf{w} = \mathbf{w} + \mathbf{v}$
  the same argument as for the associativity works again.

- Identity addition element: there exists a vector $\mathbf{0} \in V$ such that for any vector $\mathbf{v} \in V$ the equation holds: $\mathbf{0} + \mathbf{v} = \mathbf{v}$.
  In the example this $\mathbf{0}$ - vector would be the following one:

$$\mathbf{0} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \mathbb{R}^2 \tag{1.10}$$

$$\mathbf{0} + \mathbf{v} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} v_1 + 0 \\ v_2 + 0 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \mathbf{v} \tag{1.11}$$

- Additive inverse: For every vector $\mathbf{v} \in V$ there exists an additive inverse element $-\mathbf{v}$ such that $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$. In $\mathbb{R}^2$ the additive inverse vector would be the one with the additive inverse components.

- $a(b\mathbf{v}) = (ab)\mathbf{v}$

- $1 \cdot \mathbf{v}$ with $1 \in K$

- Distributivity with respect to vector addition: Let $a \in K$ be a scalar and $\mathbf{v}, \mathbf{w} \in V$ vectors. Then the following equation holds:

$$a \cdot (\mathbf{v} + \mathbf{w}) = a \cdot \mathbf{v} + a \cdot \mathbf{w} \tag{1.12}$$

  In the case of the 2-dimensional vector space $\mathbb{R}^2$, this would mean for a scalar $a \in \mathbb{R}$:

$$a \cdot (\mathbf{v} + \mathbf{w}) = a \cdot \left(\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \right) = a \cdot \begin{pmatrix} v_1 + w_1 \\ v_2 + w_2 \end{pmatrix} = \begin{pmatrix} a(v_1 + w_2) \\ a(v_2 + w_2) \end{pmatrix} \tag{1.13}$$

  and since distributivity holds on the field of real numbers $\mathbb{R}$, it also holds for vectors of $\mathbb{R}^2$.

- Distributivity with respect to scalar addition: Let $a, b \in K$ be two scalars and $\mathbf{v} \in V$ a vector. Then the following equation holds:

$$(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v} \tag{1.14}$$

  The analogy in $\mathbb{R}^2$ is similar to the distributivity with respect to vector addition.

Now that we know what a vector space is and how we can imagine one, it is time for another important concept that we will need to understand, what a quantum state. I am talking about the basis of a vector space.

1.3.2 The basis of a vector space

Let's start right with the definition of a basis.
The Basis B of a vector space V is a subset of V with the following properties:

- it is linear independent
- it spans the whole vector space

Okay, we need to take a look at those abstract formulations again. Let's start with the linear independence.
To understand, what linear independent vectors are, we first have to define, what a linear combination of vectors is.
Let V be a vector space over the field K, let $\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_n} \in V$ be a list of vectors and $a_1, a_2, ..., a_n$ a list of scalars. A linear combination is defined as the sum:

$$\sum_{i=1}^{n} a_i \mathbf{v_i} = a_1 \mathbf{v_1} + a_2 \mathbf{v_2} + ... + a_n \mathbf{v_n} \tag{1.15}$$

A vector $\mathbf{v} \in V$ is called linear dependet from some other vectors $\mathbf{v_i} \in V,\ \ i = 1, 2, ..., n$ if it can be expressed as a linear combination of those vectors:

$$\mathbf{v} = a_1 \mathbf{v_1} + a_2 \mathbf{v_2} + ... + a_n \mathbf{v_n} \tag{1.16}$$

To make this a little more clear, we return to the well known example of the $\mathbb{R}^2$ vector space. Let's take for example the following three vectors:

$$\mathbf{v} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \quad \mathbf{v_1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \mathbf{v_2} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1.17}$$

The vector $\mathbf{v}$ can be written as linear combination of the two vectors $\mathbf{v_1}, \mathbf{v_2}$:

$$\mathbf{v} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + 3 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 2\mathbf{v_1} + 3\mathbf{v_2} \tag{1.18}$$

This means that the vector $\mathbf{v}$ is linear dependent on the other two vectors. A set of vectors $\{v_1, v_2, ..., v_n\}$ is said to be linear independent, if no vector can be written as a linear combination of the other vectors in this set. Or, written as an equation:

$$a_1 \mathbf{v_1} + a_2 \mathbf{v_2} + ... + a_n \mathbf{v_n} = 0 \ \ only \ \ if \ \ a_i = 0 \ \ for \ all \ i = 1, 2, ...n \tag{1.19}$$

A linear independent set of vectors on the vector space $\mathbb{R}^2$ would be the two vectors

$$\mathbf{v_1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \mathbf{v_2} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1.20}$$

$$a_1 \mathbf{v_1} + a_2 \mathbf{v_2} = a_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + a_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1.21}$$

This results in the two equations:

$$a_1 \cdot 1 + a_1 \cdot 0 = a_1 \cdot 1 = 0 \tag{1.22}$$

$$a_2 \cdot 0 + a_2 \cdot 1 = a_2 \cdot 1 \ = 0 \tag{1.23}$$

Thus the two scalars can only fulfill equation 1.19 if they are zero. This is why the two vectors are linear independent.
Let's get to the second part of the definition of a basis in a vector state. It says, that a set of basis vectors needs to span the whole vector space.
This means, that every vector $\mathbf{v} \in V$ can be written as a linear combination of the basis vectors.
Let's say we have a set of basis vectors $\{b_1, b_2, ..., b_n\}$. For every $\mathbf{v} \in V$ we can find a set of scalars $\{\lambda_1, \lambda_2, ..., \lambda_n\}$ such that the following equation holds:

$$\mathbf{v} = \sum_{i=1}^{n} \lambda_i b_i \tag{1.24}$$

Returning to the well known $\mathbb{R}^2$ and the two vectors

$$\mathbf{v_1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \mathbf{v_2} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1.25}$$

we can see that every vector in $\mathbb{R}^2$ can be written as a linear combination of those two. So the set of vectors $\mathbf{v_1}, \mathbf{2}$ is not only linear independent, but also spans the whole $\mathbb{R}^2$ space. These two vectors are a basis of $\mathbb{R}^2$.
The number of basis vectors in a vector space determines the dimension of the space.
If we take the vector space $\mathbb{R}^2$, its basis would be the two vectors in 1.20.

To define a quantum state mathematically, we shortly have to look at a special type of vector spaces: the so-called Hilbert spaces.

1.3.3 Hilbert spaces

When von Neumann and Dirac formulated a mathematical description of quantum mechanics, they chose Hilbert spaces as the type of vector space needed for quantum physical computations.
In short, a Hilbert space is a vector space that is complete and equipped with an inner product.
Alright, so let's workt through those two new terms:

- complete
- inner product

Having a set of elements, that is complete simply means that it is possible to do calculus with the elements of this set. So let's say we have a set of functions $V = \{f : f : \mathbb{R} \to \mathbb{R}\}$ which map a real number $x \in \mathbb{R}$ on another real number $f(x) = y \in \mathbb{R}$. We can make the set of real valued functions a vector space by defining vector addition and scalar multiplication on V:

- vector Addition: $+ : V \times V \to V$
  Let $f, g \in V$ be two real valued functions. The addition would be defined by adding the real values $f(x) + g(x)$

- Scalar Multiplication: $\cdot : V \times V \to V$
  A scalar would be any real number $\lambda \in \mathbb{R}$, so the scalar multiplication would be just multiplying one real number by another $\lambda f(x)$

So a set of functions can be a vector space, and if you work with spaces like that, it's good to be able to differentiate or integrate those functions. This is why completeness is required.
The second condition for a Hilbert space was the inner product. A simple example for a Hilbert space is the 3-dimensional euclidean space $\mathbb{R}^3$. Vectors in $\mathbb{R}^3$ are written like those in $\mathbb{R}^2$ but with an extra component for the extra dimension.
An inner product of two vectors would be defined the following way:

$$\mathbf{x}, \mathbf{y} \in V \quad \mathbf{x} \cdot \mathbf{y} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = x_1 y_1 + x_2 y_2 + x_3 y_3 = \sum_{i=1}^{3} x_i y_i \tag{1.26}$$

This is also known as the scalar product, since multiplying those two vectors results in an element of the field $\mathbb{R}$ which is a scalar.

Now that we know what a Hilbert space is, we can return to the mathematical definition of a quantum state.

1.3.4 Quantum states and the superposition principle

A quantum state is a vector in a Hilbert space, which can be written as the linear combination of a set of basis vectors. The basis vectors correspond to the classical states of the physical system.
In Quantum mechanics, one uses the so-called Dirac-notation for the state vectors.

$$|\phi\rangle = a_0 |0\rangle + a_1 |1\rangle + ... + a_{N-1} |N-1\rangle \tag{1.27}$$

In this equation, $|\phi\rangle$ is the vector in a N-dimensional (since there are N basis vectors) Hilbert space H that represents the quantum state of the system. The classical states of the system are represented by the set of basis vectors $|0\rangle, |1\rangle, ..., |N-1\rangle$ and the coefficients $a_0, a_1, ...a_{N-1}$ are called amplitudes.
The fact that the state vector is a linear combination of the classical states of the system is the mathematical way to formulate the superposition principle: a quantum system can be in more than one classical state at the same time.
Usually one can write the state vector $|\phi\rangle$ as a column vector with the amplitudes being

the components.

$$|\phi\rangle = \begin{pmatrix} a_0 \\ a_1 \\ . \\ . \\ . \\ a_{N-1} \end{pmatrix} \tag{1.28}$$

In the next section we will find out which physical meaning the amplitudes of the state vector have.

1.4 What to do with a quantum state

This section will be all about the things we can do with a quantum state. We will learn what happens, when we *measure* a quantum state and how to interpret the *amplitudes* of a quantum state. In the end we will see that it is possible to act on a quantum state without actually destroying it: by applying *unitary operations*.

1.4.1 Measurement

In the section about the double slit experiment we learnt that electrons - or quantum particles in general - do not behave like we classically imagined them to behave. We cannot see them as tiny little balls flying around but have to see them as waves propagating through space and being able to interfer. Quantum particles seem to move through both slits at the same time, they are in more than one classical state at the same time.
Let's denote those classical states $|0\rangle$ for the upper slit and $|1\rangle$ for the lower slit.
The state of the electron can then be written in the form

$$|\phi\rangle_{electron} = a_0 |0\rangle + a_1 |1\rangle \tag{1.29}$$

with amplitudes a_0, a_1.
What happens when we want to observe the electron as it passes through both slits at the same time?
Let's assume we have a detector that we position right behind the double slit wall. The detector is able to measure through which one of the slits the electron flies (see Figure 1.8).

But the electrons don't want to be observed. As soon as the detector is activated the electrons stop their wave-like behavior and only pass through one single slit. They act like classical particles again, like tiny balls flying around.
This is another fundamental principle in quantum physics.
Observing a quantum system will result in the measurement of one of the classical states - the quantum system would never let us observe the superposition state.
When you measure a quantum state $|\phi\rangle$, which is a superposition of classical states $|0\rangle, |1\rangle, ..., |N-1\rangle$, it *collapses* to one of the classical states.

Measurement of a quantum system

A quantum system in general will be in a superposition state of some number of classical state vectors. As soon as one observes or measures the state of a quantum system, the state *collapes* into one of the classical states. This means that is not possible to actually *observe* a superposition state.

We cannot say with certainty, to which classical state the quantum state will collaps. We cannot precisely predict, through which one of the slits the electron will come.

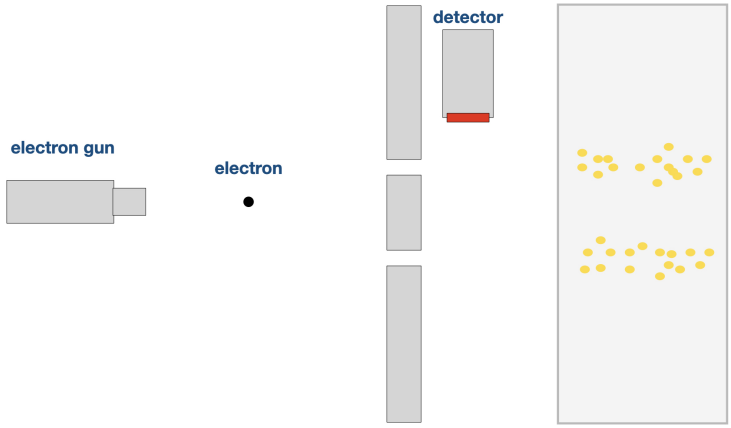


Figure 1.8: Double slit experiment with electrons, again, but this time we put a detector behind the double slit wall, to observe which one of the slits the electrons take. It seems to be a coincidence, which slit the electrons take and after some time one would see, that they did not produce an interference pattern. It looks like they behaved like tiny, classical balls again. The process of measuring their state has destroyed their state.

But we can say, with what *probability* $|\phi\rangle$ will collaps into a certain classical state $|i\rangle$, $i \in 0, 1, ..., N-1$. The probability, in which state a superposition state will collaps after measurement - is given by the so-called *Born rule*. Let's have a quick look.

1.4.2 The Born rule

The Born rule states that the amplitudes $a_0, a_1, ..., a_{N-1}$ of a superposition quantum state

$$|\phi\rangle = a_0 |0\rangle + a_1 |1\rangle + ... + a_{N-1} |N-1\rangle \tag{1.30}$$

should be interpreted as *probability amplitudes*. In case the quantum state is measured, probability amplitude tells us how probable it is for the superposition state to collaps into a certain classical state.
The probability of the state $|\phi\rangle$ collapsing into the state $|i\rangle$ is given by the term $p_i = |a_i|^2$. Let's say for example we work with the state

$$|\phi\rangle = \frac{1}{\sqrt{3}} |0\rangle + \sqrt{\frac{2}{3}} |1\rangle \tag{1.31}$$

and we measure it. It will collaps into one of the two classical states, with a probability of $\frac{1}{3}$ into $|0\rangle$ and with a probability of $\frac{2}{3}$ into the state $|1\rangle$.
In order to be interpreted as a set of probability amplitudes, the coefficients have to satisfy the following equation:

$$\sum_{i=0}^{N-1} |a_i|^2 = |a_0|^2 + |a_1|^2 + ... + |a_{N-1}|^2 = 1 \tag{1.32}$$

This equation just ensures that the total probability will always be exactly 1. When a superposition state is observed, it will collaps into at least one of the classical states that form the basis of the Hilbert space.

The Born Rule

After measuring a quantum state, it will collaps into one of the classical states. According to Born's rule, the amplitudes of a superposition state tell us how probable it will be that the state will collaps into this specific classical state.

1.4.3 Unitary evolution

We cannot only observe a quantum state (and by doing so destroying it), but we can as well manipulate it by applying some operation to it and change it to another state. Let's assume we have the state

$$|\phi\rangle = a_0 |0\rangle + a_1 |1\rangle + ... + a_{N-1} |N-1\rangle \tag{1.33}$$

and we apply some operation U to change it to the state:

$$U |\phi\rangle = |\psi\rangle = b_0 |0\rangle + b_1 |1\rangle + ... + b_{N-1} |N-1\rangle \tag{1.34}$$

We can write the two states $|\phi\rangle, |\psi\rangle$ as column vectors with the probability amplitudes as components. This changes the transformation equation to the following form:

$$U |\phi\rangle = U \begin{pmatrix} a_0 \\ a_1 \\ . \\ . \\ . \\ a_{N-1} \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ . \\ . \\ . \\ b_{N-1} \end{pmatrix} \tag{1.35}$$

To transform the vector $|\phi\rangle$ into the vector $|\psi\rangle$, U has to be a $N \times N$ matrix. To visualise that, we will take a look at the example of a quantum system with two basis states, where we have a 2-dimensional Hilbert space representing the quantum system. Let's assume our transformation operation is a 2×2 matrix

$$U = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{1.36}$$

and we apply this operation to the state vector $|\phi\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$, then we get the result:

$$U |\phi\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle \tag{1.37}$$

The transformation operation U changes the vector $|0\rangle$ to the state vector $|1\rangle$.
Matrix transformations that manipulate quantum states are *unitary transformations*.
Unitary transformations always have an inverse element U^{-1} such that $UU^{-1} = \mathbf{1}$ with $\mathbf{1}$ being the $N \times N$ identity matrix.
The fact that unitary matrices or unitary transformations have an inverse means that such operations can be undone. If one manipulates a quantum state $|\phi\rangle$ by applying the operation U to it and changing it to the state vector $|\psi\rangle$, one can always undo that process by applying the inverse transformation U^{-1} to $|\psi\rangle$.
This is the key difference between measurement and unitary transformation.

Unitary evolution

Instead of destroying a quantum state by measuring it, we could apply *unitary transformations* to change it without destroying it. Unitary transformations are (other than measuring a state) reversible and thus important for processing information in quantum computers.

When measuring a quantum state, it collapses to a classical state and the quantum state cannot be reconstructed again. Measurement is an irreversible process. A unitary transformation is not. Those two processes will later be important to develop the quantum physical analogy of logic gates.

1.5 Compound systems and entanglement

In this section we will learn how to describe systems that consist of more than one quantum system: *compound systems*. After studying the mathematics of compound systems, we will be able to apply them on so-called *entangled states*. Compound systems that are in an entangled state will be of great importance later when we will come to the quantum computing part of this course.

1.5.1 Compound systems and the tensor product

Almost always when doing quantum computing we will be interested in describing not only one quantum system but two or more quantum systems at the same time. A quantum system that consists of multiple sub-systems, is called a *compound system*. And a compound system is in a compound state.
But how do we describe the state of a compound quantum system?
Let's assume that we have two quantum systems A and B. The system A shall be in the state $|\psi\rangle$ and the state B shall be in the state $|\phi\rangle$. The compound system is said to be in the state

$$|\psi\rangle \otimes |\phi\rangle \tag{1.38}$$

with $\otimes$ denoting the tensor product. Let's have a quick look at what a tensor product is.
A tensor product is a special type of product between matrices. Consider a $n \times m$ matrix A and a $p \times l$ matrix B. The tensor product between theese two matrices will result in a $np \times ml$ matrix.
For example, if A and B are both 2×2 matrices, the tensor product between those two would be a 4×4 matrix.
The actual product is defined in the following way:

$$A \otimes B = \begin{pmatrix} A_{11}B & \dots & A_{1m}B \\ A_{21}B & \dots & A_{2m}B \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ A_{n1}B & \dots & A_{nm}B \end{pmatrix} \tag{1.39}$$

Since this definition looks pretty complex we will break it down in an example. Let A and B be 2×2 matrices:

$$A = \begin{pmatrix} 2 & 2 \\ 2 & -2 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \tag{1.40}$$

The tensor product of these two matrices would result in the following 4×4 matrix:

$$A \otimes B = \begin{pmatrix} 2 & 2 \\ 2 & -2 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 2 & 0 & 2 \\ -2 & 0 & -2 & 0 \\ 0 & 2 & 0 & -2 \\ -2 & 0 & 2 & 0 \end{pmatrix} \tag{1.41}$$

Since vectors are matrices too (a vector is a matrix with just one column), we can apply the tensor product to them. Take for example the tensor product between the two vectors:

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \tag{1.42}$$

So now we know what the tensor product is and how it is defined for matrices and vectors. But how does that apply to compound quantum systems?
Assume, the system A is represented by a n-dimensional Hilbert space H_n with n basis vectors $|x_i\rangle ; i = 1, ..., n$ and the system B corresponds to an m-dimensional Hilbert space H_m with m basis vectors $|y_j\rangle ; j = 1, ..., m$. The two states $|\psi\rangle , |\phi\rangle$ can - according to the superposition principle - be written as a linear combination of the basis vectors.

$$|\psi\rangle = \sum_{i=1}^{n} \alpha_i |x_i\rangle \quad |\phi\rangle = \sum_{j=1}^{m} \beta_j |y_j\rangle \tag{1.43}$$

The compound system $A \otimes B = H_n \otimes H_m$ (where $H_n \otimes H_m$ is the set of state vectors, that are elements of the compound system) would be in the state (The tensor product between state vectors $|\psi\rangle \otimes |\phi\rangle$ is often abbreviated by $|\psi\rangle |\phi\rangle$ or $|\psi\phi\rangle$):

$$|\psi\rangle \otimes |\phi\rangle = |\psi\rangle |\phi\rangle = \left(\sum_{i=1}^{n} \alpha_i |x_i\rangle \right) \otimes \left(\sum_{j=1}^{m} \beta_j |y_j\rangle \right) = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j |y_j\rangle |x_i\rangle \tag{1.44}$$

Okay this looks pretty complex so we better look at an example to better understand that formula. Let's say both quantum systems A and B are represented by the 2-dimensional Hilbert space H_2 with the two basis vectors $|0\rangle , |1\rangle$. So the compound system would be $H_2 \otimes H_2$. Assume that A is in a superposition state $|\psi\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$ and B in the classical state $|\phi\rangle = |0\rangle$. In which state would the compound system $A \otimes B$ be?

Compound quantum systems

A *compound system* is a combination of many quantum systems, that is mathematically described by the *tensor product* of the underlying Hilbert spaces. Describing a bunch of electrons as one connected system would happen by applying the compound system formalism.

$$
\begin{aligned}
|\psi\rangle |\phi\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \otimes |0\rangle = \\
= \frac{1}{\sqrt{2}} (|0\rangle |0\rangle + |1\rangle |0\rangle) = \\
\frac{1}{\sqrt{2}} \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) = \\
\frac{1}{\sqrt{2}} \left(\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \right) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}
\end{aligned}
\tag{1.45}
$$

The question now arises, if we can write any state vector in a compound quantum system as the tensor product of sub-space vectors. The answer is no, there exist state vectors, that cannot be *decomposed* into the tensor product of sub-vectors. Such a state is called *entangled*. Let's have a closer look at those states.

1.5.2 Entangled States

We look at the compound system $H_2 \otimes H_2$ again, but this time we assume that it is in the superposition state

$$
|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{1.46}
$$

Can we write this state vector as a tensor product of two separate vectors $|\mu\rangle, |\tau\rangle \in H_2$ with $|\mu\rangle = a_0 |0\rangle + a_1 |1\rangle$ and $|\tau\rangle = b_0 |0\rangle + b_1 |1\rangle$? If it is possible, then this state vector would be called *decomposable*.

$$
\begin{aligned}
|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \stackrel{!}{=} |\mu\rangle \otimes |\tau\rangle = \\
(a_0 |0\rangle + a_1 |1\rangle) \otimes (b_0 |0\rangle + b_1 |1\rangle) = \\
a_0 b_0 |00\rangle + a_0 b_1 |01\rangle + a_1 b_0 |10\rangle + a_1 b_1 |11\rangle
\end{aligned}
\tag{1.47}
$$

For this equation to work the coefficients must fulfill the following conditions:

$$
a_0 b_0 = a_1 b_1 = \frac{1}{\sqrt{2}} \tag{1.48}
$$

$$
a_0 b_1 = a_1 b_0 = 0 \tag{1.49}
$$

Equation 1.49 only works, if at least one number is zero, $a_1 b_0 = 0$ is true, if either a_1 or b_0 equals zero. But if only one of these coefficients equals zero, then equation 1.48 would not work anymore. Therefore the superposition state from equation 1.46 cannot be written as the tensor product of two sub-space vectors $\Rightarrow$ it is not decomposable.
States that are not decomposable are called *entangled*. The two quantum systems that form a compound system share a state vector and are linked to each other.

Compound quantum systems that are entangled have a pretty interesting property when it comes to measurement.
What will for example happen, when we go into the system A and measure the entangled state?
By measuring, the system A will collaps into one of the classical states $|0\rangle, |1\rangle$, and the whole compound system will collaps into either $|00\rangle$ or $|11\rangle$. What happens in one system, influences the events in the other system.
The crazy thing about that is, that it does not matter how far apart the two quantum systems are from each other. As long as they are entangled, they can be at different ends of the universe but would simultaneously collaps into a classical state once one of the systems is observed.

Entangled states

Two quantum systems can be linked together in a way, that they both influence the other system. What happens in one system directly impacts what happens in the other system. This link between quantum states is known as *entanglement* (or as Einstein called it: spooky action at a distance).

Let's do a real life example and take the well-known electron to explain entanglement. Electrons have a quantum property, that physcists call spin. You could somehow compare that property to the angular momentum of a rotating body, meaning that the spin of an electron contains information about its rotation. For our purposes it is not important to know what the spin is exactly. It will be enough to know that the spin of an electron is a quantum state.
The spin of an electron can either point upwards or downwards (*spin-up* and *spin-down*). Assume, we have entangled two electrons and they are in a superposition state

$$|\phi\rangle_{electron} = \frac{1}{\sqrt{2}}(|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle) \tag{1.50}$$

with $|\uparrow\rangle$ being the spin-up state and $|\downarrow\rangle$ the spin-down state. If we measure the first electron and observe that it has a spin-up state, then the other electron will instantaneously collaps to the spin-down state.
So quantum particles or quantum systems cannot only be in two classical states at the same time, but they can also be entangled to each other and seem to be able to transfer information with a velocity greater than the speed of light. This "spooky action at a distance" (how Einstein described entanglement) is a very important concept in quantum computing, that we will later use to understand how quantum teleportation and superdense coding work.

2 From QuBits to Quantum Computing

Now that we learned a little bit about quantum physics and how strange the results of experiments in this area are, we are ready to face the transition to quantum computing. How can we use our freshly learned knowledge to find out what a quantum computer would be capable of doing?
Before we can do so, we will take a look at classical computers and how they work. And as we all heard somewhere at some time in the past: Computers work with 1's and 0's, they work with so-called *bits*.

2.1 Classical Bits and Qubits

Computers work with bits, right? What is a *bit* actually?
Classical Bits, or short just bits, are the unit of information used in modern day computers and in digital technology in general. Any type of information that you can imagine is encoded into a bit, videos, pictures, text, calculations.
A bit can take one of two possible values. It can either be 1 or 0. That means that any type of information that you can imagine can be represented by a sequence of 1's and 0's: {10010101110101101010001001010010011111} is an example for a sequence of bits representing some information.
When it comes to the processing of information, the change of any information is a change of the bit values in the sequence of bits. Watching a video or adding two numbers together starts with a sequence of bits, that is changed into another sequence of bits:

$$\{10110101001010010101010\} \rightarrow \{10101010101010001010\} \tag{2.1}$$

In digital technology, the bit with the value 1 corresponds to"electrical current *on*", wheras the bit value 0 corresponds to "electrical current *off*.
The best way to visualize that is by showing it graphically, as you can see in Figure 2.1. There you can see an electric circuit powered by a battery. The current flows through a lamp and then through a strange looking device, which is called *transistor*.
Transistors are very important devices in modern electronics, since they are the cornerstore of digital technology. All logic circuits (that actually make a computer so powerful) are made of transistors.
So let's have a quick look at how a transistor works.
A transistor has three pins. The one in the middle labelled with a B is called the *basis*, the one labelled with a C is called the *collector* and the one with the E the *emitter*. You

can imagine the collector as the entrance to the transistor and the emitter as the exit. The current flows through the entrance (collector C) into the transistor and wants to go through the exit (emitter E). But to do so, there has to be another current from the basis B, to open the gate to the exit E. This is how a transistor works.
If there is a current in the basis, the current can flow from the collector to the emitter and the whole circuit is closed, such that the lamp is burning. If there is no current in the basis, the flow from the entrance to exit is not possible and the lamp remains dark.This is how we can imagine a bit.

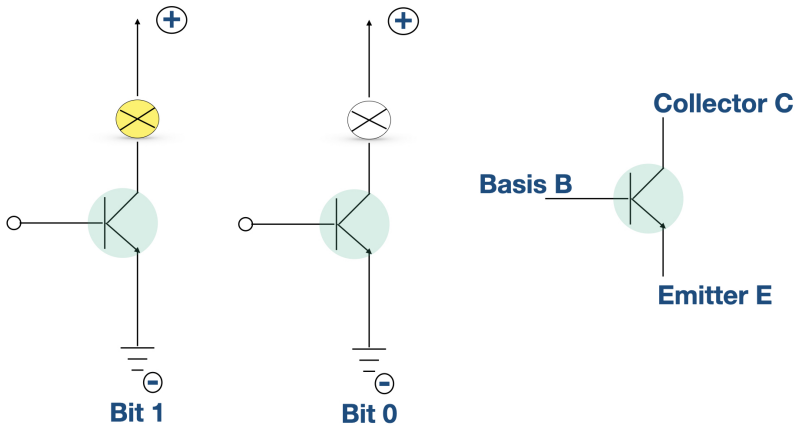


Figure 2.1: This is how a simple circuit with a transistor in it. The current flows from through the lamp and then into the collector C. If there is current in the basis B, the flow from C can go on and leave through the emitter E to close the circuit. If there is current in B, the lamp would shine and represent the bit with value 1. If there is no current at the basis, then the lamp would remain dark, representing a bit with value 0.

A shining lamp represents the bit with value 1, a dark one the bit with value 0.
There can either be a current at the basis b (and the lamp shines) or there is no current and the gate is closed.
The bit can thus take only one value, either 1 or 0.
It cannot be both.

Classical Bits

A *classic bit* or just *bit* is the basic unit of information in digital technology. It can take one of two possible values, either 1 or 0. In electronic circuits, bit values are represented by *current on/ current off*.
Flowing electricity corresponds to a bit with value 1. An off electric current represents a bit with value 0.

But a quantum bit can be both. And this is how we can define quantum bits.
A *quantum bit* - or from now on short a *qubit* - is a superposition of the two classical states $|0\rangle$ and $|1\rangle$, which means it is a superposition of the two classical bit values.
A qubit can be 1, 0 or some superposition of both.
To be more precise, a qubit is defined as a 2-dimensional Hilbert space (denoted with H_2) with the basis vectors $\{|0\rangle, |1\rangle\}$.
This means that in quantum computing, information will no longer be processed by the concept "current on/current off", but by qubits, which are quantum systems that have to classical states as basis vectors.

Quantum Bits

In order to do *quantum computing*, we will no longer just stick to the classical bits as a unit for information, since they can only take two possible values 1 and 0. Instead we define a *quantum bit* to be a quantum system, which has two classical states as basis vectors. This means, that a qubit can be in a superposition state of those two basis vectors.
When the qubit gets measured, it collapses into one of the two classical states - it turns into a classical bit.

If we want to do quantum computing, we are going to need more that one qubit. Just like in modern day computers, one bit is not enoguh to save and at the same time process information.
When we are working with more than one qubit, we are having many quantum systems which together form a compound system. And as we learned in the last chapter, a compound system is represented by the tensor product of the underlying Hilbert spaces (the Hilbert spaces representing each single quantum system). Every qubit we are working with is represented by a 2-dimensional Hilbert space H_2. So the compound system would be described by the tensor product $H_2 \otimes ... \otimes H_2$. When it comes to quantum computing, a compound system of many qubits has the name *quantum register* with dimension 2^m, where m is the number of qubits in the compound system.

Quantum Register

One qubit is not enough. We will need many qubits in order to do quantum computing, and many qubits together can be seen as compound quantum system. A compound system of m qubits is called a *quantum register* with dimension 2^m. Every qubit is represented by a 2-dimensional Hilbert space H_2. The quantum register is represented by the tensor product of m 2-dimensional Hilbert spaces.

So how could we implement a qubit or even a register of qubits in real life?
This is what physicists have been working on for some decades now and there has been some progress on this field of research. One possible way to experimentally realise a qubit or even a regsiter of qubits is by using so-called *ion traps*.

Imagine an atom. Atoms consist of a positively charged nucles and in general an equally negative charged electron cloud surrounding the nucleus.
By using a laser we could kick a single electron out of the cloud in order to make the atom apositively charged ion, as you can see in Figure 2.2

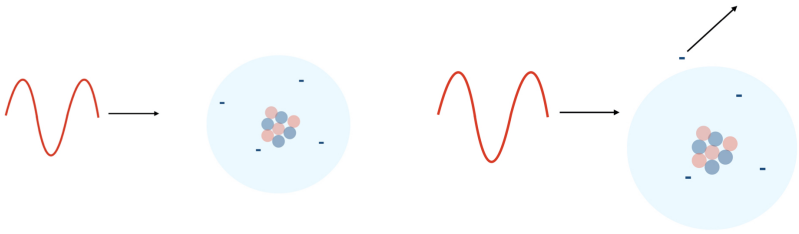


Figure 2.2: A laser is shot onto an atom and kicks an electron out of the electround cloud which surrounds the nucleus of the atom. This process is called *ionization*. Now that the whole atom is electrically charged it reacts to the use of electric fields.

Since the atom now has an electrical charge, it reacts to the use of electrical fields. And that's where the actual ion trap comes into play.
An ion trap is basically nothing else than two electrodes facing each other with a strong, electric field between them. This electric field is strong enough, to capture the wanted ions and hold them in a stable position (see Figure 2.3).

Since we don't want the captured ions to interact with anything, we put the ion trap into a vacuum chamber.
Now the ions captured in the trap are cooled down by a laser to a temperature of almost 0 Kelvin ($-273°$ Celsius).
One element, which is used in ion form to create qubits in an ion trap, is Ytterbium.
After Ytterbium is cooled down and in its groundstate, one can see that this groundstate is split up into two sub-states (due to the spin of the nucleus). Remember the definition of a qubit. A quantum bit is a quantum system with to basic states. This means that the quantum system can only collaps into one of two possible states.
The atom can be seen as a quantum system, which is the qubit and the two states of the groundstate are the classical states of the qubit (the states, the qubit can collaps into, when the qubit is measured).

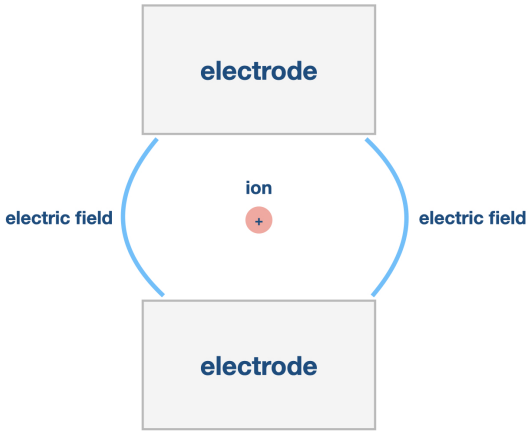


Figure 2.3: This is how one could simply imagine an ion trap in a schematic way. A positively charged ion is trapped in the electric field between to electrodes.

Qubits in experiment

How to build single qubits and registers is part of actual research. One way of realizing qubits would be by using ion traps. An *ion trap* is just an electrical field between two electrodes, that keeps some ions in a stable floating position. Then they are cooled down to almost 0 Kelvin. such that they remain calm. The whole setup is put into a vacuum chamber, so no kind of interaction can disturb the compound system of ions.
The states of the ions serve as qubits and can be changed by laser interaction.

So instead of encoding our information into bits, we can encode them into quantum bits. But information needs to be processed, and this means it needs to be changed. As we remember from the beginning of this section, in digital technology any information is represented by a sequence of 1's and 0's. Processing the information would mean changing the values of the single bits to get another sequence of 1's and 0's.
The quantum computing analogon would be having a register of qubits and changing the states of every single qubit. In the case of the ion trap qubit, changes of the state can be done by a laser. But how do we need to change the states of qubits in order to do computations?
Therefore we are first going to have a look at how classical computers process information on a fundamental level.

2.2 Logic gates and circuits

In this section we will learn, what *logic gates* are and how they are used to change sequences of bits. We will discuss the three most important logic gates AND, OR and NOT. In the end we will find out that combining logic gates gives a so-called *circuit* and that any circuit can be constructed by using only these three gates.

As we learned in the last chapter, any form of information can be represented by a sequence of bits. Processing that information means to change the sequence of bits to another sequence of bits. In order to change the whole sequence we have to change the single bits.
Changing single bits is achieved by applying *logic gates* to the sequence of bits. A logic gate is an operation, that acts on one or more bits. They are the fundamental building blocks of digtal information processing. So let's have a look at the most important logic gates.

2.2.1 The NOT-Gate

The most simple logic gate is the so-called NOT-Gate, sometimes also called an Inverter. The NOT-Gate acts on a single bit and has a single bit as an output. It *inverts* the value of the input bit, that's why it's also referred to as an inverter. If the input bit has the value 0, the output bit would take the value 1 and vice versa. In the upper-left-hand corner of Figure 2.4 you can see its symbol and below that its truth table.
On the right side of Figure 2.4 we can find the circuit diagram, which shows how we could implement a NOT-Gate in real life. The input would be the basis of the transistor, the output leads away from the collector of the transistor.
If there is current on the basis of the transistor, then the value of the input bit A will take the value 1. This means, that no current can flow to the output pin, since the current flows through the transistor. No current in the output pin means that the bit value of the output is 0.
If there is no current at the basis of the transistor, then the input bit has the value 0. The current is not able to get from the collector to the emitter so it takes the free way to the output, which has then the bit value 1.

2.2.2 The AND-Gate

The next logic gate we are having a look at is the so-called AND-Gate. The AND-Gate no longer acts only on one bit, but on two bits and results in one single bit as an output value. You can see the symbol in the upper-left-corner of Figure 2.5. The truth table below shows how the AND-Gates works. Explained in one sentence:
If both inputs A and B take the value 1, the output will take the value 1.
So only if A=1 and B=1, then the output bit will have the value 1. If only one of the

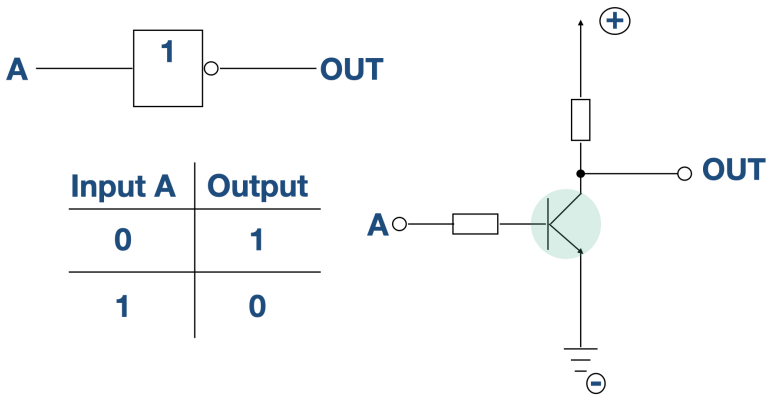
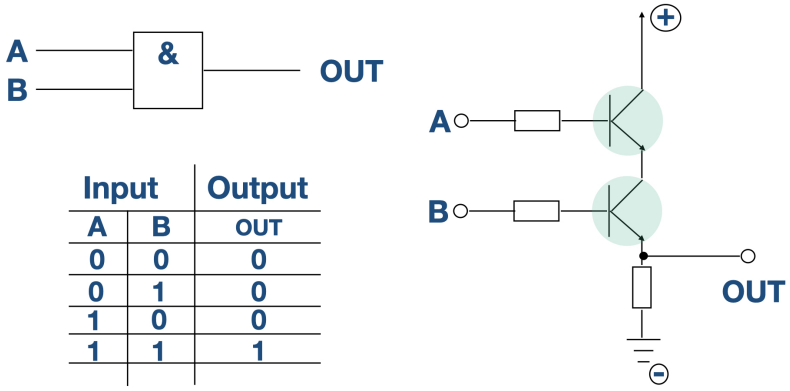


Figure 2.4: The symbol for the NOT-Gate (upper-left-hand corner) with one input and one output. The truth table (below the symbol) shows how the NOT-Gate works. If the input A has the value 0, the output OUT takes the value 1 and vice versa. On the right side you can see, how a NOT-Gate would be implemented with a transistor. If the basis of the transistor has current, then the input bit value would be 1. The current flows from the collector to the emitter and not to the output, thus the output takes the value 0. No current in the basis would block the transistor, such that the current flows to the output instead of going through the transistor.

inputs has the value 1 or none of them, then the output will remain on bit value 0. Implemented with transistors it would look like on the right side of Figure 2.5. Only if both transistors get current in their basis (when both inputs A and B have the bit value 1), then the output will get current and thus take the value 1. If the basis of the first transistor gets current, but the second one doesn't, the current flows through the first and gets stopped at the second one.

2.2.3 The OR-Gate

The last important logic gate we are having a look at is the so-called OR-Gate. Just like the AND-Gate, it acts on two input bits and results in one output bit. You can see the symbol of the OR-Gate in Figure 2.6 together with the truth table and the transistor implementation. The functionality of the OR-Gate could be described like this:
If either A ***OR*** *B or both have the value 1, then the output will take the value 1.*
So if both input bits have the value 0, then the output is also zero. If only one of the inputs takes the value 1, the output switches to bit value 1 as well.
Implemented with transistors the OR-Gates looks a little bit more tricky than the previous gates. If we follow the way of the current, we will find out that it either has to go through the first or the second transistor to get to the output. Both ways are possible and it is sufficient, if only one transistor has current in its basis.



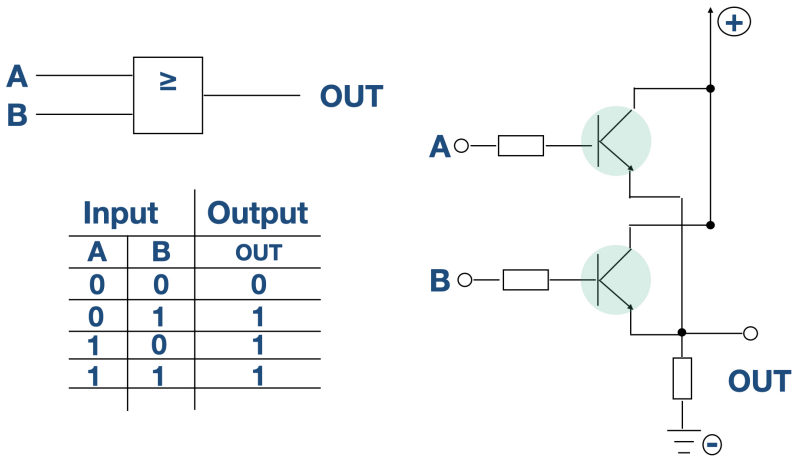
Input		Output
A	B	OUT
0	0	0
0	1	0
1	0	0
1	1	1

Figure 2.5: The AND-Gate symbol in the upper-left-hand corner with two inputs A and B and one output OUT. The truth table below shows how the AND-Gates acts on the input bits. On the right side you can see, how an AND-Gate is implemented with transistors. Only if the basis of each transistor A and B gets current, the current can flow from the collector of the first transistor through both transistors to the output.

Logic gates

To process information one needs to apply so-called *logic gates*. The three most important ones are the AND, OR and the NOT gate.
The NOT gate acts on a single bit and has one bit as an output. It is also referred to as the inverter, because it inverts the bit value of the input.
The AND gate acts on two input bits and results in one single output bit. The output only takes the value 1, if both input bits have the value 1.
The OR gate - just like the AND gate - also acts on two input bits and results in one output bit. The output takes the value 1, if at least one of the input bits has the value 1.

There are many more gates, that are being used in digital technology, but it's enough to know these three, as we will see right now. Let's recall that any type of information can be represented by a sequence of bits and that processing information means to change that sequence of bits into another one. To do so, one has to change the single bits of the sequence and this is done by applying logic gates like AND, OR and NOT. To change the whole sequence of bits we need to combine the logic gates to a so-called *circuit*. It can be mathematically proven that any circuit you can imagine can be constructed by using only those three gates. To be more precise, either the set $\{AND, NOT\}$ or the set $\{OR, NOT\}$ is sufficient to construct any possible logic circuit. This is why those gates are called *universal*. You could for example construct an OR-gate by using only AND- and NOT-gates and vice versa.



Input		Output
A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	1

Figure 2.6: The OR-Gate symbol in the upper-left corner with the two inputs A and B and the one output OUT. The truth table below shows, how the OR-Gate acts on the input bits. Other than the AND-Gate, it is sufficient to have only one input bit with value 1 to switch the output bit to 1 as well. On the right side you can see the implementation with transistors. The current can flow through both transistors to get to the output. It is sufficient, if only one transistor has current on its basis.

Every microprocessor consists entirely of these logic gates. It does not matter how you would like to change a certain type of information - it is entirely done by applying these three logic gates. Whether it is watching videos or adding numbers.
Now that we know how a modern day computer operates on a fundamental level, we are ready to apply our knowledge in the quantum world.

Circuits and Universality

Combining logic gates creates a so-called *circuit*. Any existent logic circuit can be constructed by using either the set $\{AND, NOT\}$ or the set $\{OR, NOT\}$. This makes those three gates a *universal set*. Every microprocessor processes information by using only those logic gates.

2.3 Quantum gates and quantum circuits

This section will be all about the tools we need, to theoretically build a quantum computer. We will learn what *quantum gates* are and what the difference between a *unary* and a *binary quantum gate* is. After that we will look at *quantum circuits* and how they are visualized. In the end will answer the question, is a quantum computer could do the same stuff a classical computer can do.

In the last section we learned that in order to process information (which is represented by a sequence of bits), one has to change the single bits by applying logic gates. Those logic gates can either act on one or multiple bits and connected with each other they form a circuit.
When it comes to processing quantum information (information, that is encoded in qubits), one has to change the single qubits. As we learned in the first chapter, a qubit is a quantum system, that can be represented by a 2-dimensional Hilbert space H_2 with the two basis vectors $|0\rangle$ and $|1\rangle$. Changing a qubit would mean changing the state of the qubit.
To change the state of a qubit without actually destroying it, we will have to apply *unitary transformations* on them. And this is how we can define, what a quantum gate is:
A quantum gate is a unitary transformation acting on a quantum register of length m (a compound quantum system of m qubits).
Quantum gates are reversible, which is a huge difference to classical logic gates. Take the AND-gate for example. It has two inputs and only one output, which means that one bit of information gets lost in the process. Thus, the AND-gate can not be reversed. When it comes to quantum gates, the number of inputs will equal the number of outputs, to ensure unitarity.
There are two important categories of quantum gates, *unary* and *binary quantum gates*. Unary quantum gates are acting only on one qubit (changing the state of one qubit), whereas binary gates act on two qubits. Let's have a look a those.

Quantum gates

A *quantum gate* is a unitary transformation, that acts on either one or multiple qubits. If it acts on one qubit, it is called a *unary gate*. If it acts on two qubits, it is called a *binary gate*. Since quantum gates are unitary, they are - other than classical logic gates - reversible.

2.3.1 Unary quantum gates

Assume we have a qubit represented by the Hilbert space H_2 with basis $\{|0\rangle, |1\rangle\}$. A unary quantum gate is a unitary function

$$U : H_2 \to H_2 \tag{2.2}$$

that changes the state $|\psi\rangle$ of the qubit to $U|\psi\rangle = |\phi\rangle$.
As we have seen in the first chapter, unitary transformation can be represented by matrices. A unary quantum gate is represented by a 2×2 matrix. The most important unary quantum gates are the following ones:

$$\begin{aligned} M_{NOT} &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ F &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\ H &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \end{aligned} \tag{2.3}$$

M_{NOT} is the quantum version of the classical NOT-gate. If the M_{NOT} gate acts for example on the state $|0\rangle$, it will result in the state $|1\rangle$ and vice versa.

$$\begin{aligned} M_{NOT}|0\rangle &= |1\rangle \\ M_{NOT}|1\rangle &= |0\rangle \end{aligned} \tag{2.4}$$

Let's take qubit in an arbitrary state $\alpha|0\rangle + \beta|1\rangle$ with $\alpha^2 + \beta^2 = 1$. How will the NOT-gate change that state?

$$M_{NOT}(\alpha|0\rangle + \beta|1\rangle) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix} = \beta|0\rangle + \alpha|1\rangle \tag{2.5}$$

So the NOT-gate flips the probability amplitudes. Before, the state $|0\rangle$ would have been measured with a probability of α^2, but after applying the NOT-gate, the state $|0\rangle$ will be measured with a probability of β^2.

The next gate - denoted with a F, is called the *phase flip* gate. To understand, how this gate works, we will apply it to the superposition state $|0\rangle + |1\rangle$:

$$F(|0\rangle + |1\rangle) = |0\rangle - |1\rangle \tag{2.6}$$

And the last important unary quantum gate, that we will learn about, is the so-called *Hadamard gate*, which acts the following way on the input qubit:

$$\begin{aligned} H|0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ H|1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned} \tag{2.7}$$

Unary quantum gates

Quantum gates can be represented by matrices. Since a *unary gate* acts on only one qubit (which is a 2-dimensional vector), it is represented by a 2×2- matrix. The most important unary gates to know are the *NOT*, the *phase flip* and the *Hadamard gate*.

2.3.2 Binary quantum gates

Now, that we know what a unary quantum gate is, we can have a look at binary qunatum gates. A binary quantum gate is a unitary transformation, that acts on a compound system of two qubits:

$$U : H_2 \otimes H_2 \to H_2 \otimes H_2 \tag{2.8}$$

Binary quantum gates can be written as a 4×4 matrix. Consider for example the following matrix:

$$M = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix} \tag{2.9}$$

This matrix can be seen as a binary quantum gate. Let's see how it acts on a compound quantum system of two qubits. For simplicity we will assume, that both qubits can be found in the state $|00\rangle$. Applying the transformation to this compound state will result in:

$$M|00\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix} |00\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle) \tag{2.10}$$

We get an entangled state as a result! So by applying this quantum gate we are able to entangle two qubits.

One very important type of binary quantum gates are the so-called *controlled-M* gates, where M stands for an arbitrary unary quantum gate.
Let's assume again, we have two qubits A and B, with M being a unary quantum gate acting on the qubit B. The controlled-M gate acts on the compound system $A \otimes B$ and is defined by:

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes I + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes M \tag{2.11}$$

A very important - if not *the* most important - gate of this type is the *controlled-NOT* gate M_{CNOT}:

$$M_{NOT} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{2.12}$$

$$M_{CNOT} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (2.13)$$

Okay, we have a pretty complex looking 4×4 matrix claiming to be the most important binary quantum gate. But what does it actually do?
Controlled gates act on two qubits. One qubit is the so-called *target qubit*, while the other one is called the *control qubit*. The control qubits remains unchanged during the transformation.
The CNOT - gate only inverts the target qubit, if the control qubit in in the state $|1\rangle$. This is best visualized by the truth table in Figure 2.7. If the control qubit is in the state $|0\rangle$, it does not matter, in which state the target qubit is - it stays unchanged.

Truth Table CNOT

Before		After	
Control A	Target B	Control A	Target B
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$

Figure 2.7: Truth Table for a controlled-NOT (CNOT) gate. Only if the control qubit is in the state $|1\rangle$, the target qubit will be flipped. The target qubit remains unchanged, if the control qubit is not in this state.

Now that we have learned about the most important quantum gates to know, we can go a step further and take a look at *quantum circuits*.

Binary quantum gates

Binary quantum gates are unitary transformations represented by a 4×4- matrix. The most important type of binary gates are the *controlled-M* gates, where M stays for an arbitrary unary gate. A controlled-M gate acts on twe qubits, one of which is called the control qubit, the other one the target qubit. The unary gate M only acts on the target qubit, if the control qubit is set to the state $|1\rangle$. The most important controlled-M gate is the *CNOT gate*. It only inverts the state of the target qubit, if the control qubit can be found in the state $|1\rangle$.

2.3.3 Quantum circuits

Similar to classical logic circuits, a *quantum circuit* is defined as a combination of several quantum gates acting together on a quantum register of a certain length m. Quantum circuits are usually visualized with so-called *quantum circuit diagrams*. A line represents a qubit whereas a rectangle with any given letter in it (say U) represents the action of a quantum gate (see Figure 2.8).

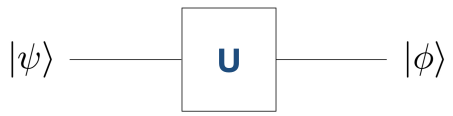


Figure 2.8: Quantum circuit diagram representing a qubit and a quantum gate U, that acts on the qubit. Qubits are represented by a line, gates by rectangles with letters in them, that specifiy the type of transformation. The gate U changes the state of the qubit from $|0\rangle$ to $|1\rangle$.

Let's have a look at how the different gates and operations on qubits, that we learned about, are represented by circuit diagram symbols (see Figure 2.9. The first ones with the letters H and F in their rectangles represent the Hadamard and the phase flip gate. The third one with a circle and a cross inside stands for the NOT-gate. The last one, which contains the letter M represents the act of measurement. In quantum circuit diagrams the measured state of a qubit is shown by a double line.

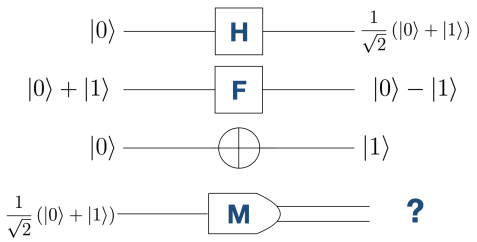


Figure 2.9: Circuit diagram symbols of unary quantum gates and operations. The first two gates are the *Hadamard* and the *phase flip* gate. The third one inverting the state of the qubit is the NOT-gate. The last one represents the process of measurement. If a qubit is measured, it is represented by a double line. In the case of the above state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, both classical states $|0\rangle$ and $|1\rangle$ are equally probable. We don't know, which result the measurement will yield.

Controlled-M gates on the other side are shown with a filled, black circle on the control qubit and a line connecting it to the target qubit. The visual representation of an arbitrary controlled-M and a controlled NOT- gate is shown in Figure 2.10.

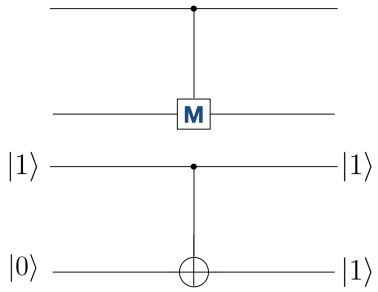


Figure 2.10: Circuit diagram symbold for a controlled-M and a controlled-NOT (CNOT) gate. The control qubit remains unchanged and is marked with a black dot and a connection to the target qubit, where the actual gate action happens.

As with classical logic circuits, there is also a quantum version of the *universality* concept. It can be proven that any quantum circuit can be constructed by CNOT - gates and all existent unary quantum gates.

Quantum circuits

A *quantum circuit* is a concatenation of quantum gates acting on a quantum register with some amount of qubits in it. Any quantum circuit can be constructed by using CNOT gates and all existen unary gates. They form a *universal set*. Quantum circuits are visualized by circuit diagrams. A qubit is represented by a single line, a measured qubit by a double line.

So now we learned everything that is needed to understand how a quantum computer could be constructed theoretically. We would need some quantum system, that can act as a qubit - just like the spin states of an atom. By applying unitary transformations we can change the states of the qubits and construct quantum circuits. Now the question may arise, what one can actually do with a quantum computer?
Can a quantum computer do the same things as a classical one?
Why is it superior to a classical computer?
Let's have look at the answer for the first question.

2.3.4 The quantum version of classical logic gates

To see, if a quantum computer could do the same things as a classical computer, we have to recall how a classical computer operates. In order to process information (which is a sequence of bits) one has to apply logic gates to change the single bits. The concatenation of gates changing the whole sequence of bits is called the circuit. Any circuit you can imagine can be constructed by using only AND, OR and NOT gates. Those gates are universal.
For a quantum computer to be able to do the same stuff a computer can do, we have to find out if there is a quantum version of these universal logic gates.
The answer is Yes, there is!
As we learned about quantum gates, they have to be unitary transformations (because it ensures the probability interpretation according to Born's rule and reversibility). So for a quantum computer to perform like a classical one, the classic logic gates have to be represented by unitary transformations.
To do so, we will need the so-called *Toffoli* gate, which is also known as the *controlled-controlled-NOT* gate (CCNOT). Other than the unary and binary quantum gates, that we learned about, the Toffoli gate acts on three qubits, two of which are control qubits. In matrix form, the Toffoli Gate would look like this:

$$M_{CCNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.14}$$

This matrix looks pretty confusing, but by taking a closer look one can see similiarties with the identity matrix. Only in the last two rows the positions of the 1's are switched. As a circuit diagram symbol, the Toffoli gate resembles a CNOT gate, it just has one additional control qubit. You can see the symbol for the Toffoli or the CCNOT gate in Figure 2.11.

But how does the Toffoli gate actually work? Only if both of the control qubits can be found in the state $|1\rangle$, the target qubit will be flipped by the NOT-gate. This can be seen in Figure 2.12. If both of the control qubits can be found in the state $|1\rangle$, the target qubit gets flipped. If the target qubit is in the state $|0\rangle$, it gets flipped to $|1\rangle$ and vice versa.

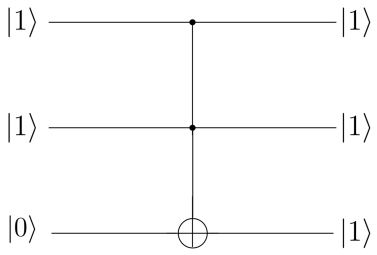


Figure 2.11: Circuit diagram symbol for the Toffoli- or CCNOT-gate. The three lines represent the three qubits, the Toffoli gate is acting on. The upper two lines are the control qubits, marked with the filled, black circle and both connected to the NOT-gate of the target qubit.

Before			After		
Control 1	Control 2	Target	Control 1	Control 2	Target
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$

Figure 2.12: Toffoli Gate truth table. Only if both of the control qubits are in the state $|1\rangle$, the target qubit will flip its state.

So let's see how we can use the Toffoli gate to construct the classical logic gates AND, OR and NOT.
So, the NOT-gate is the most simple one, since we already know the quantum version of the NOT-gate:

$$M_{NOT} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{2.15}$$

. Let's have a look at how we could construct the quantum version of the AND-gate. Recall that the AND-gate has two inputs and one output. Only if both input bits have the value 1, the output will take the value 1 too. If only one of them is zero or even both, the output bit will remain zero.
Take a look at the Toffoli gate. Only if both control qubits are in the state $|1\rangle$, the target qubit will flip its state. We could use the control qubits as input bits and the target qubit as the output. Therefore, we need to set the target qubit to the state $|0\rangle$. You can see the circuit diagram in Figure 2.13.

By having quantum versions of the NOT- and the AND-gate, we can construct any classical circuit we can imagine. This means, a quantum computer can do anything a

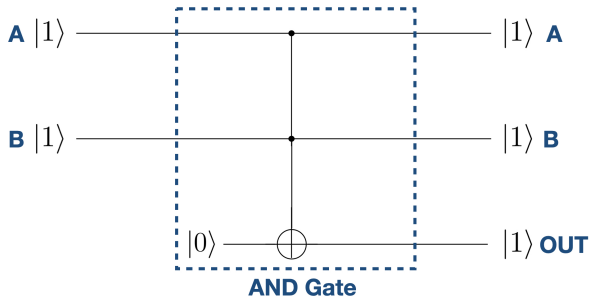


Figure 2.13: Quantum Version of the AND Gate realized with a Toffoli Gate. The control qubits act as input bits, whereas the target qubit is the output bit. Only if both control qubits /input bits are set to $|1\rangle$, will the target qubit /output bit take the value $|1\rangle$ as well.

modern computer can do.
Let's have a look at how the quantum version of the OR-gate will look like. Therefore, we need to remember that an OR-gate can be constructed by using AND- and NOT-gates. The steps are shown in Figure 2.14. First, we apply a NOT-gate to both inputs, then we act with an AND-gate on those inverted inputs and finally we invert the output again.

				Step 1		Step 2	Step 3
A	B	A and B	A or B	NOT A	NOT B	NOT A and NOT B	NOT (NOT A or NOT B)
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$

Figure 2.14: Truth table for the AND- and the OR-gate and Steps to construct an OR-gate by using AND- and NOT gates. First, both inputs are inverted, then we apply an AND-gate to the inputs and invert the outcome again.

So the quantum version of an OR-gate can be constructed similar to the AND-gate, just with a couple of additional NOT-gates. This can be seen in Figure 2.15. We again use a Toffoli gate and define the control qubits as the inputs for the OR-gate. Before they control the action on the target qubit, each of them gets inverted. The target qubit serves as an output and after the controlled action is done, the target qubit gets inverted once again.

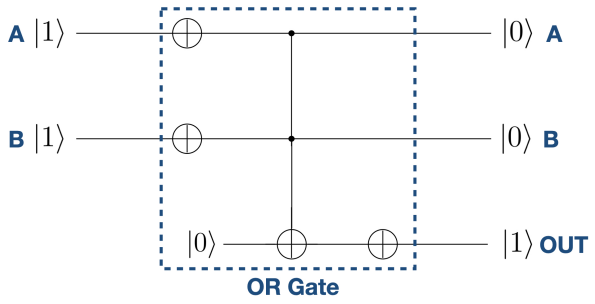


Figure 2.15: Quantum version of the OR gate, realized with a Toffoli gate and three additional NOT-gates. In this case, both of the control qubits are in the state $|1\rangle$. They get inverted to the state $|0\rangle$, so the NOT-operation does not act on the target qubit. The target qubit stays in the state $|0\rangle$, but in the end it gets inverted to $|1\rangle$. The same would happen, if only one of the control qubits is in state $|1\rangle$. If both control qubits are in the state $|0\rangle$, they get inverted to $|1\rangle$, which activates the controlled action. The target qubit state gets changed from $|0\rangle$ to $|1\rangle$, but then inverted again to $|0\rangle$.

Quantum classical logic gates

With the help of a *Toffoli gate* it is possible to construct classical AND, OR and NOT gates. The Toffoli gate acts on three qubits, two of which are the control qubits. Only if both of the control qubits are in the state $|1\rangle$, the target qubit will be inverted by the NOT gate.
The NOT gate already is a unitary transformation.
The AND-gate can be constructed by using the control qubits of a Toffoli gate as inputs and the target qubit as the output for the gate.
The OR-gate can be constructed similarily to the AND gate, the control qubits get inverted in the beginning as does the outcome of the controlled transformation.
Since it is possible to create a quantum version of those gates, a quantum computer is able to perform any task like a classical computer.

So now that we know that a quantum computer could do anything a classical computer can do, we can have a look at our second question: Why is a quantum computer or quantum technology superior to digital technology?
We will answer this question by taking a look at three interesting applications of quantum technology.

3 The power of Quantum Technology

After we learned the fundamentals of quantum computing and prooved that a quantum computer would be able to do anything that a classical computer can do, it is now time to take a look at the superiority of quantum technology. We will do so by studying two very impressive examples, that show how powerful quantum technology can be. First, we will learn about *superdense coding*, a protocol to encode the information of *two* bits into *one* single qubit. Eventually we will use all of our knowledge to work through the steps of *quantum teleportation* and learn about the *No-Cloning Theorem*.

3.1 Superdense Coding

The first example we are having a look at is the *superdense coding protocol* - a series of steps to encode the information of two bits into one single qubit, transmit the qubit and then extract the information from the qubit again.
Recall that a bit can take either the value 1 or 0. Thus, any information of two bits can be one of the following four sequences: $(0,0)$, $(0,1)$, $(1,0)$ and $(1,1)$.
Let's say, we have Alice and Bob, who want to try out the superdense coding protocol. Alice decides to be the sender and to encode the bits into a qubit, Bob will be the recipient, who decodes the sent information. In order to be able to send a qubit to Bob, there needs to be a *quantum channel* between Alice and Bob. A quantum channel is just a connection, that can be used to transfer qubits from one place to another.
A *classical channel* on the other hand would be any traditional way of sending classical bits. So anything like email, phone or even post is a classical channel.
The superdense coding protocol can be split up into three parts:

1. Encoding the bits
2. Transmitting the qubit
3. Decoding the qubit

But before the encoding can start, we need to put both Alice and Bob into an entangled state. That means we give both Alice and Bob a qubit each and then we apply a unitary transformation to entangle the states of both qubits. Alice's qubit shall be denoted A from now on, Bob's qubit will be labelled with a B. Let's assume that Alice and Bob are in the well-known entangled state:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{3.1}$$

We will say that the left qubit is Alice's and the right one is Bob's. To be more clear, we rewrite the entangled state as:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B) \tag{3.2}$$

Let's have a look at the actual steps of the superdense coding procedure. Alice has two bits a, b, that she wants to encode into her qubit, send to Bob, who then decodes them.

Step 1: We start with looking at the first bit a, which can have either the value 1 or 0. To encode this bit into her qubit, Alice performs the phase flip gate on her qubit, if the value of the bit is 1, otherwise the qubit remains unchanged.
If a=1, Alice uses the phase flip gate on her qubit. If a=0, her qubit stays unchanged.
So how will the entangled state look like after Alice performed the phase flip gate on her qubit?

$$F|0\rangle_A = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle_A \text{ and } F|1\rangle_A = -|1\rangle_A \tag{3.3}$$

So with F_A denoting the unary phase flip gate acting on the qubit A, the entangled state $|\psi\rangle$ changes to:

$$F_A|\psi\rangle = \frac{1}{\sqrt{2}}(F_A|0\rangle_A \otimes |0\rangle_B + F_A|1\rangle_A \otimes |1\rangle_B) = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B - |1\rangle_A \otimes |1\rangle_B) \tag{3.4}$$

So, if $a = 1$, the entangled state changes to

$$F_A|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \tag{3.5}$$

In Figure 3.1 you can see, how the entangled state changes dependend on the value of the first bit a.

a	state after step 1
0	$\frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$
1	$\frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$

Figure 3.1: Alice encodes the first bit a into her qubit by applying a phase flip gate in case the bit equals one, $a = 1$. If $a = 0$, the state stays the same.

Step 2: Now we take a look at the second bit b. Alice will apply the NOT-gate to her qubit, if the second bit equals 1, otherwise she doesn't to anything with her qubit. *If b=1, Alice applies the NOT gate on her qubit. If $b = 0$, she leaves her qubit unchanged.*
Mathematically, this would look like this:

$$M_{NOT} |0\rangle_A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |1\rangle_A \text{ and } M_{NOT} |1\rangle_A = |0\rangle_A \tag{3.6}$$

So if $a = 0$ and $b = 1$, the entangled state would change to:

$$\begin{aligned} & M_{NOT}^{(A)} \left(\frac{1}{\sqrt{2}} (|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B) \right) \\ &= \frac{1}{\sqrt{2}} (M_{NOT}^{(A)} |0\rangle_A \otimes |0\rangle_B + M_{NOT}^{(A)} |1\rangle_A \otimes |1\rangle_B) \\ &= \frac{1}{\sqrt{2}} (M_{NOT}^{(A)} |0\rangle_A \otimes |0\rangle_B + M_{NOT}^{(A)} |1\rangle_A \otimes |1\rangle_B) \\ &= \frac{1}{\sqrt{2}} (|1\rangle_A \otimes |0\rangle_B + |0\rangle_A \otimes |1\rangle_B) = \frac{1}{\sqrt{2}} (|10\rangle + |01\rangle) \end{aligned} \tag{3.7}$$

with the superscript (A) showing, that the NOT-gate is unary and acts on the qubit A.
If on the other side $a = 0$ and $b = 1$, the entangled state would look different after the second step:

$$\begin{aligned} & M_{NOT}^{(A)} \left(\frac{1}{\sqrt{2}} (|0\rangle_A \otimes |0\rangle_B - |1\rangle_A \otimes |1\rangle_B) \right) \\ &= \frac{1}{\sqrt{2}} (M_{NOT}^{(A)} |0\rangle_A \otimes |0\rangle_B - M_{NOT}^{(A)} |1\rangle_A \otimes |1\rangle_B) \\ &= \frac{1}{\sqrt{2}} (|1\rangle_A \otimes |0\rangle_B - |0\rangle_A \otimes |1\rangle_B) = \frac{1}{\sqrt{2}} (|10\rangle - |01\rangle) \end{aligned} \tag{3.8}$$

In Figure 3.2 you can see, how the entangled state would look like after the two bits a and b are encoded into Alice's qubit.

Step 3: Now that Alice has encoded both bits into her qubit (by applying quantum gates on her qubit), she can send the qubit via a quantum channel to Bob.

Step 4: Bob receives the qubit from Alice and now needs to extract the bits from Alice's qubit. He starts with performing a controlled-NOT gate on his own qubit B with A being the control qubit.
Bob uses a CNOT gate on his own qubit B with qubit A as the control qubit.

a	b	state after step 2
0	0	$\frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$
0	1	$\frac{1}{\sqrt{2}}(\|10\rangle + \|01\rangle)$
1	0	$\frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$
1	1	$\frac{1}{\sqrt{2}}(\|10\rangle - \|01\rangle)$

Figure 3.2: How the states would look like after both bits have been encoded into Alice's qubit.

Let's see how this would look like mathematically for the case $a = b = 0$:

$$\begin{aligned} & M_{CNOT}\left(\frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B)\right) \\ = & \frac{1}{\sqrt{2}}(M_{CNOT}\,|0\rangle_A \otimes |0\rangle_B + M_{CNOT}\,|1\rangle_A \otimes |1\rangle_B) \\ = & \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |0\rangle_B) = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \end{aligned} \tag{3.9}$$

In Figure 3.3 you can see how the state changes after Bob's CNOT operation, dependend on the bit values.

a	b	state after step 4
0	0	$\frac{1}{\sqrt{2}}(\|00\rangle + \|10\rangle)$
0	1	$\frac{1}{\sqrt{2}}(\|11\rangle + \|01\rangle)$
1	0	$\frac{1}{\sqrt{2}}(\|00\rangle - \|10\rangle)$
1	1	$\frac{1}{\sqrt{2}}(\|11\rangle - \|01\rangle)$

Figure 3.3: How the states would look like after Bob received Alice's qubit A and performed a CNOT operation on his qubit B with A as the control qubit.

Step 5: Now that Bob has performed a CNOT operation on his qubit with A as a control qubit, we are ready to take a look at the last unitary transformation, that will be

applied to the entangled state. In this step, Bob applies a Hadamard gate to the qubit A. Recall how the Hadamard gate acts on qubit states:

$$\begin{aligned} H_A \left|0\right\rangle_A &= \frac{1}{\sqrt{2}}(\left|0\right\rangle_A + \left|1\right\rangle_A) \\ H_A \left|1\right\rangle_A &= \frac{1}{\sqrt{2}}(\left|0\right\rangle_A - \left|1\right\rangle_A) \end{aligned} \tag{3.10}$$

Let's call the entangled state - after Bob performed the CNOT operation - $\left|\phi\right\rangle_{ab}$. With this logic, the entangled state for $a = b = 0$ would be $\left|\phi\right\rangle_{00}$. So let's study how the Hadamard gate changes the entangled state.

$$\begin{aligned} H_A \left|\phi\right\rangle_{00} &= \frac{1}{\sqrt{2}}(H_A \left|0\right\rangle_A \otimes \left|0\right\rangle_B + H_A \left|1\right\rangle_A \otimes \left|0\right\rangle_B) \\ &= \frac{1}{2}((\left|0\right\rangle_A + \left|1\right\rangle_A)) \otimes \left|0\right\rangle_B + (\left|0\right\rangle_A - \left|1\right\rangle_A) \otimes \left|0\right\rangle_B) \\ &= \frac{1}{2}(\left|00\right\rangle + \left|10\right\rangle + \left|00\right\rangle - \left|10\right\rangle) = \left|00\right\rangle \end{aligned} \tag{3.11}$$

$$\begin{aligned} H_A \left|\phi\right\rangle_{01} &= \frac{1}{\sqrt{2}}(H_A \left|1\right\rangle_A \otimes \left|1\right\rangle_B + H_A \left|0\right\rangle_A \otimes \left|1\right\rangle_B) \\ &= \frac{1}{2}((\left|0\right\rangle_A - \left|1\right\rangle_A)) \otimes \left|1\right\rangle_B + (\left|0\right\rangle_A + \left|1\right\rangle_A) \otimes \left|1\right\rangle_B) \\ &= \frac{1}{2}(\left|01\right\rangle - \left|11\right\rangle + \left|01\right\rangle + \left|11\right\rangle) = \left|01\right\rangle \end{aligned} \tag{3.12}$$

$$\begin{aligned} H_A \left|\phi\right\rangle_{10} &= \frac{1}{\sqrt{2}}(H_A \left|0\right\rangle_A \otimes \left|0\right\rangle_B + H_A \left|1\right\rangle_A \otimes \left|0\right\rangle_B) \\ &= \frac{1}{2}(\left|00\right\rangle + \left|10\right\rangle - \left|00\right\rangle + \left|10\right\rangle) = \left|10\right\rangle \end{aligned} \tag{3.13}$$

$$\begin{aligned} H_A \left|\phi\right\rangle_{11} &= \frac{1}{\sqrt{2}}(H_A \left|1\right\rangle_A \otimes \left|1\right\rangle_B + H_A \left|0\right\rangle_A \otimes \left|1\right\rangle_B) \\ &= \frac{1}{2}(\left|01\right\rangle - \left|11\right\rangle - \left|01\right\rangle - \left|11\right\rangle) = \left|11\right\rangle \end{aligned} \tag{3.14}$$

So we can conclude that applying the Hadamard gate on the qubit A gives us:

$$H \left|\phi\right\rangle_{ab} = \left|ab\right\rangle \tag{3.15}$$

Bob only needs to measure the entangled state and it will yield the bits as a result. This leads us to the last step.

Step 6: Since the compound system of Alice's and Bob's qubits is now in the state $|ab\rangle$, Bob only needs to measure that state. It will collaps into two classical states, which represent the encoded qubits.

Superdense Coding

Superdense coding is a quantum computing process, in which the information of two classical bits gets encoded into a single qubit, sent to a recipient and then decoded again. The superdense coding protocol basically *copies* the two classical bits (it copies the two classical states) and stores them in a qubit. At the end of the process, both Alice and Bob have a copy of the two bits.

Now that we worked through each of the steps of the superdense coding protocol, we can have a look at the quantum circuit diagram of that process (that you can see in Figure 3.4). The two classical bits are represented by double lines, since the measured state of a qubit can be seen as a classical bit. The encoding process is demonstrated by controlled-operations. Only if a=1, the phase flip gate will be applied and only if b=1, an additional NOT-gate is used for the qubit A. The qubit gets sent to Bob, who then decodes the message by using a CNOT and a Hadamard gate. In the end, both qubits get measured and yield the two encoded bits as a result.

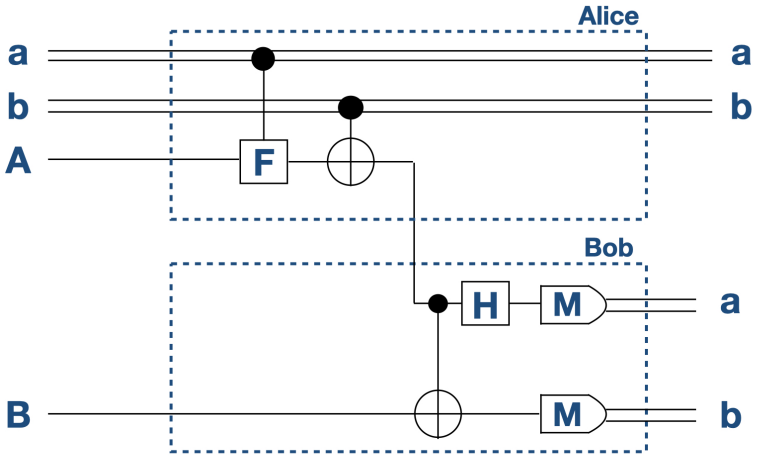


Figure 3.4: Quantum circuit diagram for the superdense coding protocol. The classical bits act like measured states of a qubit. To encode the bits into Alice's qubit, controlled-gates are used. To encode the first bit, a controlled phase flip gate is applied and to encode the second bit, a controlled-NOT gate is used. Bob decodes the qubit by using a controlled-NOT gate on his own qubit B with A as a control qubit and afterwards he applies the Hadamard gate to Alice's qubit. Measuring the outcome will yield the original two bits a and b.

3.2 Quantum Teleportation

In this last section we are going to study the *quantum teleportation protocol*. The teleportation of a quantum state can be seen as the converse to the superdense coding protocol we learned about in the last section. Recall for a second, how this protocol works again.
In superdense coding, the information of two classical bits gets encoded into a qubit, which is then transmitted via a *quantum channel*. The recipient at the other side of the channel receives the qubit and applies some quantum gates to recover the original two classical bits.
Quantum teleportation is different. Here, the state of a qubit gets encoded into two classical bits, which then get sent via a *classical channel* to the recipient, who recovers the original quantum state.
The difference to superdense coding - as we will see at the end of this section - is that a quantum state can not be copied without being destroyed. Other than superdense coding, at the end of the quantum teleportation protocol we will have only one version of the teleported qubit. The original one does not longer exist. This fact is also known as the *No-Cloning Theorem*, which we will proove later on.
For now, let's work through the steps of the teleportation protocol.
Just like in superdense coding, we start the procedure by stating we have to parties participating in the teleportation process - Alice and Bob. Alice and Bob both have qubits A and B and just like in the last section, we assume both qubits to be entangled and in the state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle_A |0\rangle_B + |1\rangle_A |1\rangle_B) \tag{3.16}$$

Alice has another qubit, which we will call T, that she wants to teleport to Bob. We assume, that the qubit T can be found in a generall superposition state of the form

$$|\phi\rangle = a\,|0\rangle_T + b\,|1\rangle_T \quad with\ \ a^2 + b^2 = 1 \tag{3.17}$$

The compound system of the three qubits $T \otimes A \otimes B$ is then in the state

$$\begin{aligned} |\phi\rangle \otimes |\psi\rangle &= (a\,|0\rangle_T + b\,|1\rangle_T)\frac{1}{\sqrt{2}}(|00\rangle_{AB} + |11\rangle_{AB}) \\ &= \frac{a}{\sqrt{2}}\,|0\rangle_T \otimes |00\rangle_{AB} + \frac{a}{\sqrt{2}}\,|0\rangle_T \otimes |11\rangle_{AB} + \frac{b}{\sqrt{2}}\,|1\rangle_T \otimes |00\rangle_{AB} + \frac{b}{\sqrt{2}}\,|1\rangle_T \otimes |11\rangle_{AB} \\ &= \frac{a}{\sqrt{2}}\,|000\rangle + \frac{a}{\sqrt{2}}\,|011\rangle + \frac{b}{\sqrt{2}}\,|100\rangle + \frac{b}{\sqrt{2}}\,|111\rangle \end{aligned} \tag{3.18}$$

Now that we are done with all the needed preparations, we can start studying the actual protocol by going through step by step.

Step 1: Alice performs a controlled-NOT operation of her qubit A while using T as a control qubit. The state of the compound system thus changes to:

$$\begin{aligned} M_{CNOT}^{(AT)} |\phi\rangle |\psi\rangle &= M_{CNOT}^{(AT)} \left(\frac{a}{\sqrt{2}} |000\rangle + \frac{a}{\sqrt{2}} |011\rangle + \frac{b}{\sqrt{2}} |100\rangle + \frac{b}{\sqrt{2}} |111\rangle \right) \\ &= \frac{a}{\sqrt{2}} |000\rangle + \frac{a}{\sqrt{2}} |011\rangle + \frac{b}{\sqrt{2}} |110\rangle + \frac{b}{\sqrt{2}} |101\rangle \end{aligned} \tag{3.19}$$

The first two terms of the above equation remain unchanged when the CNOT gate is applied, the last two terms flip the value of the qubit A.

Step 2: Now, Alice continues by performing a Hadamard transformation on the qubit to be teleported. Recall, how the Hadamard gate acts on qubits:

$$H |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad and \quad H |1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{3.20}$$

So the Hadamard gate acting on the qubit T will yield the following state for the compound system:

$$\begin{aligned} H_T |\phi\rangle |\psi\rangle &= \\ &= \frac{a}{\sqrt{2}} H_T |0\rangle_T |00\rangle_{AB} + \frac{a}{\sqrt{2}} H_T |0\rangle_T |11\rangle_{AB} + \\ &\quad \frac{b}{\sqrt{2}} H_T |1\rangle_T |10\rangle_{AB} + \frac{b}{\sqrt{2}} H_T |1\rangle_T |01\rangle_{AB} \\ &= \frac{a}{2}(|0\rangle_T + |1\rangle_T) |00\rangle_{AB} + \frac{a}{2}(|0\rangle_T + |1\rangle_T) |11\rangle_{AB} + \\ &\quad \frac{b}{2}(|0\rangle_T - |1\rangle_T) |10\rangle_{AB} + \frac{b}{2}(|0\rangle_T - |1\rangle_T) |01\rangle_{AB} \end{aligned} \tag{3.21}$$

Multiplying this out will give us:

$$\begin{aligned} H_T |\phi\rangle |\psi\rangle &= \\ \frac{a}{2} |000\rangle + \frac{a}{2} |100\rangle + \frac{a}{2} |011\rangle + \frac{a}{2} |111\rangle & \\ + \frac{b}{2} |010\rangle + \frac{b}{2} |110\rangle + \frac{b}{2} |001\rangle + \frac{b}{2} |101\rangle & \end{aligned} \tag{3.22}$$

We will rewrite the above equation and get the following as a result:

$$\begin{aligned}
&\frac{a}{2}|00\rangle_{TA}|0\rangle_B + \frac{a}{2}|10\rangle_{TA}|0\rangle_B + \frac{a}{2}|01\rangle_{TA}|1\rangle_B + \frac{a}{2}|11\rangle_{TA}|1\rangle_B \\
+&\frac{b}{2}|01\rangle_{TA}|0\rangle_B + \frac{b}{2}|11\rangle_{TA}|0\rangle_B + \frac{b}{2}|00\rangle_{TA}|1\rangle_B + \frac{b}{2}|10\rangle_{TA}|1\rangle_B \\
&= \frac{1}{2}|00\rangle_{TA}(a|0\rangle_B + |1\rangle_B) + \frac{1}{2}|10\rangle_{TA}(a|0\rangle_B - |1\rangle_B) \\
&+\frac{1}{2}|01\rangle_{TA}(a|1\rangle_B + |0\rangle_B) + \frac{1}{2}|11\rangle_{TA}(a|1\rangle_B - |0\rangle_B)
\end{aligned} \tag{3.23}$$

Step 3: After applying a CNOT and a Hadamard gate on the two qubits A and T, Alice measures both qubits. A measurement of the two qubits would result in one of the following four states with equal probability:

$$\begin{aligned}
&|00\rangle(a|0\rangle_B + b|1\rangle_B) \\
&|10\rangle(a|0\rangle_B - b|1\rangle_B) \\
&|01\rangle(a|1\rangle_B + b|0\rangle_B) \\
&|11\rangle(a|1\rangle_B - b|0\rangle_B)
\end{aligned} \tag{3.24}$$

Alice now uses the information, into which state the compound system collapsed, to encode it into two classical bits b_A and b_T, with $T \otimes A$ being in the measured state $|b_T b_A\rangle$. The states after Alice's measurement together with the corresponding bit values are shown in Figure 3.5 .

state after measurement	b_T	b_A
$\|00\rangle(a\|0\rangle_B + b\|1\rangle_B)$	0	0
$\|01\rangle(a\|1\rangle_B + b\|0\rangle_B)$	0	1
$\|10\rangle(a\|0\rangle_B - b\|1\rangle_B)$	1	0
$\|11\rangle(a\|1\rangle_B - b\|0\rangle_B)$	1	1

Figure 3.5: After measuring the qubits A and T, the compound state of the three qubits will be in one of the four states above, each with equal probability. Which state the system collapsed into determines the values of the two classical bits b_A and b_T.

Step 4: Now Alice is ready to actually "teleport" the qubit T. She does so by sending the two classical bits b_T and b_A via a classical channel to Bob.

Step 5: Now that Bob is in posession of the two classical bits he is able to recover the state of the original qubit T. Notice that at this point, T no longer is in its original state, since it has been measured. All the information about the original state of T is contained in the two bits b_T and b_A. The first operation, that Bob will perform now, is a NOT operation on his qubit, but only if $b_A = 1$. So the corresponding states change in the following way:

$$\begin{aligned} |01\rangle (a|1\rangle + b|0\rangle) &\rightarrow |01\rangle (a|0\rangle + b|1\rangle) \\ |11\rangle (a|1\rangle - b|0\rangle) &\rightarrow |11\rangle (a|0\rangle - b|1\rangle) \end{aligned} \tag{3.25}$$

In Figure 3.6 you can see, how the states would look like after Bob's first operation.

state after Bob's first operation	b_T	b_A
$\|00\rangle (a\|0\rangle + b\|1\rangle)$	0	0
$\|01\rangle (a\|0\rangle + b\|1\rangle)$	0	1
$\|10\rangle (a\|0\rangle - b\|1\rangle)$	1	0
$\|11\rangle (a\|0\rangle - b\|1\rangle)$	1	1

Figure 3.6: How the state of the compound system transformed after Bob's first operation on his qubit.

Step 6: In the last step, Bob uses a phase flip gate, but only if the second qubit $b_T = 1$. Now, no matter what values the two transported classical bits have, Bob's qubit can be found in the state $a|0\rangle + |1\rangle$.
The teleportation protocol has transformed the state of Bob's qubit B into the original state of the qubit T. Just like a copy machine, where the qubit B is the blank piece of paper, on which we like to print the qubit T. But this is not possible without measuring T, thus destroying its original state.
The whole protocol can (just like the superdense coding protocol) be visualized in a quantum circuit diagram, as you can see in Figure 3.7. The classical bits b_T and b_A act as control qubits for Bob, when he recovers the original state of the qubit to be teleported.

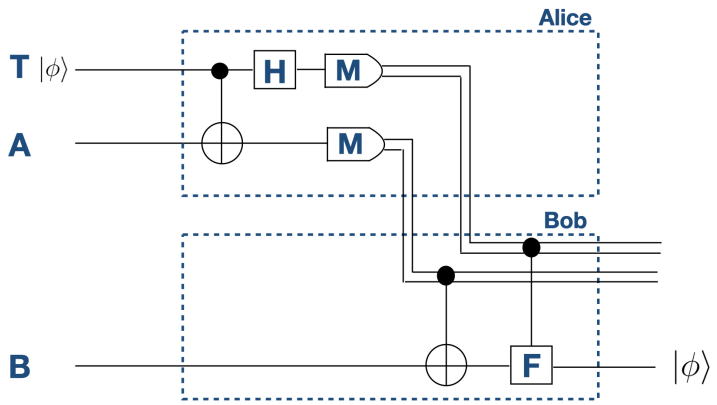


Figure 3.7: Quantum circuit diagram of the quantum teleportation protocol.

So in the end, Bob does not really have a copy of the qubit T, but the only existent version of it. Interestingly, that is not a problem of the teleportation protocol. One can say in general that no quantum superposition state can be copied, without being destroyed in the process of copying. This fact is known as the so-called *No-Cloning Theorem* and in this last section to come we will prove that theorem.

Quantum Teleportation

Quantum Teleportation is a quantum protocol, which can be seen as the converse of the superdense coding protocol. In teleportation, the state of a qubit is encoded into two classical bits. These two bits get transported via a classical channel and afterwards decoded into a quantum state again. In order for this protocol to work, the sender and the recipient have to share an entangled state. In the process of encoding the information the qubit to be teleported needs to get measured and thus destroyed. Until the two classical bits get decoded again by the recipient, the original state of the qubit does not longer exist. This is not a problem of the protocol, but a fundamental law of nature: Quantum states cannot be copied without being destroyed.

3.2.1 The No-Cloning Theorem

To prove the *No-Cloning Theorem*, we will use a method, that is called *proof by contradiction*: we will assume, that it is in fact possible to copy a superposition quantum state but going through logical steps will yield the result that it's not possible at all.
LeT's assume there exists a way to copy a quantum state without actually destroying it. Mathematically, this would be expressed by a unitary transformation acting on a compound system of two qubits.
Say we have two qubits, one of which can be found in the state $|\psi\rangle$ and the other one in the classical state $|0\rangle$. We want to copy the state $|\psi\rangle$ such that the other qubit transforms its state to $|\psi\rangle$ as well while the original qubit stays in its state. The qubit, which is in the classical state $|0\rangle$, can be seen as the blank piece of paper, on which the information of another quantum state will be imprinted.
In the form of an equation, this would look like this

$$U(|\psi\rangle\,|0\rangle) = |\psi\rangle\,|\psi\rangle \tag{3.26}$$

with U being a unitary transformation, that we don't know anything about except for that it copies the state of the first qubit onto the second one. A unitary transformation, that is able to copy a quantum state is called a *quantum copy machine*. And the No-Cloning theorem states, that no-quantum copy machine can exist.

Quantum copy machine and the No-Cloning theorem

A *quantum copy machine* is a unitary transformation (for example a combination of different unary and CNOT gates), which acts on two qubits and is able to copy the state of the first qubit and imprint it on the second one without destroying the first one. The *No-Cloning Theorem* on the other hand states that such a unitary transformation can not exist: there will alway be only one version of a quantum state.

Let's prove the No-Cloning theorem. Let $|\psi\rangle$ be the state we like to copy. Since it is the state of a qubit, we can write it as a linear combination

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle \quad with \quad a^2 + b^2 = 1 \tag{3.27}$$

with a and b being arbitrary probability amplitudes.
We now take a look at how a quantum copy machine would act on the above linear combination:

$$\begin{aligned} U(|\psi\rangle\,|0\rangle) &= ((a\,|0\rangle + b\,|1\rangle)\,|0\rangle) \\ &= a \cdot U(|0\rangle\,|0\rangle) + b \cdot U(|1\rangle\,|0\rangle) \\ &= a\,|00\rangle + b\,|11\rangle \end{aligned} \tag{3.28}$$

But on the other side, we can also write

$$\begin{aligned} U(|\psi\rangle\,|0\rangle) &= |\psi\rangle\,|\psi\rangle \\ = (a\,|0\rangle + b\,|1\rangle)(a\,|0\rangle + b\,|1\rangle) &= a^2\,|00\rangle + ab\,|01\rangle + ab\,|10\rangle + b^2\,|11\rangle \end{aligned} \tag{3.29}$$

Those two results that we get for the transformation done by the quantum copy machine, have to equal each other:

$$a^2\,|00\rangle + ab\,|01\rangle + ab\,|10\rangle + b^2\,|11\rangle \stackrel{!}{=} a\,|00\rangle + b\,|11\rangle \tag{3.30}$$

And this equation is a contradiction, because it can only be true, if $a = b = 0$, otherwise there is no way for this equation to work out. This contradiction lets us know, that such a unitary transformation can not exist.