UNIVERSITÄT GRAZ
UNIVERSITY OF GRAZ

Tina Radkohl

Quantum Teleportation and Superdense Coding

A case study with IBM's quantum computer

Bachelor Thesis

University of Graz, Austria
Institute of Physics
Supervisor: Hohenester, Ulrich, Ao.Univ.-Prof. Mag. Dr.rer.nat.
Graz, July 2021

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, July 2021

Tina Radkohl

Abstract

The concepts of quantum teleportation and superdense coding are presented, and the mathematical descriptions are provided. Moreover, both protocols are tested on IBM's Quantum Simulators and successfully implemented on IBM's Quantum Computer. Comparing the protocols shows that both rely on quantum entanglement and consist of similar building blocks. However, the order of these blocks and the data input differ, leading to contrasting concepts. Quantum teleportation uses two classical bits to transmit one qubit, while superdense coding uses one qubit to transmit two classical bits. Finally, some fields of application are outlined.

Contents

1 Introduction

In the autumn of 2018, the Quantum FET[1] Flagship was launched by the European Commission. This is a large-scale research initiative that provides €1 billion in funding within ten years. [1]

> The EU's long-term vision is the development of the Quantum Internet all over Europe: quantum computers, simulators, and sensors would be interconnected via quantum networks distributing information and quantum resources. [2]

In the ramp-up phase (until the end of 2021), 20 projects in 5 different fields explore the feasibility of building the required quantum communication infrastructure [2]. Moreover, the *Midterm Report of the Quantum Technologies Flagship* has shown remarkable progress that has been made in these projects and the synergies between them [3].

Thus, the future for European quantum research looks bright. However, to develop more complex concepts, it is essential to understand the fundamental concepts fully.

In this thesis, two quantum protocols from the early 90s are examined.
The quantum teleportation and superdense coding protocol were first proposed by Bennett et al. and Bennett and Wiesner, respectively [4, 5]. These concepts harness the power of entanglement and easily outperform everything that is classically achievable. Furthermore, both protocols proved to be enormously versatile in several fields of application and as building blocks for more complex protocols. Thus, playing a vital role not only in the past but also in the future of quantum technology and quantum information theory.

In the following, the basic concepts required to understand quantum teleportation and superdense coding are outlined. These include the used qubit bases and quantum gates, the projective measurement and measurement principles, and the no-cloning theorem.

In sections two and three, quantum teleportation and superdense coding are demonstrated, respectively. For this purpose, the concepts are explained, and a mathematical description is given. Furthermore, the protocols are tested on IBM's Quantum Simulators and implemented on IBM's Quantum Computer.

In the fourth section, a comparison between quantum teleportation and superdense coding is presented. Apart from the similarities and differences between the protocols,

[1] Future and Emerging Technologies

a short overview of their fields of application is given.

Finally, the content of the most important topics will be summarized.

1.1 Qubit Bases

Nielsen and Chuang specify in *Quantum Computation and Quantum Information* [6] that a quantum bit, or qubit, is the fundamental building block of a quantum computer, as a bit is of a classical computer. Bits and qubits can be any two-state system with two mutually exclusive states (e.g., different energy levels). However, a qubit can not only be in one of two states but also in a superposition:

$$|\text{qubit}\rangle = a\,|\text{state a}\rangle + b\,|\text{state b}\rangle\ . \tag{1}$$

Therefore, the state of a single qubit is a vector in a two-dimensional complex vector space, where the exact representation depends on the chosen basis of the vector space. The most relevant bases are given below.

Computational Basis

The z-direction is conventionally chosen to be privileged. Accordingly, the z-basis is called the computational basis and the orthonormal basis states are labeled $|0\rangle$ and $|1\rangle$, so that

$$\begin{aligned} |0\rangle &= |+z\rangle \\ |1\rangle &= |-z\rangle\ . \end{aligned} \tag{2}$$

A general qubit state in the computational basis is therefore

$$|\Psi\rangle = a\,|0\rangle + b\,|1\rangle\ . \tag{3}$$

Furthermore, the computational basis can be generalized for n qubits using the tensor product. For example, the computational basis for two qubits is given by

$$\Big\{\,|00\rangle\,,|01\rangle\,,|10\rangle\,,|11\rangle\,\Big\}_{AB}\,, \tag{4}$$

whereas $|00\rangle_{AB}$ is the short notation of the tensor product $|0\rangle_A \otimes |0\rangle_B$ and so forth. The indices of the state vectors denote the associated qubit(s), and their order is important because the tensor product is not commutative. Moreover, if all state vectors inside

parenthesis have the same order of indices, the indices are moved outside the parenthesis to increase readability.

X Basis

A special case of superposition in the computational basis is equivalent to the orthonormal x-basis states $\{|+x\rangle, |-x\rangle\}$, often also denoted as $|+\rangle$ and $|-\rangle$, respectively.

$$\begin{aligned} |+\rangle &= \frac{1}{\sqrt{2}}\Big(|0\rangle + |1\rangle \Big) \\ |-\rangle &= \frac{1}{\sqrt{2}}\Big(|0\rangle - |1\rangle \Big) \end{aligned} \tag{5}$$

Here, the state vectors in the computational basis have a 50% probability of being measured in state $|0\rangle$ and a 50% probability of being measured in state $|1\rangle$.

Bell Basis

The Bell states, or EPR pairs (after Einstein, Podolsky and Rosen), are defined as

$$\begin{aligned} \big|\Psi^+\big\rangle_{AB} &= \frac{1}{\sqrt{2}}\Big(|00\rangle + |11\rangle \Big)_{AB} \\ \big|\Psi^-\big\rangle_{AB} &= \frac{1}{\sqrt{2}}\Big(|00\rangle - |11\rangle \Big)_{AB} \\ \big|\Phi^+\big\rangle_{AB} &= \frac{1}{\sqrt{2}}\Big(|01\rangle + |10\rangle \Big)_{AB} \\ \big|\Phi^-\big\rangle_{AB} &= \frac{1}{\sqrt{2}}\Big(|01\rangle - |10\rangle \Big)_{AB} \end{aligned} \tag{6}$$

and are chosen as the Bell basis states. These composite states cannot be separated into a product of states of their component systems. In other words, the state of one qubit of the EPR pair cannot be described independently of the other qubit. This is called an "entangled" state[2].

A simple example demonstrates the correlation between two entangled qubits: If qubit A of a system in the Bell state $|\Psi^+\rangle_{AB}$ is measured in the computational basis, it has a 50% probability to be in state $|0\rangle$ and a 50% probability to be in state $|1\rangle$. However, if qubit A is measured to be in state $|0\rangle$, qubit B has a 100% probability to also be in state $|0\rangle$. Similarly, if qubit A is measured to be in state $|1\rangle$, qubit B has a 100% probability to also be in state $|1\rangle$.

[2]In fact, Bell states are maximally entangled. For details see [6].

1.2 Quantum Gates

To create a quantum circuit, it is imperative to define operators that transform the states of qubits. During the transformation process, the following two properties must be preserved: the linearity of the state space and the norm of the state (probability). This is achieved using unitary operators, which are called "gates". Another property of unitary operators, and therefore of gates, is reversibility. Consequently, because measurements are irreversible, a measurement of a qubit state cannot be represented as a quantum gate.[7]

The following qubit gates [8, 9] will be used to implement the quantum teleportation and the superdense coding circuit.

1.2.1 Single Qubit Gates

X Gate

The X gate is the quantum computational equivalent of the classical NOT gate. It can be represented by the Pauli-X matrix and corresponds to a rotation by π around the x-axis, resulting in a bit flip.

$$\begin{aligned} X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} &= |0\rangle\langle 1| + |1\rangle\langle 0| \\ X|0\rangle &= |1\rangle \\ X|1\rangle &= |0\rangle \end{aligned} \tag{7}$$

Z Gate

Similarly, the Z gate can be represented by the Pauli-Z matrix and corresponds to a rotation by π around the z-axis, resulting in a phase flip.

$$\begin{aligned} Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} &= |0\rangle\langle 0| - |1\rangle\langle 1| \\ Z|0\rangle &= |0\rangle \\ Z|1\rangle &= -|1\rangle \end{aligned} \tag{8}$$

Hadamard Gate

The Hadamard gate, or H gate, is different from the Pauli gates because it allows the

transition of a qubit in and out of superposition regarding the computational basis (the z-basis). This can also be interpreted as changing from the z-basis $\{|0\rangle, |1\rangle\}$ to the x-basis $\{|+\rangle, |-\rangle\}$, and vice versa.

$$\begin{aligned} H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} &= \frac{1}{\sqrt{2}} \Big(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1| \Big) \\ H|0\rangle = |+\rangle, H|+\rangle &= |0\rangle \\ H|1\rangle = |-\rangle, H|-\rangle &= |1\rangle \end{aligned} \tag{9}$$

1.2.2 Multiple Qubit Gates

CNOT Gate

The CNOT gate, or controlled-X gate, is a conditional two-qubit gate with the input qubits called "control" and "target". This gate applies the X gate to the target if the control is in state $|1\rangle$ and the identity gate ($I = |0\rangle\langle 0| + |1\rangle\langle 1|$) if the control is in state $|0\rangle$. The operation is comparable to the classical XOR gate with the control qubit as additional output.

$$\begin{aligned} \mathrm{CNOT(c,\,t)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} &= \Big(|00\rangle\langle 00| + |10\rangle\langle 10| + |11\rangle\langle 01| + |01\rangle\langle 11| \Big)_{tc} \\ \mathrm{CNOT(c,\,t)}\,|00\rangle_{tc} &= |00\rangle_{tc} \\ \mathrm{CNOT(c,\,t)}\,|10\rangle_{tc} &= |10\rangle_{tc} \\ \mathrm{CNOT(c,\,t)}\,|01\rangle_{tc} &= |11\rangle_{tc} \\ \mathrm{CNOT(c,\,t)}\,|11\rangle_{tc} &= |01\rangle_{tc} \end{aligned} \tag{10}$$

Controlled-Z Gate

Expectedly, the controlled-Z gate (CZ) is a conditional two-qubit gate that applies the Z gate to the target if the control is in state $|1\rangle$, while the control qubit is output

unaffected.

$$\begin{aligned}
\mathrm{CZ(c,\,t)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} &= \Big(|00\rangle\langle 00| + |10\rangle\langle 10| + |01\rangle\langle 01| - |11\rangle\langle 11| \Big)_{tc} \\
\mathrm{CZ(c\ ,t)}\,|00\rangle_{tc} &= |00\rangle_{tc} \\
\mathrm{CZ(c,\,t)}\,|10\rangle_{tc} &= |10\rangle_{tc} \\
\mathrm{CZ(c,\,t)}\,|01\rangle_{tc} &= |01\rangle_{tc} \\
\mathrm{CZ(c,\,t)}\,|11\rangle_{tc} &= -\,|11\rangle_{tc}
\end{aligned} \tag{11}$$

1.3 Measurement

Nielsen and Chuang describe the projective measurement as a special case of Postulate 3 (quantum measurement) in quantum mechanics because it requires the measurement operators to also be orthogonal projectors. This leads to the following definition [6]:

> **Projective measurement:** A projective measurement is described by an *observable, M,* a Hermitian operator on the state space of the system being observed. The observable has a spectral decomposition,
>
> $$M = \sum_m m P_m \,, \tag{12}$$
>
> where P_m is the projector onto the eigenspace of M with eigenvalue m. The possible outcomes of the measurement correspond to the eigenvalues, m, of the observable. Upon measuring the state $|\Psi\rangle$, the probability of getting result m is given by
>
> $$p(m) = \langle\Psi|P_m|\Psi\rangle \ . \tag{13}$$
>
> Given that outcome m occurred, the state of the quantum system immediately after the measurement is
>
> $$\frac{P_m\,|\Psi\rangle}{\sqrt{p(m)}} \,. \tag{14}$$

Two more principles are relevant for measuring a quantum circuit:

> **Principle of deferred measurement:** Measurements can always be moved from an intermediate stage of a quantum circuit to the end of the circuit; if the measurement results are used at any stage of the circuit then the classically controlled operation can be replaced by conditional quantum operations.[6]
>
> **Principle of implicit measurement:** Without loss of generality, any unterminated quantum wires (qubits which are not measured) at the end of quantum circuit may be assumed to be measured.[6]

1.4 No-Cloning Theorem

Classical bits are copied using the XOR operation. Thus, it stands to reason that a qubit can be copied using the quantum computational counterpart, the CNOT gate. However, an example shows that this is not the case. [6]

An unknown quantum state $|\Psi\rangle_A = \Big(a\,|0\rangle + b\,|1\rangle\Big)_A$ shall be copied onto the state $|0\rangle_B$. Therefore, the combined state of the system before the copying process is

$$|0\rangle_B\,|\Psi\rangle_A = |0\rangle_B\,\Big(a\,|0\rangle + b\,|1\rangle\Big)_A = \Big(a\,|00\rangle + b\,|01\rangle\Big)_{BA}. \tag{15}$$

A CNOT gate with qubit A as control and qubit B as target is applied, yielding

$$\text{CNOT(A, B)}\Big(a\,|00\rangle + b\,|01\rangle\Big)_{BA} = \Big(a\,|00\rangle + b\,|11\rangle\Big)_{BA}. \tag{16}$$

For the copying process to be successful, this result must be equivalent to

$$\begin{aligned} |\Psi\rangle_B\,|\Psi\rangle_A &= \Big(a\,|0\rangle + b\,|1\rangle\Big)_B\Big(a\,|0\rangle + b\,|1\rangle\Big)_A \\ &= \Big(a^2\,|00\rangle + ab\,|01\rangle + ba\,|10\rangle + b^2\,|11\rangle\Big)_{BA}. \end{aligned} \tag{17}$$

This is not the case for arbitrary qubit states, as a comparison of coefficients shows. Equality is only given if one coefficient is zero and the other coefficient is one. Hence, only the states $|0\rangle_A$ and $|1\rangle_A$ can be copied.

Generalizing this example shows that there is no unitary operator that can copy an arbitrary unknown qubit. This is called the no-cloning theorem and is proven using the

properties of unitary operators. All unitary operators are linear, whereas the copying process is not; and unitary operators preserve the norm, which is only true if the copied states are orthogonal. Thus, a qubit copier can only copy orthogonal states in a certain basis but not arbitrary qubits. [10]

2 Quantum Teleportation

First, the concept of quantum teleportation is explained and a mathematical description is provided. Afterward, a quantum circuit is created to simulate the quantum teleportation protocol. Finally, the circuit is adapted and executed on one of IBM's quantum computers.

This section mainly follows IBM Textbook [11], Bennett et al. [4], and Nielsen and Chuang [6]. The programs are written in Python, using IBM's Qiskit [12] and the QuTiP package [13]. The key parts of the program are shown in sections 2.2 and 2.3.

2.1 Concept

The concept of "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels" was first described in 1993 by Bennett et al. [4]. As stated in their paper, the goal is to teleport an unknown quantum state $|\Psi\rangle$ from one system, "Alice", to another, "Bob". This is achieved in three steps (see figure 1):
First, an EPR pair is created and shared between Alice and Bob so that each hold one qubit of the pair. This is crucial because the entanglement of the qubits is what allows the protocol to work. In addition to her qubit of the EPR pair, Alice holds the qubit in state $|\Psi\rangle$.
Second, Alice takes a joint measurement on her two qubits. Doing so, Alice does not get any information about the unknown state $|\Psi\rangle$ (which is destroyed in the process[3]) but obtains a random and purely classical result, which she sends to Bob.
Third, due to the entanglement of the EPR pair that Alice and Bob share, Bob can now use the information sent by Alice to project his qubit into the state $|\Psi\rangle$, thus completing the teleportation.

It is important to note that Alice's measurement only yields a useful result if the orthonormal set $|\Psi\rangle$ belongs to is known [4]. This will be readily seen in the mathematical description of the teleportation protocol below.

[3]Thus, the teleportation protocol does not violate the no-cloning theorem; see section 1.4.

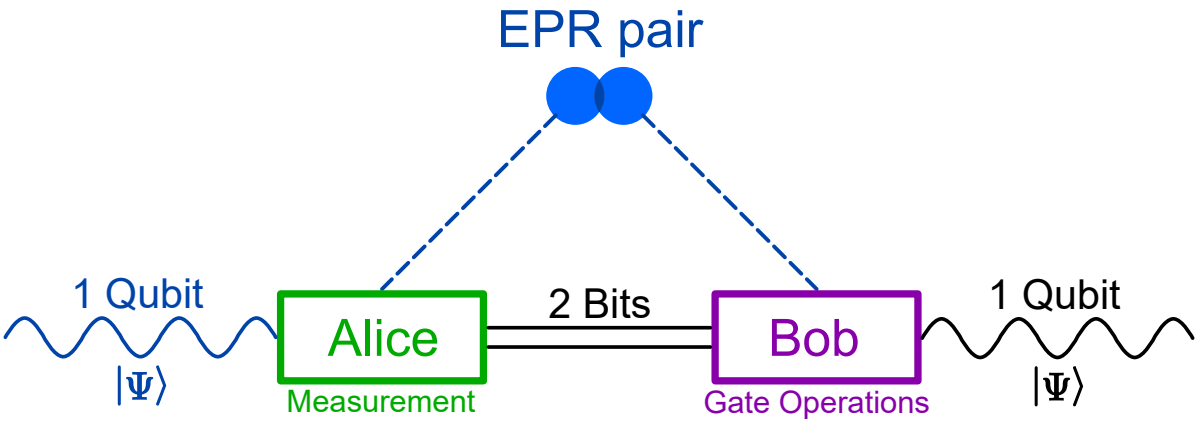


Figure 1: Concept of quantum teleportation.
Blue: Preparation of EPR pair and the state $|\Psi\rangle$ to be teleported.
Green: Alice takes a joint measurement of $|\Psi\rangle$ and her qubit of the entangled pair. The classical result (2 bits) is sent to Bob.
Purple: Bob applies the corresponding gates to his qubit to reconstruct $|\Psi\rangle$.

2.1.1 Mathematical description

To implement the quantum teleportation protocol, three qubits are required. These are labeled B (Bob), A (Alice), and S (system from which $|\Psi\rangle$ is teleported).
It must be emphasized that the significance of the qubits follows the qubit order in Qiskit, which in turn follows the standard programming convention with the most significant (qu)bit on the left: $|q_2 q_1 q_0\rangle$.

Step 1: Preparation

All qubits start out in state $|0\rangle$. From there, $|\Psi\rangle$ is initialized in an arbitrary state

$$|\Psi\rangle_S = a\,|0\rangle_S + b\,|1\rangle_S \tag{18}$$

and Alice's and Bob's qubits are entangled. This is achieved using a Hadamard gate on qubit A, and then a CNOT gate with qubit A as control and qubit B as target.

$$
\begin{aligned}
&CNOT(A,B)\Big[I_B \otimes H_A\Big]\Big(\,|0\rangle_B \otimes |0\rangle_A\,\Big) = \\
&= CNOT(A,B)\Big[\Big(\,|0\rangle\langle 0| + |1\rangle\langle 1|\,\Big)_B \\
&\qquad\qquad \otimes \frac{1}{\sqrt{2}}\Big(\,|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|\,\Big)_A\Big]\,|00\rangle_{BA} = \\
&= CNOT(A,B)\Big[\frac{1}{\sqrt{2}}\Big(\,|00\rangle\langle 00| + |00\rangle\langle 01| + |01\rangle\langle 00| - |01\rangle\langle 01| \\
&\qquad\qquad + |10\rangle\langle 10| + |10\rangle\langle 11| + |11\rangle\langle 10| - |11\rangle\langle 11|\,\Big)_{BA}\Big]\,|00\rangle_{BA} = \\
&= CNOT(A,B)\Big[\frac{1}{\sqrt{2}}\Big(\,|00\rangle + |01\rangle\,\Big)_{BA}\Big] = \\
&= \Big(\,|00\rangle\langle 00| + |10\rangle\langle 10| + |11\rangle\langle 01| + |01\rangle\langle 11|\,\Big)_{BA}\Big[\frac{1}{\sqrt{2}}\Big(\,|00\rangle + |01\rangle\,\Big)_{BA}\Big] = \\
&= \frac{1}{\sqrt{2}}\Big(\,|00\rangle + |11\rangle\,\Big)_{BA} = \big|\Phi^+\big\rangle_{BA}
\end{aligned}
\tag{19}
$$

This state is one of the Bell states (see equation 6); hence, the two qubits are indeed entangled.

After this preparation, the total state of the system is

$$
\begin{aligned}
\big|\Phi^+\big\rangle_{BA} \otimes |\Psi\rangle_S &= \frac{1}{\sqrt{2}}\Big(\,|00\rangle + |11\rangle\,\Big)_{BA} \otimes \Big(a\,|0\rangle + b\,|1\rangle\,\Big)_S = \\
&= \Big(\frac{a}{\sqrt{2}}\,|000\rangle + \frac{a}{\sqrt{2}}\,|110\rangle + \frac{b}{\sqrt{2}}\,|001\rangle + \frac{b}{\sqrt{2}}\,|111\rangle\,\Big)_{BAS} \equiv |\varphi\rangle_{BAS}\ ,
\end{aligned}
\tag{20}
$$

which shall be called $|\varphi\rangle_{BAS}$.

Step 2: Measurement

Now Alice takes a joint measurement of her two qubits A and S. Mathematically it would be the easiest to rearrange $|\varphi\rangle_{BAS}$ to

$$
\begin{aligned}
|\varphi\rangle_{BAS} &= \Big(a\,|0\rangle + b\,|1\rangle\,\Big)_B \otimes \big|\Phi^+\big\rangle_{AS} + \Big(a\,|0\rangle - b\,|1\rangle\,\Big)_B \otimes \big|\Phi^-\big\rangle_{AS} \\
&\quad + \Big(a\,|1\rangle + b\,|0\rangle\,\Big)_B \otimes \big|\Psi^+\big\rangle_{AS} + \Big(a\,|1\rangle - b\,|0\rangle\,\Big)_B \otimes \big|\Psi^-\big\rangle_{AS} = \\
&= I_B\,|\Psi\rangle_B \otimes \big|\Phi^+\big\rangle_{AS} + Z_B\,|\Psi\rangle_B \otimes \big|\Phi^-\big\rangle_{AS} \\
&\quad + X_B\,|\Psi\rangle_B \otimes \big|\Psi^+\big\rangle_{AS} + Z_B X_B\,|\Psi\rangle_B \otimes \big|\Psi^-\big\rangle_{AS}
\end{aligned}
\tag{21}
$$

and let Alice measure in the Bell basis. Equation 21 also shows, which gates Bob needs to apply to reconstruct the quantum state $|\Psi\rangle$, depending on the outcome of Alice's

measurement.

Regardless, it is not yet possible to take any measurements on IBM's quantum devices in a basis other than the computational one [11]. Thus, Alice maps the Bell basis states onto the computational basis states of two qubits: $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. For that, Alice first applies a CNOT gate with S as control and A as target and afterward a Hadamard gate on qubit S [14].
For all dimensions in this calculation to be the same, an identity gate (I) is used on non-concerned qubits. Hence, Alice's gate operations can be written as

$$\Big[I_B \otimes I_A \otimes H_S\Big]\Big[I_B \otimes CNOT(S,A)\Big]\,|\varphi\rangle_{BAS} \tag{22}$$

and are calculated analogously to equation 19.

However, knowing the effect of the CNOT gate[4], it can be readily applied to Alice's qubits. Thus, changing the state of the total system to

$$\Big[I_B \otimes I_A \otimes H_S\Big]\Big(\frac{a}{\sqrt{2}}\,|000\rangle + \frac{a}{\sqrt{2}}\,|110\rangle + \frac{b}{\sqrt{2}}\,|011\rangle + \frac{b}{\sqrt{2}}\,|101\rangle\Big)_{BAS} \tag{23}$$

which is rearranged to

$$\Big[I_B \otimes I_A \otimes H_S\Big]\Big[\Big(|00\rangle + |11\rangle\Big)_{BA} \otimes \frac{a}{\sqrt{2}}\,|0\rangle_S + \Big(|01\rangle + |10\rangle\Big)_{BA} \otimes \frac{b}{\sqrt{2}}\,|1\rangle_S \tag{24}$$

by factoring out the S qubit of the total state of the system.

Now, the Hadamard gate[5] can be easily applied to the S qubit, yielding

$$\Big(|00\rangle + |11\rangle\Big)_{BA} \otimes \frac{a}{2}\Big(|0\rangle + |1\rangle\Big)_S + \Big(|01\rangle + |10\rangle\Big)_{BA} \otimes \frac{b}{2}\Big(|0\rangle - |1\rangle\Big)_S. \tag{25}$$

[4]The CNOT or controlled-X gate applies the X gate to the target if the control is in state $|1\rangle$. $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$ (see equation 10)
[5]$H|0\rangle = |+\rangle$ and $H|1\rangle = |-\rangle$ (see equation 9)

The result is rearranged to

$$\begin{aligned}
&\frac{1}{2}\Big[\Big(a\,|0\rangle + b\,|1\rangle\Big)_B \otimes |00\rangle_{AS} + \Big(a\,|0\rangle - b\,|1\rangle\Big)_B \otimes |01\rangle_{AS} \\
&\quad + \Big(a\,|1\rangle + b\,|0\rangle\Big)_B \otimes |10\rangle_{AS} + \Big(a\,|1\rangle - b\,|0\rangle\Big)_B \otimes |11\rangle_{AS}\Big] = \\
&= \frac{1}{2}\Big[I_B\,|\Psi\rangle_B \otimes |00\rangle_{AS} + Z_B\,|\Psi\rangle_B \otimes |01\rangle_{AS} \\
&\qquad + X_B\,|\Psi\rangle_B \otimes |10\rangle_{AS} + Z_B X_B\,|\Psi\rangle_B \otimes |11\rangle_{AS}\Big]
\end{aligned} \tag{26}$$

as was done in equation 21, enabling Alice to take the joint measurement of both her qubits in the computational basis. It is important to note that the probability of all four outcomes is equal. Therefore, the measurement result does not allow to draw any conclusions about the state $|\Psi\rangle$ to be teleported. In fact, at this stage of the teleportation protocol, the state to be teleported is destroyed and is no longer in Alice's possession.

Step 3: Gate Operations

The information required for Bob to reconstruct $|\Psi\rangle$ is stored in the measurement result as can be seen in equation 26 and table 1. For example, if Alice measures her qubits to be in state $|11\rangle_{AS}$, Bob must apply an X gate followed by a Z gate to his qubit. After Bob used the appropriate gates, his qubit is in the teleported state $|\Psi\rangle$. This is shown during the statevector simulation in section 2.2.1.

Table 1: Gate operations Bob must apply to the state of his qubit to reconstruct $|\Psi\rangle$, depending on Alice's measurement outcome. [cf. 11]

Alice's Result	Bob's State	Gate Operation
$\|00\rangle_{AS}$	$a\,\|0\rangle + b\,\|1\rangle$	$I\,\|\Psi\rangle_B$
$\|01\rangle_{AS}$	$a\,\|0\rangle - b\,\|1\rangle$	$Z\,\|\Psi\rangle_B$
$\|10\rangle_{AS}$	$a\,\|1\rangle + b\,\|0\rangle$	$X\,\|\Psi\rangle_B$
$\|11\rangle_{AS}$	$a\,\|1\rangle - b\,\|0\rangle$	$ZX\,\|\Psi\rangle_B$

Adaptations of Step 3: Gate Operations for QASM Simulation

Since it is impossible for any measurement obeying the laws of quantum physics to yield a state in a superposition of the measurement basis, it is necessary to go one step further before testing the teleportation protocol on a quantum device. It must be emphasized that this step is only possible if the gates for initializing $|\Psi\rangle$ are known. Fortunately,

as all states are created using only quantum gates and since those are by definition reversible (see section 1.2), the gates for the initialization are (at least in principle) known. Thus, Bob has to apply the inverse of the initialization gates to his qubit. After that, the outcome of Bob's measurement on his qubit will invariably[6] be $|0\rangle$. This is confirmed through the QASM simulation in section 2.2.2.

Adaptations of Step 3: Gate Operations on a Quantum Computer

IBM updated their *ibmq_manhatten* backend in February 2021, enabling mid-circuit measurements [15]; however, this backend is not available for free account holders. In any case, real-time conditionals (i.e., if statements) are not yet possible [16], forcing the following adaptations to the quantum circuit.
Instead of measuring her qubits, Alice only maps them onto the computational basis. Then, the qubits are sent to Bob, who uses them as control qubits for his gate operations. He applies a CNOT gate with A as control first, followed by a controlled-Z gate with S as control, whereas Bob's qubit serves as target in both cases. This corresponds to using X or Z gates on Bob's qubit, depending on the information provided by the classical bits, in the original quantum teleportation protocol.

This adaptation takes away the benefit to teleport a quantum state via classical channels! However, it is a hardware limitation and will be worked out in the future.

The total state of the system, after Alice mapped the states onto the computational basis (see equation 25), serves as a starting point called $|\varphi\prime\rangle_{BAS}$.

$$\begin{aligned} |\varphi\prime\rangle_{BAS} \equiv \frac{1}{2}\Big[&a\,|000\rangle + a\,|110\rangle + a\,|001\rangle + a\,|111\rangle \\ &+ b\,|100\rangle + b\,|010\rangle - b\,|101\rangle - b\,|011\rangle\Big]_{BAS} \end{aligned} \tag{27}$$

Now Bob uses the control gates as described above.
First, he applies the CNOT gate[7] changing the total state to

$$\begin{aligned} \Big[CNOT(A,B)\otimes I_S\Big]\,|\varphi\prime\rangle_{BAS} = \frac{1}{2}\Big[&a\,|000\rangle + a\,|010\rangle + a\,|001\rangle + a\,|011\rangle \\ &+ b\,|100\rangle + b\,|110\rangle - b\,|101\rangle - b\,|111\rangle\Big]_{BAS}, \end{aligned} \tag{28}$$

[6] That is, as long as it is an ideal quantum device with zero error rate.
[7] The CNOT or controlled-X gate applies the X gate to the target if the control is in state $|1\rangle$. $X\,|0\rangle = |1\rangle$ and $X\,|1\rangle = |0\rangle$ (see equation 10)

which shall be called $|\varphi''_{BAS}\rangle$.

Next, Bob applies the controlled-Z gate[8], leading to

$$\begin{aligned}\Big[CZ(S,B)\otimes I_S\Big]\,|\varphi''\rangle_{BAS} = \frac{1}{2}\Big[&a\,|000\rangle + a\,|010\rangle + a\,|001\rangle + a\,|011\rangle \\ &+ b\,|100\rangle + b\,|110\rangle + b\,|101\rangle + b\,|111\rangle\Big]_{BAS}.\end{aligned} \tag{29}$$

Finally, the total state is rearranged by factoring out qubit B

$$\begin{aligned}&\frac{1}{2}\Big[\Big(a\,|0\rangle + b\,|1\rangle\Big)_B \otimes |00\rangle_{AS} + \Big(a\,|0\rangle + b\,|1\rangle\Big)_B \otimes |01\rangle_{AS} \\ &+ \Big(a\,|0\rangle + b\,|1\rangle\Big)_B \otimes |10\rangle_{AS} + \Big(a\,|0\rangle + b\,|1\rangle\Big)_B \otimes |11\rangle_{AS} = \\ &= |\Psi\rangle_B \otimes \frac{1}{2}\Big[\,|00\rangle + |01\rangle + |10\rangle + |11\rangle\Big]_{AS}.\end{aligned} \tag{30}$$

This result shows that Bob's qubit is invariably in state $|\Psi\rangle$, independent of Alice's qubits. Therefore, the adapatations are correct and the quantum teleportation protocol is successful.

2.2 IBM's Quantum Simulators

2.2.1 Statevector Simulator

IBM's Statevector Simulator calculates the state of a quantum system without destroying its superposition. Therefore, the simulation shows in which state the qubits of the quantum circuit would be in if they were not measured. More precisely, it "simulates a quantum circuit by computing the wavefunction of the qubit's statevector as gates and instructions are applied" [17].
Although this cannot be performed on a real quantum computer, the simulation is included here for two reasons. First, measuring qubits without destroying their superposition allows a unique illustration of what is happening inside the quantum circuit. And second, it verifies that the quantum circuit behaves as intended.

Before starting with the circuit, all required packages must be imported into a jupyter notebook and an empty circuit is created. To perform a state vector simulation, the circuit requires three qubits and two classical bits, which are combined into the quantum

[8] The controlled-Z gate applies the Z gate to the target if the control is in state $|1\rangle$.
$Z\,|0\rangle = |0\rangle$ and $Z\,|1\rangle = -\,|1\rangle$ (see equation 11)

circuit *QT_ circuit* as shown below. The third classical bit is commented out, which will be used for the QASM Simulation in the next section and can be ignored for now.

```
#-- setup ---------------------------------------------

# 3 qubits
qr0 = QuantumRegister(1, name="s")
qr1 = QuantumRegister(1, name="a")
qr2 = QuantumRegister(1, name="b")
# 2 classical bits
bit_z = ClassicalRegister(1, name="bit_z")
bit_x = ClassicalRegister(1, name="bit_x")
# classical bit for qasm simulation
#bit_res = ClassicalRegister(1, name="bit_res")

# create empty quantum circuit
QT_circuit = QuantumCircuit(qr0,qr1,qr2, bit_z, bit_x)#, bit_res)
```

Hereafter, functions are created for better readability:
All functions have a quantum circuit (*qc*) as an input parameter and generate gates for the circuit.

Step 1: Preparation

The first function *create_ init_ state* prepares the qubit state $|\Psi\rangle$ that will be teleported. It allows to either choose a specific state or to create a random state using the function argument *state*. Notably the function *Initialize()* first resets the qubit into state $|0\rangle$ and initializes it afterward. Therefore, the reversibility is lost and *Initialize()* is called an "instruction"[11] rather than a gate. This instruction is then added to the quantum circuit to act on the qubit in position *pos*. Lastly, the initialization and $|\Psi\rangle$ are returned to be available for later use.

```
def create_init_state(qc, pos, state):
    """initializes qubit in position 'pos' into 'state'
    use list for specific state or choose 'random'
    Note: qubit is reset to |0> first"""
    if isinstance(state, list):
        qubit = state
    if state == 'random':
        qubit = random_state(1)     # create random state
    initial = Initialize(qubit)     # reset and initialize
    initial.label = "Initialize"
```

```
    qc.append(initial, [pos])
    return qubit, initial
```

The second part of the preparation is a function (*entangle*) to create the EPR pair that Alice (a) and Bob (b) share. As shown in the mathematical description (see section 2.1.1) this is done by first applying a Hadamard gate (h), followed by a CNOT gate. The latter is called *cx()*, where the first argument is the control qubit and the second argument the target.
The command *barrier()* does not affect the qubits but adds a vertical line to the circuit, improving clarity when the circuit is drawn later on.

```
def entangle(qc, a, b):
    """creates a EPR pair (entangled) using qubits a and b"""
    qc.barrier()     # adds vertical line, no affect on qubits
    qc.h(a)          # Hadamard gate on 'a'
    qc.cx(a,b)       # CNOT with 'a' as control and 'b' as target
```

Step 2: Measurement

Alice's mapping of the Bell states onto the computational basis is performed by the function *alice_gates*. This is mandatory because measuring in the Bell basis is not yet available.

```
def alice_gates(qc, s, a):
    """maps Bell basis states onto
    computational basis states of two qubits"""
    qc.barrier()
    qc.cx(s, a)     # CNOT with 's' as control and 'a' as target
    qc.h(s)         # Hadamard gate on 's'
```

Afterward, the function *measure_and_send* measures Alice's qubits (s, a) in the computational basis. In the process, the result of qubit s is stored in the classical bit in position 0, called *bit_z*; whereas the result of measuring Alice's part of the EPR pair (qubit a) is stored in the classical bit in position 1, called *bit_x*.

```
def measure_and_send(qc, s, a):
    """measures qubits (s,a) and stores results in
    classical bits in positions 0 and 1"""
    qc.barrier()
    qc.measure(s, 0)     # measures 's', stores in pos '0'
    qc.measure(a, 1)     # measures 'a', stores in pos '1'
```

Step 3: Gate Operations

For Bob's qubit to change into state $|\Psi\rangle$, one last function *bob_gates* is necessary. Depending on the information Bob gets through the classical channels, he applies the appropriate gates to his qubit as shown in table 1 (see section 2.1.1). This is implemented via the *c_if* function of Qiskit. For example, the X gate is solely applied to Bob's qubit if the classical bit *bit_x* equals one.

```
def bob_gates(qc, b, bit_z, bit_x):
    """decides (c_if) which gates Bob has to apply to
    reconstruct teleported state and adds them
    to the quantum circuit"""
    qc.barrier()
    qc.x(b).c_if(bit_x, 1)     # apply X gate on 'b' if bit_x=1
    qc.z(b).c_if(bit_z, 1)     # apply Z gate on 'b' if bit_z=1
```

With the functions ready, the quantum teleportation circuit is created by using them in order. In this example, a random qubit state is initialized.

```
#-- QT circuit: statevector sim ----------------------
psi, initial = create_init_state(QT_circuit, 0, 'random')
entangle(QT_circuit, 1, 2)             # 2 = qubit b
alice_gates(QT_circuit, 0, 1)          # 1 = qubit a
measure_and_send(QT_circuit, 0, 1)     # 0 = qubit s
bob_gates(QT_circuit, 2, bit_z, bit_x)

QT_circuit.draw(output='mpl')#, filename='2_circ_statevec.svg')
```

The last line of this program generates a depiction of the quantum circuit, which is shown in figure 2.

To get a better idea of the state that is teleported, the initialized state vector $|\Psi\rangle$ and its representation on the Bloch sphere are shown in figure 3. The construction of the Bloch sphere was done using the QuTiP package in python [13].

With the qubit state being illustrated, it is now time for the state vector simulation. Therefore, the *statevector_simulator*, provided by *Qiskits Aer module* [12], is chosen as the backend and the quantum circuit *QT_circuit* is assembled into a quantum object that can be interpreted by the state vector simulator. The result is stored in *statevec* as the state vector of the total system.

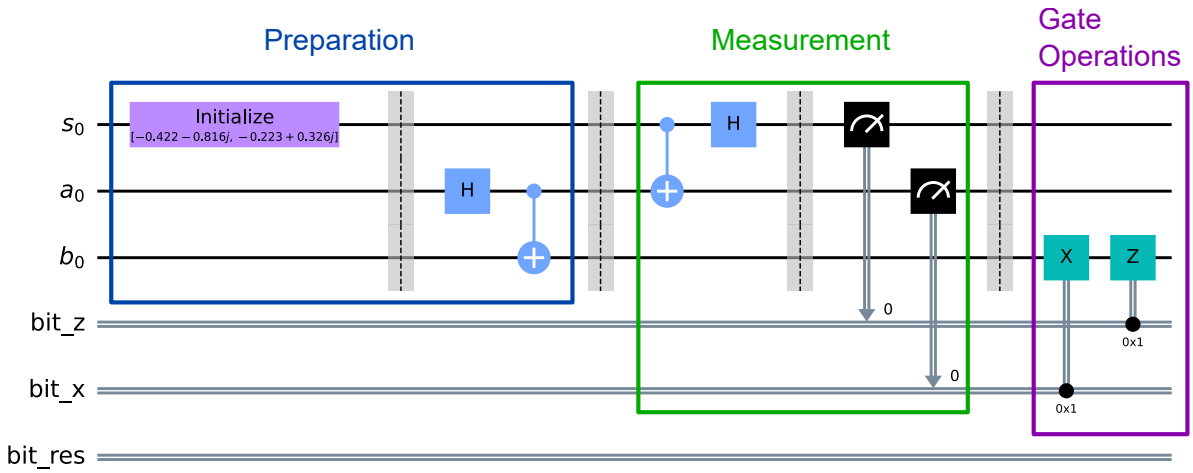
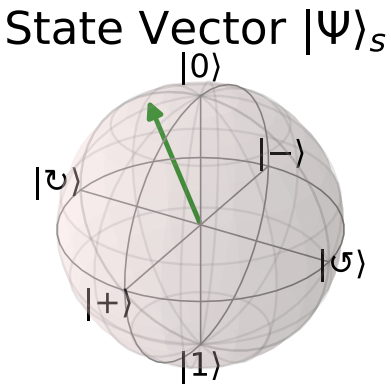


Figure 2: Quantum teleportation circuit for statevector simulation. Created with Qiskit[12] in Jupyter Notebook with annotations added using Inkscape. Code based on IBM Textbook, *Chapter 3.10: Quantum Teleportation* [11].



$$|\Psi\rangle_S = \begin{bmatrix} -0.4224 - 0.8157j \\ -0.2230 + 0.3264j \end{bmatrix}$$

Figure 3: Initialized quantum state $|\Psi\rangle_S$ to be teleported.
Left: Representation on Bloch sphere, created with QuTiP [13].
Right: State vector.

```
#-- statevector simulation ---------------------------
sv_sim = Aer.get_backend('statevector_simulator')
qobj = assemble(QT_circuit)
statevec = sv_sim.run(qobj).result().get_statevector()
```

Extracting the unique state vector of each separate qubit from the given result can be done because both of Alice's qubits are either in state $|0\rangle$ or $|1\rangle$; leaving the only non-zero entries of the combined state vector to be the state vector $|\Psi\rangle$ of the teleported qubit in Bob's possession. The positions of these entries in the combined state vector indicate which states Alice's qubits collapsed into. The algorithm below demonstrates how the separation is realized using an *if* statement. An illustration of the result is

given in figure 4.

```
#-- split combined statevec -------------------------

non0 = np.nonzero(statevec)[0]
# [0] to only get np.array of nonzero indices
# otherwise: tuple = (array([1,5], dtype=int64),)

if (non0 == [0,4]).all():     # /00psi>
    q0 = [1,0]        # /0>
    q1 = [1,0]        # /0>
elif (non0 == [1,5]).all():   # /10psi>
    q0 = [0,1]        # /1>
    q1 = [1,0]        # /0>
elif (non0 == [2,6]).all():   # /01psi>
    q0 = [1,0]        # /0>
    q1 = [0,1]        # /1>
elif (non0 == [3,7]).all():   # /11psi>
    q0 = [0,1]        # /1>
    q1 = [0,1]        # /1>

psi = [statevec[non0[0]], statevec[non0[1]]]
```

The result confirms that the teleportation of state $|\Psi\rangle$ from Alice's to Bob's qubits was successful and that the quantum circuit behaves as intended. The circuit can now be adapted for the next simulation.

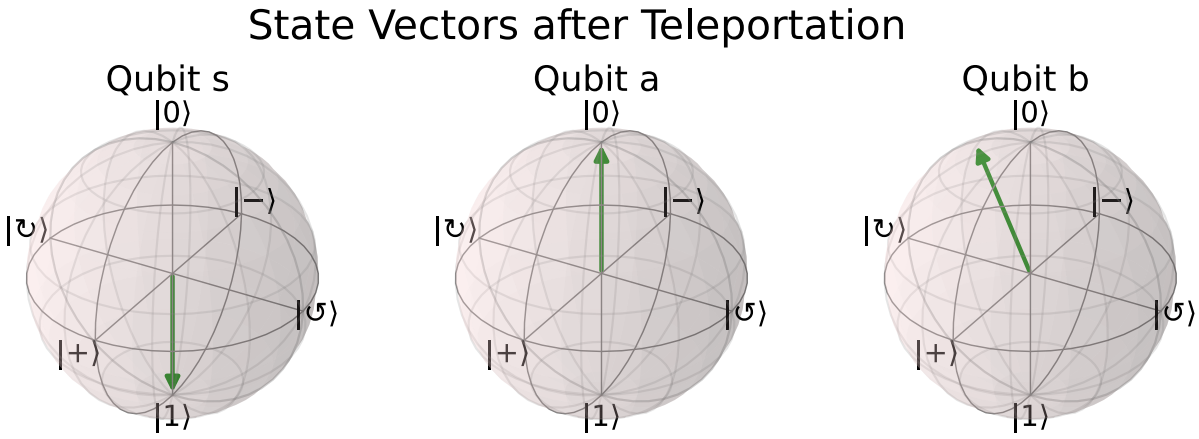
2.2.2 QASM Simulator

IBM's QASM Simulator allows ideal simulations of quantum circuits, thus collapsing the state of the measured qubits into either $|0\rangle$ or $|1\rangle$. To still be able to verify that the state $|\Psi\rangle$ is teleported successfully, some modifications to the circuit are necessary. The objective is that Bob's qubit is measured in state $|0\rangle$ invariably. [11]

First, the extra classical bit *bit_ res*, which was commented out before (see section Statevector Simulation 2.2.1), is added to the quantum circuit. This bit will store the result of Bob's measurement.

Adaptations of Step 3: Gate Operations for QASM Simulation

The new function *inverse_ initial* is defined. As the name suggests, this function creates



$$|\tau\rangle_{BAS} = \begin{bmatrix} -0.4224 - 0.8157j \\ 0 + 0j \\ 0 + 0j \\ 0 + 0j \\ -0.2230 + 0.3264j \\ 0 + 0j \\ 0 + 0j \\ 0 + 0j \end{bmatrix} \longrightarrow \begin{aligned} |q_0\rangle_S &= \begin{bmatrix} 0 + 0j \\ 1 + 0j \end{bmatrix} = |1\rangle \\ |q_1\rangle_A &= \begin{bmatrix} 1 + 0j \\ 0 + 0j \end{bmatrix} = |0\rangle \\ |\Psi\rangle_B &= \begin{bmatrix} -0.4224 - 0.8157j \\ -0.2230 + 0.3264j \end{bmatrix} \end{aligned}$$

Figure 4: Result of state vector simulation done with Qiskit [11, 12]. Alice's qubits collapsed into states $|1\rangle_S$ and $|0\rangle_A$; Bob's qubit is in the teleported state $|\Psi\rangle_B$.
Top: Representation on Bloch sphere, created with QuTiP [13].
Bottom: Combined state vector $|\tau\rangle_{BAS}$ is separated into the state vectors of the individual qubits.

gates to invert the initialization used in the *create_init_state* function. This is possible because all quantum gates are reversible. For that, Qiskit's function *gates_to_uncompute()* is used. In the quantum circuit (see figure 5), this operation is called "Disentangler".

```
def inverse_initial(qc, init):
    """creates gates to invert state of teleported qubit back to
    state |0> [inv func of create_init_state(qc, pos, state)]"""
    qc.barrier()
    inv_init = init.gates_to_uncompute()    # UNcomp !
    qc.append(inv_init, [2])
```

Lastly, the newly created inverse function and a measurement of Bob's qubit are added to the quantum circuit seen in figure 5.

```
#-- QT circuit: qasm sim (from statevec sim) ---------
inverse_initial(QT_circuit, initial)
QT_circuit.measure(2,2)
```

```
QT_circuit.draw(output='mpl')#, filename='2_circ_qasm.svg')
```

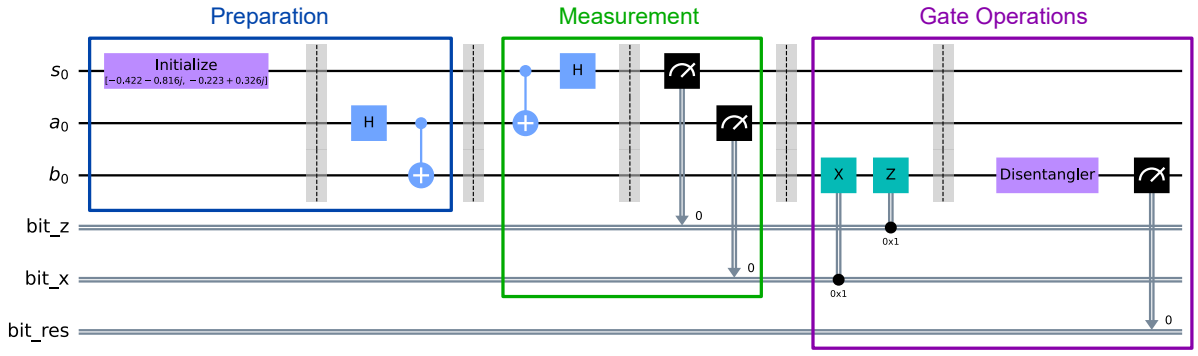


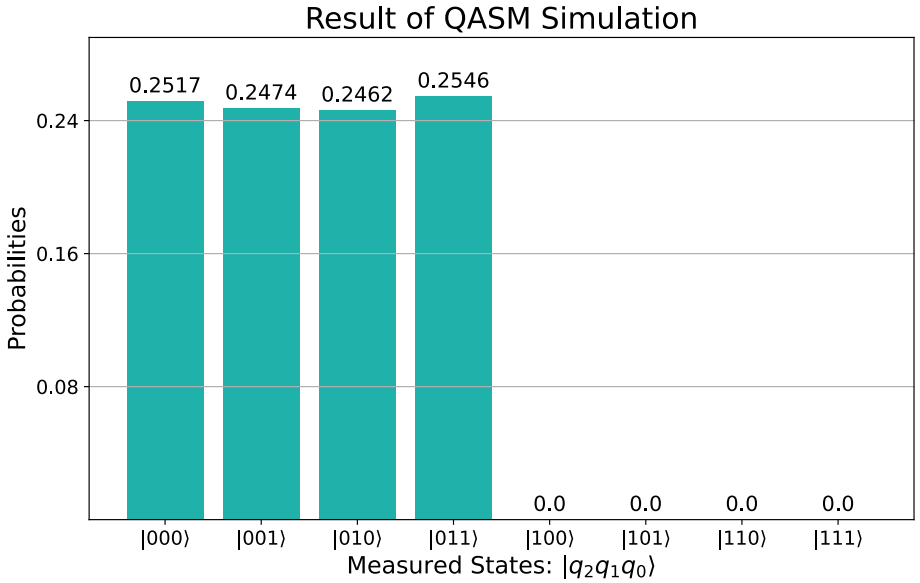
Figure 5: Quantum teleportation circuit for QASM Simulation. Created with Qiskit[12] in Jupyter Notebook with annotations added using Inkscape. Code based on IBM Textbook, *Chapter 3.10: Quantum Teleportation* [11].

Now, the circuit can be simulated using the backend *qasm_simulator* from *Qiskit's Aer module* [12]. Note that the quantum circuit (unlike before) requires to be transpiled before it can be assembled. This is due to the inverse function *gates_to_uncompute()* in the circuit. Furthermore, two settings are chosen for this simulation.
First, the circuit is set to be measured $8192 = 2^{13}$ times by setting *shots* to that amount, which is the maximum amount approved by IBM. This yields a probability distribution of the measured outcome. Since there are three qubits in this circuit there are eight possible states for the qubits. If $|\Psi\rangle$ is successfully teleported, the possible results are reduced to four since Bob's qubit is in state $|0\rangle$ whereas the other two are arbitrarily in states $|0\rangle$ and $|1\rangle$.
Second, the memory is set to *True*, so that all measureable qubit states are counted[9]. The measurement result is then plotted as a histogram (see figure 6).

```
#-- qasm simulation ----------------------------------
qasm_sim = Aer.get_backend('qasm_simulator')
t_qc = transpile(QT_circuit, qasm_sim) # new!
qobj = assemble(t_qc)
result = qasm_sim.run(qobj, shots=8192, memory=True).result()
memory = result.get_memory() # list of strings: ['0 1 1', '0 1 0', ...]
```

[9]Instead of using *get_memory()*, it is also possible to use *get_counts()*. In which case only outcomes that are measured are stored. States like $|111\rangle$, are not part of *get_counts()* whereas they are counted as 0 using *get_memory()*.

As expected, Bob's qubit is invariably measured in state $|0\rangle$, leading to the conclusion that the teleportation was again successful. Building on that, the circuit is adapted to operate on a quantum computer in the next section.



Counts	
$\|000\rangle$:	2062
$\|001\rangle$:	2027
$\|010\rangle$:	2017
$\|011\rangle$:	2086
$\|100\rangle$:	0
$\|101\rangle$:	0
$\|110\rangle$:	0
$\|111\rangle$:	0
Total:	8192

Figure 6: Measerement result of quantum teleportation circuit using QASM Simulator [11, 12]. Bob's qubit is the left-most qubit as the order is $|q_2q_1q_0\rangle_{BAS}$. Thus, Bob reconstructs the state $|\Psi\rangle$ without error.
Left: Histogram showing the probabilities of all measured outcomes.
Right: Counts of each possible outcome.

2.3 Quantum Teleportation on IBM's Quantum Computer

IBM allows everyone to create a free IBM account online, which enables access to numerous resources, one of which is using their quantum devices. All that is required is to download the unique API token that comes with the account and save it on the computer. After loading the token into the jupyter notebook the quantum devices are ready for usage. [17]
This implementation of the quantum teleportation protocol uses the backend *ibmq_ santiago v1.3.22*, which is one of the IBM Quantum *Falcon* Processors. The backend was chosen because it has the lowest CNOT error rate of the (freely) available backends with an average CNOT error of $6.023 \cdot 10^{-3}$. [17]

Adaptations of Step 3: Gate Operations on a Quantum Compter

The functions defined above are still correct for the current circuit with merely one exception: Bob's gates must be altered, as explained in the mathematical description (see section 2.1.1). Thus, in the new function *bob_ gates_ real*, Bob applies a CNOT gate (cx) followed by a controlled-Z gate (cz). Again, it is important to note that the benefit of quantum teleportation is lost in this step!

```
def bob_gates_real(qc, a, b, c):
    """gates to apply on Bobs qubit for use on real quantum computer"""
    qc.barrier()
    qc.cx(b, c) # Controlled-X
    qc.cz(a, c) # Controlled-Z
```

With these adjustments, the new quantum circuit is created below and displayed in figure 7.

```
#--QT circuit: quantum computer ----------------------
create_init_state(QT_circuit, 0, psi) # initialize to /psi> used above
entangle(QT_circuit, 1, 2)              # 2 = qubit b
alice_gates(QT_circuit, 0, 1)           # 1 = qubit a
bob_gates_real(QT_circuit, 0, 1, 2)     # 0 = qubit s    # new!
inverse_initial(QT_circuit, initial)
QT_circuit.measure(2, 0)

QT_circuit.draw(output='mpl')#, filename='2_circ_real.svg')
```

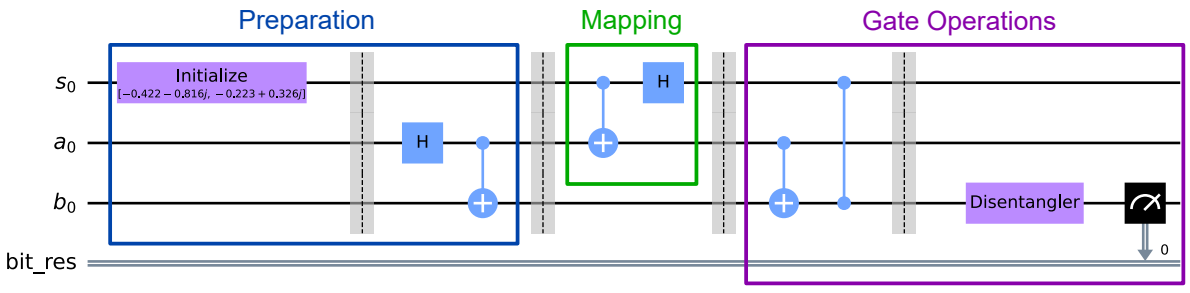


Figure 7: Quantum teleportation circuit for quantum computer. Created with Qiskit[12] in Jupyter Notebook with annotations added using Inkscape. Code based on IBM Textbook, *Chapter 3.10: Quantum Teleportation* [11]. Note that instead of taking her measurement, Alice only maps the qubit states onto the computational basis (see section 2.1.1).

The code for executing the circuit differs from the codes for the simulations. Instead of transpiling and assembling the circuit manually as before, the command *execute()* is

used, which includes all preparations. This is recommended by Qiskit, as the manual way will not be accepted in future releases [18]. Also, the *job_ monitor* is added which displays the job status in real-time, ranging from the number of jobs in the queue to the confirmation that the "job has successfully run"[12].
After that, the result is obtained and plotted with a build-in function of the *visualization* modul of *Qiskit*. The result shows how many times Bob's qubit has been measured in states $|0\rangle$ and $|1\rangle$ and is presented in figure 8. Unlike before, the state of Bob's qubit is not always $|0\rangle$ but has an error rate of 10.52%. This is due to imperfections of physical qubits and quantum noise, i.e. unwanted interactions of the qubits with the environment (for details see [6]).

```
#-- implement on quantum device ----------------------
IBMQ.load_account()
provider = IBMQ.get_provider(hub='ibm-q', group='open', project='main')
qucomp = provider.get_backend('ibmq_santiago')    # smallest error rate
job = execute(QT_circuit, backend=qucomp, shots=8192)
job_monitor(job)

#-- get counts and plot ------------------------------
exp_counts = job.result().get_counts(QT_circuit)
print(exp_counts)              # print actual amounts
plot_histogram(exp_counts)     # plot probabilities
```

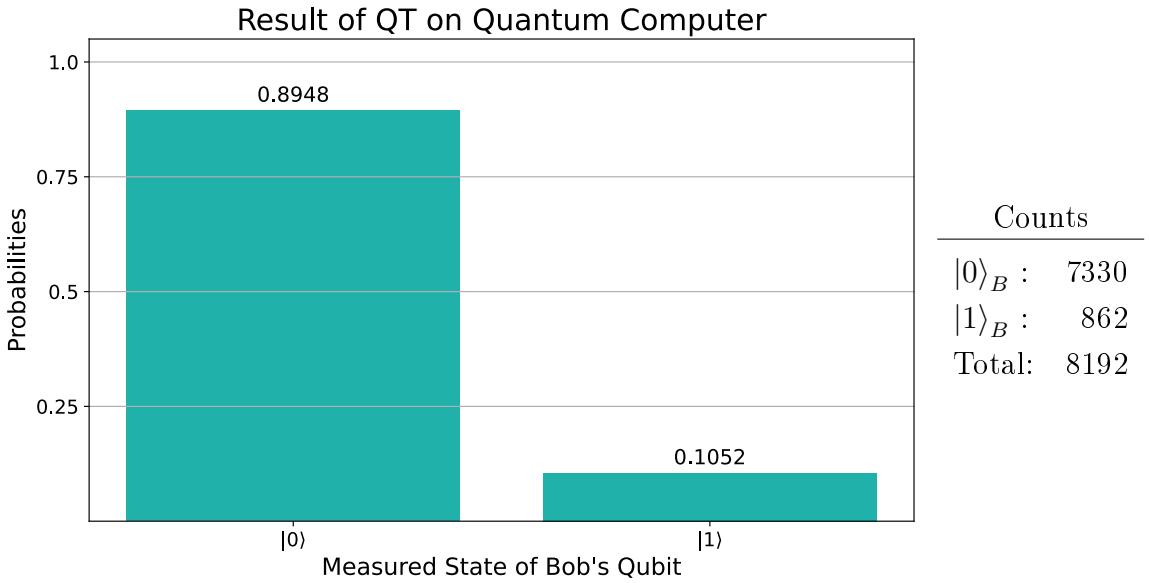


Figure 8: Measurement result of quantum teleportation circuit using the backend *ibmq_santiago v1.3.22*, which is one of the IBM Quantum *Falcon* Processors [17]. The error rate for this experiment is 10.52%.
Left: Histogram showing the probabilities of Bob's qubit to be measured in state $|0\rangle$ (successful) or $|1\rangle$ (error).
Right: Counts of each outcome.

3 Superdense Coding

First, the concept of superdense coding is explained and a mathematical description is provided. Afterward, a quantum circuit is created to simulate the superdense coding protocol. Finally, the circuit is executed on one of IBM's quantum computers.

This section mainly follows IBM Textbook [19], Bennett and Wiesner [5], and Nielsen and Chuang [6]. The programs are written in Python, using IBM's Qiskit [12] and the QuTiP package [13]. The key parts of the program are shown in sections 3.2 and 3.3.

3.1 Concept

The concept of superdense coding was first described in 1992 by Bennett and Wiesner in their paper "Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states" [5]. The goal of superdense coding is to send a two-bit message from one system, "Alice", to another, "Bob" by transferring only one qubit. This is achieved in three steps (see figure 9):
First, an EPR pair is created and shared between Alice and Bob so that each hold one qubit of the pair. This is crucial, since the entanglement of the qubits is what allows the protocol to work.
Second, Alice encodes her two-bit message by applying the appropriate gate(s) to her qubit of the EPR pair. She then sends her qubit to Bob.
Third, Bob decodes the message by taking a joint measurement of both qubits, thus completing the superdense coding protocol.

3.1.1 Mathematical description

To implement the superdense coding protocol, two qubits are required, which are labeled B (Bob) and A (Alice). The significance of the qubits, again, follows the qubit order in Qiskit with the most significant qubit on the left: $|q_1 q_0\rangle$.

Step 1: Preparation

Alice and Bob share an EPR pair, which is created using a Hadamard gate on qubit A, and then a CNOT gate with qubit A as control and qubit B as target. This calculation

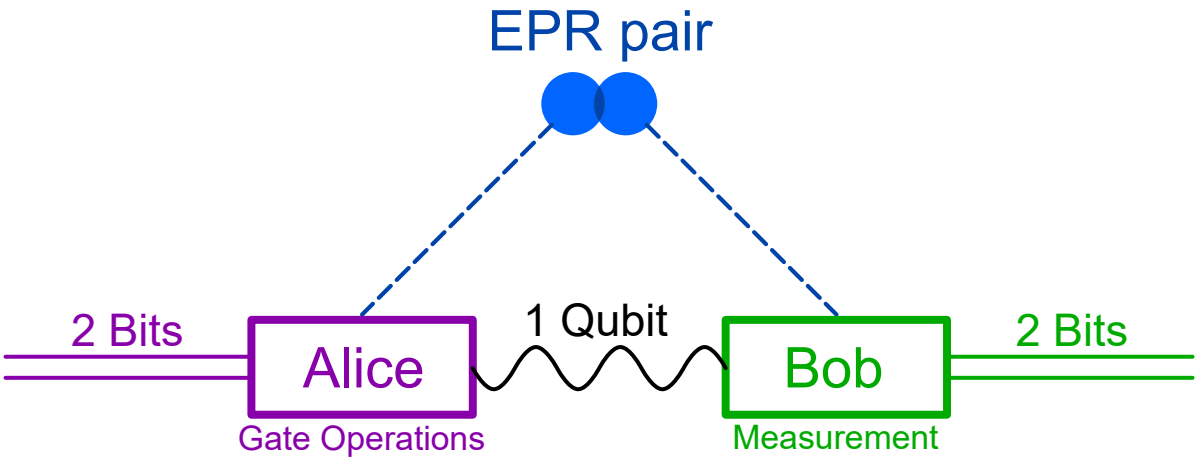


Figure 9: Concept of superdense coding.
Blue: Preparation of EPR pair.
Purple: Alice encodes a message by applying gates to her qubit and sends it to Bob.
Green: Bob takes a joint measurement to decode the message.

is analogous to equation 19 in section 2.1.1 and yields

$$\text{CNOT(A, B)}\Big[I_B \otimes H_A\Big]\Big(\,|0\rangle_B \otimes |0\rangle_A\,\Big) = \frac{1}{\sqrt{2}}\Big(\,|00\rangle + |11\rangle\,\Big)_{BA} = |\Phi^+\rangle_{BA} \tag{31}$$

Step 2: Gate Operations

Alice applies gates depending on the message she wants to encode. The appropriate operations are summarized in table 2. As a sample calculation, the message '11' was chosen because it is slightly more complex than the other possible messages.

Table 2: Gate operations Alice must apply to encode her two-bit message for Bob and the resulting state of the composite system. [cf. 19]

Message	Gate Operation	Resulting State	$(\cdot 1/\sqrt{2})$	
'00'	I	$\Big(\,\|00\rangle + \|11\rangle\,\Big)_{BA}$	$=$	$\|\Phi^+\rangle_{BA}$
'01'	Z	$\Big(\,\|00\rangle - \|11\rangle\,\Big)_{BA}$	$=$	$\|\Phi^-\rangle_{BA}$
'10'	X	$\Big(\,\|10\rangle + \|01\rangle\,\Big)_{BA}$	$=$	$\|\Psi^+\rangle_{BA}$
'11'	XZ	$-\Big(\,\|10\rangle - \|01\rangle\,\Big)_{BA}$	$=-$	$\|\Psi^-\rangle_{BA}$

To encode the message '11', Alice first applies a Z gate, followed by a X gate to her

qubit, which transforms the state of the total system to

$$\begin{aligned}\Big[I_B \otimes X_A\Big]\Big[I_B \otimes Z_A\Big]\left|\Phi^+\right\rangle_{BA} &= \Big[I_B \otimes X_A\Big]\Big[I_B \otimes Z_A\Big]\frac{1}{\sqrt{2}}\Big(\left|00\right\rangle + \left|11\right\rangle\Big)_{BA} = \\ = \Big[I_B \otimes X_A\Big]\frac{1}{\sqrt{2}}\Big(\left|00\right\rangle - \left|11\right\rangle\Big)_{BA} &= \frac{1}{\sqrt{2}}\Big(\left|01\right\rangle - \left|10\right\rangle\Big)_{BA} = -\left|\Psi^-\right\rangle_{BA}\,.\end{aligned} \tag{32}$$

Step 3: Measurement

Now Bob can, theoretically, take a Bell state measurement to decode Alice's message. However, since that is not yet possible in practice (see section 2.1.1), Bob maps the Bell basis state onto the computational basis state of two qubits and takes his measurement in the computational basis. Bob achieves that by first applying a CNOT gate with A as control and B as target, followed by a Hadamard gate to qubit A. Continuing the sample calculation, therefore, yields

$$\begin{aligned}&\Big[I_B \otimes H_A\Big]\mathrm{CNOT(A,\,B)}\frac{1}{\sqrt{2}}\Big(\left|01\right\rangle - \left|10\right\rangle\Big)_{BA} = \\ &= \Big[I_B \otimes H_A\Big]\frac{1}{\sqrt{2}}\Big(\left|11\right\rangle - \left|10\right\rangle\Big)_{BA} = I_B\left|1\right\rangle_B \otimes H_A\frac{1}{\sqrt{2}}\Big(\left|1\right\rangle - \left|0\right\rangle\Big)_A = \\ &= \left|1\right\rangle_B \otimes H_A\Big(-\left|-\right\rangle_A\Big) = -\left|11\right\rangle_{BA}\,.\end{aligned} \tag{33}$$

Bob, thus, measures his qubits in state $\left|11\right\rangle_{BA}$ and Alice's message was successfully decoded. It is important to note that although the qubits are in state $-\left|11\right\rangle_{BA}$, Bob's measurement outcome will be $\left|11\right\rangle_{BA}$ because the global phase factor cannot be measured. All possible cases for Bob's transformation and measurement are summarized in table 3, showing that it is indeed possible to send two bits of information with only one qubit.

3.2 QASM Simulation

Before the superdense coding protocol is implemented on one of IBM's quantum computers it is tested using IBM's QASM Simulator, which allows ideal simulations of quantum circuits.

First, all required packages must be imported into a jupyter notebook and an empty circuit is created. The superdense coding circuit necessitates two qubits and two classical bits, to store the measurement result, which are combined into the quantum circuit

Table 3: Bob receives the qubit with Alice's encoded message and maps both qubits onto the computational basis. To do so, Bob first applies a CNOT(A, B) gate, followed by a Hadamard(A) gate. Bob then measures the qubits in the computational basis to decode the message. [cf. 19]

After CNOT(A, B):	$(\cdot 1/\sqrt{2})$	After H_A:	Message
$\Big(\|00\rangle + \|01\rangle\Big)_{BA}$	$= \|0+\rangle_{BA}$	$\|00\rangle_{BA}$	'00'
$\Big(\|00\rangle - \|01\rangle\Big)_{BA}$	$= \|0-\rangle_{BA}$	$\|01\rangle_{BA}$	'01'
$\Big(\|10\rangle + \|11\rangle\Big)_{BA}$	$= \|1+\rangle_{BA}$	$\|10\rangle_{BA}$	'10'
$-\Big(\|10\rangle - \|11\rangle\Big)_{BA}$	$= -\|1-\rangle_{BA}$	$-\|11\rangle_{BA}$	'11'

SDC_ circuit as shown below.

```
#-- setup ------------------------------------------

# 2 qubits
qr0 = QuantumRegister(1, name="a")
qr1 = QuantumRegister(1, name="b")
# 2 classical bits
bit_a = ClassicalRegister(1, name="bit_a")
bit_b = ClassicalRegister(1, name="bit_b")

# create empty quantum circuit
SDC_circuit = QuantumCircuit(qr0,qr1, bit_a, bit_b)
```

Hereafter, functions are created according to the three steps of the protocol.
All functions have a quantum circuit (qc) as input parameter and generate gates for the circuit.

Step 1: Preparation

The function *entangle* creates the EPR pair that Alice (a) and Bob (b) share and is equivalent to the one used in the quantum teleportation circuit (see section 2.2.1).

```
def entangle(qc, a, b):
    """creates a EPR pair (entangled) using qubits a and b"""
    qc.h(a)        # Hadamard gate on 'a'
    qc.cx(a,b)     # CNOT with 'a' as control and 'b' as target
```

Step 2: Gate Operations

Alice uses the function *encode_message* to encode her message for Bob. This is a conditional function using an *if* statement to choose the appropriate gate(s), according to table 2, depending on the message Alice wants to encode.

```
def encode_message(qc, qubit, msg):
    """adds gates to 'qubit' (index of position) in quantum circuit
    'qc' depending the message 'msg' (string: '01', '10', or '11'),
    does nothing (corr to identity gate) for msg=='00'"""
    qc.barrier()          # adds vertical line, no affect on qubits
    if msg[1] == '1':     # True for '01' and '11'
        qc.z(qubit)
    if msg[0] == '1':     # True for '10' and '11'
        qc.x(qubit)
```

Step 3: Measurement

With the function *decode_message* Bob first maps the Bell states onto the computational basis by applying a CNOT gate (cx) with Alice's qubit (a) as control and his qubit (b) as target, followed by a Hadamard gate (h) on Alice's qubit (a). Then, Bob takes a joint measurement in the computational basis. In the process, the result of qubit a is stored in the classical bit in position 0, called bit_a; whereas the result of qubit b is stored in the classical bit in position 1, called bit_b.

```
def decode_message(qc, a, b):
    """maps Bell basis states onto comp basis states of two qubits and
    measures states of those qubits (corr to Bell state measurement);
    stores the result in classical bits in positions 0 and 1"""
    qc.barrier()
    qc.cx(a, b)         # CNOT with 'a' as control and 'b' as target
    qc.h(a)             # Hadamard gate on 'a'
    qc.barrier()
    qc.measure(a,0)     # measures 'a', stores in pos '0'
    qc.measure(b,1)     # measures 'b', stores in pos '1'
```

Now the superdense coding circuit is created by calling all functions in order and choosing a message. Here the sample message '11' is used to parallel the example in the mathematical description (see section 3.1.1). The quantum circuit is shown in figure 10.

```
#-- SDC circuit ---------------------------------
entangle(SDC_circuit, 0, 1)
message = '11'    # '00', '01', '10' or '11'
```

```
encode_message(SDC_circuit, 0, message)
decode_message(SDC_circuit, 0, 1)

SDC_circuit.draw(output='mpl')#, filename='SDC_11.svg')
```

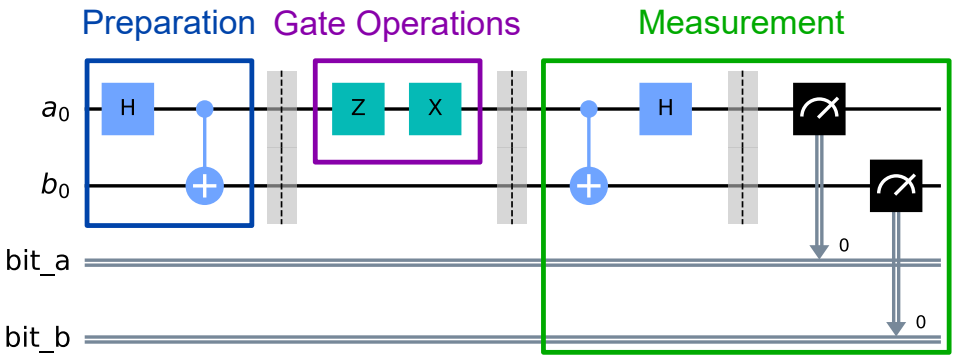


Figure 10: Superdense coding circuit for messaging '11'. Created with Qiskit [12] in Jupyter Notebook with annotations added using Inkscape. Code based on IBM Textbook, *Chapter 3.11: Superdense Coding* [19].

Finally, the superdense coding protocol is simulated using the backend *qasm_simulator* from *Qiskit's Aer module* [12]. Therefore, the circuit *SDC_circuit* is assembled into a quantum object that can be interpreted by the simulator. Furthermore, the *shots* are set to their maximum of $8192 = 2^{13}$ to obtain a probability distribution of the measurement outcome; and the memory is set to *True*, so that all measureable qubit states are counted (cf. section 2.2.2). The result is then plotted as a histogram (see figure 11).

```
# qasm simulation ----------------------------------
qasm_sim = Aer.get_backend('qasm_simulator')
qobj = assemble(SDC_circuit)
result = qasm_sim.run(qobj, shots=8192, memory=True).result()
memory = result.get_memory() # list of strings: ['0 1','1 1', ...]
```

3.3 Superdense Coding on IBM's Quantum Computer

To implement the superdense coding protocol, the backend *ibmq_santiago v1.3.22*, which is one of the IBM Quantum *Falcon* Processors, is used. This backend was chosen because it has the lowest CNOT error rate of the (freely) available backends with an average CNOT error of $6.023 \cdot 10^{-3}$ [17].

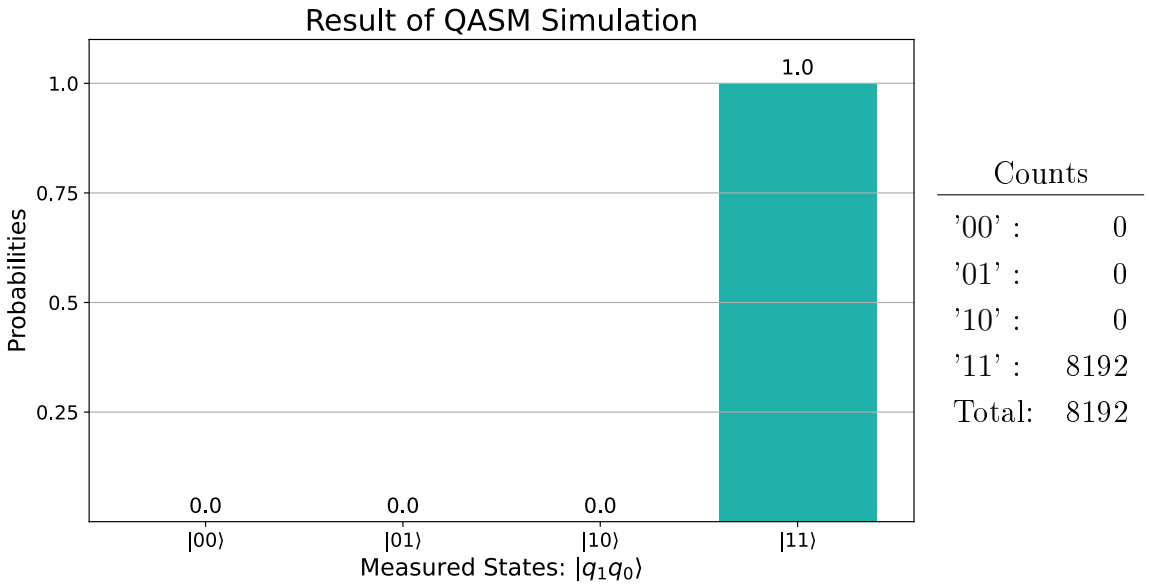


Figure 11: Measurement result of superdense coding circuit using QASM Simulator [12, 19]. Bob decodes Alice's message without error.
Left: Histogram showing the probabilities of Bob's received message.
Right: Counts of each message received.

It is noteworthy that no adjustments to the circuit are necessary for its use on a quantum computer. Therefore, the circuit is the same as for the QASM simulation depicted in figure 10.

The code below shows how the circuit is executed on a quantum computer and is analogous to the code used in section 2.3.

```
#-- implement on quantum device ----------------------
IBMQ.load_account()
provider = IBMQ.get_provider(hub='ibm-q', group='open', project='main')
qucomp = provider.get_backend('ibmq_santiago')    # smallest error rate
job = execute(SDC_circuit, backend=qucomp, shots=8192)
job_monitor(job)

#-- get counts and plot ------------------------------
exp_counts = job.result().get_counts()
print(exp_counts)              # print actual amounts
plot_histogram(exp_counts)     # plot probabilities
```

The result (see figure 12) shows that Bob cannot decode Alice's message without error anymore. This is due to noise on the quantum computer. However, with an error rate of 5.66% Bob is likely to receive the correct message.

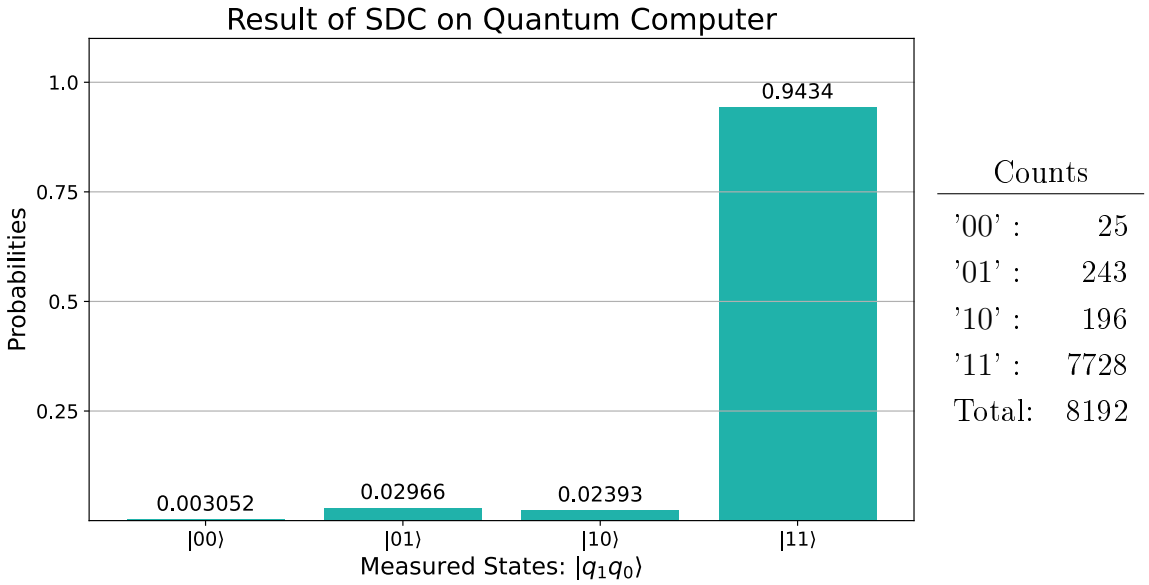


Figure 12: Measurement result of superdense coding circuit using the backend *ibmq_ santiago v1.3.22*, which is one of the IBM Quantum *Falcon* Processors [17]. The error rate in this example is 5.66%.
Left: Histogram showing the probabilities of Bob's received message. Here the message '11' was sent, thus, the outcome $|11\rangle$ shows a successful transfer of information.
Right: Counts of each outcome.

4 Comparison

In this chapter, the similarities and differences of the quantum teleportation (QT) and the superdense coding (SDC) protocol are pointed out, followed by a short overview of their fields of application. A depiction of both concepts is given in figure 13.

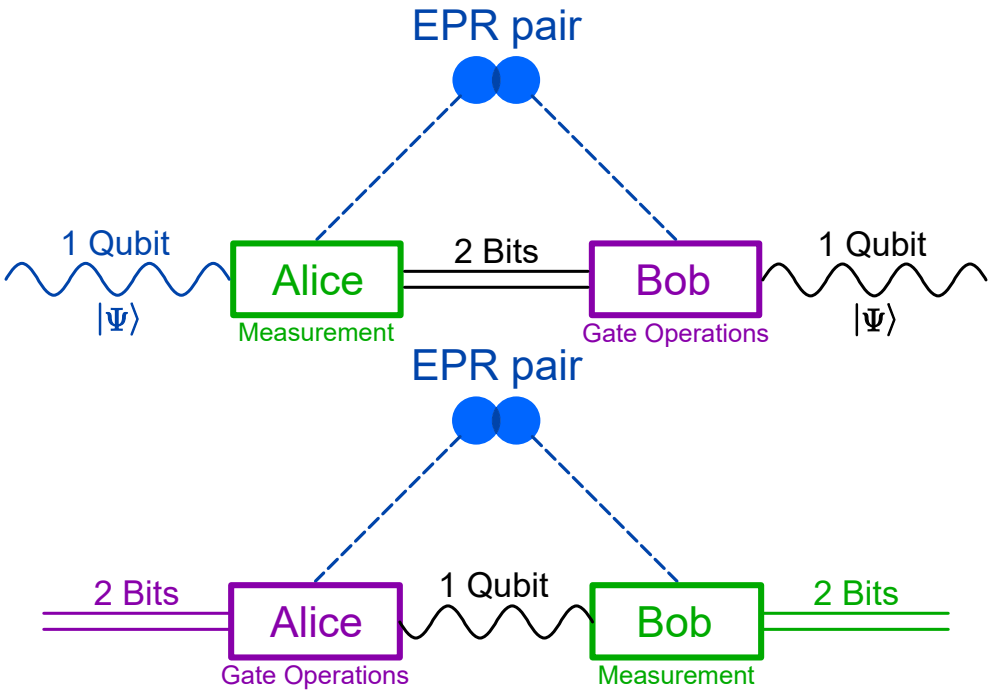


Figure 13: Top: Concept of quantum teleportation.
Bottom: Concept of superdense coding.
Blue: Preparation. Purple: Gate Operations. Green: Measurement.
The position of the Measurement and the Gate Operations are exchanged in the protocols.

4.1 Comparison of the Protocols

The overall similarity of the two concepts is immediately apparent: In both protocols, data is sent from Alice to Bob. This is achieved by sharing an entangled pair (e.g., an EPR pair) of qubits before executing the protocol. Additionally, a joint measurement is taken during both protocols with the qubit of the shared pair as part of the measurement. Thus, QT and SDC show congruity in how they use quantum mechanics to their advantage.

However, a closer look upon the data sent reveals that QT and SDC serve entirely

different purposes: While the QT protocol uses two classical bits to send one qubit, the SDC protocol uses one qubit to send two classical bits.

$$\begin{aligned} &\text{QT:} \quad 2\ \text{cbits} + 1\ \text{ebit} = 1\ \text{qubit} \\ &\text{SDC:} \quad 1\ \text{qubit} + 1\ \text{ebit} = 2\ \text{cbits} \end{aligned} \tag{34}$$

Here, cbits refer to the classical bits, and ebit refers to the entangled pair of qubits.

Another similarity is that both concepts require local gate operations built out of the X gate and the Z gate. In the QT protocol, Bob applies gate operations depending on Alice's measurement outcome, whereas in the SDC protocol, Alice uses gate operations depending on the message she intends to send. Table 4 gives a summary of the applied gates for each possible case for both protocols.

Table 4: Gate operations for quantum teleportation and superdense coding protocol (cf. tables 1 and 2). The gate operations of the two protocols are the inverse of each other.

Quantum Teleportation		Superdense Coding	
Alice's Result	Gate Operation	Message	Gate Operation
$\|00\rangle_{AS}$	I	'00'	I
$\|01\rangle_{AS}$	Z	'01'	Z
$\|10\rangle_{AS}$	X	'10'	X
$\|11\rangle_{AS}$	ZX	'11'	XZ

Nevertheless, a closer look again reveals that the gate operations are, in fact, the inverse of each other. All three gates can be interpreted as Pauli matrices which are Hermitian and unitary. A Hermitian matrix is its own conjugate transpose, which is also its inverse if the matrix is unitary. Thus, all gates are also their own inverse. This becomes more transparent in the fourth case (Alice's result: $|11\rangle_{AS}$; Message: '11'), where the order of the applied gates is reversed.

In summary, both protocols consist of similar building blocks. However, their input data are different. As a consequence, the positions of the joint measurement and the local gate operations have to be exchanged: In the QT protocol, the joint measurement is step 2, while it is step 3 in the SDC protocol. Accordingly, the gate operations in QT are step 3, and step 2 in SDC (Indicated with corresponding colors in figure 13.). Thus, QT and SDC can undoubtedly be regarded as contrasting concepts.

4.2 Comparison of the Implementations

The significant difference that is immediately apparent is that the QT protocol could not be implemented as intended due to hardware limitations. That is because the classical bits, which Bob is supposed to use for his gate operations, are Alice's measurement result and real-time conditionals are not possible yet.
However, the same problem would arise for the SDC protocol if Alice would not choose the transmitted message but instead forward the outcome of a previous measurement.

Unfortunately, this restricts further possibilities of comparing the protocols, e.g., their error rates. It would be interesting to revisit the problem and perform an in-depth error analysis once the limitations are overcome.

4.3 Fields of Application

Quantum Communication

In the subsection above, it was established that QT and SDC are contrasting concepts. Nevertheless, both protocols use entanglement-assisted communication, thus, playing a pivotal role in quantum communication [20].
There are various benefits of being able to transmit a quantum state to an arbitrarily distant location. For example, QT will allow sending a quantum state from one quantum computer to another for further calculations without the need for a physical quantum channel connecting them. A big step towards that goal was the successful teleportation of six bases states from Earth to the Micius satellite in 2017 by Ren et al. [21].
SDC, on the other hand, enables faster data transfer of classical bits. This was already pointed out by Bennett and Wiesner in their original paper about superdense coding [5]. Accordingly, as presented in the present thesis, the protocol allows doubling the peak data rate by preparing the entangled pairs beforehand. However, recent studies have shown that the data rate can be increased even further by using hyperentangled qubits [22] or combining SDC with quantum linear network coding [23].

Quantum Cryptography

With new possibilities of communication come new responsibilities of securing the transmitted information. In this respect again, QT and SDC prove their capabilities.
The QT protocol is not susceptible to eavesdropping in general. Admittedly, the classical channel transferring the two bits of information required to reconstruct the quantum

state can be intercepted. However, it is not possible to draw any conclusions from it. This is also true if multiple senders and receivers share a quantum secret, as Lee et al. demonstrated [24].
SDC is the primitive for several more complex protocols too. The most important so far are the quantum secure direct communication (QSDC) protocol [25] and the quantum key secure communication (QKSC) protocol [26], which both have been implemented on IBM's quantum computer. Furthermore, the use of hyperentangled qubits was proposed for an even higher capacity [27].

Quantum Computing

In this field, mainly the QT protocol is applied.
A significant problem of quantum computing is the decoherence of quantum states. However, there are two great tools to counter decoherence: local memory and error correction.
QT combats short decoherence times by teleporting a quantum state onto a local memory with longer decoherence times, thus, preserving the state. Singh et al. suggest a hybrid technology between superconducting qubits for computing and Nitrogen-Vacancies for the data storage to be "the most encouraging design to be scaled up into a comprehensive quantum network" [28].
The progress of large-scale error corrections was demonstrated by Luo et al. in their paper "Quantum teleportation of physical qubits into logical code-spaces" in 2020. Basically, the information on the physical qubits is teleported onto the logical qubit, a highly entangled state where error correction can be performed more effectively. [29]

Of course, these examples show only a small portion of where QT, SDC, and the more established protocols building upon them are applicable. For more information and further reading, the following papers are recommended: "Advances in quantum teleportation" (Pirandola et al., 2015) [30], "The Applications and Challenges of Quantum Teleportation" (Liu, 2020) [31], "Quantum Science and Quantum Technology: Progress and Challenges" (Wang and Alexander, 2020) [32], "Quantum Internet- Applications, Functionalities, Enabling Technologies, Challenges, and Research Directions" (Singh et al., 2021) [28].

5 Summary

The quantum teleportation (QT) protocol allows sending a qubit from Alice to Bob using only two classical bits and one shared pair of entangled qubits.
In this thesis, a random quantum state was created and successfully teleported. This was achieved on two simulators and a quantum computer.
IBM's Statevector Simulator was used to test the QT circuit. This simulator calculates the combined state vector of the circuit and shows in which states the qubits are in without measuring them. The combined state vector was separated into the state vectors of the individual qubits, thus confirming that the qubit state was teleported successfully. Following that, the quantum circuit was implemented on IBM's QASM Simulator. It allows ideal simulations and measures the qubits, thus collapsing them into one of their eigenstates. To still obtain verifiable results, an extra gate ("Disentangler") was added so that Bob's qubit should always be measured in state $|0\rangle$. This was also confirmed.
Before running the QT circuit on IBM's quantum computer, another adaptation was necessary due to hardware limitations. Since it is not yet possible to use real-time conditionals in a circuit, Bob cannot decide which gate operations to use. To circumvent this, Alice maps her qubits onto the computation basis but does not measure them. Afterward, Bob uses her qubits as control qubits for his gate operations. Finally, the "Disentangler" was added again to verify the results. The QT circuit was run on the backend *ibmq_santiago v1.3.22*, which is one of the IBM Quantum *Falcon* Processors. The teleportation was performed 2^{13} times (permitted maximum by IBM) and yielded an error rate of 10.52%.

With the superdense coding (SDC) protocol, two classical bits are transmitted from Alice to Bob using one qubit and one shared pair of entangled qubits.
In this thesis, the message '11' was chosen because it is slightly more complex than the other possible messages. First, IBMS's QASM Simulator was used to confirm that the SDC circuit worked as intended. Then the circuit was implemented on the backend *ibmq_santiago v1.3.22*, which is one of the IBM Quantum *Falcon* Processors. Again, a maximum of 2^{13} iterations was chosen for the transmission. The measured error rate was 5.66%.

Comparing QT and SCD showed that both protocols require that an entangled pair is shared beforehand. Furthermore, a joint measurement, with the qubit of the shared pair as part of the measurement, is required for successful transmission. Additionally, the qubit gates used for the local gate operations are the same.

However, the order of these building blocks and which gates are used for the local operations differ. Moreover, the input data and the transmitted data are opposites. Thus, despite all the similarities, leading to two contrasting protocols.

The outline of the fields of application showed that QT, as well as SDC, are suitable for a wide variety of uses. Additionally, QT and SDC are the building blocks of many, more complex protocols. Thus, it must be concluded that it is crucial to understand the QT and SDC protocol fully for further developing quantum technologies.

References

[1] European Commisson. *The future is quantum: EU countries plan ultra-secure communication network*. Last update: Mar. 2021. URL: https://digital-strategy.ec.europa.eu/en/news/future-quantum-eu-countries-plan-ultra-secure-communication-network.

[2] European Commission. *Quantum Flagship: a major boost for European Quantum Research*. Factsheet. July 2019. URL: https://digital-strategy.ec.europa.eu/en/library/quantum-flagship-major-boost-european-quantum-research.

[3] European Commission. *Midterm Report of the Quantum Technologies Flagship*. Research rep. Sept. 2020. URL: https://digital-strategy.ec.europa.eu/en/policies/quantum-technologies-flagship.

[4] Charles H. Bennett et al. "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels". In: *Phys. Rev. Lett.* 70 (13 Mar. 1993), pp. 1895–1899. DOI: 10.1103/PhysRevLett.70.1895. URL: https://link.aps.org/doi/10.1103/PhysRevLett.70.1895.

[5] Charles H. Bennett and Stephen J. Wiesner. "Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states". In: *Physical review letters* 69.20 (Nov. 1992), pp. 2881–2884. DOI: 10.1103/PhysRevLett.69.2881. URL: https://link.aps.org/doi/10.1103/PhysRevLett.69.2881.

[6] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. 10th Anniversary Edition. Cambridge University Press, Dec. 2010. ISBN: 9781107002173. URL: www.cambridge.org/9781107002173.

[7] Wolfgang Scherer. *Mathematik der Quanteninformatik*. Springer Spektrum, 2016. ISBN: 978-3-662-49080-8. DOI: 10.1007/978-3-662-49080-8.

[8] IBM Textbook. *Chapter 1.2: Single Qubit Gates*. URL: https://qiskit.org/textbook/ch-states/single-qubit-gates.html.

[9] IBM Textbook. *Chapter 2.2: Multiple Qubits and Entangled States*. URL: https://qiskit.org/textbook/ch-gates/multiple-qubits-entangled-states.html.

[10] William K. Wootters and Wojciech H. Zurek. "The no-cloning theorem". In: *Physics Today* 62.2 (2009), pp. 76–77.

[11] IBM Textbook. *Chapter 3.10: Quantum Teleportation*. URL: https://qiskit.org/textbook/ch-algorithms/teleportation.html.

[12] Héctor Abraham et al. *Qiskit: An Open-source Framework for Quantum Computing*. 2019. DOI: 10.5281/zenodo.2562110.

[13] J.R. Johansson, P.D. Nation, and Franco Nori. "QuTiP: An open-source Python framework for the dynamics of open quantum systems". In: *Computer Physics Communications* 183.8 (2012), pp. 1760–1772. ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2012.02.021. URL: https://www.sciencedirect.com/science/article/pii/S0010465512000835.

[14] Mark Riebe. "Preparation of Entangled States and Quantum Teleportation with Atomic Qubits". PhD thesis. Fakultät für Mathematik, Informatik und Physik der Leopold-Franzens-Universität Innsbruck, 2005. URL: https://quantumoptics.at/images/publications/dissertation/riebe_diss.pdf.

[15] Paul Nation and Blake Johnson. *How to measure and reset a qubit in the middle of a circuit execution*. IBM Research Blog. Feb. 2021. URL: https://www.ibm.com/blogs/research/2021/02/quantum-mid-circuit-measurement/.

[16] *Mid-Circuit Measurements Tutorial*. IBM Quantum Lab: Docs. URL: https://quantum-computing.ibm.com/lab/docs/iql/manage/systems/midcircuit-measurement/.

[17] *IBM Quantum*. 2021. URL: https://quantum-computing.ibm.com/.

[18] *Access systems with your account*. IBM Quantum Lab: Docs. URL: https://quantum-computing.ibm.com/lab/docs/iql/manage/account/ibmq.

[19] IBM Textbook. *Chapter 3.11: Superdense Coding*. URL: https://qiskit.org/textbook/ch-algorithms/superdense-coding.html.

[20] C. H. Bennett and P. Shor. "Quantum Information Theory". In: *IEEE Trans. Inf. Theory* 44 (1998), pp. 2724–2742.

[21] Ji-Gang Ren et al. "Ground-to-satellite quantum teleportation". In: *Nature* 549.7670 (2017), pp. 70–73.

[22] Xiao-Min Hu et al. "Beating the channel capacity limit for superdense coding with entangled ququarts". In: *Science advances* 4.7 (2018), eaat9304.

[23] Steven Herbert. "Increasing the classical data throughput in quantum networks by combining quantum linear network coding with superdense coding". In: *Physical Review A* 101.6 (2020), p. 062332.

[24] Sang Min Lee et al. "Quantum teleportation of shared quantum secret". In: *Physical review letters* 124.6 (2020), p. 060501.

[25] Saiful Islam Salim et al. "Enhancing Fidelity of Quantum Cryptography using Maximally Entangled Qubits". In: *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE. 2020, pp. 1–6.

[26] Mario Mastriani. "Quantum key secure communication protocol via enhanced superdense coding". In: *arXiv preprint arXiv:2011.05165* (2020).

[27] Wang Tie-Jun et al. "High-capacity quantum secure direct communication based on quantum hyperdense coding with hyperentanglement". In: *Chinese Physics Letters* 28.4 (2011), p. 040305.

[28] Amoldeep Singh et al. "Quantum Internet- Applications, Functionalities, Enabling Technologies, Challenges, and Research Directions". In: *arXiv preprint arXiv:2101.04427* (2021).

[29] Yi-Han Luo et al. "Quantum teleportation of physical qubits into logical codespaces". In: *arXiv preprint arXiv:2009.06242* (2020).

[30] Stefano Pirandola et al. "Advances in quantum teleportation". In: *Nature photonics* 9.10 (2015), pp. 641–652.

[31] Tao Liu. "The Applications and Challenges of Quantum Teleportation". In: *Journal of Physics: Conference Series*. Vol. 1634. 1. IOP Publishing. 2020, p. 012089.

[32] Lidong Wang and Cheryl Ann Alexander. "Quantum Science and Quantum Technology: Progress and Challenges". In: *Am. J. Electr. Electron. Eng.* 8.2 (2020), pp. 43–50.