PHYS483: Quantum information processing—Lecture Notes

Henning Schomerus, Lancaster University

Contents

I. QUANTUM MECHANICS

A. States

The state of a quantum system is described by a vector $|\psi\rangle$. These vectors form a *complex linear vector space*, which entails, in particular, the following properties: Any state $|\psi\rangle$ can be scaled by any complex number α, i.e., we can form new states $|\alpha\psi\rangle = \alpha|\psi\rangle$. Furthermore, any two states $|\psi\rangle$, $|\chi\rangle$ can be combined into new states by a forming a *superposition* $|\psi + \chi\rangle = |\psi\rangle + |\chi\rangle$.

The vector space is a *Hilbert space*, i.e., it is equipped with a *scalar product* that associates a complex number $\langle\psi|\chi\rangle$ to any pair of states $|\psi\rangle$, $|\chi\rangle$. The scalar product is positive definite, $\langle\psi|\psi\rangle > 0$ for $|\psi\rangle \neq 0|\psi\rangle$, and fulfills $\langle\psi|\chi\rangle = \langle\chi|\psi\rangle^*$. Furthermore, it is linear in the second argument, but *conjugate linear* in the first argument, i.e., $\langle\psi|\alpha\chi\rangle = \alpha\langle\psi|\chi\rangle$, $\langle\alpha\psi|\chi\rangle = \alpha^*\langle\psi|\chi\rangle$, $\langle\psi+\varphi|\chi\rangle = \langle\psi|\chi\rangle+\langle\varphi|\chi\rangle$, $\langle\psi|\varphi+\chi\rangle = \langle\psi|\varphi\rangle + \langle\psi|\chi\rangle$.

Formally, the scalar product can be interpreted as a product $\langle\psi| \cdot |\chi\rangle$ between the vectors $|\chi\rangle$ and the entities $\langle\psi|$, which form the *dual vector space*. They represent the left states in the scalar product and therefore are also conjugate linear: $\langle\alpha\psi+\beta\chi| = \alpha^*\langle\psi| + \beta^*\langle\chi|$. The particular notation introduced here is the so-called *Dirac notation*. In this notation, a dual vector is also called a *bra*, and an ordinary vector is called a *ket*, alluding to the fact that in the scalar product $\langle\psi|\chi\rangle$ they form a bracket (bra-ket).

We call $||\psi|| = \sqrt{\langle\psi|\psi\rangle}$ the *length* of the vector $|\psi\rangle$. A vector with $\langle\psi|\psi\rangle = 1$ is called *normalized*. The procedure of passing from a vector $|\psi\rangle$ to the normalized vector $|\psi\rangle/||\psi||$ is called *normalization*. Two states $|\psi\rangle$, $|\chi\rangle$ fulfilling $\langle\psi|\chi\rangle = 0$ are said to be *orthogonal to each other*.

A *basis* is a collection of vectors $|n\rangle$, $n = 1, 2, 3, \ldots, \mathcal{N}$ such that any vector can be written as a superposition $|\psi\rangle = \sum_{n=1}^{\mathcal{N}} \psi_n |n\rangle$, where the complex coefficients ψ_n are unique. The coefficients ψ_n give a *representation* of the state, and can be written as a column vector

$$\psi = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_{\mathcal{N}} \end{pmatrix}.$$

The corresponding dual vector is written as a row vector $\psi^\dagger = (\psi_1^*, \psi_2^*, \ldots, \psi_{\mathcal{N}}^*)$. While there are many possible bases, in which the same vector is represented by different coefficients, the number $\mathcal{N}$ of basis states required to obtain all vectors is always the same, and is called the *dimension* of the vector space ($\mathcal{N}$ may be ∞).

An orthogonal basis fulfills $\langle n|m\rangle = 0$ for any $n \neq m$. If furthermore $\langle n|n\rangle = 1$ for all n one speaks of an *orthonormal basis*. In such a basis, the coefficients representing a state are given by

$\psi_n = \langle n|\psi\rangle$, and the scalar product takes the form $\langle\psi|\chi\rangle = \sum_n \psi_n^* \chi_n = \psi^\dagger \chi$.

B. Operators

An operator $\hat{A}$ converts any state $|\psi\rangle$ into another state $|\hat{A}\psi\rangle = \hat{A}|\psi\rangle$. Linear operators fulfill $\hat{A}(\alpha|\psi\rangle + \beta|\chi\rangle) = \alpha\hat{A}|\psi\rangle + \beta\hat{A}|\chi\rangle$. Operators can be added according to the rule $(\hat{A}+\hat{B})|\psi\rangle = \hat{A}|\psi\rangle + \hat{B}|\psi\rangle$, and multiplied according to the rule $(\hat{B}\hat{A})|\psi\rangle = \hat{B}(\hat{A}|\psi\rangle)$.

In an orthonormal basis, linear operators are represented by $N \times N$-dimensional square matrices

$$A = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1N} \\ A_{21} & A_{22} & \dots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & \dots & A_{NN} \end{pmatrix}$$

with coefficients $A_{nm} = \langle n|\hat{A}m\rangle \equiv \langle n|\hat{A}|m\rangle$. They then act on vectors by matrix multiplication, i.e., $|\varphi\rangle = \hat{A}|\psi\rangle$ is represented by coefficients $\varphi_n = \sum_m A_{nm}\psi_m$. In a given representation, the operator addition and multiplication rules translate to the usual prescriptions of matrix addition and multiplication.

In Dirac notation, operators are written as $\hat{A} = \sum_{nm} A_{nm}|n\rangle\langle m|$, and the action of an operator is obtained from the multiplication rule $\langle m| \cdot |\psi\rangle = \langle m|\psi\rangle$.

The action of an operator is particularly simple in its *eigenrepresentation*, defined by a basis fulfilling $\hat{A}|n\rangle = a_n|n\rangle$. The numbers a_n are called *eigenvalues*, and the vectors $|n\rangle$ are called *eigenvectors*. If the eigenvectors form an orthonormal basis, the matrix A_{nm} is diagonal, $A_{nm} = 0$ if $n \neq m$ and $A_{nn} = a_n$. In Dirac notation, the operator can then be written as $\hat{A} = \sum_n a_n|n\rangle\langle n|$.

A particularly simple operator is the identity operator $\hat{I}$, which leaves all states unchanged, $\hat{I}|\psi\rangle = |\psi\rangle$. Every state is therefore an eigenstate of $\hat{I}$, with eigenvalue 1. Consequently, in any orthonormal basis this operator takes the same form $\hat{I} = \sum_n |n\rangle\langle n|$. Representations are simply obtained by multiplying out the identities $|\psi\rangle = \hat{I}|\psi\rangle$ and $\hat{A} = \hat{I}\hat{A}\hat{I}$. For a fixed orthonormal basis, it is useful to decompose the identity $\hat{I} = \sum \hat{E}_n$ as the sum of *projection operators* $\hat{E}_n = |n\rangle\langle n|$, which fulfill $\hat{E}_n\hat{E}_m$=0 if $n \neq m$, and $\hat{E}_n^2 = \hat{E}_n$.

For all operators we can define an *adjoint operator* $\hat{A}^\dagger$ by $\langle\psi|\hat{A}^\dagger\chi\rangle = \langle\hat{A}\psi|\chi\rangle$. For many operators, we can also define an inverse operator $\hat{A}^{-1}$ by $\hat{A}\hat{A}^{-1} = \hat{I}$

Two important types of operators are *hermitian* operators $\hat{H}$ and *unitary* operators $\hat{U}$. For any two states $|\psi\rangle$, $|\chi\rangle$, hermitian operator fulfill $\langle\psi|\hat{H}\chi\rangle = \langle\hat{H}\psi|\chi\rangle$, while unitary operators fulfill $\langle\hat{U}\psi|\hat{U}\chi\rangle = \langle\psi|\chi\rangle$. This entails $\hat{H} = \hat{H}^\dagger$ and $\hat{U}^\dagger = \hat{U}^{-1}$. Both classes of operators have the nice property that their sets of normalized eigenvectors form an orthonormal basis. For hermitian operators, the eigenvalues a_n are real, while for unitary operators they fulfill $|a_n| = 1$.

C. Dynamics

The time evolution of quantum states is governed by the *Schrödinger equation*

$$i\hbar\frac{d}{dt}|\psi(t)\rangle = \hat{H}(t)|\psi(t)\rangle,$$

where $\hat{H}$ is a hermitian operator called *Hamiltonian*. Given an initial state $|\psi(t_0)\rangle$, the general solution can be written as $|\psi(t)\rangle = \hat{U}(t,t_0)|\psi(t_0)\rangle$, where $\hat{U}(t,t_0)$ is a unitary operator called the *time evolution operator*. This operator fulfills the Schrödinger equation $i\hbar\frac{d}{dt}\hat{U}(t,t_0) = \hat{H}(t)\hat{U}(t,t_0)$ with initial condition $\hat{U}(t_0,t_0) = \hat{I}$.

In the particular case $\hat{H} = \mathrm{const}(t)$ of a time-independent Hamiltonian, the time evolution operator takes the form

$$\hat{U}(t,t_0) = \exp[-i(t-t_0)\hat{H}/\hbar],$$

where the exponential function of an operator is defined as $\exp\hat{A} = \sum_{k=0}^{\infty}\hat{A}^k/k!$. Using the eigenrepresentation $\hat{H} = \sum_n E_n|n\rangle\langle n|$ of the Hamiltonian we can write $\hat{U}(t,t_0) = \sum_n \exp[-i(t-t_0)E_n/\hbar]|n\rangle\langle n|$.

D. Measurements

Measurements deliver information about observable properties (*observables*) of quantum systems. Quantum mechanics associates to each observable A a hermitian operator $\hat{A}$. The eigenvalues a_n of $\hat{A}$ are the (only) possible outcomes of the experiment. Each outcome occurs with a probability given by

$$P(A = a_n) = |\langle n|\psi\rangle|^2 = \langle\psi|n\rangle\langle n|\psi\rangle = \langle\psi|\hat{E}_n|\psi\rangle,$$

where $|n\rangle$ is the eigenvector associated with a_n, and $\hat{E}_n = |n\rangle\langle n|$ is the associated projection operator. The average $\langle A\rangle_\psi = \sum_n P_n a_n$ of the outcome of many identical experiments, known as the *expectation value* of A, can be calculated directly from the quantum state using $\langle A\rangle_\psi = \langle\psi|\hat{A}|\psi\rangle$.

In the simplest case, a measurement with outcome a_n transforms the state of the system into the

eigenstate $|n\rangle$. In general, however, the values of one observable alone do not suffice to uniquely determine the state of a system. Instead, a complete description requires to measure a larger set $\hat{A}^{(l)}$ of *simultaneous observables*. Such observables commute with each other, $[\hat{A}^{(l)}, \hat{A}^{(m)}] = 0$, where $[\hat{A}, \hat{B}] = \hat{A}\hat{B} - \hat{B}\hat{A}$ is the *commutator*. This property guarantees that one can find a joint eigenbasis, given by states $|\psi_{\mathbf{a}}\rangle = |a^{(1)}, a^{(2)}, a^{(3)}, \ldots\rangle$ fulfilling $\hat{A}^{(l)}|\psi_{\mathbf{a}}\rangle = a^{(l)}|\psi_{\mathbf{a}}\rangle$. These states are only fully specified by knowledge of the eigenvalues $a^{(l)}$ of the full set of simultaneous observables, which we here grouped into a vector $\mathbf{a}$ with components $\mathbf{a}_l = a^{(l)}$.

In this situation, a specific measurement outcome $a_n^{(l)}$ for a single observable $\hat{A}^{(l)}$ delivers only incomplete information about the quantum system. In order to describe the effect of such a measurement, let us introduce the projection operator $\hat{E}_n^{(l)} = \sum_{\mathbf{a}_l = a_n^{(l)}} |\psi_{\mathbf{a}}\rangle\langle\psi_{\mathbf{a}}|$ onto all states that share the given eigenvalue $a_n^{(l)}$. A measurement with outcome $a_n^{(l)}$ then occurs with probability $P_n = \langle\psi|\hat{E}_n|\psi\rangle$, and transforms the quantum state into the (not yet normalized) state $|\chi_n\rangle = \hat{E}_n|\psi\rangle$. The normalized post-measurement state is given by $|\psi_n\rangle = \sqrt{1/P_n}|\chi_n\rangle$. Since the other observables remain undetermined, such an incomplete measurement does not force the system into a unique final state.

E. Density matrix

An *ensemble* is a large collection of physically identical quantum systems, which however can be described by different states. When all the states are identical the ensemble is said to be *pure*, otherwise it is *mixed*. In general, we specify that a fraction P_i of states is in state $|\psi_i\rangle$, where $\sum_i P_i = 1$ and $\langle\psi_i|\psi_i\rangle = 1$. Starting from a pure ensemble with all members in state $|\psi\rangle$, such a mixed ensemble is obtained, e.g., by measurement of an observable, with P_i and $|\psi_i\rangle$ obtained as described in the previous section. In the ensemble, expectation values are defined by $\langle A\rangle = \sum_i P_i\langle\psi_i|\hat{A}|\psi_i\rangle$.

By construction, a mixed ensemble cannot be described by a single quantum state. However, it is possible to define a *statistical operator* $\hat{\rho}$, most commonly known as the *density matrix*, which allows to calculate all expectation values in a given mixed ensemble. This operator is given by

$$\hat{\rho} = \sum_i P_i|\psi_i\rangle\langle\psi_i|,$$

and the expectation values are obtained by

$$\langle A\rangle_\rho = \mathrm{tr}\,(\hat{A}\hat{\rho}).$$

Here, $\mathrm{tr}\,\hat{B}$ denotes the *trace* of an operator, which in any given orthonormal basis can be calculated as $\mathrm{tr}\,\hat{B} = \sum_n \langle n|\hat{B}n\rangle = \sum_n B_{nn}$.

Normalization of states carries over to the property $\mathrm{tr}\,\hat{\rho} = 1$. Moreover, the density matrix is hermitian and positive definite. This entails that in its eigenrepresentation $\hat{\rho} = \sum_n p_n|n\rangle\langle n|$, all eigenvalues are nonnegative, $p_n > 0$; they also sum up to unity, $\sum_n p_n = 1$. (The nonvanishing eigenvalues p_n are only identical to the values P_i if the states $|\psi_i\rangle$ used to define the ensemble are orthogonal to each other.)

For a pure ensemble, $p_n = 1$ for one state, while all the other $p_m = 0$ ($m \neq n$). In this case, $\hat{\rho} = |n\rangle\langle n| = \hat{E}_n$ is a projection operator, and therefore fulfills $\hat{\rho}^2 = \hat{\rho}$. It follows that for a pure state $\mathrm{tr}\,\hat{\rho}^2 = 1$. For a mixed state, however $\mathrm{tr}\,\hat{\rho}^2 = \sum_n p_n^2 < 1$. The quantity $\mathcal{P} = \mathrm{tr}\,\hat{\rho}^2$, also known as the *purity*, therefore easily distinguishes pure from mixed states. The maximally mixed state is described by the density matrix $\hat{\rho} = \frac{1}{\mathcal{N}}\hat{I}$ (where $\mathcal{N}$ is the Hilbert space dimension), and has purity $\mathcal{P} = 1/\mathcal{N}$.

In a given representation, the density matrix of a pure state $|\psi\rangle$ can be obtained from $\rho = \psi\psi^\dagger$, which is useful for specific calculations.

The time evolution of the density matrix follows from the Schrödinger equation, and is given by $\frac{d}{dt}\hat{\rho} = \frac{i}{\hbar}[\hat{\rho}, \hat{H}]$. The general solution can be written as $\hat{\rho}(t) = \hat{U}(t, t_0)\hat{\rho}(t_0)\hat{U}^\dagger(t, t_0)$, where $\hat{U}$ is the unitary time evolution operator defined in section I.C.

F. Two-state systems

Given an orthonormal basis $|0\rangle$, $|1\rangle$ of a two-state system, each state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is represented by a two-component vector $\begin{pmatrix}\alpha\\ \beta\end{pmatrix}$. Each hermitian operator $\hat{H} = a_0\hat{I} + a_x\hat{X} + a_y\hat{Y} + a_z\hat{Z}$ can be formed from four elementary operators with matrix representation

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

The matrices X, Y and Z are the *Pauli matrices*, most familiar from the description of the spin of an electron where they are often denoted as σ_x, σ_y and σ_z, respectively. They fulfill $X^2 = Y^2 = Z^2 = I$, $XY = -YX = iZ$, $YZ = -ZY = iX$, $ZX = -XZ = iY$.

It is useful to characterize the state of a two-state system by the vector of expectation values

$$\vec{P} = (\langle X\rangle, \langle Y\rangle, \langle Z\rangle),$$

which is known as the *polarization vector*. For a normalized pure state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$,

$$\vec{P} = (2\,\mathrm{Re}\,\alpha^*\beta, 2\,\mathrm{Im}\alpha^*\beta, |\alpha|^2 - |\beta|^2)$$

is of unit length, and therefore lies on a sphere called the *Bloch sphere*. In terms of spherical polar coordinates on this sphere,

$$|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\phi}\sin(\theta/2)|1\rangle.$$

The azimuthal angle $\phi = \arg\alpha^*\beta$ of this vector is also known as the *phase* of the state. For a mixed state, $|\vec{P}| < 1$ so that the vector lies within the sphere. In terms of these expectation values, the density matrix can be written as

$$\rho = \frac{1}{2}\begin{pmatrix} 1+P_z & P_x - iP_y \\ P_x + iP_y & 1-P_z \end{pmatrix} = \frac{1}{2}(I + P_x X + P_y Y + P_z Z).$$

The purity of this density matrix is given by $\mathcal{P} = \frac{1}{2}(1+|\vec{P}|^2)$.

G. Composite systems and entanglement

An important example where simultaneous observables occur are composite systems (say, a system composed of parts 1 and 2), where incomplete information can be acquired by measuring an observable of a subsystem (say, part 1). Starting from an orthonormal basis $|n\rangle$ ($n = 1, \ldots, \mathcal{N}_1$) for system 1 and $|m\rangle$ ($m = 1, \ldots, \mathcal{N}_2$) for system 2, the joint state $|\psi\rangle = \sum_{nm}\psi_{nm}|nm\rangle$ of the composite system can be written by using combined basis states $|nm\rangle$, sometimes also written as $|n\rangle|m\rangle$ or $|n\rangle \otimes |m\rangle$. The corresponding dual basis vectors are denoted by $\langle nm|$. The Hilbert space dimension of the composite system is therefore given by $\mathcal{N} = \mathcal{N}_1\mathcal{N}_2$. General operators can be written as $\hat{A} = \sum_{nmkl} A_{nk,ml}|nk\rangle\langle ml|$. Operators acting on subsystem 1 will be denoted by $\hat{A}_1$, and have representation $\hat{A}_1 = \sum_{nmk} A^{(1)}_{nm}|nk\rangle\langle mk|$. Operators acting on subsystem 2 will be denoted by $\hat{A}_2$, and have representation $\hat{A}_2 = \sum_{nkl} A^{(2)}_{kl}|nk\rangle\langle nl|$. This results in the convenient *block matrix* form

$$A_1 = \begin{pmatrix} A^{(1)}_{11} I & A^{(1)}_{12} I & \cdots \\ A^{(1)}_{21} I & A^{(1)}_{22} I & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}, \quad A_2 = \begin{pmatrix} A^{(2)} & 0 & \cdots \\ 0 & A^{(2)} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix},$$

where I is the $\mathcal{N}_2 \times \mathcal{N}_2$-dimensional identity matrix. Here, the basis states are ordered as $|1,1\rangle, |1,2\rangle, \ldots, |1,\mathcal{N}_2\rangle, |2,1\rangle, |2,2\rangle, \ldots$.

Sometimes, the state of a composite system can still be written as the product $|\varphi\rangle|\chi\rangle$ of two states, where $|\varphi\rangle$ describes system 1 and $|\chi\rangle$ describes system 2. Such states are called *separable*. This requires that the coefficients can be written as $\psi_{nm} = \varphi_n\chi_m$. States that are not separable are called *entangled*.

In order to determine whether states are separable or entangled, it is useful to consider measurements of observables of one subsystem, say system 1. When a state is separable, $|\psi\rangle = |\varphi\rangle|\chi\rangle$, the outcome of such measurements only depends on $|\varphi\rangle$. However, when the system is entangled, measurements on one subsystem cannot be described by a single state of that system. It is then still possible to describe these measurements by a density matrix

$$\hat{\rho}_1 = \sum_{nmk}\langle nk|\psi\rangle\langle\psi|mk\rangle|n\rangle\langle m|,$$

known as the *reduced density matrix*. This means that all expectation values can be computed according to $\langle A\rangle = \operatorname{tr}\hat{A}\hat{\rho}_1$. Analogously, measurements of the second subsystem are described by a reduced density matrix $\hat{\rho}_2 = \sum_{nkl}\langle nk|\psi\rangle\langle\psi|nl\rangle|k\rangle\langle l|$. If a state $|\psi\rangle$ is separable, the reduced density matrices are pure, i.e., $\operatorname{tr}\hat{\rho}_1^2 = \operatorname{tr}\hat{\rho}_2^2 = 1$. If the state $|\psi\rangle$ is entangled, the reduced density matrices are both mixed, i.e., $\operatorname{tr}\hat{\rho}_1^2 = \operatorname{tr}\hat{\rho}_2^2 < 1$.

Reduced density matrices can also be defined when the composite system is already in a mixed state, described by a density matrix $\hat{\rho}$. They are then given by

$$\hat{\rho}_1 = \sum_{nmk}\langle nk|\hat{\rho}|mk\rangle|n\rangle\langle m|, \quad \rho_2 = \sum_{nkl}\langle nk|\hat{\rho}|nl\rangle|k\rangle\langle l|.$$

These constructions are also called *partial trace*, and then written as $\hat{\rho}_1 = \operatorname{tr}_2\hat{\rho}$, $\hat{\rho}_2 = \operatorname{tr}_1\hat{\rho}$. This designation becomes clear when one considers the block form

$$\rho = \begin{pmatrix} \rho^{(2)}_{11} & \rho^{(2)}_{12} & \cdots \\ \rho^{(2)}_{21} & \rho^{(2)}_{22} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

of the density matrix in the composite basis, where $\rho^{(2)}_{nm}$ are $\mathcal{N}_2 \times \mathcal{N}_2$-dimensional matrices. Then,

$$\rho_1 = \begin{pmatrix} \operatorname{tr}\rho^{(2)}_{11} & \operatorname{tr}\rho^{(2)}_{12} & \cdots \\ \operatorname{tr}\rho^{(2)}_{21} & \operatorname{tr}\rho^{(2)}_{22} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}, \quad \rho_2 = \sum_n \rho^{(2)}_{nn}.$$

In this more general case of a composite system with a mixed density matrix, the purities of both reduced density matrices do not need to be identical, and cannot simply be used to decide whether the system is entangled or not; this is discussed in more detail below.

As an (important) example, consider the composition of two two-state systems. Pure states can be written as $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$, and are normalized if $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$. The entanglement of such a state is often characterized by the *concurrence*

$$\mathcal{C} = 2|\alpha\delta - \beta\gamma|,$$

which fulfills $0 \leq \mathcal{C} \leq 1$. For separable states, $\mathcal{C} = 0$, i.e., the concurrences vanishes. For entangled states, $\mathcal{C} > 0$. States with $\mathcal{C} = 1$ are called maximally entangled. Examples of maximally entangled states are the four *Bell states*

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

For the pure state given above, the full density matrix

$$\rho = \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix} (\alpha^*, \beta^*, \gamma^*, \delta^*) = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

can be conveniently written in block form, where A, B, C, and D are 2×2-dimensional matrices. The reduced density matrix

$$\rho_1 = \begin{pmatrix} \mathrm{tr}\, A & \mathrm{tr}\, B \\ \mathrm{tr}\, C & \mathrm{tr}\, D \end{pmatrix}$$

can then be obtained by taking traces of the blocks, which here results in

$$\rho_1 = \begin{pmatrix} |\alpha|^2 + |\beta|^2 & \alpha\gamma^* + \beta\delta^* \\ \gamma\alpha^* + \delta\beta^* & |\gamma|^2 + |\delta|^2 \end{pmatrix}.$$

Similarly,

$$\rho_2 = A + D = \begin{pmatrix} |\alpha|^2 + |\gamma|^2 & \alpha\beta^* + \gamma\delta^* \\ \beta\alpha^* + \delta\gamma^* & |\beta|^2 + |\delta|^2 \end{pmatrix}.$$

The purity of these reduced density matrices is related to the concurrence,

$$\mathrm{tr}\, \hat{\rho}_1^2 = \mathrm{tr}\, \hat{\rho}_2^2 = 1 - \mathcal{C}^2/2.$$

Furthermore, we have the identity $\det \hat{\rho}_1 = \mathcal{C}^2/4$.

For composite systems in a pure state, the reduced density matrix also delivers the *entanglement of formation* $\mathcal{E} = -\mathrm{tr}\,(\hat{\rho}_1 \log_2 \hat{\rho}_1) = -\mathrm{tr}\,(\hat{\rho}_2 \log_2 \hat{\rho}_2)$, which is identical to the *von Neumann entropy* of the mixed states of the subsystems (see section III.B).

In their form discussed above, these measures of entanglement only apply to *pure* states of a composite system. Entanglement measures for multicomponent systems with a mixed density matrix are an active field of research. Well understood is only the case of *two composite two-level systems*, for which entanglement measures can be computed efficiently from the 4×4 dimensional density matrix ρ of the composite system in the standard basis. In order to obtain the concurrence, one needs to compute the four eigenvalues λ_i of the matrix $\rho(Y_1 Y_2)\rho^*(Y_1 Y_2)$, where Y_1 and Y_2 are the Y Pauli matrix acting on subsystem 1 and 2, respectively. When the eigenvalues are ordered such that $\lambda_1 > \lambda_2 > \lambda_3 > \lambda_4$, the concurrence is given as $\mathcal{C} = \max(0, \sqrt{\lambda_1} - \sqrt{\lambda_2} - \sqrt{\lambda_3} - \sqrt{\lambda_4})$. The entanglement of formation $\mathcal{E} = \min_{\mathrm{dec}} \sum_i P_i \mathcal{E}(\psi_i)$ is generalized by minimizing the averaged pure-state entanglement of formation over all possible decompositions $\hat{\rho} = \sum_i P_i |\psi_i\rangle\langle\psi_i|$ of the density matrix (where the states $|\psi_i\rangle$ do not need to be orthogonal). Remarkably, both entanglement measures are related by the general formula $\mathcal{E} = h(\frac{1}{2} + \frac{1}{2}\sqrt{1 - \mathcal{C}^2})$, where $h(x) = -x \log_2 x - (1-x)\log_2(1-x)$.

H. Bell inequalities

Entanglement is physically significant because it results in correlations that cannot be described by classical probabilities. These correlations can be uncovered by statistical tests, known as *Bell inequalities*. The most transparent inequality is the *CHSH inequality* due to Clauser, Horn, Shimony, and Holt. Consider the composition of two two-state systems; to be explicit, think of the spins of two electrons with basis states $|0\rangle = |\uparrow\rangle$ and $|1\rangle = |\downarrow\rangle$. On each spin we carry out two different experiments, described by observables $\hat{A}_1$, $\hat{A}_1'$, $\hat{B}_2$, and $\hat{B}_2'$, which measure whether the spin points into a particular direction. To the outcome of each experiment we designate the value 1 or -1, depending on whether the spin is found to be aligned parallel or antiparallel to the measurement direction, respectively. Now consider the expectation value of

$$\hat{F} = (\hat{A}_1 + \hat{A}_1')\hat{B}_2 - (\hat{A}_1 - \hat{A}_1')\hat{B}_2'.$$

Classically, for each combination of outcomes, F is either 2 or -2, and therefore on average $\langle F \rangle \leq 2$, which is the CHSH inequality. Quantum-mechanically, the average is obtained by an expectation value. Let us choose $\hat{A}_1 = Z$, $\hat{A}_1' = X$, $\hat{B}_2 = -\sqrt{1/2}(X + Z)$, and $\hat{B}_2' = \sqrt{1/2}(Z - X)$, so that $\hat{F} = -\sqrt{2}(X_1 X_2 + Z_1 Z_2)$, which is represented by the matrix

$$F = -\sqrt{2}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

Furthermore, assume that the system is in the Bell state $|\beta_{11}\rangle = \sqrt{1/2}(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle)$, represented by the

vector

$$\psi = \sqrt{1/2} \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix}.$$

We then find $\langle \hat{F} \rangle = \psi^\dagger F \psi = 2\sqrt{2}$, which violates the CHSH inequality. The reason are quantum-mechanical correlations that arise as a consequence of the entanglement of the Bell state. Quantum computation taps into this resource to achieve tasks that are classically impossible.

II. CLASSICAL COMPUTATION

A. von Neumann architecture

Computers store and process information. All common computers feature a combination of input and output devices, memory, processing unit, and control unit, which is known as the *von Neumann architecture*. This architecture is designed to store and process information flexibly by means of programs, which encode a sequence of instructions. These features also apply to the proposed quantum computers. The difference between both types of devices consists in how they represent and manipulate the information. In this section, we describe how this works for classical computers.

B. Binary representation of information

Classical computers represent information digitally as binary numbers $x = \sum_{n=0}^{N-1} x_n 2^n$, where the binary digits (or *bits*) x_n can take values 0 or 1, and can be presented as a sequence $x_{N-1} \ldots x_2 x_1 x_0$ (sometimes, leading zeros are dropped). For example, the binary number 11 is identical to the decimal number 3, and the binary number 100010 is identical to the decimal number 34. These numbers can be interpreted in many different ways — they may represent a letter (as in the ASCII code), or the color or brightness of a pixel on a computer monitor. They therefore represent various types of *information*.

C. Quantifying classical information

From a physical perspective, the transmission of large amounts of binary data results in an irregular sequence of discrete events which is best described in the language of statistical mechanics. This provides means to precisely quantify the information content of the data, as well as its degradation due to imperfections in transmission and manipulation processes.

Consider a register with N binary digits (bits). This can represent $\mathcal{N} = 2^N$ different numbers. When we add a register with M bits, this increases to $\mathcal{N}\mathcal{M} = 2^{N+M}$ (where $\mathcal{M} = 2^M$). However, since physically we simply added components, the data capacity of the composed register is far more conveniently specified by describing it as an $(N+M)$-bit register. The desired additive measure of storage capacity is therefore $N = \log_2 \mathcal{N}$, $M = \log_2 \mathcal{N}$, where $\log_2$ is the logarithm with base 2 (such that $\log_2 2 = 1$). This is nothing else but the entropy

$$S = -\sum_{n=1}^{\mathcal{N}} p_n \log_2 p_n$$

of a system with $\mathcal{N}$ microstates, which are all occupied with equal probability $p_n = \mathcal{N}^{-1}$. The entropy S defined here is known as the *Shannon entropy*, and differs from the entropy in thermodynamics only by a multiplicative factor $k_B \ln 2$ (where k_B is Boltzmann's constant).

The entropy is a convenient measure of information content particularly when one considers that we usually do not make best use of the register states. E.g., the ASCII code only has 7 bits (the 8th bit in a byte can therefore be used to check for errors, i.e., it adds *redundancy*). Moreover, in texts we only use printable characters, and some characters (such as e) occur far more frequently than others (such as q). The different register states then no longer occur with equal probability p_n, which is exploited by compression algorithms (frequent symbols—or combinations of symbols—are abbreviated by short bit sequences). The entropy tells us to how many bits a data stream can be ideal compressed—this is *Shannon's noiseless channel coding theorem*.

The entropy can also be used to quantify the degradation of the information content due to transmission errors. Assuming that bits are flipped randomly with error rate p, the capacity of a channel reduces by a factor $C = 1 + p \log_2 p + (1-p) \log_2 (1-p)$. Another source of errors are lost data packages—this is especially prevalent in wireless transmission and satellite communication.

Errors can be detected and corrected by adding additional bits—naively, by repeatedly sending the same data stream; more effectively, by adding other means of redundancy, such as using the 8th bit in the ASCII code for a parity check. Fortunately, *Shannon's noisy channel coding theorem* ensures the existence of error-correction codes which make the error probability arbitrarily small (see section VI.A for examples of error-correction schemes).

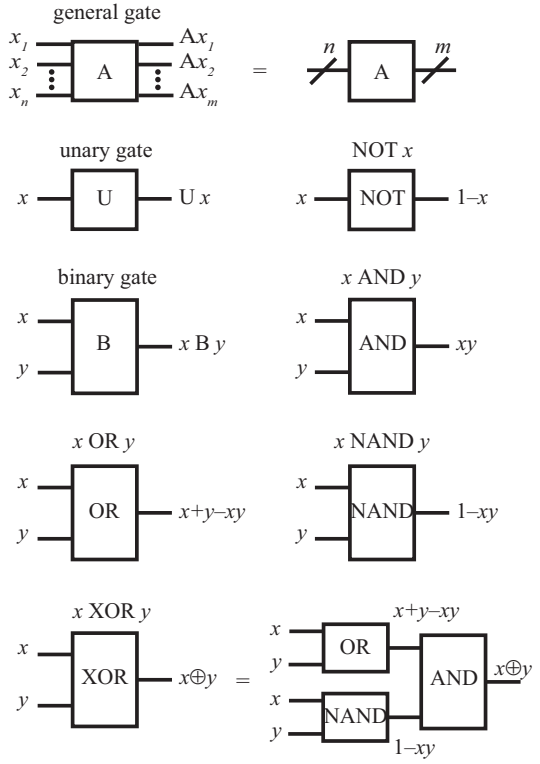

FIG. 1 Circuit representation of elementary classical gates.

D. Operations

Consider a memory unit which contains N bits. Such a unit is also called a *register*. The specified register can represent $\mathcal{N} = 2^N$ binary numbers, which range from 0 to $2^N - 1$. In order to carry out computations, we need facilities to transform any of these numbers into any other number in the same range. It turns our that this is possible with a sequence of operations which act either only on one bit (called unary operations), or on a pair of bits (called binary operations). Furthermore, because bits only take two possible values, these operations can be formulated using the concepts of Boolean logic, the mathematical discipline of TRUE and FALSE statements, which are conventionally associated with the values 1 and 0, respectively.

Each type of logical operation is called a *gate*. Graphically, a gate is represented by a box, with horizontal lines to the left representing input channels and horizontal lines to the right representing output channels (see Fig. 1). The operation of the gate is specified by a truth table, which specifies the output values as function of the input values. Below, for compactness, we express these outputs as simple algebraic functions of the inputs.

E. Unary and binary gates

There are four unary gates (i.e., gates that take one bit as input): The *identity gate* $\mathrm{Id}\, x = x$ which lets the bit intact, the *not gate* $\mathrm{NOT}\, x = 1 - x$ which inverts (or *flips*) the value of the bit, and the two *reset gates* $\mathrm{ALWAYS\,TRUE}\, x = 1$, $\mathrm{ALWAYS\,FALSE}\, x = 0$, for which the output is independent of the input.

There are 16 binary gates (gates that take two bits as input), of which the following ones are particularly noteworthy: The *and gate* $x\,\mathrm{AND}\,y = xy$, which outputs TRUE only if both x and y are TRUE, the *or gate* $x\,\mathrm{OR}\,y = x + y - xy$, which outputs TRUE if at least one of x and y are TRUE, the *exclusive-or gate* $x\,\mathrm{XOR}\,y = x + y - 2xy \equiv x \oplus y$ (where $\oplus$ denotes addition modulo 2), and the *negated-and gate* $x\,\mathrm{NAND}\,y = 1 - xy$. These gates are not independent; for example, we can write

$$x\,\mathrm{XOR}\,y = (x\,\mathrm{OR}\,y)\,\mathrm{AND}\,(x\,\mathrm{NAND}\,y).$$

Indeed, it is possible to write all unary and binary operations only using the NAND gate (e.g., $\mathrm{NOT}\, x = x\,\mathrm{NAND}\,x$). Together with the capability to replicate (or copy) information, this is sufficient to achieve all possible operations.

F. Reversible gates

With exception of Id and NOT, the gates discussed so far are irreversible: knowing only the output, it is impossible to infer the input. It is noteworthy that one can also implement classical computations with a set of reversible gates, which have additional output channels that allow to infer the input values (see Fig. 2). The most important example is the *controlled-not gate* $x\,\mathrm{CNOT}\,y = (x, x\,\mathrm{XOR}\,y)$, which flips the *target*

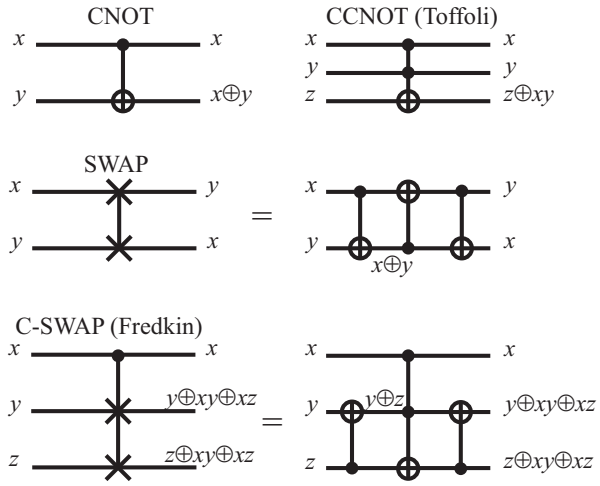

FIG. 2 Circuit representation of some reversible classical gates.

bit y if the *control bit* x is TRUE, and otherwise leaves the target bit unchanged (the first bit therefore controls whether the second bit is flipped or not). Because $x\,\mathrm{CNOT}\,0 = (x, x)$, this gate can be used to copy information. Another example is the SWAP gate $x\,\mathrm{SWAP}\,y = (y, x)$, which interchanges the values of the two bits. As indicated in Fig. 2, this can be implemented by combining three CNOT gates. The figure also shows two important reversible three-bit gate, the *Toffoli gate* (also known as CCNOT gate), which transforms $(x, y, z) \to (x, y, (x\,\mathrm{AND}\,y)\,\mathrm{XOR}\,z)$, and the Fredkin gate (also known as *controlled swap*), which transforms $(0, y, z) \to (0, y, z)$ and $(1, y, z) \to (0, z, y)$. A universal set of reversible gates needs to include at least one of these three-bit gates.

G. Complexity of computational tasks

We will later see that quantum computers can achieve certain tasks in a relatively small number of operations. In order to gauge their efficiency, it is useful to distinguish computational tasks by their *complexity*. Let us assume we want to operate on registers with N bits, and denote the number of operations required for specific a task (such as the multiplication of two numbers with $N/2$ bits) by T. If T grows as a polynomial with N, the task is relatively simple; this pertains, e.g., to arithmetic and algebraic tasks, such as matrix inversion. Problems of this kind are grouped into complexity class P. Many problems, however, are much harder, and require a number of operations T which grows exponentially with N. A notable example is the factorization of large numbers into prime factors, for which no algorithm in P is known. For this specific problem it is, however, easy to verify that the solution (once found) is correct — this simply requires multiplication of the factors, which is a task in P. Such problems are called *nondeterministic polynomial*, and grouped in class *NP*. Interestingly, there are problems to which all other *NP* problems can be reduced; such problems are called *NP-complete*, and a large number of them is known. What is *not* known is the answer to the fundamental question whether the complexity classes P and *NP* actually differ — one cannot exclude that there is an undiscovered algorithm which solves an *NP*-complete problem in a polynomial number of operations. If such an algorithm would be identified, it could be used to solve *all NP* problems much more efficiently than presently possible.

H. Practical issues

In order to build a workable computer, the theoretical concepts presented here must be complemented by practical considerations. Available computers preferably use irreversible logics based on gates which dissipate energy, not least because this provides a much larger degree of stability than reversible gates. For efficiency, they use more than a minimally required set of gates, which reduces the number of necessary operations. Furthermore, they use many different methods to physically represent the bits, e.g., by means of different voltages in components of the electronic circuit in the processing unit, or by different magnetization of domains in hard-disk storage units. Over the years, the feature size of electronic circuits has shrunken steadily, which slowly but surely brings them close to the threshold where they become susceptible to quantum effects. As we will see in the next sections (III-V), these effects are not necessarily unwelcome, as they can be exploited to achieve tasks that are hard to achieve with classical computers, which leads to the concept of a quantum computer. However, quantum computation is far more error prone than classical computation, which brings about a new range of practical issues that are discussed in section VI.B.

III. QUANTUM INFORMATION REPRESENTATION AND MANIPULATION

This section introduces quantum bits (qubits) and quantum gates, and discusses some underlying concepts such as quantum parallelism (arising from the superposition principle), entanglement as a resource for computation, and the no-cloning theorem. These concepts clearly distinguish quantum computers from classical computers, and are the reason why (in principle) the former are far more powerful than the latter.

A. Quantum bits

Quantum computers encode information into the quantum state of a composite system, consisting of a number of two-state systems known as *qubits* (quantum bits). Taken individually, each qubit can be described by a quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. The basis states $|0\rangle$ and $|1\rangle$ form the *computational basis*, and represent classical bits with values 0 and 1. What is special about qubits is that their general state $|\psi\rangle$ can also be a superposition of 0 and 1, which cannot be realized by a classical bit. In this case, $|\alpha|^2$ and $|\beta|^2$ give the probabilities to find 0 or 1 in a measurement of the qubit in the computational basis [associated with the operator

$(I - Z)/2$], which on the Bloch sphere depend on the polar angle θ. Measurements of other observables (like X or Y) give access to other combinations of α and β, which also depend on the phase $\phi = \arg(\alpha^*\beta)$ of the qubit. However, since measurements change the state of the qubit, it is not possible to directly encode information into the complex numbers α and β; this distinguishes quantum computers from analog computers.

An N-bit quantum register contains N qubits. The state of the register is then formed using the $\mathcal{N} = 2^N$ computational basis states $|x_{N-1}\dots x_1x_0\rangle$, where $x_{N-1}\dots x_1x_0$ is the binary code of numbers $x = \sum_{n=0}^{N-1} x_n 2^n$ ranging from 0 to $2^N - 1$. Here, we will not drop leading zeros (thus, for $N = 4$ we write 0010 for the decimal number 2) because they describe the state of some of the qubits. Sometimes, we denote these states by the corresponding numbers x; e.g., for a 4-bit register, $|3\rangle \equiv |0011\rangle$ and $|0\rangle \equiv |0000\rangle$.

Just as the individual qubits, the register may be brought into a superposition of the computational basis states. For instance, one can form the state

$$|\Psi\rangle \equiv 2^{-N/2} \sum_{x_n=0,1} |x_{N-1}\dots x_1x_0\rangle = 2^{-N/2} \sum_{x=0}^{2^N-1} |x\rangle,$$

which contains all binary numbers between 0 and $2^N - 1$ *at the same time*. All these numbers therefore can be manipulated *at once* by a *single* physical operation on the register. This feature is sometimes described as *quantum parallelism*, and will be frequently exploited in Section V.

The state $|\Psi\rangle$ given above is still separable, and therefore can be obtained by operating on the individual qubits. The real power of the register is unleashed when one considers that the qubits in the register can also be entangled, which can be achieved by manipulating pairs of qubits. We next explore how entanglement is linked to information and then have a look at a number of basic quantum operations on the register which enable to exploit these special features of qubits.

B. Quantum information

Quantum mechanically, the entropy of a system with density matrix $\hat\rho$ is given by the *von Neumann entropy*

$$S(\hat\rho) = -\mathrm{tr}\,\hat\rho \log_2 \hat\rho.$$

This is identical to the Shannon entropy, with the probabilities p_n replaced by the eigenvalues of $\hat\rho$. It follows that a system in a pure state has von-Neumann entropy $S = 0$.

Classically, the entropy of a subsystem A is always less than the entropy of a composite system with parts A and B: $S_A, S_B < S_{AB}$. Quantum mechanically, the entropy $S(\hat\rho_A)$ of a subsystem follows by inserting the reduced density matrix, and *can* be larger than the entropy of a composed system. This is in particular the case when an entangled systems is in a pure state. The entropy of the composite system vanishes, but it is finite for each of its parts because the reduced density matrices are mixed (see section I.G). For composite systems that are already in a mixed state, this leads to the more sophisticated measures of entanglement also mentioned in that section.

In terms of data compression, the von Neumann entropy plays a similar role as the Shannon entropy: it describes how many qubits are necessary to transmit a certain amount of (quantum) information (this can be quantified via *Schumacher's noiseless channel coding theorem*).

How much information can be transmitted via a single qubit? Assume that the incoming data stream is composed of states described by a density matrix $\hat\rho_i$, which can be assumed to be mixed because of limitations in the preparation or transmission of the qubit states. It then can be shown that the maximally accessible amount of information I per qubit cannot exceed a certain limit, $I < S(\hat\rho) - \sum_n P_i S(\hat\rho_i) \equiv \chi$, where $\hat\rho = \sum_i P_i\hat\rho_i$, and P_i is the fraction of qubits with density matrix $\hat\rho_i$. This inequality is known as the *Holevo bound*, and the quantity χ is the *Holevo information*. Note that when all $\hat\rho_i$ are pure, $S(\hat\rho_i) = 0$, and therefore $\chi = S(\hat\rho)$.

The Holevo bound implies that a single qubit cannot transmit more than one bit of classical information. However, as embodied by the superdense coding scheme discussed next, if sender and receiver possess as an additional resource a number of shared entangled qubits, it is possible to transmit *two* bits of classical information via a single (entangled) qubit. Indeed, the applications below show that entanglement is a useful, non-classical resource for communication. Entanglement can also be used for error correction schemes, as will be discussed in section VI.A.

In order to quantify entanglement of a (many-body) state $|\psi\rangle$, it is useful to determine how many maximally entangled Bell pairs one could generate (*by local operations within each subsystem, and classical communication*) if one had many copies of the state $|\psi\rangle$. This results in the *entanglement of formation*, which is an additive measure of entanglement, and therefore constitutes the appropriate counterpart to the information measure provided by the Shannon entropy. As discussed in Sec. I.G, for pure states the entanglement of formation simply reduces to the von Neumann entropy of the subsystems. This applies even for the case that each subsystem has more than two possible quan-

tum states. For mixed states, however, explicit expressions are only known for some special cases, such as for two qubit-systems (see again Sec. I.G).

C. Quantum gates as unitary operations

Since solutions of the Schrödinger equation can be written as a time-dependent unitary transformation of the initial state, all quantum gates are represented by unitary operators, $|\psi_{\rm f}\rangle = \hat{U}|\psi_{\rm i}\rangle$, where $|\psi_{\rm i}\rangle$ is the initial state of the register, and $|\psi_{\rm f}\rangle$ is the final state of the register. Computational algorithms furthermore complement these gates by mechanisms to prepare the register in an initial state, as well as mechanisms to read out the final state (which can be achieved by measurements). In the remainder of this section we concentrate on the unitary quantum gates.

First, a general observation: Because unitary operators can be inverted, all quantum gates are *reversible*: the input state can be inferred from $|\psi_{\rm i}\rangle = \hat{U}^\dagger|\psi_{\rm f}\rangle$. This also entails an important constraint onto quantum operations: The *no-cloning theorem*, according to which it is impossible to transfer the state of a control qubit to a target qubit without erasing the state of the control qubit.

Even though the quantum gates are reversible, a universal set of gates can be formed using only unary and binary gates; no three-bit gates are required. On the other hand, we need to consider a larger variety of such gates — besides achieving the classical logical tasks, we need gates that put qubits into non-classical superpositions of 0 and 1, and gates that establish entanglement between the qubits. Fortunately, this can be achieved by using a finite number of unary gates, plus a single binary gate. Circuit representations of these gates are shown in Fig. 3.

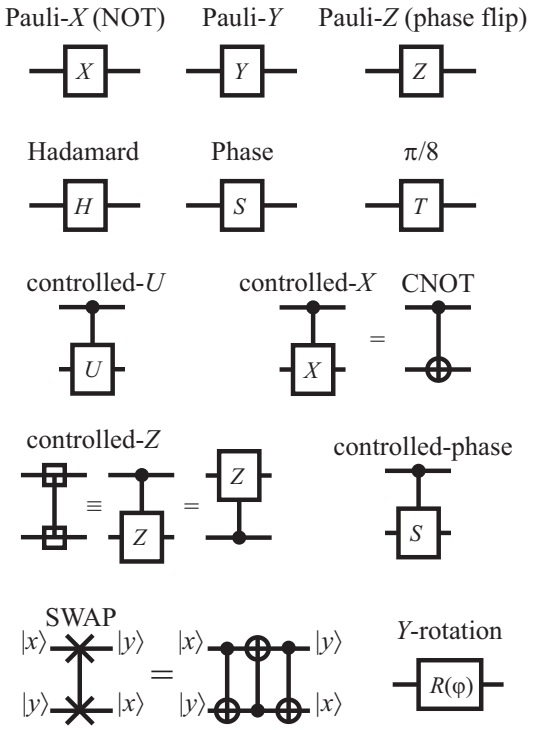

FIG. 3 Circuit representation of elementary quantum gates.

D. Single-qubit gates

All unary (single-qubit) gates can be represented as 2×2-dimensional unitary matrices, which act on the two component vector $\psi = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ representing the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Just as hermitian matrices, these can be generated by combining the Pauli matrices X, Y, and Z, as well as the identity matrix I. The latter leaves the state of the qubit unchanged, and therefore constitutes the quantum analogue to the identity gate Id. Furthermore, considering that $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$, X represents the analogue to the classical NOT gate. For a general state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ of the qubit, application of this gate yields $X|\psi\rangle = \beta|0\rangle + \alpha|1\rangle$. Because they are irreversible, the two remaining classical unary gates ALWAYS TRUE/FALSE do not have unary quantum analogues.

Clearly, I and X do not exhaust all possible transformations of a qubit. On the Bloch sphere, X represents a rotation by 180° about the x axis, while I leaves the state untouched. However, we can also operate, e.g., with Z, which flips the *phase* of the qubit (this advances the azimuthal angle ϕ by π). The most general single-qubit transformations correspond to rotations of the Bloch sphere about an arbitrary axis $\hat{\mathbf{n}} = (n_x, n_y, n_z)$, by an arbitrary angle φ. Such general rotations can be written as

$$\begin{aligned} R_{\hat{\mathbf{n}}}(\varphi) &= \exp[-i\varphi(n_x X + n_y Y + n_z Z)/2] \\ &= \cos(\varphi/2)I - i\sin(\varphi/2)(n_x X + n_y Y + n_z Z). \end{aligned}$$

Fortunately, not all rotations are independent: E.g., following from the identity $Y = iXZ$, Y (a 180° rotation about the y axis) can be obtained by combining X and Z (rotations about the x and z axis, respectively).

An example of a set of elementary rotations which can be combined to generate all possible rotations is given by the following three gates: the $\pi/8$-*gate*

$$T = \begin{pmatrix} 1 & 0 \\ 0 & (i+1)/\sqrt{2} \end{pmatrix},$$

the *phase gate*

$$S = T^2 = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix},$$

and the *Hadamard gate*

$$H = \frac{1}{\sqrt{2}}(X+Z) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Here, T and S generate $45°$ and $90°$ rotations about the z axis, respectively, while H generates a $180°$ rotation about the $(1,0,1)$ direction. Starting, say, from the initial state $|0\rangle$, these operations can be used to bring the qubit into any general superposition state $\alpha|0\rangle + \beta|1\rangle$.

Below, we will also often use φ-rotation gates about the Y axis, which have the form

$$R(\varphi) = \cos(\varphi/2)I - i\sin(\varphi/2)Y = \begin{pmatrix} \cos(\varphi/2) & -\sin(\varphi/2) \\ \sin(\varphi/2) & \cos(\varphi/2) \end{pmatrix}.$$

E. Two-qubit gates

General two-qubit gates are represented by 4×4-dimensional unitary matrices, which act on the four component vector

$$\psi = \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix}$$

representing a state $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$.

The most important gate is the quantum version of the controlled-not gate CNOT, which negates the target qubit if the control qubit is in $|1\rangle$, and leaves the target unchanged if the control qubit is in $|0\rangle$; the control qubit always remains in its initial state. If qubit 1 is the control bit, this is represented by the matrix

$$C_{12} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix};$$

for reversed roles we have

$$C_{21} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

Starting, say, from the separable state $|\psi\rangle = \sqrt{1/2}(|0\rangle + |1\rangle)|0\rangle = \sqrt{1/2}(|00\rangle + |10\rangle)$, we find $C_{12}|\psi\rangle = \sqrt{1/2}(|00\rangle + |11\rangle)$, i.e., one of the Bell states. The CNOT gate can therefore be used to entangle two qubits.

In block notation, the matrix C_{12} can also be written as $C_{12} = \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix}$. Replacing X by an arbitrary unary gate U delivers controlled versions of each unary gate, denoted as controlled-U gates. An example is the controlled phase flip, which is obtained for $U = Z$. Another notable two-qubit gate is the quantum version of the SWAP gate, which is represented by the matrix

$$S_{12} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

F. Composition of gates

In order to manipulate the information in the register, we need to apply unary and binary gates to the individual qubits. Unary gates U operating on the nth qubit in the register will be denoted by U_n. Analogously, binary gates acting on qubits n and m will be denoted by U_{nm}. Here, order of the indices matters — in general, $U_{nm} \neq U_{mn}$. In particular, for controlled gates, we will use the first index to refer to the control qubit, while the second index refers to the target qubit.

We can now combine unary and binary gates to achieve arbitrary operations on the register. For instance, the realization of the SWAP gate in terms of three CNOT gates, already known from the classical case, takes the form $S_{12} = C_{12}C_{21}C_{12}$.

The combination of quantum gates allows to achieve tasks that would require more complicated gates if done classically. Some examples are listed in Fig. 4. Consider the operation $U = H_1H_2C_{12}H_1H_2$, which first applies Hadamard gates to the two qubits, then acts with a CNOT gate,

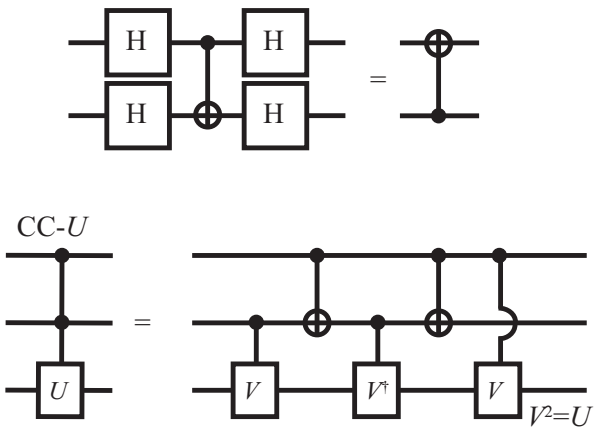

FIG. 4 Quantum circuits which implement the exchange of control and target bit, as well as the realization of a controlled-controlled-U gate.

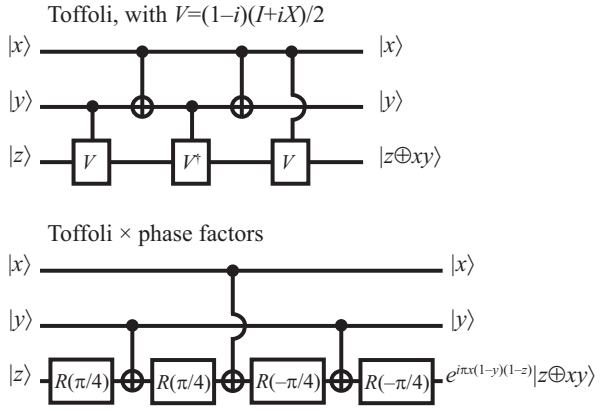

FIG. 5 Reduction of the Toffoli gate to unary and binary quantum gates. In analogy to Fig. 2, the Fredkin gate is obtained by two additional CNOT operations.

and finally again applies Hadamard gates. Using matrix multiplications, we find that for all initial states, this is equivalent to $U = C_{21}$. Therefore, amazingly, decorating the gate with single-qubit operations allows to exchange the roles of the control and target qubits. Classically, this exchange would require additional binary gates. Similarly, we can obtain classical three-bit gates by combination of unary and binary gates. For instance, as shown in the figure, a general controlled-controlled-U gate can be obtained by controlled-V gates, where $V^2 = U$.

The Toffoli gate (controlled-controlled-not, CCNOT), which classically has to be introduced separately to achieve unversal reversible computations, follows for $V = (1-i)(I+iX)/2$, such that $V^2 = X$. Just as in the classical case, the Fredkin gate is obtained by two additional CNOT operations, $F_{123} = C_{32}T_{123}C_{32}$. Up to a phase factor, the Toffoli gate can also be achieved by using Y-rotations,

$$e^{i\vartheta}T_{123} = R_3(-\pi/4)C_{13}R_3(-\pi/4)C_{23}R_3(\pi/4)C_{13}R_3(\pi/4)$$

(see Fig. 5). Here, the phase factor ϑ changes the sign of the basis state $|100\rangle$, but leaves all other basis states unchanged.

Figure 6 shows combinations of gates which serve as frequent components of the quantum algorithms discussed in the following section. (a) The gate $C_{12}H_1$ entangles qubits 1 and 2 that are initially prepared in computational basis states:

$$\begin{aligned} C_{12}H_1|00\rangle &= \sqrt{1/2}(|00\rangle + |11\rangle) \equiv |\beta_{00}\rangle, \\ C_{12}H_1|01\rangle &= \sqrt{1/2}(|01\rangle + |10\rangle) \equiv |\beta_{01}\rangle, \\ C_{12}H_1|10\rangle &= \sqrt{1/2}(|00\rangle - |11\rangle) \equiv |\beta_{10}\rangle, \\ C_{12}H_1|11\rangle &= \sqrt{1/2}(|01\rangle - |10\rangle) \equiv |\beta_{11}\rangle. \end{aligned}$$

(b) Application of $U = \prod_n H_n$ (i.e., acting with the Hadamard gate on each qubit) transforms an N-qubit register with initial state $|0\rangle = |000\ldots000\rangle$ into

$$U|\psi\rangle = 2^{-N/2} \sum_{x_n=0,1} |x_{N-1}\ldots x_2 x_1 x_0\rangle \equiv |\Psi\rangle,$$

the superposition of all states of the computational basis, which represents all binary numbers in the range of the register. (c) When the initial state is another computation basis state $|x\rangle$, the action of a Hadamard gate on an individual qubit can be written as

$$H|x_n\rangle = \frac{|0\rangle + (-1)^{x_n}|1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \sum_{z_n=0,1} (-1)^{x_n z_n}|z_n\rangle.$$

When we act with Hadamard gates on all the qubits in the register, we obtain the expression

$$\prod_n H_n|x\rangle = 2^{-N/2} \sum_z (-1)^{x\cdot z}|z\rangle,$$

where $x \cdot z = x_0 z_0 + x_1 z_1 + \ldots + x_{N-1} z_{N-1}$ denotes the *bitwise product* of x and z.

G. Function gates

An important class of gates encountered in the following applications implement functions $f : x \to f(x)$, where x is an N-bit input, and $f(x)$ is an M-bit output (see Fig. 7). In general, such functions are not invertible. The bottom panel of the figure shows a strategy to implement reversible versions of these functions: add auxiliary output channels which keep track of the input, and auxiliary input channels which when set to 0 gives the desired outcome. (The auxiliary channels are called *ancillas*, and also feature prominently in error correction, section VI.A.) This can be achieved using the bitwise XOR operation

$$\begin{aligned} f \oplus y &= f_M \ldots f_1 f_0 \oplus y_M \ldots y_1 y_0 \\ &= (f_M \oplus y_M)\ldots(f_1 \oplus y_1)(f_0 \oplus y_0). \end{aligned}$$

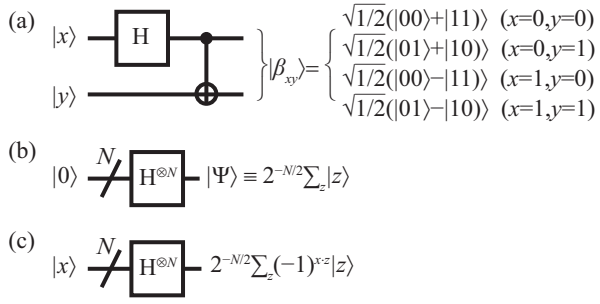

FIG. 6 Creating entanglement and superpositions using Hadamard gates.

Functions with single-bit output are called *Boolean functions* (because their output can be interpreted as 0=FALSE, 1=TRUE). If the input x is also only a single bit, there are exactly four function gates, which can be implemented in terms of elementary gates as shown in Fig. 8. If the auxiliary input state is set to $|y\rangle = |0\rangle$, they implement reversible versions of the four classical unary operators discussed in section II.E.

IV. QUANTUM COMMUNICATION

Quantum communication concerns the transfer of information, and uses entanglement as a resource in order to achieve classically impossible tasks. In the sections below we discuss three examples: the transmission of two bits of information by sending a single qubit (superdense coding); the transfer of the quantum state of one qubit to another qubit at a distant location (teleportation), and the generation of unbreakable encryption codes for secure communication. All these tasks can be achieved with current technology.

A. Superdense coding

Superdense coding is a simple scheme which illustrates how entanglement can facilitate the transmission of information. The scheme is illustrated in Fig. 9(a). Here and in the following, double-lines indicate classical transmission or control channels. By convention, sender and receiver are designated the names Alice and Bob (or A and B), respectively. Initially, Alice and Bob each are in possession of a single qubit. In the distant past (say), they met and prepared these qubits in the entangled Bell state $|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Now, Alice and Bob are located at distant locations (this handed down tale is intended to convey how the scheme will exceed classical expectations, but most of these embellishments are not

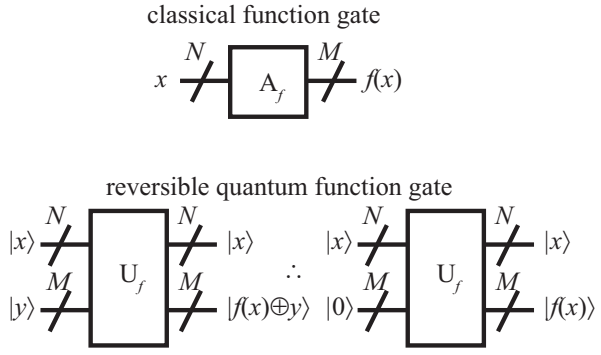

FIG. 7 Function gates.

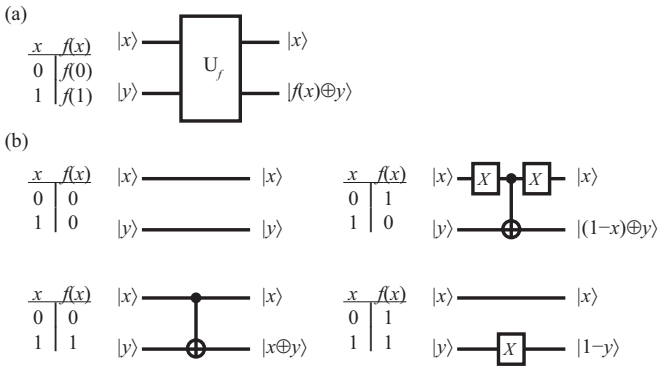

FIG. 8 (a) Boolean function gates with single-bit input. (b) Implementation in terms of elementary gates.

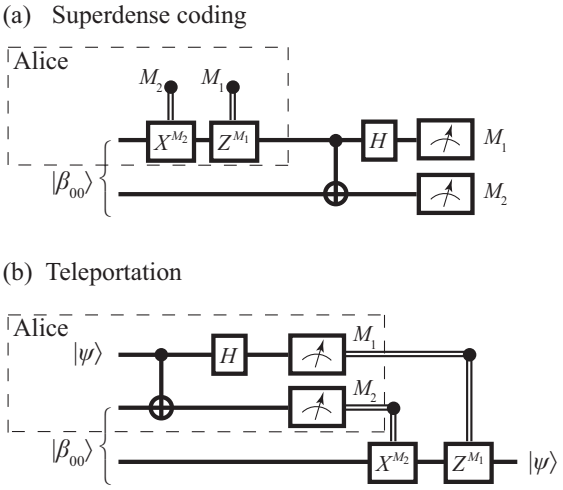

FIG. 9 Circuit representations of (a) superdense coding and (b) quantum teleportation. Double-lines denote classical communication channels. The dashed box indicates operations carried out by the sender (Alice); all other operations are carried out by the receiver (Bob).

necessary—e.g., A&B could have received their entangled qubits from an appropriate, distantly located two-photon source). Alice can now decide whether she wishes to carry out two operations on her qubit—first a NOT operation (X), then a phase flip (Z). This results in the state $Z_1^{M_1} X_1^{M_2} |\beta_{00}\rangle$, where $M_i = 1$ if the operation was carried out, and $M_i = 0$ if it was not carried out. Considering each case separately, we see that this transforms the state into $|\beta_{M_1 M_2}\rangle$, i.e., into one of the four Bell states. Alice then sends her qubit—just the single one—to Bob, who can use a CNOT and a Hadamard gate to transform the qubit pair back to the computational basis states, $H_1 C_{12} |\beta_{M_1 M_2}\rangle = |M_1 M_2\rangle$ [this is just the inverse of the entanglement procedure in Fig. 6(a)]. A measurement of both qubits in the computational basis therefore allows him to infer both M_1 and M_2. In effect, using entanglement as a resource, Alice has sent Bob two

bits of information.

B. Quantum teleportation

Quantum teleportation addresses the task of transferring the *unknown and arbitrary* state of one qubit to another, possibly distant qubit. Because of the no-cloning theorem, this requires to erase all information from the first qubit. How this can be achieved is shown in Fig. 9(b). Before considering the details, note the striking similarities to the circuit for superdense coding—most notably, Alice and Bob again share a Bell pair, and the set of operations they carry out is just interchanged. However, Alice is now also in possession of an extra qubit, whose state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ she wishes to transmit to Bob. She *does not know* the complex amplitudes α and β, and she cannot send the qubit itself, but she can send classical information along a transmission line (e.g., by phone). To succeed, she first entangles the two qubits in her possession by applying a CNOT and a Hadamard gate. This results in the three-qubit state

$$\begin{aligned} H_1 C_{12}|\psi\rangle|\beta_{00}\rangle &= \frac{1}{2}[\alpha(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) \\ &\quad +\beta(|0\rangle - |1\rangle)(|10\rangle + |01\rangle)] \\ &= \frac{1}{2}[|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) \\ &\quad +|10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)], \end{aligned}$$

where the second line follows by reordering the terms. Next she measures the two qubits in her possession, and sends her measurement results M_1 and M_2 to Bob. This allows him to infer how the post-measurement state $|\varphi\rangle$ of his qubit is related to the former state $|\psi\rangle$:

$$\begin{aligned} M_1 M_2 = 00 \quad &\Rightarrow \quad |\varphi\rangle = \alpha|0\rangle + \beta|1\rangle = |\psi\rangle \\ M_1 M_2 = 01 \quad &\Rightarrow \quad |\varphi\rangle = \alpha|1\rangle + \beta|0\rangle = X|\psi\rangle \\ M_1 M_2 = 10 \quad &\Rightarrow \quad |\varphi\rangle = \alpha|0\rangle - \beta|1\rangle = Z|\psi\rangle \\ M_1 M_2 = 11 \quad &\Rightarrow \quad |\varphi\rangle = \alpha|1\rangle - \beta|0\rangle = XZ|\psi\rangle. \end{aligned}$$

Therefore, in order to obtain $|\psi\rangle$, he simply applies $Z^{M_1}X^{M_2}$ to his qubit.

C. Secure communication

Secure quantum communication schemes rely on the no-cloning theorem, which prevents an eavesdropper ('Eve') to listen to a communication line (either, Eve will not acquire any information, or her actions can be detected).

Secure communication based on superdense coding.—If Eve would intercept the qubit sent between Alice and Bob, she only possesses a qubit with reduced density matrix $\frac{1}{2}\hat{I}$, which is independent on the two bits M_1 and M_2 that Alice is sending. However, Eve could still send an arbitrary qubit to Bob, which would result in him obtaining random values for M_1 and M_2, as well. This can be detected when Alice and Bob compare parts of their messages.

Quantum key distribution.—To make communication both secure *and* reliable, one can generate an *encryption key* shared by Alice and Bob. The message can then be sent classically using a simple encryption technique (e.g., by using XOR operations, which flips bits according to a shared sequence of 0's and 1's.).

(a) The BB84 protocol (due to Bennett and Brassard) is a protocol that does not require entanglement. Alice prepares qubits randomly in one of the following four states:

$$|0\rangle, \quad |1\rangle, \quad |\pm\rangle = \sqrt{1/2}(|0\rangle \pm |1\rangle).$$

These are the eigenstates of the Z and X operator. Alice makes sure that she uses a preparation method that tells her which state she has prepared (e.g., this can be done by making random X and Z measurements on qubits with density matrix $\hat{\rho} = \frac{1}{2}\hat{I}$). She then sends these qubits to Bob, who, for each qubit, randomly measures either X or Z. When he measures X and the qubit was in state $|+\rangle$, he will obtain 1 with certainty, while if the state was $|-\rangle$, he will obtain -1 with certainty. If the state was $|0\rangle$ or $|1\rangle$, he will obtain 1 or -1 with 50% probability. In contrast, if Bob measures Z, he will obtain certain measurement outcomes 1 and -1 for the states $|0\rangle$ and $|1\rangle$, respectively, while the states $|\pm\rangle$ result in random measurement outcomes. Bob now tells Alice which measurements he did for each qubit, and Alice tells Bob which measurements she did to prepare the qubits (they only communicate the *type* of measurement, X or Z, but not their outcomes). For these instances, their measurements will have resulted in the same outcome, which results in a shared sequence of 0's and 1's that they can use for encryption.

An eavesdropper listening to the key distribution would be able to make measurements on an intercepted qubit, but without knowledge of its preparation would not be able to then forward the same qubit to Bob. This would introduce errors into the key (at a rate of 25%), which can be detected when Alice and Bob compare small samples of their key (this part of the key would then be discarded).

The BB84 scheme is simple and robust, and has been implemented experimentally using photons, for which the four states correspond to polarizations in $\hat{x}$, $\hat{y}$, $\hat{x}+\hat{y}$, and $\hat{x}-\hat{y}$ direction. (E.g., in 1997, the scheme was demonstrated using a 23 km long transmission line beneath Lake Geneva.)

(b) The EPR protocol (due to Eckert) uses pairwise entangled qubits prepared in the Bell state

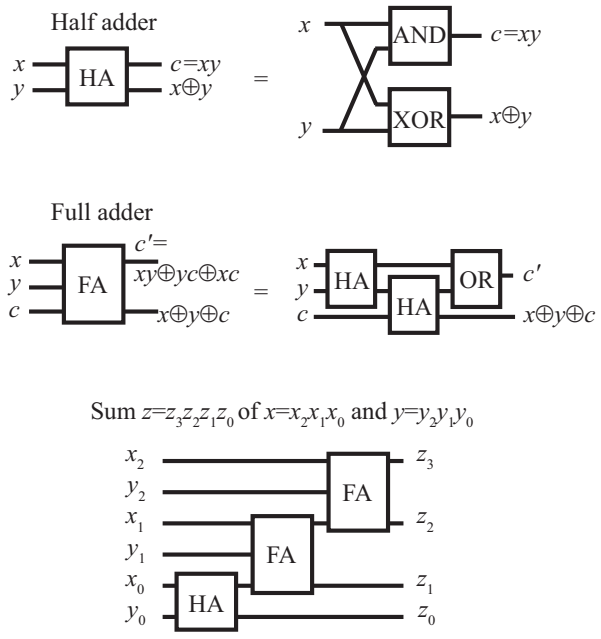

FIG. 10 Classical adder circuits.

$|\beta_{00}\rangle = \sqrt{\frac{1}{2}}(|00\rangle + |11\rangle)$, which can also be written as $|\beta_{00}\rangle = \sqrt{\frac{1}{2}}(|+\rangle|+\rangle + |-\rangle|-\rangle)$. Alice and Bob make random measurements of X and Z on each of their qubits, and then communicate to each other to find out for which qubits they did the same measurement. For each of these instances, they are then guaranteed to have found the same outcome. Again, an eavesdropper would introduce errors, and can be detected by sacrificing a small random sample of the key.

In combination with classical encoding schemes that encode messages into longer bit sequences (see error correction, in section VI.A) these protocols can be made reliable against a finite error rate, and secure even against sophisticated eavesdroppers attacks that involve the collective manipulation of the whole qubit stream.

V. QUANTUM COMPUTATION

This section discusses the most important quantum-computation algorithms, associated with the names Deutsch and Josza, Shor, and Grover. To get a feeling of algorithms we start with the simple, classical example of adding two numbers.

A. Adding numbers

The discussion in Section III implies that quantum computers can achieve, at least, the same tasks as a classical computer. Since this requires reversible logics, we first illustrate this by an example, namely, the computation of the sum of two numbers.

Binary numbers can be added just like decimal numbers, bit by bit starting with the least significant bits, and carrying over bits to the next more significant bit. Consider, e.g., the sum $3+3$, which in binary code is $11 + 11$. We first add the last two bits (i.e., the least significant bits), resulting in $1 + 1 = 10$. The last bit of this (0) is the last bit of our result, while the first bit needs to be carried over. We then add the first two bits, and to this the carry-on bit, giving $1+1+1 = 11$. This again results in a carry-on bit. There are no further bits to be processed, so the carry-on bit becomes the most significant bit. Our result now has three bits: 110, which is the same as $4 + 2 + 0 = 6$.

In classical computers, these operations can be carried out using *half-adder* and *full-adder* components, which can be realized using elementary binary gates as shown in Fig. 10. Because of $x + y = y + x$, the depicted circuit is irreversible (furthermore, it requires to copy bits). Figure 11 shows that the same operations can also be achieved quantum mechanically using a combination of reversible CNOT and CCNOT gates, which generate additional outputs that can be used to reverse the calculation.

Adding two numbers with reversible gates does not increase the efficiency of the algorithm. This is in contrast to the following examples, which concern true quantum algorithms that solve specific tasks more efficiently than classical algorithms.

B. Deutsch-Josza algorithm

Efficient quantum algorithms generally exploit the quantum parallelism to carry out many calculations in one step, which is particularly useful if one is interested in a global property of these calculations. This is nicely illustrated by the

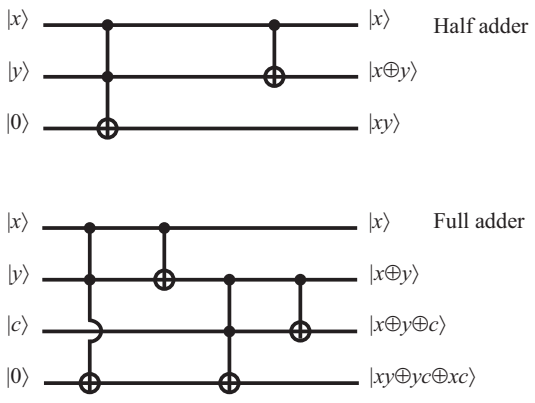

FIG. 11 Reversible half adder and full adder circuits.

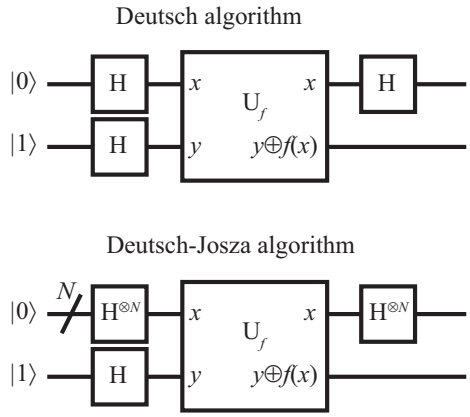

FIG. 12 Deutsch and Deutsch-Josza algorithms.

Deutsch-Josza algorithm, which allows to establish a specific feature of certain Boolean functions $f : x \to \{0, 1\}$ that specifically fall into one of the following two classes: either the function is *constant* (i.e., the output bit f is either always 1, or always 0), or it is *balanced* (i.e., the output bit is 0 for one half of the input values, and 1 for the remaining half).

Assume that we are given a function $f(x)$ which is guaranteed to be either constant or balanced, but at the outset we do not know to which of the two classes it belongs. Classically, we need at least two function evaluations to find out that a function is balanced—and this is only in the fortunate case that we immediately pick two inputs for which the outputs differ. In order to certify that a function is constant we need $2^N/2 + 1 = 2^{N-1} + 1$ function evaluations—that's because if the function is balanced, we may by chance first pick $2^N/2$ inputs that all result in the same output. In contrast, using quantum gates, the Deutsch-Josza algorithm, shown in Fig. 12, requires only a single function evaluation to determine whether the function is constant or balanced. (The upper panel of the figure shows the Deutsch algorithm, the special case of a function $f(x)$ with single-bit input, which is addressed on worksheet 3.)

Starting from the initial state $|\psi_0\rangle = |0 \ldots 001\rangle$, Hadamard gates are used to form the superposition

$$|\psi_1\rangle = \prod_n H_n |\psi_0\rangle = 2^{-N/2} \sum_x |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |\Psi\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

(see also Fig. 6(b)). If $f(x) = 1$, the function gate changes the state of the last qubit into $\frac{|1\rangle - |0\rangle}{\sqrt{2}} = -\frac{|0\rangle - |1\rangle}{\sqrt{2}}$; for $f(x) = 0$, the state of this qubit remains $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$. Therefore, the function gate introduces factors $(-1)^{f(x)}$ in front of each basis state $|x\rangle$, resulting in

$$|\psi_2\rangle = U_f |\psi_1\rangle = 2^{-N/2} \sum_x (-1)^{f(x)} |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

In the final step, Hadamard gates are applied to all but the last qubit. Using the result shown in Fig. 6(c), this delivers the final state

$$|\psi_3\rangle = 2^{-N} \sum_z \sum_x (-1)^{x \cdot z + f(x)} |z\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

Let us now have a look at the amplitude of the state with $|z\rangle = |0\rangle$, which is given by $2^{-N} \sum_x (-1)^{f(x)}$. If $f(x)$ is balanced, this amplitude vanishes. On the other hand, if $f(x) = c$ is constant, this amplitude takes the value ± 1 (depending on whether $c = 0$ or $c = 1$). Therefore, if f is constant, a measurement of all qubits will find them all taking the value 0, while this will never happen if the function is balanced. Remarkably, using quantum parallelism, the distinction of both cases has been achieved with a single operation of the function gate.

A notable ingredient of the Deutsch-Josza is the introduction of conditional phase factors (such as $(-1)^{f(x)}$) in front of each of the computational basis states. This strategy is also at the core of the practically more important algorithms discussed next.

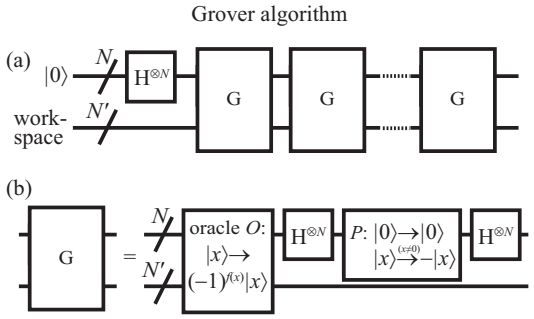

FIG. 13 Grover search algorithm.

C. Grover's quantum search algorithm

Grover's quantum search algorithm exploits quantum parallelism to speed up the search for solutions among a large set of candidate solutions. A prime example is the search for entries in a database which match to a given key. Let us enumerate all entries by an integer index x, and assume for simplicity that the size of the database is $\mathcal{N} = 2^N$. The entries matching the key can then be characterized by an *oracle function* $f(x)$, which is a Boolean function that returns $f(x) = 1$ if entry x matches to the key; otherwise, $f(x) = 0$.

Assume that there are $\mathcal{M} \ll \mathcal{N}$ entries matching the key. Classically, we need to make $\approx \mathcal{N}/\mathcal{M} \gg 1$ queries of the database (or *calls of the oracle function*) to find one of these entries. The Grover algorithm, shown in Fig. 13, only requires $O(\sqrt{\mathcal{N}/\mathcal{M}})$ calls of the oracle function—not an exponential speedup, but still sizeable when the database is large.

The first step initializes the index register in the now very familiar equal-superposition state $|\Psi\rangle$. This is followed by a sequence of operations G, known as the Grover iterate, in which the oracle gate $\hat{O}$ is called once. Its action is defined to flip the phase of all solutions: $\hat{O}|x\rangle = (-1)^{f(x)}|x\rangle$. As we have seen in the Deutsch-Josza algorithm, this can be achieved with a function gate acting as $\hat{O}_f|x\rangle|q\rangle = |x\rangle|f(x) \oplus q\rangle$, where the oracle qubit q is initialized as $\sqrt{1/2}(|0\rangle - |1\rangle)$ (Fig. 13 reserves N' workspace qubits for the implementation of the oracle). The Grover iterate also contains a conditional phase gate $P = 2|0\rangle\langle 0| - \hat{I}$, which inverts the phase of all computational basis states with exception of the state $|0\rangle$. This is embedded into Hadamard gates, resulting into

$$P' = (\prod_n H_n)(2|0\rangle\langle 0| - \hat{I})(\prod_n H_n) = 2|\Psi\rangle\langle\Psi| - \hat{I}.$$

The Grover iterate can therefore be written as

$$G = (2|\Psi\rangle\langle\Psi| - \hat{I})\hat{O}.$$

The purpose of the Grover iterate is to rotate the initial state into the direction of the equal superposition of solutions $|X\rangle \equiv \frac{1}{\sqrt{\mathcal{M}}}\sum_{m=1}^{\mathcal{M}}|x_m\rangle$, where we have enumerated all solutions as x_m, $m = 1, 2, 3, \ldots, \mathcal{M}$. This rotation takes place in a plane spanned by $|X\rangle$ and $|Y\rangle = \frac{1}{\sqrt{\mathcal{N}-\mathcal{M}}}\sum_{m=1}^{\mathcal{N}-\mathcal{M}}|y_m\rangle$, the equal superposition of all non-solutions y_m, $m = 1, 2, 3, \ldots, \mathcal{N} - \mathcal{M}$. Figure 14 illustrates how this works. The initial state can be written as

$$|\Psi\rangle = \sqrt{\mathcal{M}/\mathcal{N}}|X\rangle + \sqrt{1 - \mathcal{M}/\mathcal{N}}|Y\rangle,$$

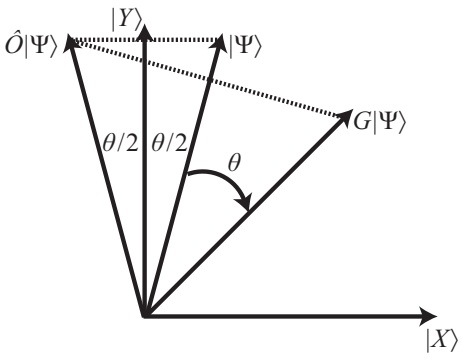

FIG. 14 Geometric interpretation of the Grover search.

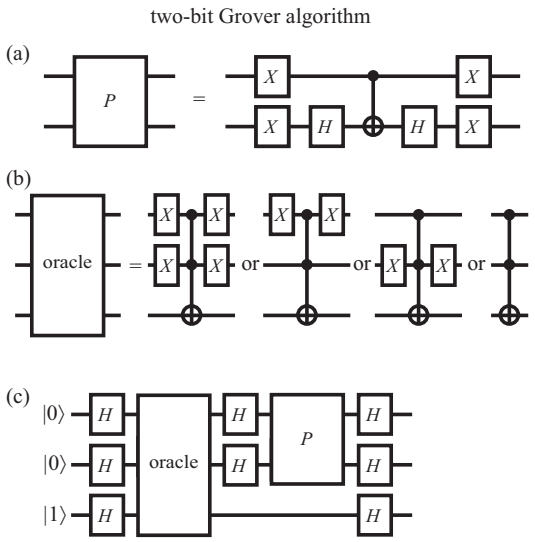

FIG. 15 Two-bit Grover search algorithm.

and therefore lies in the X,Y plane. The unit vector

$$\Psi = (\sqrt{\mathcal{M}/\mathcal{N}}, \sqrt{1 - \mathcal{M}/\mathcal{N}}) = (\sin\theta/2, \cos\theta/2)$$

can be parameterized in terms of the angle $\theta/2$ between Ψ and the Y axis. For $\mathcal{M} \ll \mathcal{N}$, $\theta = 2\arccos\sqrt{1 - \mathcal{M}/\mathcal{N}} \approx 2\sqrt{\mathcal{M}/\mathcal{N}}$ is small, such that Ψ is almost parallel to the Y axis.

Per definition, the oracle function acts as

$$\hat{O}(\alpha|X\rangle + \beta|Y\rangle) = -\alpha|X\rangle + \beta|Y\rangle,$$

which amounts to a reflection about the Y axis. The phase gate P' performs another reflection, now about the Ψ axis. As a result, the state is rotated by an angle θ towards the X axis. Repeating this rotation $\approx (\pi/4)\sqrt{\mathcal{N}/\mathcal{M}}$ times rotates the state vector close to the X axis—the misalignment will be less than $\theta/2$. With high probability, a measurement of the final state will therefore deliver a solution x_m of the search problem.

An instructive example is a two-bit search with one solution x_1, depicted in Fig. 15. Panel (a) shows a realization of the P gate with H, X, and CNOT gates (as a matter of fact, this realizes $-P$, but an overall phase factor of a quantum state is non-detectable). Depending on the binary representation of x_1, the oracle function takes one of four possible forms, which can be implemented using Toffoli (CCNOT) and X (NOT) gates as shown in panel (b). For $\mathcal{N} = 4 = 2^2$, $\mathcal{M} = 1$, the angle of the initial state vector Ψ with the Y axis is $\theta/2 = 30°$. The Grover iterate rotates the state by $\theta = 60°$. A single iteration therefore rotates the state onto the X axis, which immediately identifies the matching entry x_1. In contrast, a classical search would on average require 2.25 oracle calls before the solution is found.

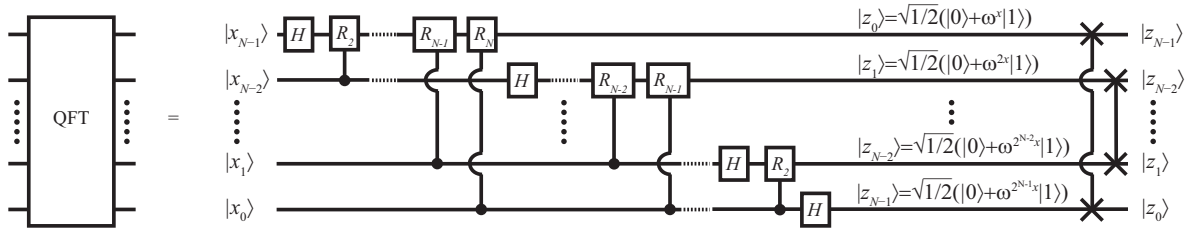

FIG. 16 Quantum Fourier transformation. The indicated swap gates simply invert the order of the output qubits, which is convenient for subsequent applications.

D. Quantum Fourier transformation

Consider the Fourier transformation

$$\tilde{f}(z) = 2^{-N/2} \sum_{x=0}^{2^N-1} \omega^{xz} f(x)$$

of an N-bit function f. Here, we abbreviated $\omega = e^{2\pi i/2^N}$, such that $\omega^{xz} = e^{2\pi i xz/2^N}$ (where xz is an ordinary, not bitwise, multiplication). Classically, implementation of the Fourier transformation takes $O(2^N)$ elementary gate operations. In contrast, its quantum-mechanical analogue, the *quantum Fourier transformation*

$$U_{QFT}|x\rangle = 2^{-N/2} \sum_{z=0}^{2^N-1} \omega^{xz}|z\rangle \equiv |\tilde{x}\rangle,$$

can be implemented with $O(N^2)$ gate operations, which constitutes an exponential increase in efficiency that can be exploited in a range of algorithms.

The corresponding circuit is shown in Fig. 16. The verification that the depicted circuit computes the quantum Fourier transformation can be based on the *product representation*

$$|\tilde{x}\rangle = |z_{N-1}\rangle|z_{N-2}\rangle \cdots |z_0\rangle,$$

where $|z_n\rangle = \sqrt{1/2}(|0\rangle + \omega^{2^n x}|1\rangle)$. Because of the 2π periodicity of the phase, the phase factors can be written as $\omega^{2^{N-n}x} = e^{i2\pi 0.x_{n-1}x_{n-2}\cdots x_0}$, where we introduced the *binary fractions*

$$x/2^n = \sum_{k=0}^{N-1} x_k 2^{k-n} \equiv x_{N-1}\cdots x_n.x_{n-1}\cdots x_0$$

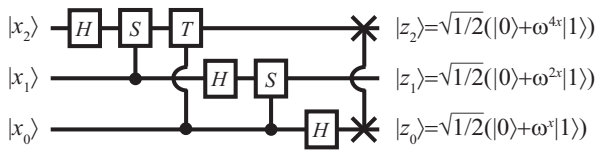

FIG. 17 Three-bit quantum Fourier transformation.

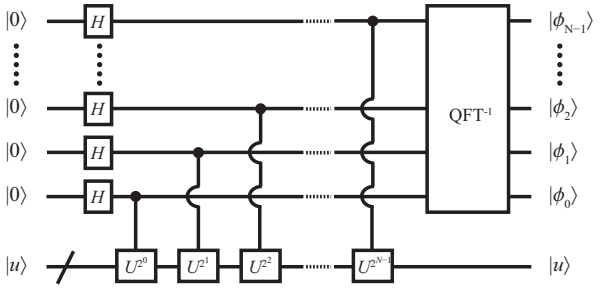

FIG. 18 Phase estimation circuit.

in a notation analogous to the one denoting fractional decimal numbers. As shown in the figure, the required transformation of each qubit can be achieved efficiently by first applying a Hadamard gate, followed by phase rotation gates

$$R_n = \begin{pmatrix} 1 & 0 \\ 0 & \exp(2\pi i/2^n) \end{pmatrix}$$

that are controlled by the less significant qubits.

An instructive example is the three-bit quantum Fourier transformation, where $\omega = \exp(2\pi i/8) = \sqrt{i}$. In this case, $R_1 = T$ is the $\pi/8$ gate, and $R_2 = S$ is the phase gate. The corresponding circuit is shown in Fig. 17.

E. Applications: From phase estimation to prime factorization

Phase estimation.—A key application of the quantum Fourier transformation is the estimation of the (reduced) phase ϕ of an eigenvalue $\lambda = e^{2\pi i\phi}$ of a unitary operator U, where $0 \le \phi < 1$. This can be used for a problem called order finding, which in itself is a central step in the prime factorization of numbers. Phase estimation can also be used for a problem know as quantum counting. These problems are briefly discussed later; at the moment it suffices to know that they involve different operators U.

Given the eigenstate $|u\rangle$ corresponding to λ, ϕ can be estimated efficiently as an N bit binary fraction $\phi \approx 0.\phi_{N-1}\phi_{N-2}\cdots\phi_0$ using the circuit shown

in Fig. 18. Since $U|u\rangle = \lambda|u\rangle$ and $\lambda = \omega^{2^N\phi}$, where $\omega = e^{2\pi i/2^N}$, the controlled-U^{2^n} operations in the first part of the algorithm transform the qubits into the state $\sqrt{1/2}(|0\rangle + \omega^{2^n(2^N\phi)}|1\rangle)$. With N-bit precision, $2^N\phi \approx \phi'$, where the integer number ϕ' has the N-bit representation $\phi' = \phi_{N-1}\phi_{N-2}\cdots\phi_0$. The intermediate state can therefore be approximated by the product representation

$$|z_{N-1}\rangle|z_{N-2}\rangle\cdots|z_0\rangle = U_{QFT}|\phi'\rangle = |\tilde{\phi'}\rangle$$

of the Fourier-transformed N-bit estimate of $2^N\phi$, whose binary digits ϕ_n are then recovered by an inverse Fourier transformation.

Quantum counting.—A straightforward application of the phase estimation algorithm is the determination of the number $\mathcal{M}$ of solutions in an $\mathcal{N}$-item search problem, as encountered in the Grover search. In the X-Y plane, the rotation by one Grover iteration can written as the

$$G = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix},$$

which is a unitary matrix with eigenvalues $e^{\pm i\theta}$. We also know that $|\Psi\rangle$ lies in the X-Y plane, and therefore is a superposition of the two corresponding eigenstates, which can be fed into the slot for $|u\rangle$.

Order finding.—Phase estimation can also be used to solve the following number-theoretic problem: given two numbers a and M without common divisors, what is the *order* r of $a \bmod M$, defined as the smallest positive integer r such that $a^r \bmod M = 1$? Here, the modulo operation determines the remainder of the division by M. The order can be obtained by estimating the phase of the eigenvalues $\lambda_n = \exp(2\pi i n/r)$ of the operator $U|x\rangle = |ax \bmod M\rangle$ (for $x < M$; otherwise $U|x\rangle = |x\rangle$, which guarantees that U is unitary). Conveniently, the sum of eigenfunctions

$$|u_n\rangle = \frac{1}{\sqrt{r}}\sum_{z=0}^{r-1}\exp(-2\pi i n z/r)|a^z \bmod M\rangle$$

is independent of r, $\sum_n |u_n\rangle = |1\rangle$. Therefore, initializing $|u\rangle = |1\rangle$, the phase estimation algorithm will deliver an approximation of n/r, where each value n in the range $0 \le n < r$ appears with equal probability $1/r$. This suffices to reconstruct the order r (most efficiently, by a method based on continued fractions).

Period finding.—Note that the order r is the *period* of the function $f(x) = a^x \bmod M$, i.e., $f(x) = f(x+r)$. It is noteworthy that a variant of the procedure above can be used to find the period of any integer-valued function. The corresponding circuit is shown in Fig. 19. The indicated intermediate

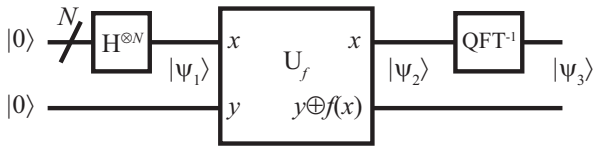

FIG. 19 Period finding circuit. The indicated states are specified in the text.

states are $|\psi_1\rangle = |\Psi\rangle|0\rangle$ and

$$|\psi_2\rangle = 2^{-N/2}\sum_{x=0}^{2^N-1}|x\rangle|f(x)\rangle = \frac{1}{\sqrt{r}}\sum_{n=0}^{r-1}|\widetilde{(n/r)'}\rangle|\hat{f}(n)\rangle,$$

where, as before, $(n/r)'$ denotes the N-bit estimate of $2^N(n/r)$, and the tilde indicates the Fourier-transformed state. Furthermore, we abbreviated

$$|\hat{f}(n)\rangle = \frac{1}{\sqrt{r}}\sum_{x=0}^{r-1}\exp(-2\pi n i x/r)|f(x)\rangle.$$

The inverse Fourier transformation converts this into the final state

$$|\psi_3\rangle = \frac{1}{\sqrt{r}}\sum_{n=0}^{r-1}|(n/r)'\rangle|\hat{f}(n)\rangle,$$

so that the measurement of the output qubits in the first register delivers an N-bit approximation of $2^N(n/r)$.

Prime factorization.—In a loose mathematical sense, the factors of an integer number M can also be interpreted as 'periods' of that number. Indeed, it turns out that the order-finding problem is equivalent to the problem of prime number factorization, and therefore can be solved efficiently using phase estimation. This is embodied in Shor's factorization algorithm. Because the accurate description of this algorithm requires various number-theoretic concepts, we here only mention one key ingredient: assume we have randomly chosen a number a and found that the order r of $a \bmod M$ is even (otherwise, start again with another randomly chosen a). Then $b = a^{r/2}$ is still an integer, which fulfills

$$b^2 \bmod M = 1 \Rightarrow (b+1)(b-1)\bmod M = 0.$$

Furthermore assume that $b \bmod M \neq -1$ (otherwise, start again...). It then follows that $b_+ = (b+1)$ or $b_- = (b-1)$ shares a nontrivial factor with M. The largest factor (the greatest common divisor, gcd) can be found efficiently using Euclid's algorithm: given $c > d$, iterate the identity $\gcd(c,d) = \gcd(d, c \bmod d)$ until the smaller number is a divisor of the larger number; the smaller number is then the gcd of c and d, which delivers a factor of M.

VI. ERROR CORRECTION AND PRACTICAL ISSUES

A. Errors and error correction

Errors occur both in computation as well as in communication, and generally degrade information content. In classical computation, typical errors are flipped bits or lost data packages. A primary goal of hardware design is to make such errors unlikely, which can be done, e.g., by utilizing dissipation to stabilize the outcomes of irreversible gate operations. This goal competes with other goals such as speed, capacity, size, stability, longevity, energy consumption, and, of course, costs, which along with practical limitations means that a certain amount of errors must be tolerated. To cope with them, classical computation algorithms and communication protocols introduce a certain amount of overhead (*redundancy*) into the information. This can be used to detect whether errors have occurred (a step called *syndrome diagnosis*), and preserves enough of the information so that the errors can be corrected (by *recovery operations*).

A simple example is the code

$$0_L \to 000, \quad 1_L \to 111,$$

which represents one logical bit of information (subscript L) in terms of three physical bits. If one of the physical bits is flipped this can be detected by comparison to the other bits, and subsequently corrected following the rules

$$000, 001, 010, 100 \to 0_L; \quad 011, 101, 110, 111 \to 1_L.$$

Following this scheme, only errors affecting two or all three of the physical bits will result in an error of the logical bit. A single-bit error probability p will thus result in an error probability $p^2(3-2p)$ for the logical bit, which is much less than p if p is small.

Classical error correction of many independent single-bit errors can be achieved when k logical bits of information are encoded into a sufficiently large number n of physical bits. The number of differing bits of two code words is called their distance, and the smallest distance occurring in a code is the distance of the code, d. Correction of multiple errors then requires to identify the logical code word with the smallest distance to the register state, which is reliable unless the number of errors exceeds $[(d-1)/2]$ (here $[\cdot]$ denotes the integer part of a number). In other words, d determines the number of erroneous physical bits that can be tolerated for reliable identification of the logical bits. Classical error correction codes of this kind are therefore often characterized by the triple $[n, k, d]$. It should be noted that the distance is a useful characteristic only if the errors are not correlated, i.e., if they only affect one bit at a time. This does not apply, e.g., to a data package loss, which yields undetermined values of a number of consecutive bits. The design of a resilient error correction scheme therefore also depends on an adequate *error model*, which identifies the types of errors that are most likely to occur.

Quantum computation is sensitive to a wide range of additional types of errors that affect the amplitudes of individual or collective qubit states. The implementation of gates is delicate because of their linear and reversible nature, which prevents the use of dissipation to stabilize the outcomes. In particular, multi-qubit gate operations typically require precise control of interactions, which can also leak on to other qubits. Furthermore, multi-qubit gates tend to propagate errors—e.g., an error in the control bit of a CNOT gate will result in an error of the target bit. To a larger extent than classical codes, therefore, quantum codes must rely on a good error model.

Consider, for example, a system designed to realize the Pauli X gate (NOT) by time evolution with a Hamiltonian aX, which must be sustained for a set time $\Delta t = \hbar\pi/2a$ (see worksheet 2). Imperfections in the duration of this action will introduce errors into the final states. Furthermore, the physically realized Hamiltonian may differ from aX (e.g., it may feature contributions proportional to Y and Z, as well as many-qubit terms arising from interactions), and all terms may fluctuate temporally.

An important source of these contributions is the dynamics of the *environment*, i.e., all physical components that are not directly participating in the computational tasks. For example, the uncontrolled motion of charge carriers in parts of a device results in a fluctuating electromagnetic field. External influences of this kind are worrisome because they generally cause the state of the quantum register to become mixed: the register state will depend on the environment, and therefore becomes entangled with it; its reduced density matrix will therefore describe a statistical mixture. This phenomenon, called *dephasing* or *decoherence*, is undesirable because it negatively affects the usable amount of entanglement (as we have seen on worksheet 2 for the example of the equal mixture of all four Bell states). Even more directly, perturbed relative phases of quantum states also negatively affect their superposition, i.e., quantum parallelism.

Consequently, error correction schemes for quantum information need to cope with a much larger variety of errors—in principle, a *continuum* of errors, which can affect the phase and magnitude of the amplitudes of the state. Moreover, they cannot establish redundancy by copying the information, which would violate the no-cloning theo-

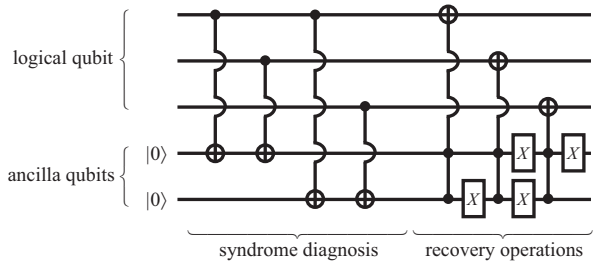

FIG. 20 Three-qubit code for correction of single-qubit flip (X) errors.

rem. Surprisingly, resilient quantum error correction strategies not only exist, but also get by with a *finite* set of diagnosis and recovery operations—a phenomenon known as *error discretization*.

In order to see how this comes about, let us specialize to single-qubit errors. Such errors can generally be represented by a unitary 2×2 matrix $U = a_0 I + a_x X + a_y Y + a_z Z$, which transforms an error-free physical qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ into the defective state $U|\psi\rangle$. The contribution X flips the qubit with a probability depending on the coefficient a_x. An individual flip can be detected and corrected by adapting the classical three-bit scheme: Encode the two logical basis states into three physical qubits, such that $|0_L\rangle = |000\rangle$ and $|1_L\rangle = |111\rangle$. A logical-qubit state is then described by a physical state $|\psi\rangle = \alpha|000\rangle + \beta|111\rangle$. This is *not* a threefold copy of the logical qubit, which would correspond to a separable product state—instead, we here deal with an entangled three-qubit state.

Erroneous flips of the nth qubit (due to X_n) result in admixture of the other three-qubit basis states. This can be detected by measurements of the *error syndromes* $S_1 = Z_1 Z_2$ and $S_2 = Z_1 Z_3$, and corrected by applying an operator U following the rules

$$\begin{array}{llll} S_1 = 1, & S_2 = 1 & \to & U = I, \\ S_1 = 1, & S_2 = -1 & \to & U = X_3, \\ S_1 = -1, & S_2 = 1 & \to & U = X_2, \\ S_1 = -1, & S_2 = -1 & \to & U = X_1. \end{array}$$

Figure 20 shows how these conditional operations can be achieved without doing any measurements, but instead utilizing CNOT and Toffoli gates involving two ancilla qubits. How does this circuit cope with the continuous set of possible errors? A single-qubit error moves the quantum state into a subspace spanned by the computational basis states with distance 0 and 1 to the logical qubits, and in this basis is specified by four complex amplitudes. In the circuit, these amplitudes simply determine the probabilities that the control bits trigger the various CNOT operations. At the end of the procedure, the complex amplitudes are transferred to the two ancillary qubits, whose joint state also resides in a four-dimensional space.

Analogously, errors of the type Z can be corrected using a three-qubit code $|0_L\rangle = |{+}{+}{+}\rangle$, $|1_L\rangle = |{-}{-}{-}\rangle$, where $|\pm\rangle = \sqrt{1/2}(|0\rangle \pm |1\rangle)$ are the eigenstates of the X operator. Effectively, the roles of X and Z are then interchanged.

Once we can cope with individual X and Z errors, schemes can be fused via *concatenation* to yield a code that can also correct combined errors. This naturally leads to the *Shor code*, which uses 9 qubits to encode one logical qubit according to

$$|0_L\rangle = \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}},$$
$$|1_L\rangle = \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}.$$

Note how X errors can be detected by comparing three consecutive qubits, while Z errors are detected by the relative phase of every third qubit; the latter can be diagnosed using the syndromes $X_1 X_2 X_3 X_4 X_5 X_6$ and $X_4 X_5 X_6 X_7 X_8 X_9$.

Since $Y = iXZ$, the Shor code also allows to correct Y errors—therefore, it offers protection against *arbitrary* single-qubit errors. On the other hand, considering that here $[n, k] = [9, 1]$, this is achieved with a rather large overhead. There are more sophisticated codes that achieve the same task with less than 9 qubits, the minimum being 5. This still exceeds what was required classically, and moreover involves many more syndrome diagnosis and recovery operations. Considering that these operations are also prone to errors, it is clear that quantum error correction is still a challenging task.

More general schemes—in particular, codes based on the *stabilizer formalism*, which utilizes group theory—allow to cope with complex types of errors affecting a collection of qubits, and also take care of errors accumulated by faulty gate operations. If the initial error rate falls below a certain threshold, these codes can be scaled up by adding more and more overhead, thereby allowing (in principle) to achieve arbitrarily good (*fault-tolerant*) quantum computation. However, while present technology has advanced sufficiently to enable reliable quantum communication, a universal quantum computer is still far removed from reality.

B. Practical requirements

This brief concluding section juxtaposes the main technological requirements for a workable quantum computer and the key features of some specific physical implementations.

As is clear from the preceding section, quantum information processing poses serious techni-

cal challenges. E.g., fault tolerant computation requires that the error rate of gate operations falls below a certain threshold, and can only be implemented when the system can be scaled up by adding more and more components. The various challenges have been canonized by diVincenzo into a set of five core requirements, which are known as the *DiVinzenco criteria*:

1. *Well-defined qubits.* This requires to identify physical systems whose quantum dynamics is essentially constrained to two quantum levels. Examples for naturally occurring two-level systems are the spin of electrons and certain nuclei, as well as the polarization of photons. In many proposed systems, however, the reduction to two levels is only approximate. Examples are atoms in ion traps, photons stored in microcavities, the magnetic flux penetrating a superconducting ring, and electrons confined to (normal conducting or superconducting) solid-state devices, such as quantum dots. In all these cases, care has to be taken that the system does not populate the other available energy levels (i.e., one needs to avoid *leakage*), which can be best done by making these levels energetically inaccessible.

   It is of course possible to design quantum computation schemes that are not binary, and therefore make use of more than two levels in the energetically accessible range ΔE of the register components. However, adding qubits provides much better scalability. Each additional qubit multiplies the register state dimension by a factor of two (resulting in 2^N levels), and each qubit can be addressed individually, which requires energy resolution $\sim \Delta E/N$ instead of resolution $\sim \Delta E/2^N$ for a comparable multi-level system.

   By convention, if there is a clear energy separation between the two levels, the state with the lower energy is designated $|0\rangle$, and the state with the higher energy is designated $|1\rangle$. As discussed in the context of error correction, the physical qubits can then be used to encode logical qubits, which allows to take care of the most likely sources of errors specific to the chosen implementation.

2. *Initialization to a pure state.* The quantum register must start in a well-defined state. It is sufficient to have a reliable method to prepare at least one such state, since a universal quantum computer would be able to transform this into any other state. Utilities to prepare a larger variety of states further improves the efficiency of quantum computation.

   Using the convention of labeling qubit levels according to their energy, the register state $|0\rangle$ often corresponds to the ground state of the system. This state can be prepared by allowing the system to equilibrate (relax) at low temperature. Other states may be enforced by relaxation in presence of external fields (such as a magnetic field for nuclear spins), or dynamically (e.g., by pumping of atomic transitions).

   It is not necessary, however, that the preparation process is deterministic. E.g., a viable strategy is to make a hard, complete measurement of the register, thereby forcing it into a pure state that is completely determined by the recorded measurement outcomes.

   If initialization is not perfect, it can be combined with error correction schemes to enhance its accuracy. In particular, if the state is not entirely pure, the entropy can be transferred into ancilla qubits, so that the register state become purified. Such procedures also allow to carry out initialization in multiple steps.

3. *Universal set of quantum gates.* As discussed in section III, a universal set of quantum gates can be obtained using single-qubit rotations on the Bloch sphere, and at least one type of two-qubit operations (such as CNOT). Alternative constructions use a sufficiently large number of multi-qubit gates. Using facilities to swap qubits, it is not necessary that each pair of qubits can be coupled directly; still, the coupling network needs to be sufficiently interconnected, and also should be scalable to a large number of qubits. This is a severe problem for many proposed implementations.

   For each implementation, the precision of gate operations can be increased not only via error correction, but also using insight into the specific quantum dynamics of the system. E.g., *echo* and *refocussing* techniques in nuclear magnetic resonance employ judiciously timed magnetic-field pulses to average out the effects of spurious qubit couplings and unwanted single-qubit terms in the Hamiltonian. This exemplifies the natural tradeoff between precision and speed of gate operations, which is a general obstacle in all implementations.

4. *Qubit-specific measurement.* Ideally, to determine the outcome of a computation one should be able to carry out ideal measurements on each physical qubit. In practice, a finite degree of imperfection can be tolerated. This may be because the computation can be repeated, or because it can be carried out on many systems in parallel. An interesting simplification occurs because algorithms often use quantum parallelism only during the calculation, but are designed to deliver a classical bit sequence x_n as output.

Such results can be amplified using a *quantum fanout* operation of the type $\alpha|x_1\rangle|00\rangle+\beta|x_2\rangle|00\rangle \rightarrow \alpha|x_1x_1x_1\rangle + \beta|x_2x_2x_2\rangle$, which enhances the measurement fidelity because the desired information is now encoded in additional qubits. (Note how the fanout operation differs from a copy operation prohibited by the no-cloning theorem; rather, it closely resembles error correction procedures.)

5. *Long coherence times.* This statement subsumes various requirements for the protection of the quantum register state throughout the computation. In particular, one needs to preserve the capacity to use superposition and entanglement as computational resources. As discussed before, this capacity is in particular degraded by spurious internal and external interactions. These effects can be broadly categorized depending on whether they affect the population probabilities or interference of the register states (i.e., the modulus or complex phase of the amplitudes): *Relaxation* (on a time scale T_1) affects the probabilities, and often is combined with energy loss or gain (dissipation). *Dephasing* (on a time scale T_2) affects the phases, and generally reduces the purity and entanglement of the register state (decoherence).

   In most systems, $T_1 \gg T_2$, i.e., the operability is limited by dephasing. A viable quantum computer needs to carry out >10000 gate operations during this time. The polarization of photons and the spin of electrons in solid-state devices are two types of qubits which possess reasonably long dephasing times (the latter lends a main incentive to the field of *spintronics*). On the other end of the scale, charge (as, e.g., carried by electrons confined in a quantum dot) couples strongly to electromagnetic fluctuations, which discounts this degree of freedom for quantum computation purposes.

These requirements are supplemented by two additional criteria for reliable quantum communication:

6. *Convert stationary and flying qubits.* Stationary qubits reside in registers, while flying qubits propagate along quantum transmission lines. Photons make ideal flying qubits, while nuclei and atoms typically serve as stationary qubits. In this respect, electrons in solid state devices are particularly flexible because they can move through conducting regions, but can also be confined electrostatically.

7. *Transmit flying qubits between distant locations.* This can be achieved with high fidelity for photons, but is far more challenging for electrons. In spintronics, e.g., the electronic spin can be flipped by scattering off magnetic impurities in the transmission line.

At present, none of the various physical candidate platforms score well on all of the core requirements. The key challenge is to overcome the natural trade-off between easy access of qubits (initialization, control, readout), a high degree of isolation (coherence), and scalability. On the other hand, this trade-off can also be exploited to balance strengths in certain areas (e.g., a long coherence time) against weaknesses in other areas (e.g., imprecise gates, whose errors can then be corrected by running a more sophisticated, time-consuming error correction scheme). That said, any viable quantum computer is likely to be a hybrid device which combines the specific strengths of the various physical platforms.

VII. FURTHER READING

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).

[2] J. Stolze and D. Suter, *Quantum Computing—A Short Course from Theory to Experiment*, 2nd edition (Wiley VCH, 2008).

[3] J. Preskill, Lecture notes, available at http://www.theory.caltech.edu/~preskill/ph219/index.html#lecture