

A 16 Introduction to Quantum Computation and Quantum Information Theory

G. Arnold and M. Richter

Zentralinstitut für Angewandte Mathematik (ZAM)

Forschungszentrum Jülich GmbH

Contents

1	Introduction	2
1.1	Why quantum computation?	2
1.2	Complexity classes	2
2	Quantum computation	3
2.1	The qubit	3
2.1.1	The Bloch sphere	4
2.2	The circuit model of quantum computation	5
2.3	Single-qubit gates	6
2.3.1	Rotations of the Bloch sphere	7
2.4	Two-qubit gates	8
3	Quantum algorithms	9
3.1	Deutsch's algorithm: constant or balanced function	9
3.2	Grover's algorithm: How to find a needle in a haystack?	10
3.2.1	Geometric visualization	11
3.3	Shor's algorithm: Factoring of numbers	13
4	How to build a quantum computer?	15
4.1	Ion traps	15
4.2	Nuclear magnetic resonance	15
4.3	Quantum dots	17
4.4	Outlook	17

1 Introduction

These lecture notes cover the topics quantum computation and quantum information theory on an introductory level. We assume that the reader is familiar with the basic concepts of quantum mechanics (see for instance [8]). Instead of getting lost in details we tried to present the underlying ideas in a comprehensible manner. The only exception is Grover's algorithm which has been described more thoroughly in order to achieve a deeper understanding of the workflow of a typical quantum algorithm. Suggestions for further readings are given in the references, of which we particularly recommend [4,5].

1.1 Why quantum computation?

According to Moore's law [1] the number of transistors of an integrated circuit, with respect to minimum component costs, doubles approximately every 18 – 24 months. Up to now, this exponential growth has not saturated and by simply extrapolating this behavior the space for storing a single bit of information will scale down to the atomic size around 2020. At that point, quantum effects will become unavoidably dominant and instead of pushing the silicon-based transistor to its physical limits it might be more reasonable to exploit the *principles of quantum mechanics* in an intrinsic way.

The power of quantum computation is due to typical quantum phenomena, such as the *superposition* of quantum states and *entanglement*. There is an inherent quantum parallelism associated with the superposition principle. In simple terms, a quantum computer can process a large number of classical inputs in a single run. On the other hand, this would lead to a large number of possible outputs. It is the task of quantum algorithms to amplify the desired output by interference of all states. To be useful, quantum computers require the development of appropriate quantum algorithms. In specific cases — like the factoring of numbers (Shor's algorithm) or the searching of an unstructured data base (Grover's algorithm) — a remarkable speedup in comparison with classical solutions can be achieved. We discuss the most prominent examples of such algorithms in chapter 3.

Around 1980 Richard Feynman suggested that a quantum computer would be ideal for simulating quantum mechanical systems. Because of the exponentially large Hilbert space of such systems the simulation on conventional computers is extremely memory consuming.

Last but not least, the unitary evolution in quantum mechanics is reversible, thus there is no energy dissipation. Therefore, no a priori limitation upon the length scale of devices is given by power requirements.

A large number of different proposals to build quantum computers exist. They range from cold-ion traps to nuclear magnetic resonance and quantum-dots on semiconductor basis. Even though in some cases elementary quantum gates have been implemented and quantum algorithms with a small number of qubits have been performed, it is too early to say what type of implementation will be the most suitable to build a scalable piece of quantum hardware. A short overview of different hardware types is given in chapter 4.

1.2 Complexity classes

Complexity theory addresses the question how difficult it is to solve a given mathematical problem. Problems are classified according to the increase of time, a classical computer would need

to solve the task, when increasing its “size”. For example, the size can be the number of bits that define the problem.

We say that a problem belongs to the computational class **P** if it can be *solved* in polynomial time, i.e., in a number of steps that is polynomial in the input size. Instead, the computational class **NP** is the class of problems whose solution can be *verified* in polynomial time. It is clear that **P** is a subset of **NP**. Here we find all the problems whose solution can be easily verified and that are also easy to solve. It is an open problem whether $\mathbf{P} \neq \mathbf{NP}$. If this were the case, there would be problems hard to solve but whose solution could be easily checked. For instance, the integer factoring problem belongs to the class **NP**, since it is easy to check if a number m is a prime factor of an integer N , but no algorithm is known that allows to efficiently compute the prime factors of N on a classical computer. Therefore, it has been conjectured, though not proven, that the integer-factoring problem does not belong to the class **P**.

2 Quantum computation

The elementary unit of quantum information and the basic building block of quantum computation is the qubit, a two-level quantum system that can be prepared, manipulated, and measured in a controlled way. A quantum computer can be seen as a collection of n qubits and therefore its wave function resides in a 2^n -dimensional complex Hilbert space. As far as coupling to the environment may be neglected, its evolution in time is unitary and governed by the Schrödinger equation.

A quantum computation is composed of three basic steps: preparation of the input state, implementation of the desired unitary transformation acting on this state, and measurement of the output state. The output of the measurement process is inherently probabilistic and the probabilities of the different possible outputs are set by the basic postulates of quantum mechanics. Therefore, in a quantum algorithm we must, in general, repeat the algorithm several times to obtain the correct solution of our problem with probability as close to one as desired. In this sense, quantum algorithms are analogous to classical probabilistic algorithms. However, the superposition principle and quantum entanglement open up new possibilities for computation. Quantum computers are potentially more powerful than classical (deterministic or probabilistic) computers due to quantum interference and entanglement.

2.1 The qubit

A classical bit is a system that can exist in two distinct states, which are used to represent 0 and 1, that is, a single binary digit. The only possible operations (gates) in such a system are the identity ($0 \rightarrow 0, 1 \rightarrow 1$) and NOT ($0 \rightarrow 1, 1 \rightarrow 0$). In contrast, a quantum bit (*qubit*) is a two-level *quantum* system, described by a two-dimensional complex Hilbert space. In this space, one may choose a pair of normalized and mutually orthogonal quantum states,

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad (1)$$

to represent the values 0 and 1 of a classical bit. These two states form a computational basis. From the superposition principle, any state of the qubit may be written as

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \quad (2)$$

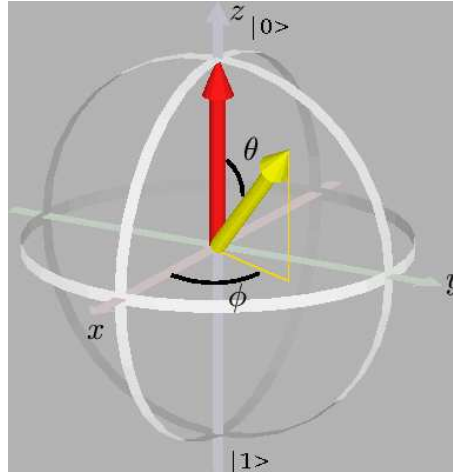


Fig. 1: The state of a qubit can be represented graphically by a point on the Bloch sphere.

where the amplitudes α and β are complex numbers, constrained by the normalization condition

$$|\alpha|^2 + |\beta|^2 = 1. \quad (3)$$

Since state vectors are defined only up to a global phase of no physical significance, one may choose α real and positive (except for the basis state $|1\rangle$, for which $\alpha = 0$). Thus, the generic state of a qubit may be written as

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle = \begin{pmatrix} \cos \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} \end{pmatrix} \quad (0 \leq \theta \leq \pi, 0 \leq \phi < 2\pi). \quad (4)$$

Therefore, unlike the classical bit, which can only be set equal to 0 or 1, the qubit resides in a vector space, parameterized by the continuous variables α and β (or θ and ϕ). Thus, a continuum of states is allowed. At this stage, one might be tempted to say that a single qubit could be used to store an infinite amount of information. However there is a catch: to extract this information we must perform a measurement which gives 0 with chance $|\alpha|^2$ and 1 with probability $|\beta|^2 = 1 - |\alpha|^2$. Thus, we obtain only a *single* bit of information. Infinitely many measurements on identically prepared states are required to obtain α and β .

2.1.1 The Bloch sphere

The Bloch sphere provides a geometric picture of the qubit and of the transformations that operate on its state: Due to the normalization condition (3), the qubit's state can be represented by a point on a sphere of unit radius, called the *Bloch sphere*. Fig. 1 illustrates such a graphical representation of a state vector. The angles shown correspond to those denoted in eqn. (4).

For the generic state of a qubit

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \quad \text{with } \alpha \geq 0 \quad (5)$$

the angles θ and ϕ can be calculated according to

$$\theta = 2 \arccos(\alpha) \quad (6)$$

$$\phi = \begin{cases} \arctan\left(\frac{\text{Im}(\frac{\beta}{|\beta|})}{\text{Re}(\frac{\beta}{|\beta|})}\right) + \begin{cases} 0 & \text{Re}(\frac{\beta}{|\beta|}) > 0 \\ \pi & \text{Re}(\frac{\beta}{|\beta|}) < 0 \end{cases} \\ \text{Im}(\frac{\beta}{|\beta|}) \frac{\pi}{2} & \text{Re}(\frac{\beta}{|\beta|}) = 0 \end{cases} . \quad (7)$$

The condition $\alpha \geq 0$ of eqn. (5) can always be fulfilled by extracting a global phase.

2.2 The circuit model of quantum computation

In contrast to classical computers, which are assembled from different functional components like the CPU, memory and a hard disk drive, a quantum computer may be thought of as a finite collection of n qubits, a *quantum register* of size n . Thus, there is no subdivision in CPU and memory, the qubits are stored in the quantum register and also manipulated there. Hence, the state of an n -qubit quantum computer can be described by

$$\begin{aligned} |\psi\rangle &= \sum_{i=0}^{2^n-1} c_i |i\rangle \\ &= \sum_{i_{n-1}=0}^1 \cdots \sum_{i_1=0}^1 \sum_{i_0=0}^1 c_{i_{n-1}, \dots, i_1, i_0} |i_{n-1}\rangle \otimes \cdots \otimes |i_1\rangle \otimes |i_0\rangle , \end{aligned} \quad (8)$$

in which the complex numbers c_i obey the normalization condition

$$\sum_{i=0}^{2^n-1} |c_i|^2 = 1 . \quad (9)$$

The wave function $|\psi\rangle$ resides in a 2^n -dimensional Hilbert space which is constructed as the tensor product of n 2-dimensional Hilbert spaces, one for each qubit. Thus, the number of basis states grows *exponentially* with the number of qubits which limits the simulation of quantum computers by the memory of the conventional computer used [9].

As an example, we consider a 2-qubit quantum computer described by a generic state

$$\begin{aligned} |\psi\rangle &= c_0 |0\rangle + c_1 |1\rangle + c_2 |2\rangle + c_3 |3\rangle \\ &= c_{0,0} |0\rangle \otimes |0\rangle + c_{0,1} |0\rangle \otimes |1\rangle + c_{1,0} |1\rangle \otimes |0\rangle + c_{1,1} |1\rangle \otimes |1\rangle \\ &= c_{00} |00\rangle + c_{01} |01\rangle + c_{10} |10\rangle + c_{11} |11\rangle . \end{aligned} \quad (10)$$

In the last line we have used the shorthand notation $|i_1 i_0\rangle = |i_1\rangle \otimes |i_0\rangle$ which allows us to write the state (8) in a more compact form

$$|\psi\rangle = \sum_{i_{n-1}, \dots, i_1, i_0=0}^1 c_{i_{n-1} \dots i_1 i_0} |i_{n-1} \dots i_1 i_0\rangle . \quad (11)$$

While n classical bits can store only a single integer i , the n -qubit quantum register can be prepared also in a superposition of those states. In contrast to classical computers where different inputs require separate runs, a quantum computer can handle exponentially many inputs in a single computation.

A quantum computation consists of the following steps:

1. *Preparation* of the quantum computer in a well-defined initial state $|\psi_i\rangle$, for instance $|0 \dots 00\rangle$.
2. *Manipulation* of the wave function in terms of unitary transformations $|\psi'\rangle = \mathcal{U} |\psi\rangle$. The sequence of unitary transformations corresponds to the quantum program.
3. *Measurement* of the polarization of each qubit at the end of the algorithm.

The state $|\psi\rangle$ of the quantum computer evolves according to the Schrödinger equation. As a result, the time-evolution is described by a unitary operator. In the following we neglect non-unitary decoherence effects, due to the undesired coupling of the qubits to the environment, which is inherent in physical realizations of quantum computers (see chapter 4), and focus on the *ideal* quantum computer.

Even though the evolution of an n -qubit wave function is described by a $2^n \times 2^n$ unitary matrix, it is always possible to decompose this matrix into a product of single- and two-qubit operations. These are the elementary *quantum gates* of the circuit model of quantum computation.

2.3 Single-qubit gates

The operations on a qubit must preserve the normalization condition (3) and are thus described by *unitary* 2×2 matrices. In this section, we will introduce the *Hadamard* and the *phase-shift* gates and show that they are sufficient to perform any unitary operation on a single qubit.

The Hadamard gate is defined as

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (12)$$

and we have

$$\begin{aligned} H|0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \equiv |+\rangle \\ H|1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \equiv |-\rangle. \end{aligned} \quad (13)$$

Thus, the computational basis $\{|0\rangle, |1\rangle\}$ is turned into the superposition states $\{|+\rangle, |-\rangle\}$.

We define the phase-shift gate as

$$R_z(\delta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\delta} \end{pmatrix} \quad (14)$$

and get

$$\begin{aligned} R_z(\delta)|0\rangle &= |0\rangle \\ R_z(\delta)|1\rangle &= e^{i\delta}|1\rangle. \end{aligned} \quad (15)$$

Since global phases are physically unobservable, the states of the computational basis, $|0\rangle$ and $|1\rangle$ remain unchanged. However, the action of $R_z(\delta)$ on a generic single-qubit state $|\psi\rangle$, gives (cf. eqn. (4))

$$R_z(\delta)|\psi\rangle = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\delta} \end{pmatrix} \begin{pmatrix} \cos \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} \end{pmatrix} = \begin{pmatrix} \cos \frac{\theta}{2} \\ e^{i(\phi+\delta)} \sin \frac{\theta}{2} \end{pmatrix}. \quad (16)$$

This corresponds to a counterclockwise rotation of $|\psi\rangle$ through an angle δ about the z -axis of the Bloch sphere (see Fig. 1).

By using only Hadamard and phase-shift gates, we can construct *any* unitary operation on a single qubit. To demonstrate this, we show that the generic state (4) can be reached by starting from $|0\rangle$:

$$R_z(\frac{\pi}{2} + \phi) H R_z(\theta) H |0\rangle = e^{i\frac{\theta}{2}} (\cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle) \quad (17)$$

More generally, the operator

$$R_z(\frac{\pi}{2} + \phi_2) H R_z(\theta_2 - \theta_1) H R_z(-\frac{\pi}{2} - \phi_1) \quad (18)$$

transfers the state parameterized by (θ_1, ϕ_1) into the one given by (θ_2, ϕ_2) .

2.3.1 Rotations of the Bloch sphere

We now consider *rotations* of the Bloch sphere about the axes of the Cartesian coordinate system. First of all, let us define the *Pauli matrices* σ_x , σ_y and σ_z as

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (19)$$

Since $\sigma_{\square}^2 = \mathbf{1}$ for $\square = x, y, z$, we have for $k \in \mathbb{N}$

$$\sigma_{\square}^k = \begin{cases} \mathbf{1} & k \text{ even} \\ \sigma_{\square} & k \text{ odd} \end{cases}. \quad (20)$$

Hence, we obtain for the Taylor expansion

$$\begin{aligned} e^{-i\frac{\delta}{2}\sigma_{\square}} &= \left[1 + \frac{1}{2!} \left(\frac{\delta}{2}\right)^2 + \dots \right] \mathbf{1} - i \left[\frac{\delta}{2} + \frac{1}{3!} \left(\frac{\delta}{2}\right)^3 + \dots \right] \sigma_{\square} \\ &= \cos\left(\frac{\delta}{2}\right) \mathbf{1} - i \sin\left(\frac{\delta}{2}\right) \sigma_{\square}. \end{aligned} \quad (21)$$

For example by choosing $\square = z$ we get

$$e^{-i\frac{\delta}{2}\sigma_z} = \cos\left(\frac{\delta}{2}\right) \mathbf{1} - i \sin\left(\frac{\delta}{2}\right) \sigma_z = e^{-i\frac{\delta}{2}} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\delta} \end{pmatrix} \equiv R_z(\delta), \quad (22)$$

which corresponds exactly to the phase-shift gate defined in the last section. Geometrically, $R_z(\delta)$ induces a counterclockwise rotation through the angle δ about the z -axis of the Bloch sphere. This can be seen by applying $R_z(\delta)$ to $|\psi\rangle$ given by eqn. (4)

$$R_z(\delta) |\psi\rangle = \cos \frac{\delta}{2} |0\rangle + e^{i(\phi+\delta)} \sin \frac{\delta}{2} |1\rangle. \quad (23)$$

Analogously, one can obtain the unitary matrices corresponding to counterclockwise rotations about the other axes of the Cartesian coordinate system

$$\begin{aligned} e^{-i\frac{\delta}{2}\sigma_x} &= \begin{pmatrix} \cos \frac{\delta}{2} & -i \sin \frac{\delta}{2} \\ -i \sin \frac{\delta}{2} & \cos \frac{\delta}{2} \end{pmatrix} \equiv R_x(\delta) \\ e^{-i\frac{\delta}{2}\sigma_y} &= \begin{pmatrix} \cos \frac{\delta}{2} & -\sin \frac{\delta}{2} \\ \sin \frac{\delta}{2} & \cos \frac{\delta}{2} \end{pmatrix} \equiv R_y(\delta) \\ e^{-i\frac{\delta}{2}\sigma_z} &= e^{-i\frac{\delta}{2}} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\delta} \end{pmatrix} \equiv R_z(\delta). \end{aligned} \quad (24)$$

For those who remember the lectures in quantum mechanics: a rotation about the x -, y - or z -axis is generated by

$$\mathbf{G}_{\square} = \frac{\hbar}{2} \boldsymbol{\sigma}_{\square} \quad (25)$$

and the rotation operator in counterclockwise direction is given by (cf. for instance [8])

$$R_{\square}(\delta) = e^{-\frac{i}{\hbar} \mathbf{G}_{\square} \delta}, \quad (26)$$

where \square stands for x , y or z , respectively. This is in direct agreement with eqn. (24).

2.4 Two-qubit gates

In order to prepare an entangled¹ state one needs inter-qubit *interactions* which can be generated by 2- or 3-qubit gates. The *controlled-NOT* gate is the prototypical two-qubit gate that is able to generate entanglement. Here, the first qubit acts as a *control* and the second as a *target*. The gate flips the state of the target qubit if the control qubit is in the state $|1\rangle$ and leaves the target position unchanged if the control qubit is equal to $|0\rangle$. If we denote the basis states as column vectors

$$|0\rangle = |00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |1\rangle = |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad |2\rangle = |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |3\rangle = |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad (27)$$

we have the following matrix representation of the CNOT gate

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (28)$$

This gate acts on the basis states as the classical XOR gate: $\text{CNOT}(|x\rangle|y\rangle) = |x\rangle|x \oplus y\rangle$, with $x, y = 0, 1$ and \oplus indicating addition modulo 2. Furthermore, the CNOT can be used to generate entangled states. For example,

$$\text{CNOT}(\alpha|0\rangle + \beta|1\rangle)|0\rangle = \alpha|00\rangle + \beta|11\rangle, \quad (29)$$

which is non-separable as far as $\alpha, \beta \neq 0$.

In contrast to the CNOT gate, the *controlled phase shift* has no classical analog:

$$\text{CPHASE}(\delta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\delta} \end{pmatrix} \quad (30)$$

This gate applies a phase shift to the target qubit only when the control qubit is in the state $|1\rangle$ and the target position is equal to $|1\rangle$: $\text{CPHASE}|11\rangle = e^{i\delta}|11\rangle$.

Without proof, we emphasize that the *Hadamard* gate (12), the *phase-shift* gate (14) and the *CNOT* gate (28) form a *set of universal operations* [2], i.e. by using only those universal quantum gates we can construct any arbitrary quantum operation.

¹An entangled state can not be represented as a product of subsystem wave functions $|\psi\rangle \neq |\psi_1\rangle \otimes \dots \otimes |\psi_n\rangle$.

3 Quantum algorithms

3.1 Deutsch's algorithm: constant or balanced function

Deutsch's problem illustrates the computational power of quantum *interference*. We consider a black box called the *oracle* which evaluates the function $f : \{0, 1\} \rightarrow \{0, 1\}$. There are four of such functions which are listed in the following table:

x	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

Those functions can be classified according to a *global* property: two of them are *constant* (f_0 and f_3) and two *balanced* (f_1 and f_2). *The problem is to decide whether a given function is constant or balanced.* On a classical computer this task requires two queries of the oracle. A quantum computer can solve the same problem with only one oracle query:

We need one ancillary qubit $|y\rangle$. On a quantum level the oracle corresponds to a unitary transformation U_f

$$U_f |x\rangle|y\rangle = |x\rangle |y \oplus f(x)\rangle \quad (31)$$

where \oplus denotes addition modulo 2. That is, the second qubit is flipped if and only if $f(x) = 1$. In the case that the second qubit is in a superposition we obtain

$$\begin{aligned} \frac{1}{\sqrt{2}} |x\rangle (|0\rangle - |1\rangle) &\xrightarrow{U_f} \frac{1}{\sqrt{2}} |x\rangle (|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle) \\ &= \begin{cases} \frac{1}{\sqrt{2}} |x\rangle (|0\rangle - |1\rangle) & \text{if } f(x) = 0 \\ \frac{1}{\sqrt{2}} |x\rangle (|1\rangle - |0\rangle) & \text{if } f(x) = 1 \end{cases} \\ &= (-1)^{f(x)} \frac{1}{\sqrt{2}} |x\rangle (|0\rangle - |1\rangle). \end{aligned} \quad (32)$$

Both qubits remain in their primary state with $(-1)^{f(x)}$ acting as a global phase factor. For a superposition of both qubits we obtain

$$\begin{aligned} \frac{1}{2} (|0\rangle + |1\rangle)(|0\rangle - |1\rangle) &\xrightarrow{U_f} \frac{1}{2} ((-1)^{f(0)} |0\rangle + (-1)^{f(1)} |1\rangle) (|0\rangle - |1\rangle) \\ &= \frac{(-1)^{f(0)}}{2} (|0\rangle + (-1)^{f(0) \oplus f(1)} |1\rangle) (|0\rangle - |1\rangle). \end{aligned} \quad (33)$$

The relevant information is now coded in the relative phase of the superposed states of the first register. By applying a Hadamard operation we get

$$\frac{(-1)^{f(0)}}{2} (|0\rangle + (-1)^{f(0) \oplus f(1)} |1\rangle) (|0\rangle - |1\rangle) \xrightarrow{H} (-1)^{f(0)} |f(0) \oplus f(1)\rangle |1\rangle \quad (34)$$

which corresponds to $|0\rangle|1\rangle$ if f is constant and $|1\rangle|1\rangle$ in the case that f is balanced. Therefore, a global property of the function $f(x)$ has been encoded in a single qubit after a single call of f . This is because a quantum computer can evaluate both $f(0)$ and $f(1)$ simultaneously. The main point is that these two alternative “paths” are combined by the final Hadamard gate, giving the desired interference pattern. The interference is constructive for the outcome $f(0) \oplus f(1)$ and destructive for the alternative outcome.

3.2 Grover's algorithm: How to find a needle in a haystack?

Imagine that you have a telephone number of a person who is living in your town and you want to find out whose number it is by just using your telephone book. The best you can do, classically, is to go through the names one by one until you find the corresponding number. If the phone book contains N entries, you would have to check on average $N/2$ numbers. Fortunately, a *quantum search* can do better: in 1996 Lov Grover could show [10] that only \sqrt{N} queries are needed. Furthermore, Grover's algorithm is known to be *optimal* [11], i.e., no classical or quantum algorithm can solve the problem (of searching an unstructured database) faster. Since the quantum algorithm does not lie in a different complexity class than the best classical one,² the speedup is still quadratic.

The underlying idea of Grover's algorithm is to start with a superposition of *all* states and amplify the amplitude of the state searched for step by step by repeatedly applying a certain sequence \mathcal{G} of operations. After a fixed number k of iterations the amplitude of this state has gained a value close to 1, which means a measurement yields the searched element with a high probability.

For simplicity³ let us assume that we have an unstructured database that contains $N = 2^n$ different elements. We label the items as $\{0, 1, \dots, N-1\}$ and x_0 is the element searched for. The result of the quantum search process is stored in a register $|x\rangle$ which yields the index x_0 with a high probability when measured. In addition a single ancillary qubit $|y\rangle$ is needed to store the result of the oracle query. The oracle \mathcal{O} computes the n -bit binary function

$$f : \{0, 1\}^n \rightarrow \{0, 1\}, \quad (35)$$

defined as

$$f(x) = \begin{cases} 1 & \text{if } x = x_0, \\ 0 & \text{otherwise.} \end{cases} \quad (36)$$

Grover's algorithm in detail:

- Start with the state

$$|x\rangle|y\rangle = |00\dots 0\rangle|1\rangle. \quad (37)$$

- Apply $H^{\otimes n+1}$ ($n+1$ Hadamard gates) \implies equal superposition of all basis states:

$$|00\dots 0\rangle|1\rangle \rightarrow \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \quad (38)$$

- Evaluate the oracle function⁴ $|x\rangle|y\rangle \xrightarrow{\mathcal{O}} |x\rangle|y \oplus f(x)\rangle$.

This flips the sign of the $|x_0\rangle$ amplitude:

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle (|0\rangle - |1\rangle) \quad (39)$$

²Both grow polynomial in time with the number of database elements.

³Grover's algorithm does also work with partially identical elements, but this issue is a little bit more complicated (see e.g. [2]). In the case $2^{n-1} < N < 2^n$ the database can be filled up with distinguishable items so that $\tilde{N} = 2^n$ holds.

⁴ \oplus means addition modulo 2 which corresponds to a XOR operation.

- Let us *define* the Grover iteration \mathcal{G} , with $\mathcal{G} = \mathcal{I}_M \mathcal{O}$, where \mathcal{O} denotes the oracle query and

$$\mathcal{I}_M = -H^{\otimes n}(\mathbf{1} - 2|0\rangle\langle 0|)H^{\otimes n} = -\underbrace{(\mathbf{1} - 2|S\rangle\langle S|)}_{\mathcal{R}_{|S\rangle}} \quad (40)$$

is often referred to as the “inversion about the mean”.⁵ The uniform superposition $|S\rangle$ is given by $|S\rangle \equiv H^{\otimes n}|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$ and the operator⁶ $\mathcal{R}_{|S\rangle}$ mirrors a given state at the hyperplane perpendicular to $|S\rangle$ (cf. section 3.2.1).

- Now apply \mathcal{I}_M (remember: we have already used \mathcal{O} , so one Grover iteration \mathcal{G} is now completed).
- (For $N > 4$):⁷ Apply \mathcal{G} several times until a measurement of $|x\rangle$ gives x_0 with maximal probability.⁸
- Perform a measurement of the first register in the computational basis, giving outcome $x = \bar{x}$. If $f(\bar{x}) = 1$ the search was successful, otherwise repeat the algorithm.

3.2.1 Geometric visualization

The underlying idea of Grover’s algorithm gets much clearer in terms of a geometric interpretation. Since this visualization goes back to the fact that geometrically a reflection operator induces a mirroring, let us start with a short warm-up.

Preliminary consideration: reflection operator and mirroring Consider a bidimensional space spanned by the vectors $\{|x_0\rangle, |x_0^\perp\rangle\}$ and a generic vector $|\psi\rangle = \alpha|x_0\rangle + \beta|x_0^\perp\rangle$. The action of the reflection operator $\mathcal{R}_{|x_0\rangle} = \mathbf{1} - 2|x_0\rangle\langle x_0|$ on $|\psi\rangle$ is $\mathcal{R}_{|x_0\rangle}|\psi\rangle = -\alpha|x_0\rangle + \beta|x_0^\perp\rangle$. Therefore, $\mathcal{R}_{|x_0\rangle}$ changes the sign of the $|x_0\rangle$ amplitude. Geometrically this corresponds to a mirroring at the axis $|x_0^\perp\rangle$, that is, at the hyperplane perpendicular to $|x_0\rangle$ (see Fig. 2).

Next, let us prove that $-\mathcal{R}_{|S\rangle} = \mathcal{R}_{|S^\perp\rangle}$. We consider a generic vector $|u\rangle = \mu|S\rangle + \nu|S^\perp\rangle$. Application of $\mathcal{R}_{|S\rangle}$ yields $-\mu|S\rangle + \nu|S^\perp\rangle$ while $\mathcal{R}_{|S^\perp\rangle}|u\rangle = \mu|S\rangle - \nu|S^\perp\rangle = -\mathcal{R}_{|S\rangle}|u\rangle$. As a result, the “inversion about the mean” operator \mathcal{I}_M in eqn. (40) can be written as

$$\mathcal{I}_M = \mathcal{R}_{|S^\perp\rangle} . \quad (41)$$

Grover’s algorithm begins with the uniform superposition state (38)

⁵The operation \mathcal{I}_M applied to a general state $\sum_x \alpha_x |x\rangle$ yields $\sum_x [-\alpha_x + 2\langle\alpha\rangle] |x\rangle$, where $\langle\alpha\rangle \equiv \sum_x \alpha_x / N$ is the mean value of the amplitudes α_x .

⁶A projection Operator $\mathcal{P} = |a\rangle\langle a|$ satisfies $\mathcal{P}^2 = \mathcal{P}$. Since $(\mathbf{1} - \mathcal{P})^2 = \mathbf{1} - \mathcal{P}$, the right side of this equation is also a projection operator. In contrast, $\mathbf{1} - 2\mathcal{P}$ is a *reflection* operator, $(\mathbf{1} - 2\mathcal{P})^2 = \mathbf{1}$, which changes the sign of the projection onto $|a\rangle$.

⁷In the case $N = 4$ we are already done. This means that quantum mechanically we can search an unsorted data base containing 4 different elements with a single query. Classically we can handle only 2 different items with one question (e.g. by asking: “Is this the element that I want?”). As a remarkable coincidence nature uses also *four* different nucleotide bases to code the genetic information in DNA. This gave rise to the speculation that quantum search processes might be involved on a genetic level [12].

⁸The exact number of iterations k will be derived in section 3.2.1.

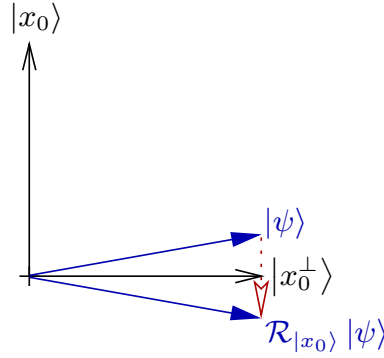


Fig. 2: The reflection operator $\mathcal{R}_{|x_0\rangle}$ flips the sign of the $|x_0\rangle$ amplitude: $\mathcal{R}_{|x_0\rangle}(\alpha |x_0\rangle + \beta |x_0^\perp\rangle) = -\alpha |x_0\rangle + \beta |x_0^\perp\rangle$. This mirrors $|\psi\rangle$ at the $|x_0^\perp\rangle$ axis.

$$|\psi_0\rangle \equiv |S\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \quad (42)$$

(for simplicity, we drop the second register whose value will not change during the rest of the algorithm). Since the plane spanned by $\{|S\rangle, |x_0\rangle\}$ can also be generated by⁹ $\{|x_0\rangle, |x_0^\perp\rangle\}$, we have

$$|\psi_0\rangle \equiv |S\rangle = \sin \theta |x_0\rangle + \cos \theta |x_0^\perp\rangle \quad (43)$$

where θ denotes the angle between the vectors $|x_0^\perp\rangle$ and $|S\rangle$. As illustrated in Fig. 3, the oracle \mathcal{O} mirrors $|\psi\rangle$ at $|x_0^\perp\rangle$ and afterwards the result $\mathcal{O}|\psi\rangle$ is mirrored at $|S\rangle$ by virtue of \mathcal{I}_M . Since both mirrorings take place in the same plane the result of the whole operation is a rotation. Fig. 3 demonstrates that the Grover iteration \mathcal{G} rotates a generic vector $|\psi\rangle$ by an angle of 2θ towards the searched element $|x_0\rangle$. After j steps of Grover's iteration the n -qubit state is given by

$$|\psi_j\rangle \equiv \mathcal{G}^j |\psi_0\rangle = \sin((2j+1)\theta) |x_0\rangle + \cos((2j+1)\theta) |x_0^\perp\rangle \quad (44)$$

since all rotations take place in the primary plane. The process must stop after k steps, where k is such that $|\psi_k\rangle$ is very close to the marked state $|x_0\rangle$. This is the case when $\sin((2k+1)\theta) \approx 1$. The smallest integer k that fulfills this condition is determined by

$$(2k+1)\theta \approx \frac{\pi}{2}, \quad (45)$$

which implies

$$k = \text{round} \left(\frac{\pi}{4\theta} - \frac{1}{2} \right), \quad (46)$$

where round specifies the nearest integer. Since we started from the uniform superposition state (42) we have

$$\sin \theta = \langle x_0 | \psi_0 \rangle = \frac{1}{\sqrt{N}}, \quad (47)$$

which leads to the following relation between the number of Grover iterations k and the volume of the database N :

$$k = \text{round} \left(\frac{\pi}{4 \arcsin(1/\sqrt{N})} - \frac{1}{2} \right) \quad (48)$$

⁹In fact, this condition defines $|x_0^\perp\rangle$.

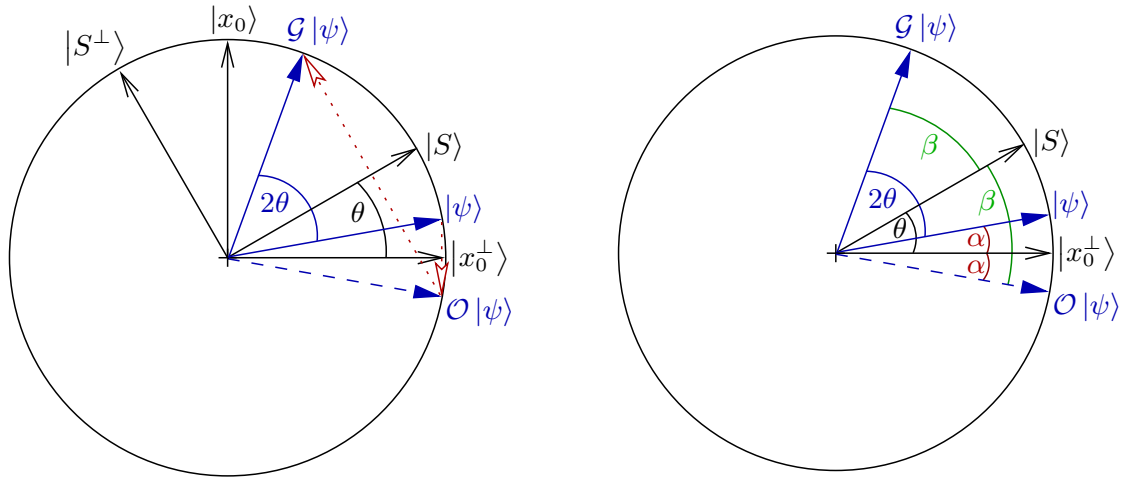


Fig. 3: Left figure: Geometric visualization of the Grover iteration \mathcal{G} . The oracle query \mathcal{O} mirrors $|\psi\rangle$ at $|x_0^\perp\rangle$ (which is achieved by $\mathcal{R}_{|x_0\rangle}$). The “inversion about the mean” \mathcal{I}_M mirrors $\mathcal{O}|\psi\rangle$ at the uniform superposition $|S\rangle$, by virtue of $\mathcal{R}_{|S^\perp\rangle}$ or the application of $-\mathcal{R}_{|S\rangle}$ (cf. eqn. (40)). Right panel: Since $\theta = \beta - \alpha$, we have $2\theta = \beta + \theta - \alpha$. Thus, $\angle(|\psi\rangle, \mathcal{G}|\psi\rangle) = 2\theta$ so that the Grover iteration \mathcal{G} rotates $|\psi\rangle$ by 2θ towards $|x_0\rangle$.

For large N the approximative behavior $\arcsin(1/\sqrt{N}) \approx 1/\sqrt{N}$ holds and we can demonstrate the quadratic speedup of Grover’s algorithm

$$k = \text{round} \left(\frac{\pi}{4} \sqrt{N} - \frac{1}{2} \right) = \mathcal{O}(\sqrt{N}) \quad (49)$$

in contrast to classical algorithms in which the number of database queries grows like $\mathcal{O}(N)$.

3.3 Shor’s algorithm: Factoring of numbers

In 1994 Peter Shor — employed by the US company AT&T Labs Research — published a quantum algorithm which allows to compute the prime factors of a given number with an exponential speed up [13]. The time needed by a classical computer to solve this problem grows exponentially with the number of digits. For instance, to factorize a number with 130 digits, about 10^{18} operations are needed. If we assume 10^{12} floating point operations per second (1 Tflops) this would take 42 days. In the case we would double the number of digits to 260 the calculation would consist of about 10^{25} operations which would last 1 million years. In practice it is not possible to solve the task. On the other hand, the verification of the result is trivial, we just have to multiply the numbers. This asymmetry is the crucial point that makes the RSA encryption scheme so successful.

Let us examine an illustrative example: 15 is the product of the prime numbers 3 and 5. In this case it is possible to find the factors by trial and error: first we try 2 as a factor which fails. Next we try 3 and succeed. If we denote the number we want to factorize by N , it takes \sqrt{N} trials to find the factors in the worst case. According to the binary representation of the number, which we assume to have L digits, this corresponds to $2^{L/2}$ attempts which means that the time for this simple algorithm grows exponentially with the number of digits. The best algorithm known grows like $e^{L/3}$. In contrast, Shor’s algorithm grows polynomial in time like L^3 .

It has been known that the factorization of a number is connected to the period of a certain function of that number. Unfortunately, the finding the period of a given function is also exponentially time consuming. However, quantum mechanically this can be done in polynomial time.

In the following we will focus on the underlying idea of Shor's algorithm. For further details see, e.g., [2]. The factorization of N is equivalent to finding the period of

$$f(x) = a^x \mod N, \quad (50)$$

where a is an arbitrary fixed number $a < N$ which does not divide N and $x \in \mathbb{N}$. Let's reconsider the example $N = 15$. We choose $a = 7$, which yields $f(x) = 7^x \mod 15$. The following table shows $f(x)$ in dependence on x :

x	1	2	3	4	5	6	...
$f(x)$	7	4	13	1	7	4	...

In this case we have the period $r = 4$. Once the period of $f(x)$ has been found, we can calculate factors of $N = p \cdot q$ according to

$$\begin{aligned} p &= \gcd(a^{r/2} + 1, N) \\ q &= \gcd(a^{r/2} - 1, N), \end{aligned} \quad (51)$$

where \gcd denotes the *greatest common divisor*. In case of our example we get

$$\begin{aligned} p &= \gcd(7^{4/2} + 1, N) = \gcd(50, 15) = 5 \\ q &= \gcd(7^{4/2} - 1, N) = \gcd(48, 15) = 3 \end{aligned} \quad (52)$$

and thus $15 = 5 \cdot 3$. Finally, we remark that the computation of the \gcd is only of polynomial complexity (already 300 B.C. Euclid found an algorithm for this task).

Classically, the period finding of $f(x)$ is as hard as the factoring of N itself. Fortunately, quantum mechanics helps in this case. As in Grover's algorithm we need a second register. We start with the uniform superposition

$$|0\rangle|0\rangle \rightarrow \frac{1}{\sqrt{2^L}} \sum_{x=0}^{2^L-1} |x\rangle|0\rangle. \quad (53)$$

In the next step the value of the function f is calculated and stored in the second register. Since we started from a uniform superposition state all function values are calculated in a single step

$$\frac{1}{\sqrt{2^L}} \sum_{x=0}^{2^L-1} |x\rangle|0\rangle \rightarrow \frac{1}{\sqrt{2^L}} \sum_{x=0}^{2^L-1} |x\rangle|f(x)\rangle. \quad (54)$$

In a third step we carry out a measurement on the second register. As we will illustrate on our example, as a result of this measurement the first register will be in a superposition of states with the periodicity searched for. The period can be extracted with a special technique called *fast quantum Fourier transform*.

In the case $N = 15$ and $a = 7$ we have

$$|0\rangle|0\rangle \rightarrow |1\rangle|0\rangle + |2\rangle|0\rangle + |3\rangle|0\rangle + \dots \quad (55)$$

due to the initialization process (for simplicity we omit the normalization factor). In the next step the value of the function $f(x)$ is written in the second register

$$\rightarrow |1\rangle |7\rangle + |2\rangle |4\rangle + |3\rangle |13\rangle + |4\rangle |1\rangle + |5\rangle |7\rangle \dots \quad (56)$$

Now we measure the second register. Since all of the amplitudes occur with equal probability we get 7, 4, 13 or 1. Let us assume that the result is 7. In this case the first register corresponds to

$$|\psi\rangle \propto |1\rangle + |5\rangle + |9\rangle + \dots \quad (57)$$

Those states differ by 4 which is the period we are looking for. The period can be extracted by the fast quantum Fourier transform.

The crucial point is that the period manifests itself as a *global* property of the wave function. This global feature can efficiently be read out. This leads to the remarkable exponential speed-up in comparison to classical algorithms.

4 How to build a quantum computer?

4.1 Ion traps

In 1995 I. Cirac and P. Zoller – both working at the time at Innsbruck University – proposed the implementation of quantum gates on the basis of ion traps [14]. Charged ions are trapped in a small region by a time dependent electrical field. The sophisticated arrangement of potentials allows only the motion in one direction and leads to a linear alignment of the ions. The typical distance between the ions is about $10\mu m$ which is large enough to address them individually by laser pulses.

Each ion represents a qubit in which two of the *internal* states correspond to $|0\rangle$ and $|1\rangle$: the state $|0\rangle$ is represented by the ground state while $|1\rangle$ is given by a metastable excited level. In order to implement an 1-qubit gate one has to focus an ion by a certain laser pulse with a frequency that corresponds to the difference of the $|1\rangle$ and $|0\rangle$ energies.

The interaction between qubits can be realized by exploiting the *external* degrees of freedom. Since the ions are electrically charged they repel each other. In order to keep them at rest the ions have to be cooled strongly by laser cooling so that the temperature is in the range of a few milli-Kelvin. If the energy of the ion chain is increased, the whole chain starts to oscillate. Depending on the energy, different oscillation modes can occur: for example all ions can oscillate in the same direction or antipodal to each other. It is also possible that the ion in the middle stays at rest while the outer qubits oscillate in reversed directions on the right and left side. This collective motion is used as a bus to implement 2-qubit gates between arbitrary ions. The excitation of the first ion is transferred to the chain as oscillatory energy; afterwards the second ion is addressed in a way that the ion chain transfers their energy to this qubit.

4.2 Nuclear magnetic resonance

Nuclear magnetic resonance (NMR) is based on the resonant excitation of nuclear magnetic moments. Since nuclear spins are the central resource in this field, NMR seems to be a promising technology for the construction of quantum computers. The first investigations on this field were carried out by D. Cory *et al.* in 1996 and N. Gershenfeld in collaboration with I. Chuang in 1997 [15].

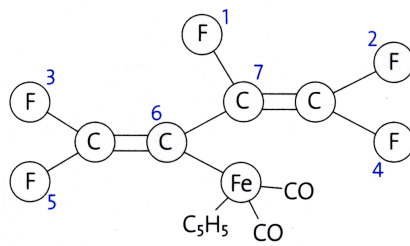


Fig. 4: Left: The molecule that has been used by L. Vandersypen *et. al* [16] in order to implement Shor's algorithm on an NMR quantum computer. Right: Isaac L. Chuang at the IBM Almaden Research Center holding a sample of those molecules.

In case of quantum computation on NMR basis the qubits are represented by the nuclear spins of certain atoms in a specially designed molecule. A strong magnetic field is used to align the spins in a specific orientation. The parallel adjustment according to the B -field is energetically favored and thus interpreted as $|0\rangle$ while the antiparallel orientation corresponds to state $|1\rangle$.

As a crucial difference to other technologies, NMR is operated at room temperature which makes it much easier to handle than experiments which need extreme cooling. On the other hand this advantage implies also a drawback: in the thermal equilibrium only a few more spins are oriented parallel to the magnetic field than antiparallel. Therefore, the initialization of the initial state is a nontrivial issue.

One-qubit operations can be easily implemented by standard NMR techniques: the state of the qubit is rotated by a resonant oscillating field perpendicular to the primary field. The frequency of the oscillating pulse is in the range of 500 MHz and corresponds to the energy gap between the initial and final state. Since each of the nuclei is located in a slightly different chemical environment, their resonance frequencies differ by a small amount. This allows to address each of them separately.

In order to implement 2-qubit operations interaction between two nuclear spins must be established: this is possible because two parallel spins have different physical properties than antiparallel ones. Due to the coupling the first spin rotates faster (slower) if the second is parallel (antiparallel) aligned. This dependence of the transformation property of one qubit by a second one is exactly the idea underlying the CNOT gate, which can be implemented in this manner.

So far NMR has been very promising: it was possible to implement different algorithms. Shor's algorithm was carried out on seven qubits in order to factorize the number $15 = 3 \cdot 5$. The molecule that has been used by L. Vandersypen *et. al* [16] for this task is shown in Fig. 4. Fluorine atoms represent five of the qubits and the further two correspond to carbon.

In spite of this success NMR does not seem to be the technique of choice for future applications since it does not scale: On the one hand it is difficult to design special molecules with a larger number of qubits, on the other hand the strength of the signal decreases exponentially with the number of qubits.

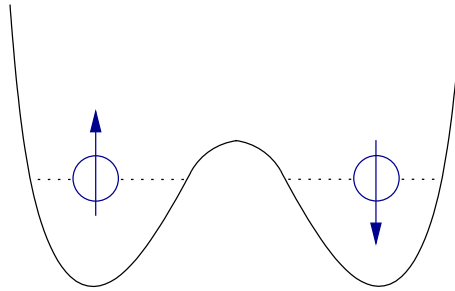


Fig. 5: A simple two quantum dot model. The interaction of the qubits can be achieved by decreasing the potential barrier in the middle.

4.3 Quantum dots

Promising ideas for the implementation of quantum computers arise also from solid state physics even though the realization is beyond today's technical feasibility. The hope is to advance semiconductor technology in order to utilize it for quantum devices.

D. Loss and D. DiVincenzo proposed in 1998 to use quantum dots in semiconductors as qubits [17]. A quantum dot is a structure of a few cubic nanometers in which a single electron can be trapped. Quantum dots behave like artificial atoms. The spin of the single electron is used as a qubit. Due to a local magnetic field the spin can be rotated – in this manner one qubit gates can be carried out. Two neighboring spins are separated by a potential barrier. By lowering this barrier tunneling processes take place and the two spins are able to interact with each other. Through a sophisticated steering of the barrier it is in principle possible to construct two qubit gates. A simple model of two quantum dots is shown in Fig. 5.

4.4 Outlook

It is difficult to foresee the future development of quantum information. Realistically, it will not be possible to build a quantum computer with one hundred qubits within the next few years. At the moment quantum cryptography has been developed on a level which is much closer to commercial application. New ideas about precision measurements as a quantum technology are also an important issue. Nevertheless, a huge benefit from the present research on quantum information is a deeper understanding of quantum phenomena.

References

- [1] G. Moore, *Cramming more components onto integrated circuits*, Electronics 38 (1965) 114.
- [2] M.A. Nielsen and I.L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge (2000).
- [3] *Information, Science, and Technology in a Quantum World*, Los Alamos Science, Number 27 (2002).
<http://www.fas.org/sgp/othergov/doe/lanl/pubs/number27.htm>
- [4] G. Benenti, G. Casati and G. Strini, *Principles of Quantum Computation and Information, Volume I: Basic Concepts*, World Scientific, Singapore (2004).
- [5] D. Bruß, *Quanteninformationstheorie*, lecture notes WS2004/2005 (in German),
<http://www.thphy.uni-duesseldorf.de/~ls3/QI-Skriptum-050210.pdf>
D. Bruß, *Quanteninformation*, Fischer, Frankfurt (2003) (in German).
- [6] A. Steane, *Quantum computing*, Rept. Prog. Phys. 61 (1998) 117, quant-ph/9708022.
- [7] D. Aharonov, *Quantum computation*, Annual Reviews of Computational Physics, ed. Dietrich Stauffer, World Scientific, vol VI, 1998, quant-ph/9812037.
- [8] J.J. Sakurai, *Modern Quantum Mechanics*, Addison-Wesley, Reading (1985).
- [9] G. Arnold, Th. Lippert, N. Pomplun and M. Richter, *Large Scale Simulation of Ideal Quantum Computers on SMP-Clusters*, to appear in the proceedings of the PARCO05 conference.
- [10] L.K. Grover, *A fast quantum mechanical algorithm for database search*, in Proc. of the 28th Annual ACM Symposium on the Theory of Computing, p. 212, ACM Press, New York (1996), quant-ph/9605043; L.K. Grover, *Quantum Mechanics helps in searching for a needle in a haystack*, Phys. Rev. Lett. 79 (1997) 325, quant-ph/9706033.
- [11] C. Zalka, *Grover's quantum search algorithm is optimal*, Phys. Rev. A60 (1999) 2746, quant-ph/9711070.
- [12] A. Patel, *Quantum Algorithms and the Genetic Code*, Pramana 56 (2001) 365, quant-ph/0002037; *Live Force*, New Scientist, April 15th (2000).
- [13] P.W. Shor, *Algorithms for quantum computation: discrete logarithms and factoring*, Proceedings of the 35th Annual Symposium on Foundations of Computer Science, IEEE Press, Los Alamitos (1994).
- [14] J.I. Cirac and P. Zoller, *Quantum Computations with Cold Trapped Ions*, Phys. Rev. Lett. 74 (1995) 4091.
- [15] D.G. Cory, A.F. Fahmy and T.F. Havel, *Nuclear magnetic resonance: an experimentally accessible paradigm for quantum computing*, Proceedings of PhysComp '96 (1996) 87. N. Gershenfeld and I. Chuang, *Bulk Spin-Resonance Quantum Computation*, Science 277 (1997) 1689.

-
- [16] L.M.K. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, M.H. Sherwood and I.L. Chuang, *Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance*, Nature 414 (2001) 883.
- [17] D. Loss and D.P. DiVincenzo, *Quantum Computation with Quantum Dots*, Phys. Rev. A 57 (1998) 120.