Faculty of mathematics and physics
University of Ljubljana

SEMINAR

# Quantum computing

author: Gregor Resman
adviser: prof. Anton Ramšak

Dragomelj, november 2001

## Contents



## 1. Abstract

The aim of this seminar is to inform the reader of the basics of quantum computation. How information is implemented is described as well as actions on logical one and zero in quantum systems. Efficient algorithms for quantum computers are mentioned. I wanted to focus the work on how these computers are actually made. Unfortunately each implementation could be a seminar on its own, so only the basics are covered.

## 2. Introduction

As the world evolves human being is eager to develop and use new, better, faster technologies. Since R.P. Feynman 1982 showed the potential of quantum systems, the research in this direction has not been stopped. Boost for the research was the fact of quantum systems being in linear combination of eigenstates. First just theories on how to use this parallelism were developed and now we already have simple but functioning quantum computers working with eight qubits.

## 3. Qubit

Classical information is stored in bits, strings of ones and zeroes representing numbers, letters, etc. In quantum informational technology we use quantum bits - qubits[8]. Logical one and zero are represented by two orthogonal states of the system, which form a computational basis. So any other state of the system can be written as a linear conbination of these two: $\alpha|1> +\beta|0>$. We can create a superposition of qubits. With two qubits we can for example produce a product(separable) state

$$\frac{1}{\sqrt{2}}(|0>+|1>)\otimes\frac{1}{\sqrt{2}}(|0>+|1>).$$

In our discussion we will ignore the normalizing factor, as the equations are easier to comprehend without it. It can be, of course, restored easily. Expanding the written example we gain a superposition of all possible states of the system

$$|0>|0>+|0>|1>+|1>|0>+|1>|1>$$

As written, a quantum register can store all the numbers between $0$ and $2^n-1$ at once, if $n$ is the number of qubits in the system. Be aware that if there are $n$ qubits in the system, the system has $2n$ states. If we follow a notation $|a>|b>=|ab>$, we get

$$|00>+|01>+|10>+|11>,$$

what in decimal equals

$$|0>_D+|1>_D+|2>_D+|3>_D=\sum_{i=0}^{2^n-1}|i>_D\ .$$

The written state however cannot be measured. When a measurement is performed the state function is at one of the pure states. We can measure the probability for a specific state. As written it would seem as the probabilities are the same, but they aren't. They in fact change with time. Each coefficient is namely time dependent with its own oscilating frequency. Storing all the numbers from the example should be written as

$$|0>_D e^{-itE_{00}/\hbar}+|1>_D e^{-itE_{01}/\hbar}+|2>_D e^{-itE_{10}/\hbar}+|3>_D e^{-itE_{11}/\hbar}=\sum_{i=0}^{2^n-1}|i>_D e^{-itE_i/\hbar}.$$

Some strictly make a difference between entangled and separable states. Entangled states are the states, that cannot be written as a product:

$$|11> + |00> \ or \ |10> + |01>$$

while separable can

$$\alpha|11> + \beta|10> = |1>(\alpha|1> + \beta|0>).$$

There are however some differences i.e. entangled states are harder to produce. Their measurement on the other hand is more accurate [9, p.139].
When seeking for a proper implementation of a computer, we must also consider interaction between qubits, which is a must. A typical qubit is therefore polarization of a photon, an array of quantum dots, a nuclear spin or any other state of an atom. As optical systems are big and therefore not promising, they are not of our interest. We will discuss implementation of qubits using the last two techniques.

### 3.1. Big molecules and spins

Nuclear spin has been interesting because of its isolation from electronic and vibrational mechanisms[2]. A suggestion was to create an apparatus to address single nuclear spins one at a time. This approach would avoid the thermal problem because by definition a single system is always in a pure quantum state. Unfortunately the experimental approach for this is very difficult to realize because of the high sensitivity required. The method would be an atomic resolution magnetic scanning force probe. The signal induced by a single nucleus is namely extremely weak.
A better way than addressing single nucleus is to use a bigger molecule, a multispin molecular system such as (2,3) dibromothiophene.

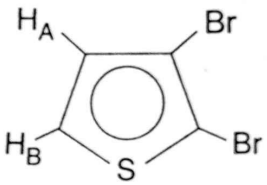



Figure 1: (2,3) dibromothiophene, which is used as a two qubit quantum computer.

The two hydrogen atoms have similar precession frequencies (200 MHz; B=5T). They differ however for a few kHz. Now we can address the whole molecule with slightly different frequencies to act on specific qubit. Spins of the same molecule interact through dipolar interactions. That is the way to a two qubit computer.
As we need a stronger signal we use more molecules of the same kind $(10^{23})$. Now statistics come in. In thermal equilibrium the $N$ spins of each molecule are arranged in some distribution of energy eigenstates. The lower the energy the more populated the state is. We describe this populations with a density matrix [2]. Let us have a look at a matrix for two state (one qubit) system

$$\rho = \begin{bmatrix} p\downarrow & 0 \\ 0 & p\uparrow \end{bmatrix}.$$

where $p\uparrow$ and $p\downarrow$ denote the population probabilities for the two energy levels: $p_i = 1/2\exp(-\frac{E_i}{2kT})$. Since $\frac{E_i}{2kT} << 1$, the matrix can be linerised:

$$\rho = \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} -E_1/2kT & 0 \\ 0 & -E_2/2kT \end{bmatrix}$$

The second part is only a small deviation ($10^{-5}B[T]$), but this is the thing we measure.

$$\rho = \frac{I}{2^{2N}} + \rho_\Delta$$

For example of our molecule (two qubit system) the deviation of the density matrix can be

$$\rho_\Delta = \alpha \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

This representation means that $1/4 + 3\alpha$ of all molecules are in the state $|00>$, $(1/4 - \alpha)$ in $|01>$, $(1/4 - \alpha)$ in $|10>$ and $(1/4 - \alpha)$ in $|11>$. The system is an ensemble of molecules, but behaves like a pure state.
Everything seems right, we use more and more qubits, the molecule grows bigger and bigger. A problem does appear. There has not been found a molecule of more than 10 spins yet, that could be addressed with different frequencies. That is why computers of this type are not possible when we want to use more than 10 qubits.

### 3.2. Ion traps

Chirac and Zoller[3] have proposed this scheme, which is scalable to an arbitrary number of qubits. Each qubit is comprised of a pair of the ions' internal states. Ions are trappe in a harmonic potential, that has to be narrow, so that the ions do not move a lot.

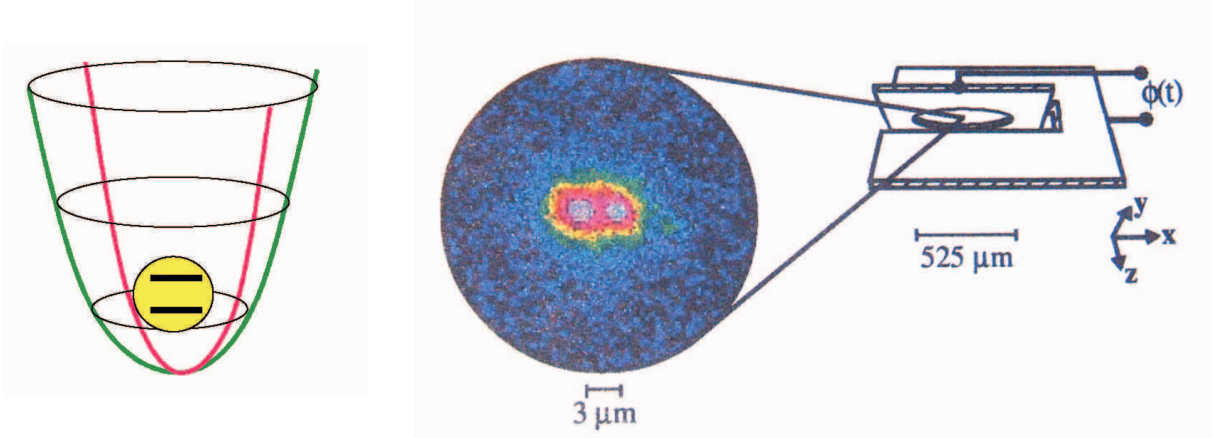



Figure 2: Ions are trapped in the paraboloidal potential[10]. They are a few $\mu m$ apart[4], so that we can adress a single one with a laser. Here two ions are trapped.

Applying a potential $\Phi$, the ions align themselves in the plane of the ring electrode. The ring electrode is eliptical, so that the ions are aligned on a line.

Ions, usually $(^6Be^+)$, are confined in a coaxial resonator based radio frequency (rf) trap, named Paul trap[5].

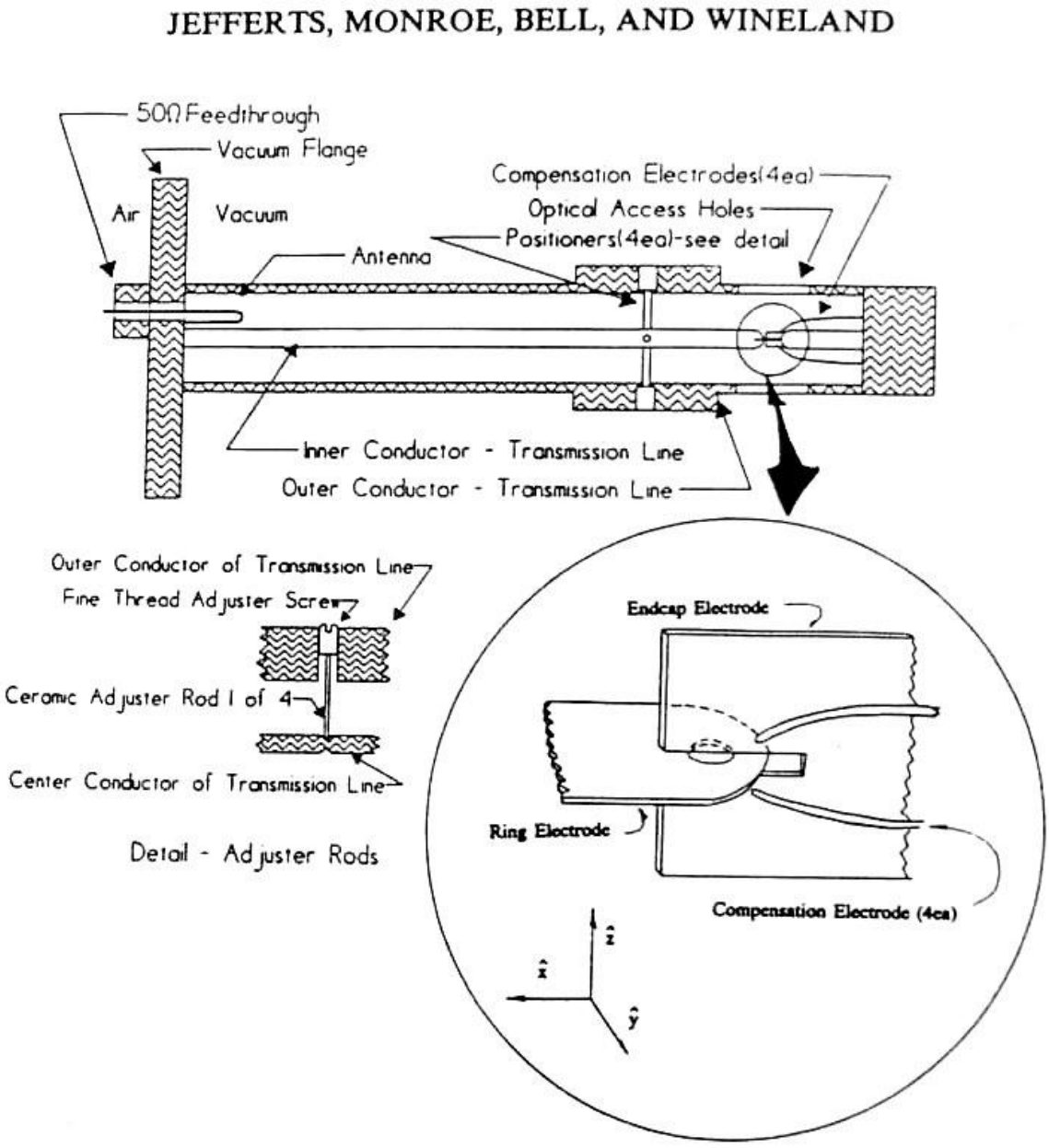



Figure 2[4]: Ions are trapped in the ellipsoidal part of the ring electrode. This kind of trap is capable of strong confinement. This means, that the ions cannot move(oscillate) more than $\lambda/2\pi$, where $\lambda$ is the wavelength of the exciting radiation($300nm$).

## 4. Gates and networks

Qubits are useless if there is nothing to do with them. That is why we seek for operators which will somehow calculate with them.
A quantum logic gate is a device that performs a fixed unitary operation on selected qubits in a fixed period of time.
A quantum network is a device consisting of quantum logic gates whose computational steps are synchronized in time.

### 4.1. Unitary operators

Why does a gate have to be unitary? If we take an arbitrary gate U, which is in general time dependent and use it on the eigenfunction, we get a propagated function, which has to satisfy the Schroedinger equation

$$H(U|\psi>) = i\hbar\frac{\partial}{\partial t}(U|\psi>).$$

So we obtain an operator equation $H\ U = i\hbar\frac{\partial}{\partial t}U$ and the solution

$$U(t) = e^{\ -i/\hbar\ Ht}.$$

If we express the exponential function into series, we see that $U^\dagger = e^{i/\hbar\ Ht}$ And so prove that $U$ is unitary operator if $H$ is hermitian ($H = H^\dagger$):

$$U^\dagger U = e^{\ -i/\hbar\ Ht}\ e^{i/\hbar\ Ht} = 1$$

### 4.2. Acting on qubits

In spin systems we act on qubits using radiofrequency pulses. We actually do NMR. Depending on the length (strength) of a pulse we apply $2\pi; \pi; \pi/2..$ pulses, which make the spin precede around the external static magnetic field.
Ions in the trap change states when we apply an electromagnetic pulse. As they are enough apart ($\mu m$), we can focus a laser beam on the ion we want.

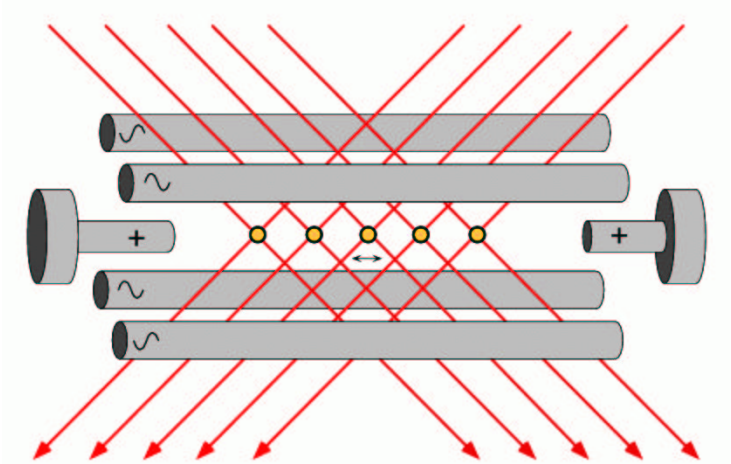



Figure 3[6]: A laser beam is separated and optically lead to a specific ion. There are two partial beams per ion. These are needed as will be seen futher in the seminar.

### 4.3. Hadamard gate

For production of superposition of qubits from a pure state we use Hadamard gate.

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$|x> \text{—}\boxed{\text{H}}\text{—} \frac{1}{\sqrt{2}}(-1)^x|x> + |1-x>$$

This is how we construct previously described register of all qubits from a pure state $|0>$

$|0> \text{—}\boxed{\text{H}}\text{—} |1> + |0>$

Notice the effect of using Hadamard twice:

$|0> \text{—}\boxed{\text{H}}\text{—} |1> + |0> \text{—}\boxed{\text{H}}\text{—} |0>$

$|1> \text{—}\boxed{\text{H}}\text{—} |1> - |0> \text{—}\boxed{\text{H}}\text{—} |1>$

Hadamard gate can be easily implemented in NMR by $\pi$ rotation around axes tilted at $\pi/4$ between the z and x axes. Such a rotation can be achieved directly using standard NMR techniques.

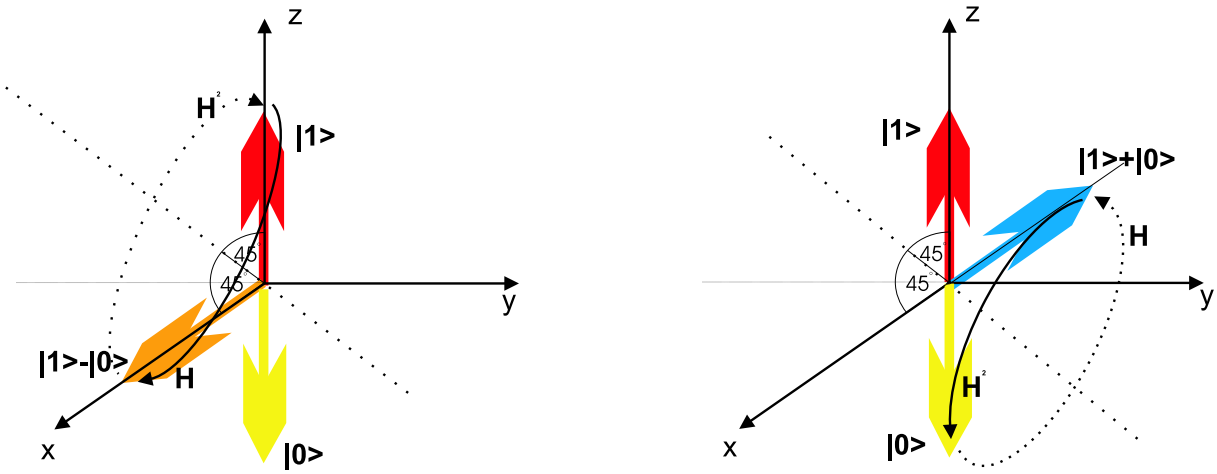



Figure 4: Hadamard gate acting on spins. First picture represents action on qubit $|1>$ and the second on $|0>$. Notice, that $H^2 = I$.

### 4.4. Phase gate

To construct a general state of a system we need phase gate:

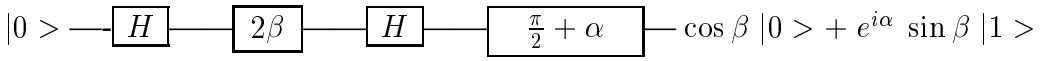
$$\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix}$$

$$|x> \text{—}\boxed{\alpha}\text{—} \frac{1}{\sqrt{2}}\, e^{ix\alpha}\, |x>$$

Construction of this gate is simplest of all, but is time consuming. We have to change the relative phase between qubits in the linear combination:

$$|0> e^{-itE_0/\hbar} + |1> e^{-itE_1/\hbar} = e^{-itE_0/\hbar}(|0> + |1> e^{-it(E_1-E_0)/\hbar})$$

Since the energies differ just a little, the frequency of oscillation $\omega = (E_1 - E_0)/\hbar$ is small (kHz; B(T)) and we have to wait long (in the aspect of computers) if we want the phase factor to come in by itself.
In combination with Hadamard gate we can now create any state we want




### 4.5. Boolean operators

When trying to create gates that are not bijective, we must remember that they should be unitary and so reversible. We must think of a gate that is a logical gate but can be reversed. A simple gate that does the trick is C-not or XOR gate. It takes two qubits, which can be entangled, so the computational basis is

$$\{|00>,\ |01>,\ |10>,\ |11>\}$$

This gate flips the second qubit if the first is $|1>$.

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$|x>\ |y> \xrightarrow{C_{not}} |x>\ |y \oplus x>$$

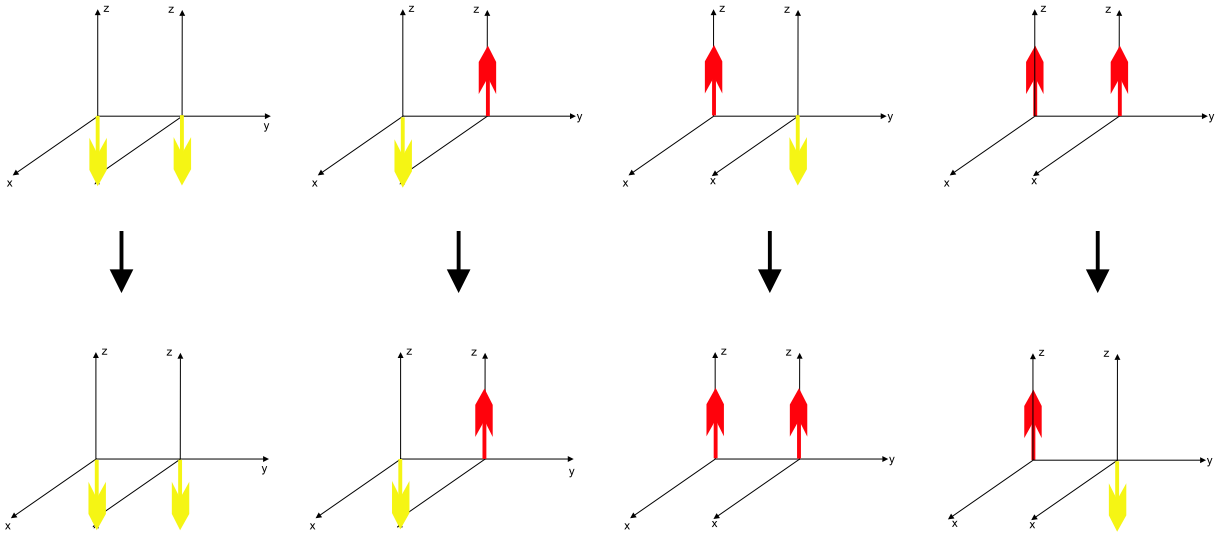


Figure 5: C-not gate flips the second qubit (spin) if the first is in logical $|1>$.

Here we see that we have to have an interaction between qubits. This is the only way to flip the second qubit on basis of the first. In bulk spin-resonance computers this interaction is spin coupling. In ion traps ions interact with phonons. This is the quantized centre-of-mass motion. Phonons (interactions) can be controlled with two different angles of the lasers. The photon gives its momentum to the data bus. When computer absorbs another photon coming from opposite direction, the center-of-mass stops.

# 5. Algorithms

Algorithm is a set of instructions which are applied to the system to obtain a solution. The algorithms are that make quantum computers useful, since some things can be done faster with quantum parallelism, while other not. The biggest difference is in NP problems. In informational sciences we have two kind of algorithms, P and NP. First, polynomial, are such that the time is scaled as a power of input. NP problems (not polynomial) are problems, for which a polynomial algorithm does not exist. With quantum computers some NP problems can be solved in polynomial time. This is the only advantage of quantum computers. The most well known problems of this kind are [1]:

- Factoring a number (Shor 1994)
  The best known factorization algorithm can factor a number $n$ in time $t$[1]:

  $$t \approx e^{1.9(\ln n)^{1/3}(\ln \ln n)^{2/3}}$$

  Shor's (quantum) algorithm, on the other hand, can factor numbers in time

  $$t \approx (\ln n)^2(\ln \ln n)(\ln \ln \ln n)$$

  This is a polynomial time algorithm. And what is this in practice:

| number of digits of the argument | Classical computer | Quantum computer |
|---|---|---|
| 130 | one week | one week |
| 400 | $10^9$ years | 1 year |

- Search algorithm (Grover)
  Imagine that we want to search for a person in a phonebook and all we have is his phone number. We have to check the number at each name, since the phonebook is sorted accordnig to names, until we come to the right one. This takes, on average, time $n/2$, if there are n entries in the phonebook. With quantum computers the process is faster. We prepeare a register containg the whole phonebook and send it through the delta function. This delta function returns one if the phone number is right. We then eliminate the state, on which the function has evaluated one and there it is. So we needed only $\log_2(n)$ qubits and evaluation of the function on these.

- Deutsch's algorithm

In order to accomplish an algorithm, measurement is necessary.

## 5.1. Deutsch's problem

This is the first problem, which was "invented" to prove the potential of quantum computing[1]. Our aim is to determine whether an unknown one bit function is constant or balanced. This means does the output of the function change with the input or not.

We have these four possibilities for the function f:

| f(0) | f(1) |
|---|---|
| 1 | 1 |
| 0 | 0 |
| 0 | 1 |
| 1 | 0 |

Classically we have to evaluate the function twice, on logical 1 and 0 and from the output determine the nature of unknown function f. This can be time consuming if the function does not evaluate quickly. The quantum algorithm determines the nature of the function with its single evaluation.
As quantum computers use reversible logic, it is not possible to implement $f$ directly. So we design a propagator $U_f$:

$$|x> |y> \xrightarrow{U_f} |x> |y \oplus f(x)>$$

Calculating this operator on superposition yields:

$$|x> (|0> -|1>) \xrightarrow{U_f} |x> (|0 \oplus f(x)> -|1 \oplus f(x)>) = \begin{cases} |x> (|0> -|1>);\ if\ f(x) = 0, \\ |x> (|1> -|0>;\ if\ f(x) = 1. \end{cases}$$

$$= (-1)^{f(x)} |x> (|0> -|1>)$$

This way the value of the function stays in the phase factor. With using $|x>$ as a superposition, we can do better:

$$|0> |1> \xrightarrow{H_1+H_2} (|0> +|1>)(|0> -|1>) \xrightarrow{U_f}$$

$$\xrightarrow{U_f} (-1)^{f(0)} |0> (|0> -|1>) + (-1)^{f(1)} |1> (|0> -|1>)) =$$

$$= ((-1)^{f(0)} |0> +(-1)^{f(1)} |1>)(|0> -|1>) =$$

$$= (-1)^{f(0)} (|0> +(-1)^{f(0) \oplus f(1)} |1>)(|0> -|1>) \xrightarrow{H_1+H_2}$$

$$\xrightarrow{H_1+H_2} (-1)^{f(0)} (|0> +|1> +(-1)^{f(0) \oplus f(1)} (|0> -|1>))|1>$$

If we look at the first qubit, we get exactly what we wanted:

$$|0> (1 + (-1)^{f(0) \oplus f(1)}) + |1> (1 - (-1)^{f(0) \oplus f(1)} =$$

$$= \begin{cases} |0>;\ if\ f(0) \oplus f(1) = 0, \\ |1>;\ if\ f(0) \oplus f(1) = 1. \end{cases}$$

What can be written as the easy detectable result.

$$|f(0) \oplus f(1)>$$

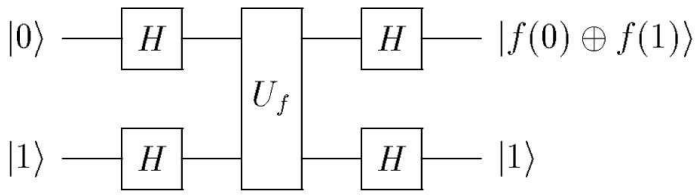



Figure 6: A quantum circuit for solving Deutsch problem

With knowing how to implement Hadamard gate it is now not difficult to build a network, solving the Deutsch's problem.

## 6. Errors

To keep a quantum computer functioning properly we need error correcting codes. There are many sources of error:

- Phase error
  The qubit may gain an arbitrary phase. For example

  $$|1> \longrightarrow -|1>$$

  Phase of the whole function is of no significance but a relative phase $|0> +|1> \longrightarrow |0> -|1>$ is catastrophic.

- Small errors
  can occur because of limited precision of the instrumentation as well as interactions between qubits. $|0> +|1> \longrightarrow (1-\delta)|0> +(1+\delta)|1>$

- Large errors - Bit flip error
  $\alpha|0> +\beta|1> \longrightarrow \alpha|1> +\beta|0>$ This error cannot occur on part of the superposition: $\alpha|0> +\beta|1> \alpha|1> +\beta|1>$

- Measurement case disturbance
  We need to measure the qubits to detect whether there are errors. But we can't measure the qubits without disturbing them.

- Cloning
  We cannot copy qubit registers.

### 6.1. Error correcting codes

For correcting we need some extra information and must encode a single qubit using three qubits $|abc>$. This way the probability for an error is bigger, but we know how to correct it.

$$|0> \longrightarrow |000>$$

$$|1> \longrightarrow |111>$$

- Correcting bit flip error:
  To do this we have to measure two qubits at once. This operation projects the state of the whole qubit back on itself. Let us have a look at an example

  $$|\psi> = \alpha|000> +\beta|111>$$
  $$|\psi'> = \alpha|001> +\beta|110>$$

  We have to measure $(b \oplus c, a \oplus c)$. The two bit number we get tells us which qubit has been flipped. In our example that is the third:

  $$(b \oplus c, a \oplus c) = (1,1)_B = 3_D$$

- Correcting small errors:
  Suppose we have a modified qubit:

  $$|\psi'' >= \alpha|000 > +\beta|111 > +\delta(\alpha|000 > +\beta|111 >)$$

  When measuring the same quantities as the previous time $(b \oplus c, a \oplus c)$, we get a projection of the qubit on itself. With probability $1 - |\delta|^2$ we obtain the measurement (0,0), which projects the damaged qubit onto the undamaged qubit $|\psi >$. The other possible measurement is (1,1), with probability $|\delta|^2$ which means that $|\psi >$ is projected onto $|\psi' >$, what we can repair.

- Correcting phase errors:
  For this more advanced technique, we need 9 qubits to represent a single one.

This algorithms were developed by Shor and do not disturb the qubit. From measurement we gain information and project the qubit on a set of discrete states. Using the measured information we know, onto which state the qubit has been projected on and can reconstruct it.

## 7. Conclusion

Since Feynman showed the possibilities of quantum computers the research has not stopped. Theory is well developed. With the three described gates we can do any qubit transformation we want. Some powerful algorithms have been developed. These are the only advantage of the quantum computers. They are however useless when we want to calculate everyday arithmetic. Just some special problems can be efficiently solved by a quantum system. That is the main reason, why they probably will not be in the stores in the next decade. There are of course other restrictions, which in contrary can and will be passed. These are implementation problems.

The biggest computer build was a 7 qubit spin system. Now it is known that there is no future in the spin systems, so attention of the researchers is directed towards ion traps and quantum dots.

## 8. References


1. Basics of Quantum Computation; Nichols A. Romeo; Massachusetts 02139; May 1998
2. Bulk Spin-Resonance Quantum Computation; Gershenffeld, Chuang; Science vol.275,p.350; 1997
3. Quantum Computation with cold trapped ions; Cirac, Zoller; Physical rev. let., vol 74, p. 4091;1995
4. Cooling the Collective motion of trapped ions to initialize a quantum register; King, Wood,..; Physical rev. let., vol 81, p. 1525; 1998
5. Coaxial-resonator-driven rf (Paul) trap for strong confinement; Jefferts,Monroe,..;Physical review A, vol. 51, p. 3112; 1995
6. Quantum computing; Andrew Steane; University of Oxford; arXiv:quant-ph/9708022 v2 24 Sept 1997
7. Implementation of Quantum Algorithm to solve Deutsch's problem on a nuclear magnetic resonance quantum computer; Jones, Mosca; arXiv:quant-ph/9801027 v2 30 Apr 1998
8. Basic concepts in quantum computation; Ekert, Hayden, Inamory; University of Oxford; 16.1.2000
9. Lecture Notes for Physics 229: Quantum Information and Computation; John Preskill; California Institute of Techology; September 1998
10. Quantum Information Processing; Jaewan Kim; Dept of Physics, KAIST; 1998