

УДК 512.6, 517.9, 519.6

**STRUCTURE DESIGN TOOLKIT OF QUANTUM ALGORITHMS. PT 1.****Reshetnikov Andrey<sup>1</sup>, Tyatyushkina Olga<sup>2</sup>, Ulyanov Sergey<sup>3</sup>, Degli Antonio Giovanni<sup>4</sup>**

<sup>1</sup>PhD in informatics, associate professor;  
Dubna State University,  
Institute of system analysis and management;  
141980, Dubna, Moscow reg., Universitetskaya str., 19;  
e-mail: agreshetnikov@gmail.com.

<sup>2</sup>PhD, associate professor;  
Dubna State University,  
Institute of system analysis and management;  
141980, Dubna, Moscow reg., Universitetskaya str., 19;  
e-mail: tyatushkina@mail.ru.

<sup>3</sup>Doctor of Science in Physics and Mathematics, professor;  
Dubna State University,  
Institute of system analysis and management;  
141980, Dubna, Moscow reg., Universitetskaya str., 19;  
e-mail: ulyanovsv@mail.ru.

<sup>4</sup>PhD, professor;  
Polo Didattico e di Ricerca di Crema;  
Via Bramante, 65-26013, Crema (CR), Italy;  
e-mail: gda@dsi.unimi.it.

*The bases of quantum computation are three operators on quantum coherent states as following: superposition, entanglement and interference. The coherent states are the solutions of corresponding Schrodinger equations described the evolution states with minimum of uncertainty (in Heisenberg sentence it is quantum states with maximum classical properties), the Hadamard transform creates the superpositon on classical states, and quantum operators as CNOT create robustness entangled states, interference is created by quantum fast Fourier transform. Structures of these operators are described.*

**Keywords:** quantum computing, universal quantum gates, quantum operators, matrix transformation

**ИНСТРУМЕНТАРИЙ ПРОЕКТИРОВАНИЯ КВАНТОВЫХ АЛГОРИТМОВ. Ч. 1.****Решетников Андрей Геннадьевич<sup>1</sup>, Тятюшкина Ольга Юрьевна<sup>2</sup>, Ульянов Сергей Викторович<sup>3</sup>, Джиованни дели Антонио<sup>4</sup>**

<sup>1</sup>Кандидат технических наук, доцент;  
ГБОУ ВО МО «Университет «Дубна»,  
Институт системного анализа и управления;  
141980, Московская обл., г. Дубна, ул. Университетская, 19;  
e-mail: agreshetnikov@gmail.com.

<sup>2</sup>Кандидат технических наук, доцент;  
ГБОУ ВО МО «Университет «Дубна»,  
Институт системного анализа и управления;  
141980, Московская обл., г. Дубна, ул. Университетская, 19;  
e-mail: tyatyushkina@mail.ru.

<sup>3</sup>Доктор физико-математических наук, профессор;  
ГБОУ ВО МО «Университет «Дубна»,  
Институт системного анализа и управления;  
141980, Московская обл., г. Дубна, ул. Университетская, 19;

e-mail: ulyanovsv@mail.ru.

<sup>4</sup>Доктор наук, профессор;  
Поло дидактико, Крема, факультет информационных технологий;  
Via Браманте, 65-26013, Крема, Италия;  
e-mail: gda@dsi.unimi.it.

Основой квантовых вычислений являются три оператора квантовых когерентных состояний: суперпозиция, запутывание и интерференция. Когерентные состояния являются решениями соответствующих уравнений Шредингера, описывающих эволюционные состояния с минимумом неопределенности (в предложении Гейзенберга это квантовые состояния с максимальными классическими свойствами), преобразование Адамара создаёт суперпозицию на классических состояниях, а квантовые операторы как CNOT передают надёжность этим состояниям, интерференция создаётся квантовым быстрым преобразованием Фурье. Описаны структуры этих операторов.

Ключевые слова: квантовые вычисления, универсальные квантовые вентили, квантовые операторы, матрицы преобразования

## Introduction

The efficient implementations of a number of operations for quantum computation include controlled phase adjustment of the amplitudes in a superposition, permutation, approximations of transformations and generalizations of the phase adjustments to block matrix transformations. These operations generalize those used in quantum algorithms. Moreover, the Hadamard transform (H), the phase (Ph), and the CNOT generate the Clifford group. A quantum computation using only operations from this group can be simulated efficiently on a classical computer. The addition of just the Toffoli gate to this group is sufficient to make this group universal. We demonstrate application of this approach to the Benchmarks as Deutsch, Deutsch – Josza, Simon, Shor, and Grover algorithms [1-18].

## Physical Principles for Quantum Computation

Here, are the five essential criteria, which we perceive for the physical implementation of quantum computing:

- |                                       |
|---------------------------------------|
| 1.Hilbert space control               |
| 2.State preparation                   |
| 3.Low decoherence                     |
| 4.Controlled unitary transformations  |
| 5.State-specific quantum measurements |

The *first criterion* means that the available states must be precisely enumerated, and it must be known how to confine the state vector of the quantum system to this part of Hilbert space. In addition, Hilbert space should be extendable, preferable with a simple tensor-product structure, by adding particles to the system. For example,  $n$  spin-1/2 particles have a simple spin Hilbert space of  $2^n$  dimension.

The *second criterion*. Within this Hilbert space, it must be possible to set the state vector initially to a simple fiducially starting state. A simple example of this, in the spin system, would be to set all the spins in the spin-down state. Frequently this only requires being able to bring the system to sufficiently low temperature that it is in its ground state. This is more difficult in some examples than in others.

The *third criterion*. The coupling to the environment (i.e., to all the rest of the Hilbert space of the world) should be sufficiently weak that quantum interference in the computational Hilbert space is not spoiled. Given our current understanding of error correction and fault-tolerant quantum computation, and given fairly benign assumptions about the nature of the decoherence (e.g., that it acts independently on each quantum bit) reliable computation is possible if the decoherence time exceeds the switching time by  $10^6$ . More clever fault-tolerant techniques may well in making this rather demanding threshold number more relaxed in the future.

The *fourth principle*. This is the fairly obvious central requirement of quantum computing: it must be possible to subject the computational system to a controlled sequence of precisely defined unitary transformations. The precision requirements are closely related to the decoherence threshold; imprecision of unitary operations is a form of a decoherence. For convenience of programming, it is very desirable that the elementary unitary transformations be implementable as discrete one- and two-qubit operations.

The *fifth principle*. The readout of a quantum computation, which would consist of some ordinary bit string, is to be the result of a sequence of quantum measurements performed on the computational quantum system. It is very desirable (although not necessary) that these measurements be the textbook projection subsystems of individual quanta. It is essential that these measurements can be made on specific, identified subsystems of the computational state; in the simplest case, this means that it should be possible to do a projection measurement on each qubit individually. If many identical copies of the quantum computer are available, then weaker, ensemble measurements, rather than projection measurements are adequate. It is still necessary that these ensemble measurements be subsystem-specific, though.

## Tools for Quantum Computation and Quantum Networks.

To exploit the power of quantum computation and create algorithms of new complexity classes, we need to use building blocks that do not have classical analogs but instead take advantage of quantum parallelism through modifying and mixing amplitudes in superposition. *Two* sorts of tools have been used effectively in design of quantum algorithms that have been developed so far.

*First*, transformations that can mixing amplitudes, such as the Walsh-Hadamard and Fourier transforms.

*Second*, selective adjustment of the phases of certain states that, when combined with a mixing transform, promote amplitude cancellation or amplification. Such phase adjustments form the basis of search algorithms for NP problems.

Here, we discuss efficient implementations of mixing transformations and of relative phase changes that combine amplitude from only a small number of states. The choice of phase and which states to mix depends on a classically efficient computable function  $f$  that in general will remain necessarily abstract. We discuss implementations of approximations of transformations, of phase changes, of permutations, and generalizations of the phase change techniques to block matrix transformations. For each of these transformations, we describe the resources in terms of time, number of calls to  $f$ , and number of additional qubits needed for the implementation.

We first concern with transformations that change the relative phases of components that make up superposition. Such transformations correspond to acting on the state with a diagonal matrix  $D$ . Conversely, because quantum operations are unitary, any operation described by a diagonal matrix will consist of such phase adjustments. Since a global phase change has no physical meaning, so the matrix is only well defined up to multiplication by a constant.

In implementing quantum algorithms it will be useful to have a variety of techniques depending on whether number of bits or coherence time (number of operations) is the main limiting factor. For implementing relative phase changes to components of an  $n$  – qubit quantum state several methods can be represented as  $2^n \times 2^n$  diagonal matrices  $D$ .

If  $D$  is decomposable, in that can be written as a tensor product of single qubit rotations, it can be implemented trivially in  $O(n)$  steps without any additional qubits or function calls.

We describe a sufficient and necessary condition for the decomposability of the matrix  $D$ .

In general form suppose that a unitary matrix  $U$  is an  $N \times N$  unitary matrix, where  $N$  is an even number. Then one can always express  $U$  in the form

$$U = \begin{bmatrix} L_0 & 0 \\ 0 & L_1 \end{bmatrix} D \begin{bmatrix} R_0 & 0 \\ 0 & R_1 \end{bmatrix}, \quad (1)$$

where the left and right side matrices  $L_0, L_1, R_0, R_1$  are  $\frac{N}{2} \times \frac{N}{2}$  unitary matrices and

$$D = \begin{bmatrix} D_{00} & D_{01} \\ D_{10} & D_{11} \end{bmatrix}, D_{00} = D_{11} = \text{diag} \left( C_1, C_2, \dots, C_{\frac{N}{2}} \right), D_{01} = -D_{10} = \text{diag} \left( S_1, S_2, \dots, S_{\frac{N}{2}} \right).$$

For all  $i \in \left\{ 1, 2, \dots, \frac{N}{2} \right\}$ ,  $C_i = \cos \theta_i$  and  $S_i = \sin \theta_i$  for some angle  $\theta_i$ .

Given any approximation CSD of  $U$ , it is possible to find another CSD of  $U$  for which the angles  $\theta_i$  are in non-decreasing order and they are contained in the interval  $[0, 90^0]$ . If one partitions  $U$  into four blocks  $U_{ij}$  of the size  $\frac{N}{2} \times \frac{N}{2}$ , then we can obtain  $U_{ij} = L_i D_{ij} R_j$ , for  $i, j \in \{0, 1\}$ . Thus,  $D_{ij}$  gives the singular values of  $U_{ij}$ .

*Example.* The operator  $D$  (diffusion – inversion about average) in Grover algorithm is

$$D = \frac{1}{\sqrt{2^n}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{\otimes n} \cdot \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}^{\otimes n} \cdot \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{\otimes n}$$

and have the form (1).

The availability of single quantum systems has fed the interest in quantum networks: A quantum computation is a network of individual quantum systems, where any two of the nodes can interact with each other. Most quantum private communication scheme is networks of two or three nodes. In addition to these information processing and communication related applications, networks avoided crossings in multilevel systems and have been considered in order to study higher dimensional quantum interference effects. Such networks can be experimentally realized by various active (energy-consuming) or passive (energy-preserving) components. It is found that transitions do take place and we are able to give them a clear physical interpretation. For active systems the transition is related to quantum and classical, for passive systems to adiabatic and diabatic behavior of the network. The transition phenomenon is clearly reflected in observable quantities and shows a relation to symmetry, which can be of general significance for quantum networks.

In general form the quantum network is represented in Fig. 1.

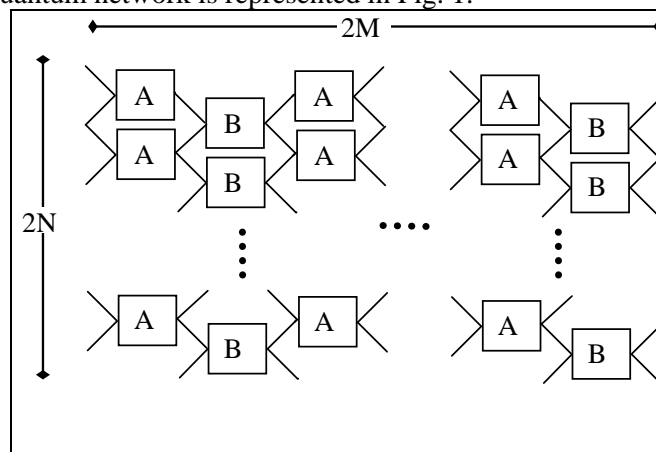


Figure 1: A quantum network where the nodes are connected to their neighbor. The boxes A and B denote the transitions performed at the nodes)

And can be described by a  $2^n \times 2^n$  plane rotation matrix  $P$ , which is

$$P = \begin{pmatrix} \boxed{A} & 0 & 0 & \dots & 0 \\ & & 0 & 0 & \dots \\ 0 & 0 & \boxed{A} & & \\ & & & & \\ 0 & 0 & & & \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & 0 & 0 & \dots & \boxed{A} \end{pmatrix} \begin{pmatrix} B_{22} & 0 & 0 & \dots & B_{21} \\ 0 & & & & \\ & \boxed{B} & & & \\ 0 & & & & \\ \vdots & & & \boxed{B} & \\ & & & & \ddots \\ B_{12} & & & & B_{11} \end{pmatrix}$$

$$\boxed{A} = \begin{pmatrix} \cosh \theta & i \sinh \theta \\ -i \sinh \theta & \cosh \theta \end{pmatrix}, \quad \boxed{B} = \begin{pmatrix} \cosh \phi & i \sinh \phi \\ -i \sinh \phi & \cosh \phi \end{pmatrix}, \quad \coth \theta = \cosh \phi$$

The form of the matrix  $P$  suggests immediately a quantum-network analog. The matrices  $A$  and  $B$  can be interpreted to describe the evolution of a two-state or two-mode system. The matrix  $P$  is then the evolution operator over period in the network of Fig.1. The input of the network are mixed pair-wise according to the transition  $B$ , and then the pairs are let to interact with the neighboring ones by applying the shifted set of operations  $A$ . By repeating this  $l$  times, a  $2n \times 2l$  – dimensional network can be constructed. Since  $P$  contains all the physical information of the quantum computation, i.e., the phase transitions, we may expect analogous phenomena in the quantum network described by  $P$ .

*Remark.* The physical realization of the quantum network  $P$  can be divided into groups. When the angle  $\phi$  is real,  $A$  and  $B$  are  $SU(1,1)$ - type matrices describing energy – consuming (active) operations. Imaginary  $\phi$  leads to  $SU(2)$  matrices, which correspond to energy – preserving (passive) manipulations of the two modes or two states. Note that  $A \rightarrow I_2$  when the angle  $\phi \rightarrow \infty$ , and  $B \rightarrow I_2$  when  $\phi \rightarrow 0$  ( $I_2$  is the two-dimensional unit matrix). That is, in both of these limits the network decomposes into sets of non-interacting modes. Thus  $P$  describes a quantum network where only nearest neighbors interact, and where a single parameter  $\phi$  determines the relative importance of the interactions, i.e., the network character of the system.

The transitions in the network are determined by the eigenvalues of  $P$ . The only problem in diagonalizing  $P$  is the relative shift between the set of  $A$  and  $B$ . This can be solved by the discrete FFT (see below) as  $(F_n)_{kl} = \exp\{i2\pi kl/n\}\sqrt{n} \equiv \omega^{kl} / \sqrt{n}$ , because the FFT of the shift matrix  $(S_n)_{kl} = \delta_{k+1,l} + \delta_{k,n-1}\delta_{l0}$  is diagonal:  $F_n^* S_n F_n = D_\omega$ , where  $(D_\omega)_{kl} = \omega^k \delta_{kl}$ .

The whole network matrix  $P$  thus decomposes into

$$P = (F_n \otimes I_2) \begin{pmatrix} \boxed{K_0} & 0_2 & \dots \\ 0_2 & \ddots & \\ \vdots & & \boxed{K_{n-1}} \end{pmatrix} (F_n^* \otimes I_2)$$

where  $(K_n)_{11} = (K_n)_{22}^* = C(\theta)C(\phi) + S(\theta)S(\phi)\omega^{-n}$ ,  $C \equiv \cosh$  and  $S \equiv i \sinh$  and  $(K_n)_{12} = (K_n)_{21}^* = -C(\phi)S(\theta) + C(\theta)S(\phi)\omega^{-n}$ .

*Example.* The usual Mach – Zehnder interferometer (see below) affects the input states by unitary transformation  $U_{M-Z}$ , which can be formally written as

$$U_{M-Z} = (F_2 \otimes I_1) \begin{pmatrix} K_0 & 0 \\ 0 & K_1 \end{pmatrix} (F_2^* \otimes I_1),$$

where  $K_n = \exp\{i\phi n\}$  is determined by a chosen phase  $\phi$ . Thus this network-type we consider acts like an  $n$  – dimensional interferometer where, instead of one-mode phase shift, two-mode rotations are performed in between the  $n$  – dimensional mixers  $F_n$  and  $F_n^*$ .

## General Structure of Quantum Gates

For this case we use coherent initial classical states and three quantum operators: *superposition*, *entanglement*, and *interference*. These operators are generators of Clifford group, together with Toffoli gate are universal and can be efficiently realized on classical computer.

*Model Description of Three Quantum Operators.* Quantum Algorithms are global random algorithms based on the quantum mechanics principles, laws, and quantum effects. In the quantum search, each design variable is represented by a finite linear superposition of classical initial states, with a sequence of elementary unitary steps manipulate the initial quantum state  $|i\rangle$  (for the input) such that a measurement of the final state of the system yields the correct output. It begins with elementary classical preprocessing, and then it applies the following quantum experiment: starting in an initial superposition of all possible states, it computes a classical function, applies a Quantum Fast Fourier Transform (*QFFT*), and finally performs a measurement. Depending on the outcome it may carry out one more similar quantum experiments, or complete the computation with some classical post-processing. Usually, three principle operators, i.e. *linear superposition (coherent states)*, *entanglement*, and *interference*, are used in the quantum search algorithm and briefly described below.

*Linear Superposition.* Linear superposition is closely related to the familiar mathematical principle of linear combinations of vectors. Quantum systems are described by a wave function  $\psi$  that exists in a Hilbert space.

The Hilbert space has a set of states,  $|\phi_i\rangle$ , that form a basis, and the system is described by a quantum state,  $|\psi\rangle = \sum_i c_i |\phi_i\rangle$ . The vector  $|\psi\rangle$  is said to be in a linear superposition of the basis states  $|\phi_i\rangle$ , and in the general case, the coefficients  $c_i$  may be complex. Use is made here of the Dirac bracket notation, where the ket  $|\cdot\rangle$  is analogous to a column vector, and the bra  $\langle\cdot|$  is analogous to the complex conjugate transpose of the ket. In quantum mechanics the Hilbert space and its basis have a physical interpretation, and this leads directly to perhaps the most counterintuitive aspect of the theory. The counter intuition is this (at the microscopic level), the state of the system is described by the wave function  $\psi$ , that is, as a linear superposition of all basis states (i.e. in some sense the system is in all basis states at ones). However, at the macroscopic or classical level the system can be in only a single basis state. For example, at the quantum level an electron can be in a superposition of many different energies; however, in the classical realm this obviously cannot be.

*Remark: Coherence and Decoherence.* *Coherence* and *decoherence* are closely related to the idea of linear superposition. A quantum system is said to be *coherent* if it is in a linear superposition of its basis states. A result of quantum mechanics is that if a system is in a linear superposition of states interacts in any way with its environment the superposition is destroyed. This loss of coherence is called *decoherence* and is governed by the wave function  $\psi$ . The coefficients  $c_i$  are called *probability amplitudes*, and  $|c_i|^2$  gives the probability of  $|\psi\rangle$  collapsing into state  $|\phi_i\rangle$  if it is decoherence. Note that the wave function  $\psi$  describes a real physical system that must collapse to exactly one basis state. Therefore, the probabilities governed by the amplitudes  $c_i$  must sum to unity. This necessary constraint is expressed as the unitary condition  $\sum_i |c_i|^2 = 1$ . In the Dirac notation, the probability that a quantum state  $|\psi\rangle$  will collapse into an eigenstate  $|\phi_i\rangle$  is written  $|\langle\phi_i|\psi\rangle|^2$  and is analogous to the dot product (projection) of two vectors.

Consider, for example, a discrete physical variable, called spin. The simplest spin system is a two-state system, called a spin-1/2 system, whose basis states are usually represented as  $|\uparrow\rangle$  (spin up) and  $|\downarrow\rangle$  (spin down). In this simple system the wave function  $\psi$  is a distribution over two values (up and down) and a coherent state  $|\psi\rangle$  is a linear superposition of  $|\uparrow\rangle$  and  $|\downarrow\rangle$ .

*Example.* One such state might be  $|\psi\rangle = \frac{2}{\sqrt{5}}|\uparrow\rangle + \frac{1}{\sqrt{5}}|\downarrow\rangle$ . As long as a system maintains its quantum coherence it cannot be said to be either spin up or spin down. It is in some sense both at ones. Classically, of course, it must be one or the other, and when this system decoherence the results is, for example, the  $|\uparrow\rangle$  state with probability  $|\langle\uparrow|\psi\rangle|^2 = \left(\frac{2}{\sqrt{5}}\right)^2 = 0.8$ . A simple two-state quantum system, such that as the spin-1/2 system just introduced, is used as the basis unit of quantum computation. Such a system is referred to as a quantum bit or *qubit*, renaming the two states  $|0\rangle$  and  $|1\rangle$  it is easy to see why this is so.

*Remark: The Operators.* Operators on a Hilbert space describe how one wave function is changed into another. Here they will be denoted by a capital letter with a hat, such as  $\hat{A}$ , and they may be represented as matrices acting on vectors. Using operators, an eigenvalue equation can be written  $\hat{A}|\phi_i\rangle = a_i|\phi_i\rangle$ , where  $a_i$  is the eigenvalue. The solutions  $|\phi_i\rangle$  to such an equation are called eigenstates and can be used to construct the basis of a Hilbert space. In the quantum formalism, all properties are represented as operators whose eigenstates are the basis for the Hilbert space associated with that property and whose eigenvalues are the quantum allowed values for that property. It is important to note that operators in quantum mechanics must be linear operators and further that they must be unitary so that  $\hat{A}^*\hat{A} = \hat{A}\hat{A}^* = \hat{I}$ , where  $\hat{I}$  is the identity operator, and  $\hat{A}^*$  is the complex conjugate transpose, or joint, of  $\hat{A}$ .

*Interference.* Interference is a familiar wave phenomenon. Wave peaks that are in phase interfere *constructively* (magnify each other's amplitude) while those that are out of phase interfere *destructively* (decrease or eliminate each other's amplitude). This is a phenomenon common to all kinds of wave mechanics from water waves to optics. The well known double slit experiment demonstrates empirically that at the quantum level interference also applies to the probability waves of quantum mechanics.

*Example.* As a simple example, suppose that the wave function described in (3) is represented in vector form as  $|\psi\rangle = \frac{1}{\sqrt{5}}\begin{pmatrix} 2 \\ 1 \end{pmatrix}$  and suppose that it is operated upon by an operator  $\hat{H}$  (Hadamard transform) described by the following matrix,  $\hat{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ . The result is  $\hat{H}|\psi\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\frac{1}{\sqrt{5}}\begin{pmatrix} 2 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{10}}\begin{pmatrix} 3 \\ 1 \end{pmatrix}$  and therefore now  $|\psi\rangle = \frac{3}{\sqrt{10}}|\uparrow\rangle + \frac{1}{\sqrt{10}}|\downarrow\rangle$ . Notice that the amplitude of the  $|\uparrow\rangle$  state has *increased* while the amplitude of the  $|\downarrow\rangle$  state has *decreased*.

This is due to the wave function interfering with itself through the action of the operator – *the different parts of wave function interfere constructively or destructively according to their phases just like any other kind of wave.*

*Entanglement.* Entanglement is the potential for quantum states to exhibit correlation that cannot be accounted for classically. From a computational standpoint, entanglement seems intuitive enough (it is simply the fact that correlation can exist between different qubits) for example if one qubit is in the  $|1\rangle$  state, another will be in the  $|1\rangle$  state. What makes it so powerful is the fact that since quantum states exist as superpositions, these correlation somehow exist in superposition as well. When the superposition is destroyed, the proper correlation is somehow communicated between the qubits, and it is this 'communication' that is the crux of entanglement. Mathematically, entanglement may be described using the

density matrix formalism. The density matrix  $\rho_\psi$  of a quantum state  $|\psi\rangle$  is defined as  $\rho_\psi = |\psi\rangle\langle\psi|$ . For example, the quantum state  $|\xi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|01\rangle$  appears in a vector form as

$$|\xi\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

and it may also be represented as density matrix

$$\rho_\psi = |\psi\rangle\langle\psi| = \frac{1}{2} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

while the state  $|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ , is represented as

$$\rho_\psi = |\psi\rangle\langle\psi| = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix},$$

and the state  $|\zeta\rangle = \frac{1}{\sqrt{3}}|00\rangle + \frac{1}{\sqrt{3}}|01\rangle + \frac{1}{\sqrt{3}}|11\rangle$  as

$$\rho_\zeta = |\zeta\rangle\langle\zeta| = \frac{1}{3} \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix},$$

where the matrices and vectors are indexed by the state labels 00,...11. Now, notice that  $\rho_\zeta$  can be factorized as  $\rho_\zeta = \frac{1}{\sqrt{2}} \left( \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \right)$ , where  $\otimes$  is a normal tensor product. On the other hand,  $\rho_\psi$  cannot be factorized. States that cannot be factorized are said to be entangled, while those that can be factorized are not. Notice that  $\rho_\zeta$  can be partially factorized two different ways, one of which is

$$\rho_\zeta = \frac{1}{\sqrt{3}} \left( \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \right)$$

(the other involves above equation and a different remainder); however, in both cases the factorization is not complete. Therefore,  $\rho_\zeta$  is also entangled, but not to the same degree as  $\rho_\psi$  (because  $\rho_\zeta$  can be partially factorized but  $\rho_\psi$  cannot). Thus there are different degrees of entanglement.

*Remark.* It is interesting to note from a computational standpoint that quantum states that are superpositions of *only* basis states that are maximally far apart in terms of Hamming distance are those states with the greatest entanglement. For example,  $\rho_\psi$  is a superposition of only the states  $|00\rangle$  and  $|11\rangle$ , which



have a maximum Hamming spread, and therefore  $\rho_{\psi}$  is maximally entangled. Finally, it should be mentioned that while interference is a quantum property that has a classical cousin, entanglement is a completely quantum phenomenon for which is no classical analog.

*Quantum Networks.* Quantum networks are one of the several models of quantum computation. Others include quantum Turing machines, and quantum cellular automata. In the quantum networks model, each unitary operator is modeled as a quantum logic gate that affects one, two or more qubits. Schematically this is represented as a set of quantum ‘wires’ entering and leaving quantum gates, reminiscent of classical logic networks.

For example, Fig. 2 shows a network that operates on three qubits, which are represented as lines.

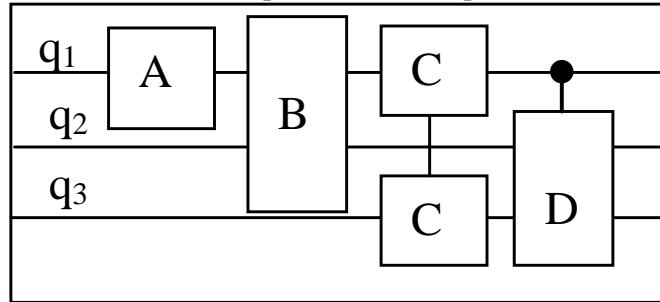


Figure 2. Quantum networks with three qubits.

By convention the logic flows from left to right. The gates are represented as boxes and labeled with the operators that they represent. A dot on a quantum “wire” represents a conditional upon that qubit.

Therefore, in the quantum network shown in Fig. 2 an operator  $\hat{A}$  represents a single qubit quantum gate,  $\hat{B}$  and  $\hat{C}$  represent 2-qubit quantum gates, and  $\hat{D}$  represents a conditional 3-qubit gate. Suppose that  $\hat{A}$  is an operator that flips the state of qubit,  $\hat{B}$  is an operator that exchanges the states of two qubits if they are equal, and  $\hat{D}$  is an operator that exchanges the states of two qubits if a third qubit in the  $|1\rangle$  state. When three qubits “enter” the quantum logic network, the one labeled  $q_1$  first has its state flipped; then  $q_1$  and  $q_2$  exchange states,  $q_1$  and  $q_3$  have their states flipped if they are equal, and finally  $q_2$  and  $q_3$  exchange states if  $q_1$  is in the state  $|1\rangle$ . Of course, if the qubits “entering” the logic array did not exist in a superposition states, this would be no different than a classical logic sequence. However, the qubits do exist in a superposition of states; therefore, these gates or operations are applied to all the states in the superposition simultaneously, resulting in what has been called *quantum parallelism*. Recall that the quantum logic gate arrays are simply a schematic way to represent the time evolution of a quantum system. They are not meant to imply that quantum computation can be physically realized in a manner similar to classical logic networks. Alternatively, the network could be represented as a product of quantum operators. Since operators are applied right to left, the network of Fig. 2 would be represented as the operator product  $\hat{D}\hat{C}\hat{B}\hat{A}$  in what follows, both the network and the product of operator representations will be used.

*Interference and Quantum Parallelism.* Because of the entanglement or quantum correlation between the  $n$  quantum particles, the state of the system cannot be specified simply describing the state of each of the  $n$  particles. Instead, the state of  $n$  quantum bits is a complicated superposition of all  $2^n$  basis states, so  $2^n$  complex coefficients are needed in order to describe it. This exponentially of a Hilbert space is a crucial ingredient in quantum computation. To gain more understanding of advantages of the exponentiality of the space, consider the following superposition on  $n$  quantum bits,  $\frac{1}{\sqrt{2^n}} \sum_{i_1, i_2, \dots, i_n=0}^1 |i_1, i_2, \dots, i_n\rangle$ . This is a uniform

superposition of all possible basis states of  $n$  qubits. If we now apply the unitary operation  $U$  which computes  $f$  for this state, we will get, simply from linearity of quantum mechanics:

$$\frac{1}{\sqrt{2^n}} \sum_{i_1, i_2, \dots, i_n=0}^1 |i_1, i_2, \dots, i_n\rangle \rightarrow \frac{1}{\sqrt{2^n}} \sum_{i_1, i_2, \dots, i_n=0}^1 |f(i_1, i_2, \dots, i_n)\rangle$$

Applying the unitary operator  $U_f$  once, computes  $f$

simultaneously on all the  $2^n$  possible inputs  $i$ , which is an enormous power of parallelism. It is tempting to think that exponential parallelism implies exponential computational power, but this is not the case. In fact, classical computations can be viewed as having exponential parallelism as well. The problem lies in the question of how to *extract the exponential information out of the system*.

In quantum computation, in order to extract quantum information one has to *observe* the system. The measurement process causes the famous *collapse of the wave function*. In nutshell, this means that after the measurement the state is projected to only one of the exponentially many possible states, so that the exponential amount of information, which has been computed, is completely lost. In order to gain advantage of exponential parallelism, one needs to combine it with another quantum feature, known as interference. Interference allows the exponentially many computations done in parallel to cancel each other, just like destructive interference of waves or light. The goal is to arrange the cancellation such that only the computations, which we are interested in remain, and all the rest canceled out.

The combination of exponential parallelism and interference is what makes quantum computations powerful and plays an important role in quantum algorithms. The Fourier transform indeed manifests *interference* and *exponentiality*.

*Logic Gates and Quantum Computations.* In classical computations and in digital electronics, one deals with sequences of elementary operations (operations such as AND, OR and NOT). These sequences are used to manipulate an array of classical bits. The operations are elementary in the sense that they act on only a few bits (one or two) at a time. We will (some times) refer to sequences as products and to operations as operators, matrices, instructions, steps and gaits. Further more we will introduce the sequences of elementary operations as basis of quantum computation. In quantum computation one also deals with sequence of elementary operations - SEO (with operations such as controlled-NOT's and qubit rotations), but for manipulating qubits instead of classical bits. Quantum sequences of elementary operations are often represented graphically by qubit circuits. In quantum computation, one often knows the unitary operator  $U$  that describes the evolution of an array of qubits. One must then find a way to reduce  $U$  into the sequence of elementary operations. The algorithm can be applied to any unitary operator  $U$ . It is useful to define certain unitary operator  $U_{N_B}$  for all  $N_B \in \{1, 2, 3, \dots\}$ , where  $U_{N_B}$  is a  $2^{N_B} \times 2^{N_B}$  matrix and  $N_B$  is the number of bits. Some  $U_{N_B}$  are known to be expressible as a sequence of elementary operations whose lengths (i.e. whose number of elementary operations) is a polynomial in  $N_B$ . Two examples are the  $N_B$  bit Hadamard transform (HT) matrix and the  $N_B$  bit discrete Fourier transform (DFT) matrix. The HT matrix is known to be expressible as a sequence of elementary operations of lengths  $Order(N_B)$ . The DFT matrix is known to be expressible (using the Fast Fourier Transform (FFT) algorithm) as a sequence of elementary operations of lengths  $Order(N_B^2)$ . Presented algorithm achieves both of these sequences of elementary operations (SEO) lengths as benchmarks. Even better the SEO often called the "Quantum FFT algorithm" is exactly reproduced by this algorithm.

Previously another algorithms have described for reducing a unitary operator into SEO, and their algorithm can be applied to any unitary operator  $U$ . However, it is very unlikely that their algorithm can be efficient in producing short SEO's unless farther optimizations are added to it. And such optimizations, if they exist, have not been specified by anyone.

## ***Benchmarks of Quantum Gates for Quantum Algorithm's Design***

We represent the general approach to design of quantum gates for quantum algorithms. The results of this approach are described in Table 1.

*Table 1: Benchmarks of Quantum Gates for Quantum Algorithms*

Name	Algorithm	Gate : Symbolic Form
Deutsch – Jozsa	$m = 1;$ $Int = {}^n H \otimes I;$ $k = 1;$ $h = 0$	
Simon	$m = n;$ $Int = {}^n H \otimes {}^n H;$ $k = O(n);$ $h = 0$	
Shor	$m = n;$ $Int = QFT_n \otimes {}^n H;$ $k = O(n^2);$ $h = 0$	
Grover	$m = n;$ $Int = D_n \otimes I;$ $k = 1;$ $h = O(2^{n/2})$	

Using this approach the methodology of quantum algorithm simulation on classical computer is developed. The concrete result of simulation Deutsch – Jozsa, Simon, Shor and Grover algorithms are described in main text. The results of this approach in Table 1 are described.

*Any relations with Walsh-Hadamard Transformation.* The Hadamard (rotation) transformation is one of the simplest and most common fault-tolerant operations. The gate equation is  $\frac{\partial U(t)}{\partial t} = -\frac{i}{\hbar} H(t)U(t)$ , which is subject to initial condition  $U(0) = I$  and target condition  $U(T) = U^0$ . Here  $U^0$  is the matrix of the desired gate.

The wave function  $|\psi(t)\rangle$  is related to the evolution operator  $U(t)$  and the initial wave function  $|\psi(0)\rangle$  by  $|\psi(t)\rangle = U(t)|\psi(0)\rangle$ . We choose the various quantum gates from physics point of view in such a way that they can be described by simple Hamiltonians. The gates correspond to the unitary operation  $(W \rightarrow R \rightarrow H)$  effected by the Hamiltonian  $H_w = \frac{\pi}{2} \left[ \frac{1}{2}(\sigma_x - \sigma_z) + 1 \right]$  acting for one unit at time.

Similarly the gate  $U_y = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$  generated by the Hamiltonian  $H_y = -\frac{\pi}{4} \sigma_y$  acting for one unit of time. The solutions of corresponding Schrodinger equations with these types of Hamiltonians are described the evolution with coherence states (with *minimum of uncertainty*), i.e., the gates are *robustness* and *fault-tolerant*.

*Remark: Physical and Geometrical Interpretations of Walsh-Hadamard Transformations.* It is natural to think of quantum computations as mutiparticle processes. Just as classical computations are processes involving several “particles” or bits. It turns out that viewed quantum computation as multiparticle interferometry leads to a simple and unifying picture of known quantum algorithms. In this language quantum computation are basically multiparticle interferometers with phase shift that result from operations of some quantum logic gates. To illustrate this point consider, for example, a Mach-Zehndler interferometer in Fig. 3a.

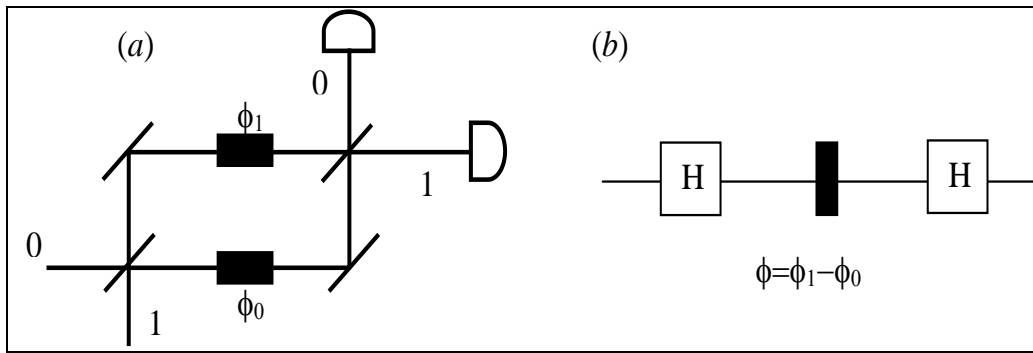


Figure 3. Scheme of a Mach-Zehnder interferometer with two phase shifter

The interference pattern depends on the difference between the phase shifts in different arms of the interferometer. (a) The corresponding quantum network representation. (b) mirror which directs the particle to one of the two detectors. Along each path between the two half-silvered mirrors is a phase shifter. If the lower path is labelled as state  $|0\rangle$  and the upper one as state  $|1\rangle$ , then the state of the particle in between the half-silvered mirrors and after passing through the phase shifters is a superposition of the type *Principle Description of Mach-Zehnder Interferometer*. A particle, say a photon, impinges on a half-silvered mirror, and, with some probability amplitudes, propagates via two different paths to another half-silvered  $\frac{1}{\sqrt{2}}(|0\rangle + e^{i(\varphi_1 - \varphi_0)}|1\rangle)$ , where  $\varphi_0$  and  $\varphi_1$  are the setting of the two phase shifters. The phase shifters in the two paths can be tuned to effect any described relative phase shift  $\varphi = \varphi_1 - \varphi_0$  and to direct the particle with probabilities  $\frac{1}{2}(1 + \cos \varphi)$  and  $\frac{1}{2}(1 - \cos \varphi)$ , respectively, to detectors “0” and “1”. The second half-silvered mirror effectively erases all information about the path taken by the particle (path  $|0\rangle$  or path  $|1\rangle$ ) which is essential for observing quantum interference in the experiment.

*Physical Interpretation of Walsh-Hadamard Transformation.* According to this description let us now rephrase the experiment in terms of quantum logic gates. We can identify the half-silvered mirrors with single-qubit Walsh-Hadamard transform. We view the phase shifter as a single-qubit gate (see Fig. 3, b). The phase shift can be “computed” with the help of an auxiliary qubit (or a set of qubits) in a prescribed state  $|u\rangle$  and some controlled- $U$  (c- $U$ ) transformation where  $U|u\rangle = e^{i\varphi}|u\rangle$  (see Fig. 3, b). Here, the controlled - $U$  transformation depends on the logical value of the control qubit; for example, we can apply the identity transformation to the auxiliary qubits (i.e. do nothing) when the control qubit is in state  $|0\rangle$  and apply a prescribed  $U$  when the control qubit is in state  $|1\rangle$ . The controlled - $U$  operation must be followed by a transformation which brings all computational paths together, like the second half-silvered mirror in the Mach-Zehnder interferometer. This last step is essential to enable the interference of different computational paths to occur – for example, by applying a Walsh-Hadamard transformation. In this example, we obtain the following sequence of transformations on the two qubits:

$$\begin{aligned}
 |0\rangle|u\rangle &\xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|u\rangle \xrightarrow{c-U} \frac{1}{\sqrt{2}}(|0\rangle + e^{i\varphi}|1\rangle)|u\rangle \\
 &\xrightarrow{H} \left( \cos\left(\frac{1}{2}\varphi\right)|0\rangle - i \sin\left(\frac{1}{2}\varphi\right)|1\rangle \right) e^{i\varphi}|u\rangle
 \end{aligned}$$

The state of the auxiliary register  $|u\rangle$ , being an eigenstate of  $U$ , is not altered along this network, but its eigenvalue  $e^{i\varphi}$  is “kicked back” in front of the  $|1\rangle$  component in the first qubit. This sequence is the exact simulation of the Mach-Zehnder interferometer and the kernel of quantum algorithms.

*Geometrical Interpretation of Walsh-Hadamard Transformation.* The Hilbert space of one qubit (i.e. a two-dimensional Hilbert space) equipped with a standard basis denoted by  $B = (|0\rangle, |1\rangle)$ . The dual basis of  $B$  denoted by  $(|0'\rangle, |1'\rangle)$  is defined by  $|0'\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ,  $|1'\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ . Thus  $H^2 = I$  and  $H$  interchange the standard and dual bases. In terms of real geometry, the dual basis lies on the  $45^\circ$  lines between the orthogonal directions  $|0\rangle$  and  $|1\rangle$  (see, Fig. 4) and  $H$  is the transformation given by reflection in a line at angle  $\frac{1}{8}\pi$  to the  $|0\rangle$  direction. Thus the eigenvectors of  $H$  (parallel and perpendicular to the mirror line) are  $\cos\left(\frac{1}{8}\right)|0\rangle \pm \sin\left(\frac{1}{8}\right)|1\rangle$  belonging to  $\lambda = \pm 1$ , respectively.

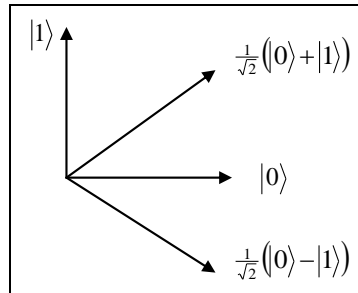


Figure 4. The relations between standard and dual bases

If  $x = (x_1, \dots, x_n)$  and  $y = (y_1, \dots, y_n)$  are in  $B^n$ , then by the inner product between two bit strings  $x$  and  $y$ , we mean the inner product modulo 2, that is:  $(x, y) = x \cdot y = (x_1 y_1 \oplus \dots \oplus x_n y_n)$ . This value can also be viewed as the parity of a subset of the bit string  $y_1 \dots y_n$  (the parity of the number of places where  $x$  and  $y$  both have a bit value of 1). This subset is described by the characteristic vector  $x$  and its size equals the Hamming weight  $\|x\|$  of the bit string  $x_1 \dots x_n$ . The parity basis is the now familiar result of the one bit Hadamard gate (see, Fig. 4).

The Hadamard transformation of a sequence of bits  $y_1 \dots y_n$  and the inner product function are closely related to each other: for any  $y \in \{0,1\}^n$  it holds that  $H^{\otimes n} |y\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{(x,y)} |x\rangle$ .

Because  $H$  is its own inverse, we can apply again a sequence of  $n$  Hadamard transforms on the these states and thus obtain the original bit string  $y_1 \dots y_n$  again:  $H^{\otimes n} \left( \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{(x,y)} |x\rangle \right) = |y\rangle$ . The above leads to the observation that if we want to know the string  $y_1 \dots y_n$ , it is sufficient to have a superposition with phase values of the form  $(-1)^{(x,y)}$ , for every  $x \in \{0,1\}^n$ . This is a well-known result in quantum computation and has been used several to underline the differences between quantum and classical information processing.

The scalar product modulo 2 as  $(x, y) = x \cdot y = (x_1 \wedge y_1) \oplus \dots \oplus (x_n \wedge y_n)$  is the XOR of the bitwise AND of the two strings  $x$  and  $y$ , and equivalent to performing a one-qubit Hadamard transform on each of the  $n$  qubits individually.

*Applications of Walsh-Hadamard Transformation in Mixing Operations.* For effective quantum algorithms, we also need to be able to effective mix amplitudes in a superpositions so as to increase the change of a desired reading being made. One way to achieve this mixing is to combine an efficiently implementable diagonal matrix with a decomposable mixing matrix. For instance, a number of existing algorithms make use of mixing matrices of the form  $HDH$  where  $D$  is a diagonal matrix and  $H$  is the

Walsh-Hadamard transformation given by  $H = W_{xy} = \frac{1}{\sqrt{2^n}} (-1)^{|x \cdot y|}$ . We have described below efficient implementations for certain diagonal matrices that can be combined with the Walsh-Hadamard transformation or other mixing matrices to achieve desirable amplitude interference.

*Remark.* More general form of Walsh-Hadamard transform is the  $n$ -bit Sylvester-Hadamard matrix is defined by

$$H_1 = \begin{matrix} 2 \times 2 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & -1 \end{matrix}; H_2 = \begin{matrix} 4 \times 4 & 00 & 01 & 10 & 11 \\ 00 & 1 & 1 & 1 & 1 \\ 01 & 1 & -1 & -1 & -1 \\ 10 & 1 & 1 & -1 & -1 \\ 11 & 1 & -1 & -1 & -1 \end{matrix}; H_{r+1} = H_1 \otimes H_r; H_n^2 = nI_n.$$

The desired  $H_n$  gate acts on a quantum register by sending each qubit individually into a separate  $H_1$  gate. The unitary transformation induced by an  $H_n$  is given by the formula  $H_n = \otimes_n H_1$ . If  $n > 1$  then a nice recursive definition is defined as

$$H_n = H_1 \otimes H_{n-1} = \begin{pmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{pmatrix}.$$

*Simple Operations of Hadamard Transform with Simple gates.* Let us see what an  $H$  gate does to  $X, Y$ , and  $Z$

$$HXH^* = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = Z; HZH^* = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = X;$$

$$HYH^* = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = -Y.$$

Therefore, applying  $H$  bitwise will switch all the  $X$ 's and all the  $Z$ 's, and give a factor of  $-1$  for each  $Y$ . This is a valid fault-tolerant operation.

Thus, the Hadamard gate implements the Hadamard transform; it is the single qubit gate  $H$  performing the unitary transformation (see, Fig. 5)

$$H = W = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \Leftrightarrow |x\rangle \rightarrow \boxed{H} \rightarrow (-1)^x |x\rangle + |1-x\rangle$$

Figure 5: The Hadamard gate representation. The diagram on the right provides a schematic representation of the gate  $H$  acting on qubit in state  $|x\rangle$ , with  $x = 0,1$

Consider the following unitary transformation (and associated gate):

$$P = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \Leftrightarrow (|1\rangle) \rightarrow \boxed{P} \rightarrow (-|1\rangle)$$

If a qubit is set to 0 nothing happens, but if is set to 1 the amplitudes is multiplied by  $-1$ . This gate “encodes” the value of the qubit into the sign of the amplitude.

Another same common bitwise operation is the  $i$  phase:  $P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ . On the basic operations  $X, Y$ , and  $Z$  it acts as follows:

$$\begin{aligned}
 PXP^* &= \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} 0 & -i \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = iY; \quad PZP^* = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = Z; \\
 PYP^* &= \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} 0 & i \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix} = iX.
 \end{aligned}$$

This switches  $X$  and  $Y$ , but with extra factors of  $i$ , so there must be a multiple of 4  $X$ 's and  $Y$ 's for this to be a valid operation. Note that a factor of  $i$  appears generically in any operation that switches  $Y$  with  $X$  or  $Z$ , because  $Y^2 = -1$ , while  $X^2 = Z^2 = +1$ . These operations actually permute Pauli matrices:  $\sigma_x = X, \sigma_z = Z$ , and  $\sigma_y = iY$ . The most general single qubit operation can be viewed as a rotation of the Bloch sphere permuting the three coordinate axes.

*Remark.* The one-qubit operations correspond to the six automorphisms of the group  $D_4$  of products of  $I, X, Y$ , and  $Z$  (or direct product of copies of  $D_4$  for multiple-qubit gates) given by  $\{I, H, P, Q = P^*HP^*, T = PQ^*, T^2\}$ . So all one-qubit operations are covered. Given any such automorphism, we first substitute  $iY$  for  $Y$  to get the actual transformation. For instance, consider the cyclic transformation  $T = X \rightarrow iY \rightarrow Z \rightarrow X$ . Since  $Z \rightarrow X$ ,  $|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ . Also,  $X \rightarrow iY$ , so

$$|1\rangle \rightarrow \frac{i}{\sqrt{2}}Y(|0\rangle + |1\rangle) = -\frac{i}{\sqrt{2}}(|0\rangle - |1\rangle). \text{ Thus, the matrix for } T \text{ is } T = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ 1 & i \end{pmatrix}.$$

We can perform a similar procedure to determine the matrix corresponding to a multiple-qubit transformation.

*Example.* We can make a copy of entering qubit. The superposition is now  $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ , i.e. entangled state. If we let the first bit go through the Hadamard gate and do nothing to the second one, the result becomes:

$$(H \otimes I) \left( \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \right) = \frac{1}{2}(|00\rangle + |10\rangle) + \frac{1}{2}(|01\rangle - |11\rangle)$$

Each vector has equal norm so we will observe, after measuring, each vector with probability  $\frac{1}{4}$ , and in particular the first bit will be perfectly random. Now let us see what happens if we also make the copy bit go through a Hadamard gate. Will it affect the randomness of the output?

$(H \otimes H) \left( \frac{1}{\sqrt{2}}( 00\rangle +  11\rangle) \right)$	$= \frac{1}{\sqrt{2^3}}(( 0\rangle +  1\rangle) \otimes ( 0\rangle +  1\rangle)) + \frac{1}{\sqrt{2^3}}( 0\rangle -  1\rangle) \otimes ( 0\rangle -  1\rangle)$
	$= \frac{1}{\sqrt{2}}( 00\rangle +  11\rangle)$

Randomness of the first bit remains intact.

*Controlled Two - Bit Gates.* The two-bit gates of the network correspond to *controlled phase shift*. These are represented in the canonical basis  $B = \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$  by the unitary operator

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \text{ generated by the Hamiltonian } H = \pi P_{|1\rangle,i} \otimes P_{|1\rangle,j}, \text{ where } i \text{ and } j \text{ designate the two}$$

qubits on which the gate acts and  $P_{|1\rangle} = \frac{1}{2}(1 + \sigma_z)$  is the projector on state  $|1\rangle$ . In this case a state  $|i, j\rangle$

picks up a phase  $\pi$  if and only if (iff) both qubits are in state  $|1\rangle$ . Variants on this gate can be obtained by replacing either or both of the projectors with  $P_{|0\rangle}$  in the definition of the Hamiltonian.

The conditional phase shift is the two-bit gate (see, Fig. 6)

$$P(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\phi} \end{pmatrix} \Leftrightarrow \left. \begin{array}{c} |x\rangle \\ |y\rangle \end{array} \right\} e^{ixy\phi} |x\rangle |y\rangle$$

Figure 6. The two-bit conditional phase shift gate. The matrix written in the basis  $B = \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ . The diagram on the right shows the structure of the gate

*Controlled Two - Qubit Gates.* To see how such unitary operators may be constructed from a few elementary ones we must also consider the *controlled-NOT* (or *XOR*) gate. Just as any classical computation can be expressed as a sequence of one- and two-bit operations (e.g., NOT and AND gates), any quantum computation can be expressed as a sequence of one- and two-qubit quantum gates, i.e., unitary operations acting on one or two qubits at a time (see, Fig. 7).

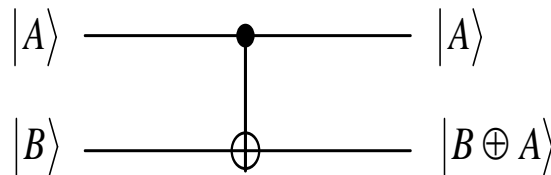


Figure 7. Quantum circuit diagram for an XOR gate. The lower bit  $|B\rangle$  is flipped whenever the upper bit  $|A\rangle$  is set

The standard two-qubit is the controlled-NOT or XOR gate, which flips its second (or “target”) input if its first (“control”) input is  $|1\rangle$  and does nothing if the first input is  $|0\rangle$ . In other words its interchanges  $|10\rangle$  and  $|11\rangle$  while leaving  $|00\rangle$  and  $|01\rangle$  unchanged. Writing the two-particle basis states as the vectors

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}; |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}; |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}; |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix},$$

we may represent the XOR gate by  $4 \times 4$  unitary matrix as a unitary operator

$$U_{XOR} = CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \equiv \begin{pmatrix} I & 0 \\ 0 & NOT \end{pmatrix}. \quad CNOT : \begin{array}{|c|c|c|} \hline |00\rangle & \rightarrow & |00\rangle \\ \hline |01\rangle & \rightarrow & |01\rangle \\ \hline |10\rangle & \rightarrow & |11\rangle \\ \hline |11\rangle & \rightarrow & |10\rangle \\ \hline \end{array}.$$

The XOR gate is a prototype interaction between two quantum particles (systems), and illustrate several key features of quantum information, in particular the impossibility of “cloning” an unknown quantum state, and the way interaction produces entanglement. Here the first particle acts as a conditional gate to flip the state of the second particle. It is easy to check that the state of the second particle corresponds to the action of the XOR gate. The quantum circuit for the XOR gate is illustrated in Fig. 8. This circuit is equivalent to the elementary instruction: *if* ( $|x\rangle = 1$ ) *then* ( $|y\rangle = NOT|y\rangle$ ), which may be thought of as example of quantum computer code.



$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \Leftrightarrow \begin{array}{ccc} |x\rangle & \text{---} \bullet \text{---} & |x\rangle \\ & | & \\ |y\rangle & \text{---} \oplus \text{---} & |x \oplus y\rangle \end{array}$$

Figure 8. Quantum circuit diagram for an CNOT (XOR) gate. The lower bit  $|y\rangle$  is flipped whenever the upper bit  $|x\rangle$  is set

*Remark.* The quantum controlled-NOT gate is not a universal quantum gate but a universal quantum gate can be constructed by a combination of the controlled-NOT and simple unitary operations on a single qubit. This means that the XOR gate, together with the set of one-bit gates, form a universal set of primitives for quantum computation; that is, any quantum computation can be performed using just this set of gates without an undue increase in the number of gates used. Unlike one-qubit gates, two-qubit gates are difficult to realize in the laboratory, because they require two separate quantum information carriers to be brought into strong and controlled interaction (see below).

If the CNOT (XOR) is applied to Boolean data in which the second qubit is 0 and the first is 0 or 1 (see also, Fig. 8) the effect is to leave the first qubit unchanged while the second becomes a copy of it:  $U_{CNOT}|x,0\rangle = |x,x\rangle$  for  $x = 0$  or  $1$ . One might suppose that the CNOT operation could also be used to copy superpositions, such as  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ , so that  $U_{CNOT}|\psi,0\rangle$  would yield  $|\psi,\psi\rangle$ , but this is not so. The unitary of quantum evolution requires that the superposition of input states evolve to a corresponding superposition of outputs. Thus the result of applying  $U_{CNOT}$  to  $|\psi,0\rangle$  must be  $\alpha|00\rangle + \beta|11\rangle$ , an entangled state in which neither output qubit alone has definite state. If one of the entangled output qubits is lost (e.g., discarded, or allowed to escape into the environment), the other thenceforth behaves as if it had acquired a random classical value 0 (with probability  $|\alpha|^2$ ) or 1 (with probability  $|\beta|^2$ ). Unless the lost output is brought back into play, all record of the original superposition  $|\psi\rangle$  will have been lost. This behavior is characteristic not only of the CNOT gate but of unitary interactions generally: their typical effect is to map most non-entangled initial states of the interacting systems into entangled final states, which from the viewpoint of either system alone causes an unpredictable disturbance.

*Remark.* The classical controlled – NOT gate is a reversible logic gate operating on the two bits  $\varepsilon_1$  and  $\varepsilon_2$ ;  $\varepsilon_1$  is called the *control bit* and  $\varepsilon_2$  - the *target bit*. The value of  $\varepsilon_2$  is negated if  $\varepsilon_1=1$ , otherwise  $\varepsilon_2$  is left unchanged (in both cases the control bit  $\varepsilon_1$  remains unchanged). The quantum controlled – NOT gate  $C_{12}$  is the unitary operation on two qubits, i.e. state in  $H_2$ , which in a chosen orthonormal basis  $\{|0\rangle, |1\rangle\}$  reproduces the controlled – NOT operation:  $|\varepsilon_1\rangle|\varepsilon_2\rangle \xrightarrow{C_{12}} |\varepsilon_1\rangle|\varepsilon_1 \oplus \varepsilon_2\rangle$ , where  $\oplus$  denotes addition modulo 2. Here and in the following the first subscript of  $C_{ij}$  always refers to the control bit and the second to the target bit. Thus, for example,  $C_{21}$  denotes the unitary operation defined by  $|\varepsilon_1\rangle|\varepsilon_2\rangle \xrightarrow{C_{21}} |\varepsilon_1 \oplus \varepsilon_2\rangle|\varepsilon_2\rangle$ .

*The properties of controlled – NOT gate.* Let us specify any interesting properties of the quantum controlled – NOT gate. The CNOT gate is the idealized discrete operation for producing entangled states.

The quantum controlled-NOT gate transforms *superpositions* into

$$C_{12} : (a|0\rangle + b|1\rangle) \rightarrow \underbrace{a|00\rangle + b|11\rangle}_{\text{quantum entanglement}}$$

As Fig. 9 indicates, a particular product-state input to the gate as shown, using two states from non-orthogonal bases (related by a Hadamard transform), produces at the output the non-product state:  $\frac{1}{2}(|01\rangle - |10\rangle)$ , a state equivalent to *EPR-Bohm* pair.

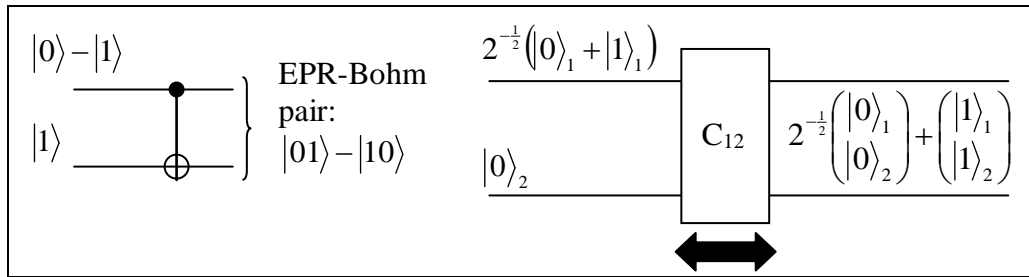


Figure 9. *CNOT* produces perfectly entangled quantum states from unentangled ones (a); The *CNOT* gate applied to two qubits. The right hand state entangled; it is not possible to associate a definite state with one qubit independent of the other, as is the case for the left hand side. The reversibility of the gate means that it can be run left to right or right to left (b)

Thus it acts as “*the measurement gates*” because if the target bit  $\varepsilon_2$  is initially in state  $|0\rangle$  then this bit is in effect an apparatus that performs a perfectly accurate non-perturbing (*quantum non-demolition* (QND) measurement type) measurement of  $\varepsilon_1$ . This is illustrated in Fig. 10: if the object is to measure state of the upper qubit (that is, whether it is in the  $|0\rangle$  state or the  $|1\rangle$  state), we may *CNOT* it with a second bit started in the  $|0\rangle$  state; then a measurement of the second bit will reveal the desired outcome.

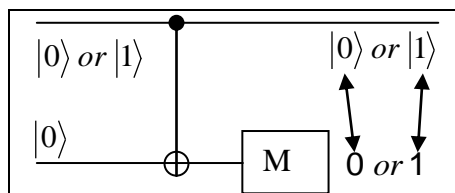


Figure 10. *CNOT* functions as an ideal non-demolition measurement apparatus for a qubit.

This may not appear to be much of an advantage over measuring the first qubit directly. However, it has the feature of being a “non-demolition” measurement in which the original quantum state remains in existence after the measurement. It only remains undistributed if it started in the  $|0\rangle$  or the  $|1\rangle$  state; if it started in a superposition, then the state is “collapsed” by the measurement.

*Example.* To appreciate the real power of the non-demolition capability of the *CNOT* gate, consider the simple quantum circuit of Fig. 11.

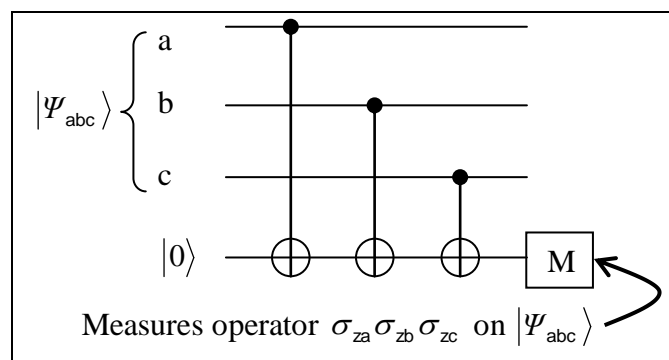


Figure 11. A circuit of *CNOT*s can be used to do a non-demolition measurement of the three-particle operator shown

The effect of the three successive CNOTs followed by a measurement of the target qubit is to accomplish a highly non-trivial non-demolition measurement of the three-particle Hermitian operator  $\sigma_{za}\sigma_{zb}\sigma_{zc}$ . Previous discussions of such three-particle operators always assumed that it would necessarily be done in a “demolishing” fashion where each of the one-particle operators were measured separately. This property forms the basis of the use of the CNOT gate in the implementation of error correction and hence in fault-tolerant quantum computation.

This transformation of superpositions into entanglements can be reversed by applying the same controlled-NOT operation again. If we define a conjugate qubit basis by  $|0'\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$  and  $|1'\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ , then it is easy to show that when both input qubits are considered in the conjugate basis, then the effective gate action is an CNOT but with the source and target bits reversed. Hence it can be used to implement the Bell measurement on the two bits by *disentangling the Bell states*.

For the four Bell states we get for product states

$$C_{12} \frac{1}{\sqrt{2}}(|0\rangle|0\rangle \pm |1\rangle|1\rangle) = \frac{1}{\sqrt{2}}(\overbrace{|0\rangle \pm |1\rangle}^{\text{disentangled}})|0\rangle; C_{12} \frac{1}{\sqrt{2}}(|0\rangle|1\rangle \pm |1\rangle|0\rangle) = \frac{1}{\sqrt{2}}(\overbrace{|0\rangle \pm |1\rangle}^{\text{disentangled}})|1\rangle.$$

Thus the Bell measurement on the two qubits is *reduced* to the simple sequence of two independent two dimensional measurements: in the basis  $\{|0\rangle, |1\rangle\}$  for the control qubit and in the basis  $\frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$  for the target qubit. The realization of the Bell measurement is the main obstacle in the practical implementation of quantum teleportation and the dense quantum coding. If just the target bit is represented in the conjugate basis, then the action of the CNOT is completely symmetric on the two qubits, having the form

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \text{ This “phase-shift” form of the gate (controlled-Rotation - CROT) is the one which}$$

has been discussed in the cavity quantum-electrodynamic implementation of a two-bit quantum gate by Turchette, 1995.

The XOR gate allows us to move information around as illustrated in Fig. 12.

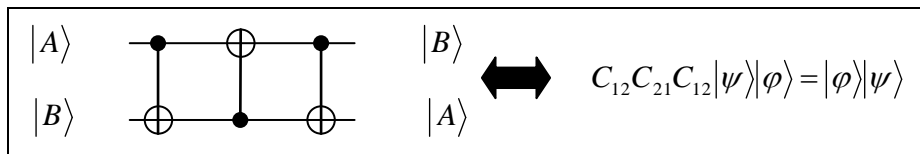


Figure 12. Circuit for swapping a pair of bits

Quantum state *swapping* can be achieved by cascading three quantum controlled-NOT gates for arbitrary states  $|\psi\rangle$  and  $|\phi\rangle$ .

*Remark:* The quantum controlled-NOT gate is not a universal gate, however, the universal quantum gate can be constructed by a simple extension of the controlled-NOT gate to the *controlled-controlled-NOT* gate combined with simple unitary operations on a single qubit (see *Theorem 1*).

It is straightforward to show that the CNOT gate induces the following transformation:

$X \otimes I \xrightarrow{CNOT} X \otimes X$	$Z \otimes I \xrightarrow{CNOT} Z \otimes I$
$I \otimes X \xrightarrow{CNOT} I \otimes X$	$I \otimes Z \xrightarrow{CNOT} Z \otimes Z$

It is easy to see here how amplitudes are copied forwards and phases are copied backwards. For two arbitrary unitary transformations  $U_1$  and  $U_2$ , the “conditional” transformation  $|0\rangle\langle 0| \otimes U_1 + |1\rangle\langle 1| \otimes U_2$  is also unitary. The CNOT gate can be defined by

$$C_{NOT} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X .$$

The transformation CNOT is unitary since  $CNOT^* = CNOT$  and  $CNOT \cdot CNOT = I$ . The CNOT gate cannot be decomposed into tensor product of two single-bit transformations. A CNOT-operation can be performed in an ion trap by a sequence of three operations: *Controlled-ROT Gate, Controlled-SWAP Gate, and Basis Transformations*.

Another types of controlled –  $U$  gates can be described as in Table 2.

Table 2. Typical controlled –  $U$  gates and Its Matrix Representation Forms

<i>Title of Operations</i>	<i>Matrix Presentation Forms</i>
<i>CNOT</i>	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
<i>CROT</i>	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$
<i>SWAP</i>	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Using the operations from the Table 2 it is possible defined the new operations as in Table 3.

In general form this gate can be written as the unitary transformation in matrix form as in Fig. 13.

Table 3. Relations between the Operations CROT and SWAP

<i>Title of Operations</i>	<i>Re presentation Forms</i>
$CROT(x, y) =$	$SWAP(y, x) * CROT(x, y) * SWAP(y, x)$
$SWAP(y, x) =$	$CNOT(y, x) * CNOT(x, y) * CNOT(y, x)$

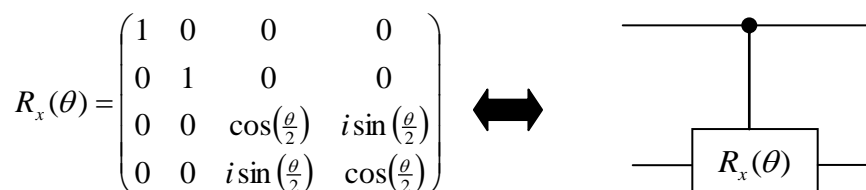


Figure 13. CROT gate and its Graphical Representation Form

This gate rotates the target (lower) qubit by  $R_x(\theta)$  (angle  $\theta$  about the  $x$  – axis, if control (upper) qubit is  $|1\rangle$ ), otherwise it does nothing. This gate can be implemented using  $CNOT$  gates and two single qubit rotations.

### Basis Transformations and Reduction of Any Unitary Matrices. State Permutations.

We introduce short definitions of permutation theory and applications to the reduction problem of any unitary matrix into product of qubit rotations and  $CNOT$ 's.

*Permutations.* A *permutation* is a 1 – 1 onto map from a finite set  $X$  onto itself. The set of permutations on set  $X$  is a group if group multiplication is taken to be function composition.  $S_n$ , the *symmetric group in  $n$  letters*, is defined as the group of all permutations on any set  $X$  with  $n$  elements. If  $X = Z_{1,n}$ , then a permutation  $G$  which maps  $i \in X$  to  $a_i \in X$  (where  $i \neq j$  implies  $a_i \neq a_j$ ) can be represented by a matrix with entries  $(G)_{i,j} = \delta(a_i, j)$ , for all  $i, j \in X$ . Note that all entries in any given row or column equal zero except for one entry which equals one. Hence, the rows of  $G$  is an orthogonal matrix. An alternative notation for  $G$  is  $G = \begin{pmatrix} 1 & 2 & 3 & \dots & n \\ a_1 & a_2 & a_3 & \dots & a_n \end{pmatrix}$ . The product of two symbols of the this type is defined by function composition.

*Example.* For example,

$$\begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ a_1 & a_2 & a_3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ b_1 & b_2 & b_3 \end{pmatrix}.$$

A cycle is a special type of permutation.

If  $G \in S_n$  maps  $a_1 \rightarrow a_2, a_2 \rightarrow a_3, \dots, a_{r-1} \rightarrow a_r, a_r \rightarrow a_1$ , where  $i \neq j$  implies  $a_i \neq a_j$  and  $r < n$ , then we call  $G$  a *cycle*.

Another way to represent  $G$  it is by  $G = (a_1, a_2, \dots, a_r)$ . We say that the cycle has in this case length  $r$ . Cycles of length 1 are just the *identity* map. A cycle of length 2 is called a *transposition*. The product of two cycles need not be another cycle.

*Example.* For example,  $(2, 1, 5)(1, 4, 5, 6) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 1 & 3 & 2 & 6 & 5 \end{pmatrix}$  cannot be expressed as a single cycle.

Any permutation can be written as a product of cycles.

*Example.* For example,

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 1 & 3 & 2 & 6 & 5 \end{pmatrix} = (5, 6)(1, 4, 2).$$

The cycle on the right side of this equation is *disjoint*; i.e., they have no elements in common. Disjoint cycles commute.

Any cycle can be expressed as a product of transpositions (assuming a group with  $\geq$  two elements), by using identities such as:

$$(a_1, a_2, \dots, a_n) = (a_1, a_2)(a_2, a_3) \dots (a_{n-1}, a_n), (a_1, a_2, \dots, a_n) = (a_1, a_n) \dots (a_1, a_3)(a_1, a_2).$$

Another useful identity is  $(a, b) = (a, p)(p, b)(a, p)$ . This last identity can be applied repeatedly.

*Example.* For example, applied twice, it gives

$$(a, b) = (a, p_1)(p_1, b)(a, p_1) = (a, p_1)(p_1, p_2)(p_2, b)(p_1, p_2)(a, p_1).$$

Since any permutation equals a product of cycles, and each cycles can be expressed as a product of transpositions, all permutations can be expressed as a product of transpositions (assuming a group with  $\geq$  two elements). The decomposition of a permutation into transpositions is not unique. However, the number of transpositions whose product equals a given permutation is always either even or odd. An even (ditto, odd) permutation is defined as one, which equals an even (ditto, odd) number of transpositions.

*Basis Transformations and Reduction of Any Unitary Matrices.* Unlike ideal classical gates, ideal quantum gates do not have high-impedance inputs. In fact, the role of “control” and “target” are arbitrary – they depend on what basis you think of a device as operating in. We have given the truth table for a CNOT and shown the control qubit does *not* get changed in the classical 00, 01, 10, 11 basis. However, in reality, the control qubit does *change*: its phase flipped depending on the state of the “target” qubit. One of example is demonstrated in Fig. 14.

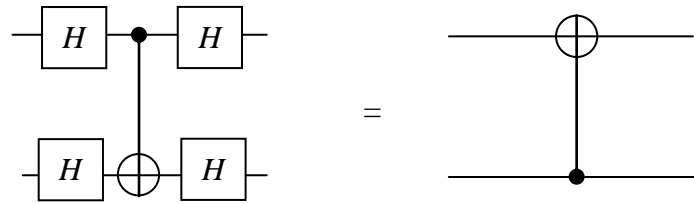


Figure 14. Basis Transformation of Hadamard gates and CNOT.

In order to realize arbitrary unitary transformations, single bit rotations need to be included. It was shown that CNOT with all one-bit quantum gates is a universal gate set. It suffices to include the following single bit rotations and phase shift transformations

$$\begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix}, \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{-i\alpha} \end{pmatrix}, \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{i\alpha} \end{pmatrix}.$$

We consider in this section the method that can to reduce any unitary matrix into a product of qubit rotations and CNOT`s. A qubit rotation acts on a single qubit at a time. We will discuss gates as CNOT`s that are state permutations that act on two bits at a time. Consider first the case when there are only two bits. Then there are four possible states: 00, 01, 10, 11. With these four states, one can build six distinct transpositions:

$$(00,01) = \begin{pmatrix} \sigma_x & \\ & I_2 \end{pmatrix} = P_0 \otimes \sigma_x + P_1 \otimes I_2 = \sigma_x(0)^{\bar{n}(1)} \tag{1}$$

$$(00,10) = \begin{pmatrix} P_1 & P_0 \\ P_0 & P_1 \end{pmatrix} = I_2 \otimes P_1 + \sigma_x \otimes P_0 = \sigma_x(1)^{\bar{n}(0)} \tag{2}$$

$$(00,11) = \begin{pmatrix} & 1 \\ I_2 & \\ 1 & \end{pmatrix}, \tag{3}$$

$$(01,10) = \begin{pmatrix} 1 & \\ & \sigma_x \\ & & 1 \end{pmatrix}, \tag{4}$$

$$(00,10) = \begin{pmatrix} P_0 & P_1 \\ P_1 & P_0 \end{pmatrix} = I_2 \otimes P_0 + \sigma_x \otimes P_1 = \sigma_x(1)^{n(0)} \tag{5}$$

$$(00,01) = \begin{pmatrix} I_2 & \\ & \sigma_x \end{pmatrix} = P_0 \otimes I_2 + P_1 \otimes \sigma_x = \sigma_x(0)^{n(1)}, \tag{6}$$

where matrix entries left blank should be interpreted as zero. The rows and columns of the above matrices are labelled by binary numbers in increasing dictionary order. Note that the four transpositions (Positions (1), (2), (5), and (6)) change only one bit value. The other two transpositions (Positions (3) and (4)) change both bit value.

We will call (00, 11) the *Twin - to - Twiner* and (01, 10) the *Exchanger*. Expressions such as  $\sigma_x(\beta)^{n(\alpha)}$  where  $\alpha \neq \beta$  are a special case of  $M_1(\beta_1)^{M_2(\beta_2)}$ , which was defined above. In this case  $\sigma_x(\beta)^{n(\alpha)}$  equals  $\sigma_x(\beta)$  when it acts on a state for which  $n(\alpha) = 1$ , whereas it equals 1 if  $n(\alpha) = 0$ ;  $\alpha$  is called the *control bit* and  $\beta$  the *flipper bit*.

*Remark.* The *Exchanger* has four possible representations as a product of *CNOT*:

1	$(01,10) = (01,00)(00,10)(01,00) = \sigma_x(0)^{\bar{n}(1)} \sigma_x(1)^{\bar{n}(0)} \sigma_x(0)^{\bar{n}(1)}$
2	$(01,10) = (01,11)(11,01)(01,11) = \sigma_x(0)^{n(1)} \sigma_x(1)^{n(0)} \sigma_x(0)^{n(1)}$
3	$(01,10) = (10,00)(00,01)(10,00) = \sigma_x(1)^{\bar{n}(0)} \sigma_x(0)^{\bar{n}(1)} \sigma_x(1)^{\bar{n}(0)}$
4	$(01,10) = (01,11)(11,10)(01,11) = \sigma_x(1)^{n(0)} \sigma_x(0)^{n(1)} \sigma_x(1)^{n(0)}$

Note that one can go from Position 1 to Position 2 by exchanging  $n$  and  $\bar{n}$ ; from Position 1 to Position 3 by exchanging bit positions 0 and 1; from position 1 to Position 4 by doing both, exchanging  $n$  and  $\bar{n}$ , and exchanging bit positions 0 and 1.

We will often represent *Exchanger* by  $E(0,1)$ . It is easy to show that

1	$E^T(0,1) = E(0,1) = E^{-1}(0,1)$
2	$E(0,1) = E(1,0)$
3	$E^2(0,1) = 1$

Furthermore, if  $X$  and  $Y$  are two arbitrary  $2 \times 2$  matrices, then, by using the matrix representation of *Exchanger*, one can show that  $E(0,1) \circ (X \otimes Y) = Y \otimes X$ . Thus, *Exchanger* exchanges the position of matrices  $X$  and  $Y$  in the tensor product.

*Twin - to - Twiner* operator also has four possible representations as a product of *CNOT*. One is  $(00,11) = (00,01)(01,11)(00,01) = \sigma_x(0)^{\bar{n}(1)} \sigma_x(1)^{n(0)} \sigma_x(0)^{\bar{n}(1)}$ . As with *Exchanger*, the other three representations are obtained by exchanging: (1)  $n$  and  $\bar{n}$ ; (2) bit positions 0 and 1; (3) both.

*Toffoli Gate.* Tom Toffoli (1980) inspired by Bennet reversibility, investigate how reversible computing could be done in the traditional language of Boolean logic gates. He showed that a set of modified gates could be used in place of the traditional Boolean logic gates like AND, OR, etc. One of these, which has turned out to be of central importance in the subsequent quantum gate work is the CNOT gate and was discussed above. The simple retention of the source bit makes the CNOT gate reversible – the input is a unique function of the output. The target bit is transformed to the exclusive – NOT (XOR), while the source bit is unchanged. The CNOT gate is not universal for Boolean computation. Toffoli sought another reversible gate, which could play the role of a universal gate.

*Remark.* A “universal” logic gate is one from which one can assemble a circuit which will evaluate any arbitrary Boolean function. In ordinary (irreversible) Boolean logic, NAND (or AND supplemented by NOT) is one choice for the universal gate.

The *Toffoli gate* (double CNOT) requires three bits, symbolized in Fig. 15.

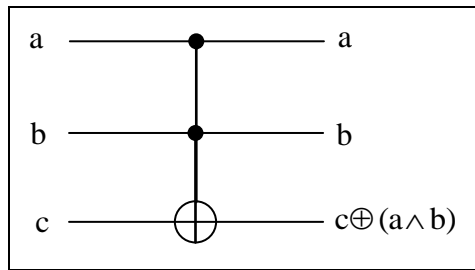


Figure 15. The three – bit Toffoli gate is universal for reversible Boolean logic. The action of the gate on the three input bits is indicated

Any universal reversible Boolean logic gate must have at least three bits. In essence, this gate is an AND gate in which both input bits are saved; as Fig. 15 indicates, bits  $a$  and  $b$  are unchanged, while bit  $c$  is “toggled” by  $a \wedge b$ . The Toffoli gate is reversible classical gate which is universal for classical computation. The XOR is also a classical gate, but the only classical functions that can be constructed with it are linear Boolean functions; it takes three bits to provide a reversible classical gate, which is universal for classical computation (recall that all quantum gates must be reversible).

We will show to do both  $\pi/2$  rotations and Toffoli ( $T$ ) gates fault – tolerant. The three – bit controlled – controlled –NOT (Toffoli) gate is also an instance of this conditional definition:

$$T = |0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes CNOT .$$

$T$  can be used to construct complete set of Boolean connectives in that it can be used to construct the NOT and AND operators in the following way:

$T 1,1,x\rangle =  1,1,-x\rangle$
$T x,y,0\rangle =  x,y,x \wedge y\rangle$

The quantum Toffoli gate is a three – qubit gate, as follows:

$ 000\rangle$	$\rightarrow$	$ 000\rangle$
$ 001\rangle$	$\rightarrow$	$ 001\rangle$
$ 010\rangle$	$\rightarrow$	$ 010\rangle$
$ 011\rangle$	$\rightarrow$	$ 011\rangle$
$ 100\rangle$	$\rightarrow$	$ 100\rangle$
$ 101\rangle$	$\rightarrow$	$ 101\rangle$
$ 110\rangle$	$\rightarrow$	$ 111\rangle$
$ 111\rangle$	$\rightarrow$	$ 110\rangle$

The  $T$  gate is sufficient to construct arbitrary combinatorial circuits.

*Example.* Consider the trivial example of a double controlled – NOT (Toffoli) gate,  $T$ , that computes the conjunction of two values as in Fig. 16.

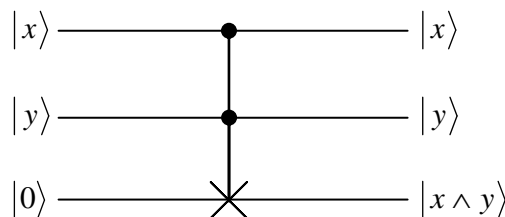


Figure 16. Calculation scheme of the conjunction of two values with Toffoli gate

Now take as input a superposition of all possible bit combinations of  $x$  and  $y$  together with the necessary 0:



$$\begin{aligned}
 H|0\rangle \otimes H|0\rangle \otimes |0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle \\
 &= \frac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |110\rangle)
 \end{aligned}$$

Superposition of inputs leads to superposition of results, namely

$$T(H|0\rangle \otimes H|0\rangle \otimes |0\rangle) = \frac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |111\rangle)$$

The resulting superposition can be viewed as a truth table for the conjunction, or more generally as the graph of a function. In the output the values of  $x$ ,  $y$ , and  $x \wedge y$  are *entangled* in such a way that measuring the result will give one line of the truth table, or more generally one point of the function graph. Note that the bits can be measured in any order: measuring the result will project the state to a superposition of the set of all input values for which the function  $f$  produces this result; measuring the input will project the result to the corresponding function value.

*Example.* Consider the application of  $T$  gate in quantum computation. Suppose we want to build a dedicated quantum device to factor large integers. The quantum factorization contains two major operations: quantum exponentiation (computing  $a^x \bmod N$ ) followed by the Quantum Fourier transform. Quantum exponentiation can be decomposed into a sequence of squaring,  $a^x = a^{2^0 x_0} \cdot a^{2^1 x_1} \cdot \dots \cdot a^{2^{l-1} x_{l-1}}$ , where  $x_0, x_1, \dots$  are the binary digits of  $x$ . Squaring is achieved by multiplication and multiplication by a sequence of additions. Following this reduction procedure we end up with a quantum adder as a basic unit for the whole network. However, a quantum adder is different from a classical adder. Any unitary operation is reversible which is why quantum network for basic arithmetic cannot be directly deduced from their classical Boolean counterparts (classical logic gates such as AND or OR are clearly irreversible – reading 1 at the output of the OR gate does not provide enough information to determine the input which could be either (0,1) or (1,0) or (1,1)). Quantum arithmetic must be build *ab initio* from the reversible logical components.

A simplified quantum adder shown in Fig. 17.

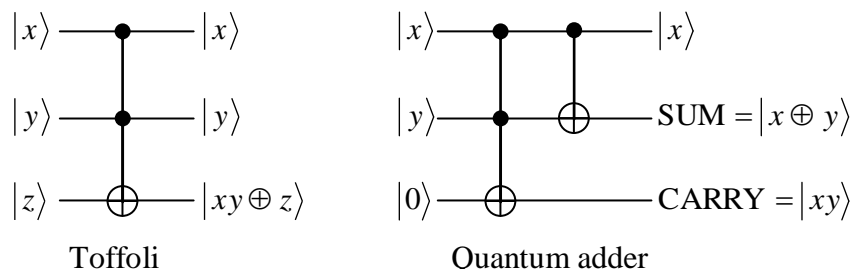


Figure 17. Diagrammatic representation of the Toffoli gate and a simplified quantum adder. States  $|x\rangle, |y\rangle$  and  $|z\rangle$  belong to the computational basis  $x, y, z = 0$  or  $1$  and both addition  $\oplus$  and multiplication  $x \cdot y$  are performed modulo 2

The Toffoli gate is a basic unit which features prominently in the network implementing elementary quantum arithmetic, i.e. in quantum adders, multipliers etc. It can be decomposed and written as a quantum network of elementary two – qubit and one – qubit gates. The following quantum circuit, for example, implements a 1 bit full adder using  $T$  and CNOT gates (Fig. 18):

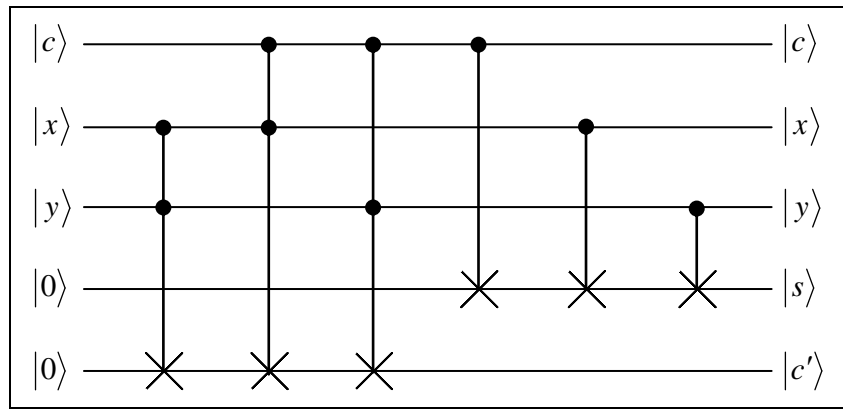


Figure 18. An one – bit full adder using Toffoli and CNOT gates

Where  $x$  and  $y$  are the data bits,  $s$  is their sum (modulo 2),  $c$  is the incoming carry bit, and  $c'$  is the new carry bit. It is possible define more complex circuits that include in – place addition and modular addition.

A simplified quantum adder is a starting point for constructing more elaborate networks.

*Example.* How do we construct the Toffoli gate? One major problem with this gate is that it requires three bits in and three out. Quantum mechanically, this seems to correspond to a scattering process involving three – particle collisions calling for a (possible) unreasonable control of the particles. Fortunately, the Toffoli gate may be constructed by two – particle scattering processes. In particular, we show a construction here involving the CNOT gate and some one – bit gates  $U_\theta$  (Fig. 19).

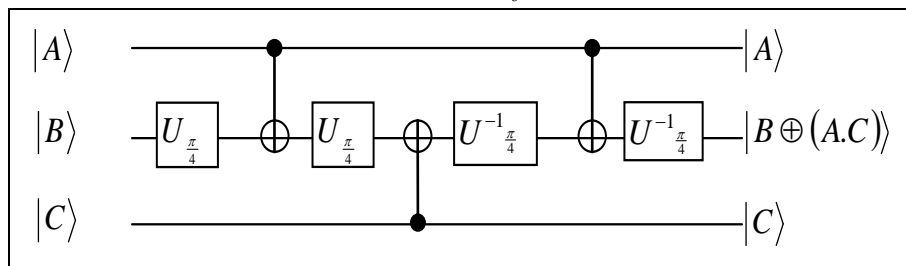


Figure 19. The Toffoli gate built from two – bit CNOT gates plus some one – bit gates. This circuit introduced some extra signs in the unitary matrix  $U_{CNOT}$  which may be removed at a later stage

Not only is the CNOT sufficient for all logic operations in quantum computation, but it can be used to construct arbitrary unitary transformations on any finite set of bits.

*The Fredkin Gate (F).* The  $F$  gate is a “controlled SWAP” and can be defined as

$$F = |0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes S$$

where  $S$  is the SWAP operation:  $S = |00\rangle\langle 00| + |01\rangle\langle 10| + |10\rangle\langle 01| + |11\rangle\langle 11|$ . The Fredkin ( $F$ ) gate has the truth table as in Fig. 20.

Inputs			Outputs		
A	B	C	A'	B'	C'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	1	1	1

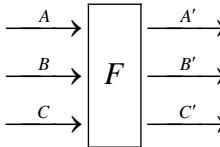


Figure 20. The truth table of Fredkin gate

The following table shows that  $F$ , like  $T$ , is complete for combinatorial circuits:

$F x,0,1\rangle$	=	$ x, x, \neg x\rangle$
$F x, y, 1\rangle$	=	$ x, y \vee x, y \vee \neg x\rangle$
$F x, 0, y\rangle$	=	$ x, y \wedge x, y \wedge \neg x\rangle$

The CNOT gate can be constructed from two Fredkin gates.

While the  $T$  and  $F$  gates are complete for combinatorial circuits, their cannot achieve arbitrary quantum state transformations. In order to realize arbitrary transformations, single bit rotations need to be included.

*Decomposition of controlled -  $U, V$  Gates and Universality for Quantum Gates.* As claimed above, the CNOT gate, when supplemented by a repertoire of one – bit quantum gates, is sufficient to perform any arbitrary quantum computation. Many important quantum computations are formulated quite naturally using this repertoire. This repertoire (CNOT plus one – bit gates) is “universal” for quantum computation in the sense that the Toffoli gate above is universal for reversible Boolean computation. We were able to make use of another important early discovery of Deutsch (1989), which was that three – qubit gates  $U_D$  are universal for quantum gate constructions, where  $U_D$  has the “double - controlled” form

$$U_D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & u_{11} & u_{12} \\ 0 & 0 & 0 & 0 & 0 & 0 & u_{21} & u_{22} \end{pmatrix}$$

Here the  $u_{ij}$  constitute a generic  $U(2)$  matrix. This gate is a quantum generalization of the Toffoli gate. It is fortunate for the prospects for physical implementation of quantum computation that, unlike in Boolean reversible computation, the Deutsch gate can indeed be broken down into simple parts. The simplest means of achieving this decomposition is shown in Fig. 21.

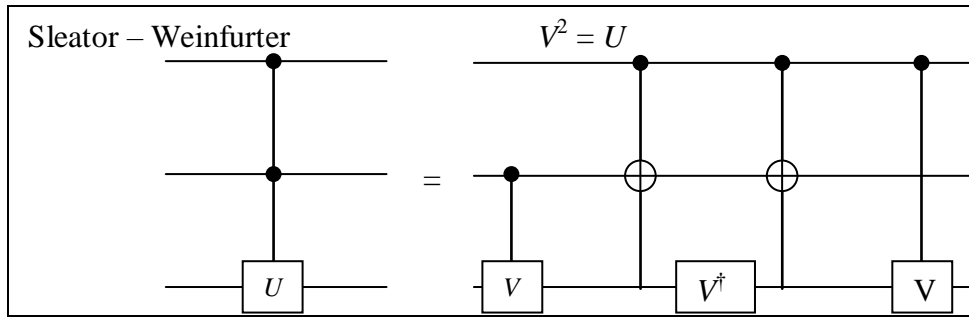


Figure 21. The construction showing the Deutsch three – qubit gate can be broken down into a series of two – qubit operations.

The first step of the decomposition, discovered by Sleator & Weinfurter, uses two CNOT gates and three “controlled - V” gates, whose matrix description is

$$U_V = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & V_{11} & V_{12} \\ 0 & 0 & V_{21} & V_{22} \end{pmatrix}$$

Here V is a  $U(2)$  matrix such that  $V^2 = U$ . These two –bit controlled – V gates could be further broken down, as shown in Fig. 22.

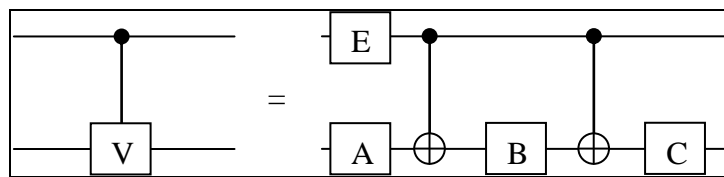


Figure 22. Decomposition of the controlled – V gate into CNOT`s and one – bit gates

Here A, B, C, and E are one – bit gates for which can be obtained explicit formulae (see abovementioned description).

Thus, the circuits, elements which would be needed in any quantum computation in which we are currently interested can be readily be simulated by short sequences of two – bit gates. For instance, the Toffoli gate, which would be the basis of much of the ordinary Boolean logic which is needed for large sections of, for example, Shor prime factoring, can be obtained with just six CNOT`s and eight one – bit gates by using the constructions above (this is shown in Fig. 23).

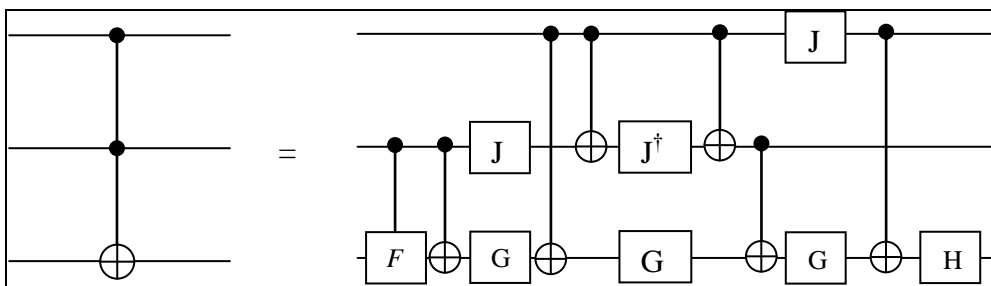


Figure 23. Simplest decomposition of the Toffoli gate into six CNOT gates and eight one – bit gates (operations F, G, H, and J in text are described)

The unitary operators for one – bit gates in this construction are

$F = \begin{pmatrix} e^{i\pi/4} \cos \frac{1}{8}\pi & e^{i\pi/4} \sin \frac{1}{8}\pi \\ -e^{i\pi/4} \sin \frac{1}{8}\pi & e^{-i\pi/4} \cos \frac{1}{8}\pi \end{pmatrix}$	$G = \begin{pmatrix} \cos \frac{1}{8}\pi & -\sin \frac{1}{8}\pi \\ \sin \frac{1}{8}\pi & \cos \frac{1}{8}\pi \end{pmatrix}$
$H = \begin{pmatrix} e^{-i\pi/4} & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$	$J = \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\pi/4} \end{pmatrix}$

It may be noted that in this construction the gates can be grouped into a sequence of just five two – bit operations: first a 2 – 3 operation, then 1 –3, 1 – 2, 2 – 3 and finally 1 – 3 (numbering the qubits 1 – 2 – 3 from the top). Simulations have indicated that the Toffoli gate can be obtained with no fewer than five two – bit quantum gates of any type.

In a related result, Margolus has found an “almost” Toffoli gate which requires even less resources, as shown in Fig. 24: just three CNOT’s and four one – bit gates (three two – bit gates over).

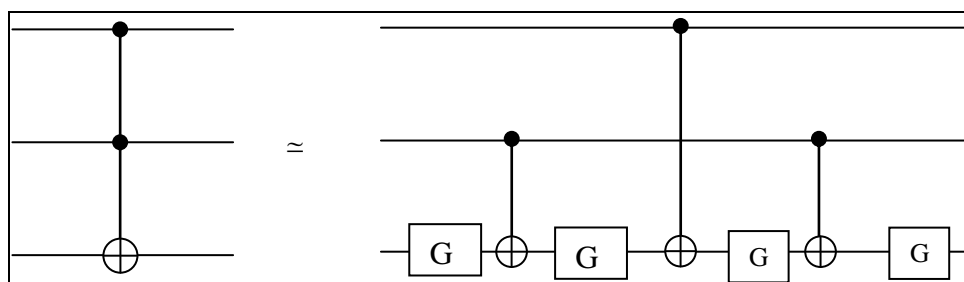


Figure 24. Margolus`s simplified Toffoli gate construction, if just one of the quantum phases is allowed to be changed

It is “almost” in the sense that one of the matrix elements of the Toffoli gate is changed from 1 to – 1 (the one corresponding to the  $|100\rangle$  state). This is not generally acceptable for quantum computation, where all the phases must be correct. However, in many quantum programmes the Toffoli gates appear in pairs, so that the “wrong” phase of the Margolus construction can be arrange to cancel out.

## References

1. Gruska J. Quantum computing. – Advanced Topics in Computer Science Series, McGraw-Hill Companies, London. – 1999.
2. Nielsen M.A. and Chuang I.L. Quantum computation and quantum information. – Cambridge University Press, Cambridge, Englandю – 2000.
3. Hirvensalo M. Quantum computing. – Natural Computing Series, Springer-Verlag, Berlinio – 2001.
4. Hardy Y. and Steeb W.-H. Classical and quantum computing with C++ and Java Simulations. – Birkhauser Verlag, Basel. – 2001.
5. Hirota O. The foundation of quantum information science: Approach to quantum computer (in Japanese). – Japan. – 2002.
6. Pittenberg A.O. An introduction to quantum computing and algorithms. – Progress in Computer Sciences and Applied Logic. – Vol. 19. – Birkhauser. – 1999.
7. Brylinski F.K. and Chen G. (Eds). Mathematics of quantum computation. – Computational Mathematics Series. – CRC Press Co. – 2002.
8. Lo H.-K., Popescu S. and Spiller T. (Eds). Introduction to quantum computing and information. – World Scientific Publ. Co. – 1998.

9. Berman G.P., Doolen G.D., Mainieri R. and Tsifrinovich V.I. Introduction to quantum computers. – World Scientific Publ. Co. – 1999.
10. Rieffel E. and Polak W. An introduction to quantum computing for non-physicists // ACM Computing Surveys. – 2000. – Vol. 32. – No 3. – pp. 300 – 335.
11. Hogg T., Mochon C., Polak W. and Rieffel E. Tools for quantum algorithms // International Journal of Modern Physics. – 1999. – Vol. C10. – No 7. – pp. 1347 – 1361.
12. Uesaka Y. Mathematical principle of quantum computation (in Japanese). – Corona Publ. Co. Ltd. – 2000.
13. Marinescu D.C. and Marinescu G.M. Approaching quantum computing. – Pearson Prentice Hall, New Jersey. – 2005.
14. Benenti G., Casati G. and Strini G. Principles of quantum computation and information. –Singapore: World Scientific. – Vol. I. – 2004; – Vol. II. – 2007.
15. Nakahara M. and Ohmi T. Quantum computing: From Linear Algebra to Physical Realizations. – Taylor & Francis. – 2008.
16. Stenholm S. and Suominen K.-A. Quantum approach to informatics. – Wiley- Interscience. A J. Wiley&Sons, Inc. – 2005.
17. Jaeger G. Quantum Information: An Overview. – N.Y.: Springer Verlag. – 2007.
18. McMahon D. Quantum computing explained. – Wiley- Interscience. A J. Wiley&Sons, Inc. – 2008.