Turk J Phys
27 (2003) , 343 – 370.
© TÜBİTAK

QCE: A Simulator for Quantum Computer Hardware[*]

Kristel MICHIELSEN, Hans De RAEDT
Department of Applied Physics-Computational Physics,
Materials Science Centre, University of Groningen,
Nijenborgh 4, NL-9747 AG Groningen, The Netherlands
kristel@phys.rug.nl and deraedt@phys.rug.nl
http://www.compphys.org

Received 28.08.2003

Abstract

The Quantum Computer Emulator (QCE) described in this paper consists of a simulator of a generic, general purpose quantum computer and a graphical user interface. The latter is used to control the simulator, to define the hardware of the quantum computer and to debug and execute quantum algorithms. QCE runs in a Windows 98/NT/2000/ME/XP environment. It can be used to validate designs of physically realizable quantum processors and as an interactive educational tool to learn about quantum computers and quantum algorithms. A detailed exposition is given of the implementation of the CNOT and the Toffoli gate, the quantum Fourier transform, Grover's database search algorithm, an order finding algorithm, Shor's algorithm, a three-input adder and a number partitioning algorithm. We also review the results of simulations of an NMR-like quantum computer.

Key Words: Quantum computation, computer simulation, educational software

1. Introduction

Recent advances in quantum information processing has opened new avenues for using quantum phenomena to perform computation [1]. Quantum computers (QCs) may solve problems such as factoring integers [2, 3] and searching databases [4, 5] faster than a conventional computer. The potential power of a QC stems from the fact that a quantum system can be in a superposition of states, allowing exponentially many computations to be done in parallel [1, 6, 7, 8, 9, 10]. Candidate technologies for building quantum gates include ion traps, cavity QED, Josephson junctions, and NMR technology [1, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36].

Just as simulation is an integral part of the design process of each new generation of microprocessors, software to emulate physical models of quantum processors may prove essential. In contrast to conventional digital circuits where the internal working of each basic unit is irrelevant for the logical operation of the whole machine, the internal quantum dynamics of each elementary gate is a key ingredient of the QC itself. Therefore it is essential to incorporate into a simulation model, the physics of the elementary units that make up the QC.

Theoretical work on quantum computation usually assumes the existence of units that perform highly idealized unitary operations. However, in practice these operations are difficult to realize: Disregarding decoherence, physical QCs will perform unitary operations that are more complicated than those considered

[*]Lectures given by K. Michielsen and H. De Raedt at the Summer School *Quantum Computation at the Atomic Scale 2003*, Istanbul, Turkey.

in most theoretical work. Therefore it is important to have theoretical tools to validate designs of physically realizable quantum processors.

The Quantum Computer Emulator (QCE) described in this paper, is software designed to simulate physical models of QC hardware. The QCE consists of a simulator of a generic, general purpose QC and a graphical user interface [37, 38]. The simulator simulates the physical processes that govern the operation of the hardware quantum processor, strictly according to the laws of quantum mechanics. The graphical user interface is used to control the simulator, to define the physical realization of the QC and to debug and execute quantum algorithms (QAs). The QCE runs in a Windows 98/NT/2000/ME/XP environment [38]. It can be used as a research tool to validate designs of physically realizable quantum processors and as an educational tool to learn about QCs and QAs in an interactive way. In the current version of QCE (8.1.0) the maximum number of qubits is sixteen.

This paper contains a first detailed account of how the QCE can be used to simulate ideal and physical QCs. A description of the internal working of the simulator itself can be found elsewhere [37, 39]. We discuss the implementation of basic QC gates (CNOT and Toffoli gate) and more complicated applications such as Grover's search algorithm and the quantum Fourier transform on an ideal QC. We also present the implementation of algorithms to find the order of a permutation, Shor's factoring algorithm, a three-input adder and an algorithm to solve the number partitioning problem on an ideal QC. We briefly review some of our earlier work on the simulation of a physical NMR-like model of a QC [40].

2. Quantum Computers

2.1. Qubits

In a QC, the basic unit of information is a quantum bit or qubit. Whereas bits can take the logical values 0 and 1, a qubit can be in any linear superposition of the two states $|0\rangle$ and $|1\rangle$

$$Q = \alpha|0\rangle + \beta|1\rangle \quad ; \quad \alpha, \beta \in C, \tag{1}$$

where $|\alpha|^2$ and $|\beta|^2$ denote the frequencies of occurrence of both states $|0\rangle$ and $|1\rangle$, respectively. Hence, in one qubit an infinite amount of classical information can be encoded. The principles of quantum mechanics, however, do not allow to retrieve this information. If a measurement is performed on the qubit, the state $|0\rangle$ ($|1\rangle$) is measured with a frequency which is proportional to $|\alpha|^2$ ($|\beta|^2$). What is then so special about a QC? Consider a classical memory register consisting of three bits. At a certain point in time this register can store one out of $2^3 = 8$ possible numbers, namely in digital notation 000, or 001, or 010, ... or 111. A quantum register with three qubits, however, can store all eight possible numbers in a superposition at a certain point in time. If the quantum register is in the superposition of the eight states, operations can be performed on the eight numbers simultaneously. This is called quantum parallelism. On a classical computer, the same operation needs to be performed eight times or the operation has to be done on eight processors of a parallel computer. The QC is thus a massive parallel computer.

The qubits on an ideal QC are ideal two-state quantum systems. Therefore, the operation of an ideal QC does not depend on the intrinsic dynamics of its qubits. A physically realizable QC, on the other hand, is a many-body system in which the dynamics of the qubits is essential to its operation. In what follows we represent the qubit by a spin-1/2 system. The two basis states spanning the Hilbert space are denoted by $\langle\uparrow| \equiv (10) \equiv \langle 0|$ and $\langle\downarrow| \equiv (01) \equiv \langle 1|$.

2.2. Models

QC hardware can be modeled in terms of qubits that evolve in time according to the time-dependent Schrödinger equation (TDSE)

$$i\frac{\partial}{\partial t}|\Phi(t)\rangle = H(t)|\Phi(t)\rangle, \tag{2}$$

in units such that $\hbar = 1$. For pedagogical reasons we consider a three-qubit QC. The state

$$|\Phi(t)\rangle = a(\downarrow,\downarrow,\downarrow;t)|\downarrow,\downarrow,\downarrow\rangle + a(\downarrow,\downarrow,\uparrow;t)|\downarrow,\downarrow,\uparrow\rangle + \ldots + a(\uparrow,\uparrow,\uparrow;t)|\uparrow,\uparrow,\uparrow\rangle, \tag{3}$$

describes the state of the whole QC at time t. The complex coefficients $a(\downarrow,\downarrow,\downarrow;t),\ldots,a(\uparrow,\uparrow,\uparrow;t)$ completely specify the state of the quantum system. For the time-dependent Hamiltonian we take

$$H(t) = -J_{12}^z(t)S_1^zS_2^z - J_{13}^z(t)S_1^zS_3^z - J_{23}^z(t)S_2^zS_3^z - \sum_{j=1}^{3}\sum_{\alpha=x,y,z} H_j^\alpha(t)S_j^\alpha, \tag{4}$$

where S_j^α denotes the α-th component of the spin-1/2 operator representing the j-th qubit, $J_{jk}^\alpha(t)$ determines the strength of the interaction between the spins labeled j and k, and $H_j^\alpha(t)$ represents the external field acting on the j-th spin. In terms of spin matrices, the expectation values Q_j^α of the three components of qubit j are given by

$$Q_j^\alpha = \frac{1}{2} - \langle S_j^\alpha\rangle. \tag{5}$$

The physical system defined by (4) is sufficiently general to serve as a physical model for a generic (three-qubit) QC at zero temperature without coupling to other degrees of freedom (e.g. interactions with the environment). For instance, it includes the simplest (Ising) model of a universal QC [7, 41]

$$H_{ideal} = -J(S_1^zS_2^z + S_1^zS_3^z + S_2^zS_3^z) - \sum_{j=1}^{3}\sum_{\alpha=x,y,z} H_j^\alpha(t)S_j^\alpha. \tag{6}$$

Specific candidate hardware realizations of (4) include linear arrays of quantum dots [29], Josephson junctions [21] and NMR systems [15, 16, 17, 18, 31, 32, 33, 34]. An approximate model for the linear arrays of quantum dots reads

$$H(t) = -\sum_{j=1}^{2} E_jS_j^zS_{j+1}^z - \sum_{j=1}^{3} H_j^x(t)S_j^x + E_0\sum_{j=1}^{3} P_j(t)S_j^z, \tag{7}$$

where $E_j = E_0$ ($E_j = 2E_0$) when j is odd (even) and $H_j^x(t)$ and $P_j(t)$ are external control parameters [29].

Projection of the Josephson-junction model onto a subspace of two states per qubit yields [20, 25]

$$H(t) = -2E_I(t)\sum_{j=1}^{2} S_j^yS_{j+1}^y - E_J\sum_{j=1}^{3} S_j^x - \sum_{j=1}^{3} H_j^z(t)S_j^z, \tag{8}$$

where the energy of the Josephson tunneling is represented by E_J and $E_I(t)$ denotes the energy associated with the inductive coupling between the qubits [20, 25]. Here $H_j^z(t)$ and $E_I(t)$ may be controlled externally.

The model Hamiltonian for a three-qubit NMR-QC is given by (4) whith

$$H_j^\alpha(t) = h_j^\alpha + \widetilde{h}_j^\alpha \sin(2\pi f_j^\alpha t + \varphi_j^\alpha), \tag{9}$$

where h_j^α and $\widetilde{h}_j^\alpha$ represent the static (magnetic) and periodic (RF) field acting on the j-th spin respectively. The frequency and phase of the periodic field are denoted by f_j^α and φ_j^α. The simple choice for the time dependence of the pulses is convenient for theoretical work. In practice, NMR experiments use much more complicated pulses of electromagnetic radiation [42, 43].

3. QCE: Quantum Computer Emulator

The model parameters in the above Hamiltonians are determined by the specific experimental systems that are simulated. The present version of QCE (8.1.0) can simulate models that are described by the Hamiltonian

$$H(t) = -\sum_{i\neq j}\sum_{\alpha=x,y,z} J_{ij}^\alpha(t)S_i^\alpha S_j^\alpha - \sum_{j}\sum_{\alpha=x,y,z} H_j^\alpha(t)S_j^\alpha, \tag{10}$$

with $H_j^\alpha(t)$ given by Eq.(9). This Hamiltonian is sufficiently general to include e.g. models for NMR-like QCs.

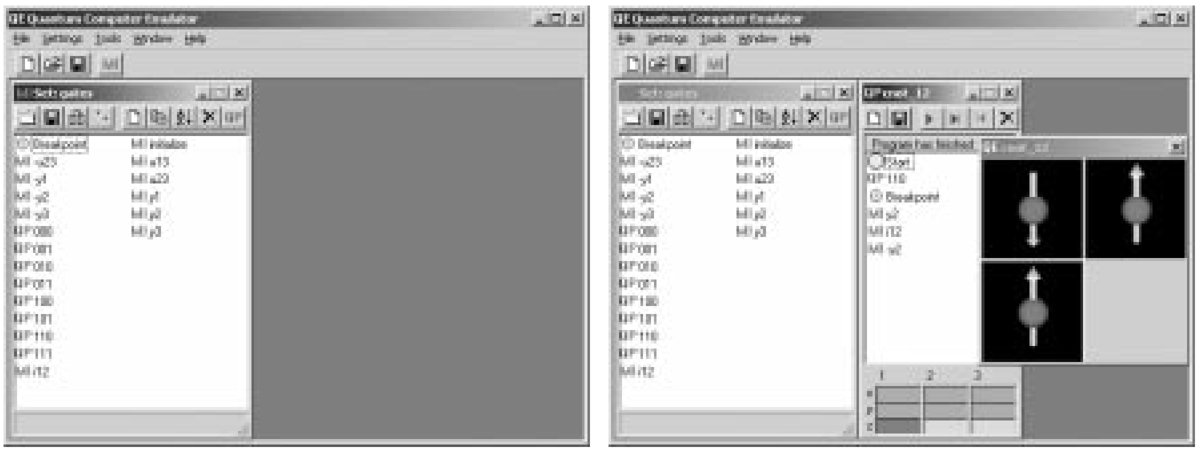

QCE Quantum Computer Emulator
File Settings Tools Window Help
Breakpoint
MI initialize
QP110
Program has finished
Start
1 2 3

Figure 1. Left: The main QCE window with a set of micro instructions implementing an ideal three-qubit quantum computer. Right: Set of micro instructions implementing an ideal three-qubit quantum computer and a window with the quantum program for the CNOT gate operating on the $|110\rangle$ state. The qubits in the final state of the quantum computer (i.e. the expectation value of the spin operators), are shown at the bottom of the program window (dark gray (red on screen) = $|1\rangle$, light gray (green on screen) = $|0\rangle$). QCE has an option to visualize the time evolution of the state of the quantum computer in terms of arrows representing the expectation values of the qubits (see menu item "Settings" on the top bar of the main window).

3.1. Quantum algorithms

A QA for QC model (10) consists of a sequence of elementary operations that change the state $|\Phi\rangle$ of the quantum processor according to the TDSE, i.e. by (a product of) unitary tranformations. We will call these elementary operations micro instructions (MIs) in the sequel. They are not exactly playing the same role as MIs do in digital processors. They merely represent the smallest units of operation of the quantum processor. The action of a MI on the state $|\Phi\rangle$ of the quantum processor is defined by specifying how long it acts (i.e. the time interval it is active), and the values of all the J's and h's appearing in (10). A MI transforms the input state $|\Phi(t)\rangle$ into the output state $|\Phi(t+\tau)\rangle$ where τ denotes the time interval during which the MI is active. During this time interval the only time-dependence of $H(t)$ is through the time-dependence of the external fields on the spins.

The QCE solves TDSE (2) by a Suzuki product-formula [37, 44, 45] in terms of elementary unitary operations. For all practical purposes, the results obtained by this technique are indistinguishable from the exact solution of the TDSE.

3.2. Graphical user interface

The graphical user interface (GUI) of QCE has been developed to facilitate the specification of the MIs (to model the QC hardware) and the execution of the quantum programs (QPs). The QCE runs in a Windows 98/NT/2000/ME/XP environment. Using the GUI requires no skills other than the basic ones needed to run a standard MS-Windows application. The QCE is freely distributed as a self-installing executable, containing the program, documentation, and various QPs [38]. The file help.htm [46] contains information about how to install and how to start the QCE. The QCE starts up in full screen mode and requires the choice of a MI set. Any of the listed MI sets can be chosen. It can be modified or renamed afterwards.

The main window of the QCE (see Fig. 1) contains a window showing the set of MIs that is currently active. Tooltips appear when the mouse moves over the buttons. The top bar of the main window contains a menu item "File" and buttons to create, open and save QPs and menu items to control ("Tools", "Settings", "Window") QCE. The menu item "Settings" offers the option to turn on the visualization of the time evolution of the state of the QC and to generate text files with the numerical results for further processing. Note that the window, showing the arrows representing the qubits (see Fig. 1), appears for the first time when a QP is started. The name of the window corresponds to the QP that is executed. QP windows within the main window can be rearranged using the menu item "Window" or by pressing Ctrl-A. The button labeled "MI" can be used to create a completely new MI set.

The MI set defines the hardware realization of a QC. Each MI set has two reserved MIs: "Breakpoint" to allow a QP to pause at a specified point and "initialize" to specify the initial state of the QC (normally the first MI in a QP). The top bar of the MI set window contains buttons to open, save and copy an MI set, to create, copy and delete a MI, to sort MIs and to add QPs. The number of qubits of the QC can be specified by clicking the button showing two qubits on the top bar of the MI set window.

The QCE supports the (non-recursive) use of QPs as MIs (see Fig. 1). QPs can be added to a particular MI set through the button labeled "QP" (top right of the MI set window). During execution, a QP that is called from another QP will call either another QP or a genuine MI from the currently loaded set of MIs. The QCE will skip all initialization MIs except for the first one. This facilitates the testing of QPs that are used as sub-QPs. A QP calling a MI or QP that cannot be found in the current MI set will generate an error message and force termination of the execution.

Writing a QA on the QCE from scratch is a two-step process. First one has to specify the MIs, taking into account the particular physical realization of the QC that one wants to emulate. A new MI set is created by clicking on the "MI" button at the top of the main window of the QCE. The MI set window offers all necessary tools to edit and manipulate the MIs. New MIs in the MI set window are obtained by clicking on the "white page" button at the top of this window. The parameters specifying the MI can then be filled out. Existing MIs can be edited by double-clicking the MI icon.

The second step, writing a QP, consists of dragging and dropping MIs onto a QP window. New QP windows are created by clicking on the "white page" button at the top of the main window of the QCE, old QPs are opened by clicking on the "folder" button at the top of the main window of the QCE or by double-clicking on a QP in a MI set window.

Execution of a QP on the QCE is realized by the control buttons at the top of the QP program. The results of executing a QP appear in color-coded form at the bottom of the corresponding QP window. For each qubit the expectation value of the three spin components (see Eq.(5)) are shown: green (light gray) corresponds to 0, red (dark gray) to 1. Usually one row of values (the z-component) will be of interest. Note that in general only QPs can be executed that contain MIs and QPs that are defined in the active MI set.

4. Ideal Quantum Computer

This section discusses the use of QCE as an ideal QC.

4.1. Single-qubit operations

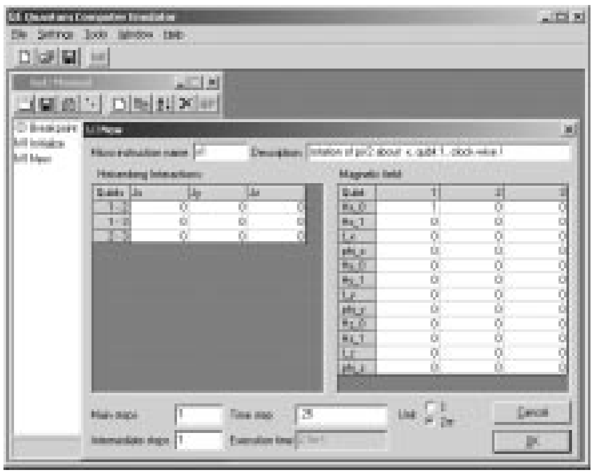
The three components of the spin-1/2 operator $\mathbf{S}$ acting on the Hilbert space spanned by the states $|0\rangle$ and $|1\rangle$ are defined by

$$S^x = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad S^y = \frac{1}{2}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad S^z = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{11}$$

in units such that $\hbar = 1$. By convention this representation is chosen such that $|0\rangle$ and $|1\rangle$ are eigenstates of S^z with eigenvalues +1/2 and -1/2, respectively. In general a rotation of spin j about an angle ϕ around an axis β can be written as ($\hbar = 1$)

$$S^\alpha(\phi,\beta) = e^{i\phi S^\beta} S^\alpha e^{-i\phi S^\beta} = S^\alpha \cos\phi + \epsilon_{\alpha\beta\gamma} S^\gamma \sin\phi, \tag{12}$$

where use has been made of $[S^\alpha, S^\beta] = i\epsilon_{\alpha\beta\gamma}S^\gamma$, $\epsilon_{\alpha\beta\gamma}$ is the totally asymmetric unit tensor ($\epsilon_{xyz} = \epsilon_{yzx} = \epsilon_{zxy} = 1$, $\epsilon_{\alpha\beta\gamma} = -\epsilon_{\beta\alpha\gamma} = -\epsilon_{\gamma\beta\alpha} = -\epsilon_{\alpha\gamma\beta}$, $\epsilon_{\alpha\alpha\gamma} = 0$) and the summation convention is used. Rotations of



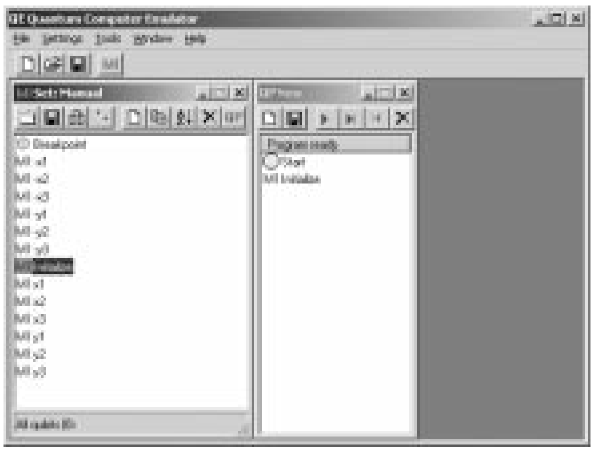


Figure 2. Left: MI set window "Manual" together with the MI window "New" which is edited to become MI "X1". Right: MI set window "Manual" together with the QP window "New" which is edited to become QP "000".

spin j about $\pi/2$ around the x and y-axis are often used as basic QC operations. In matrix notation, they are given by

$$X_j \equiv e^{i\pi S_j^x/2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}, \quad Y_j \equiv e^{i\pi S_j^y/2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}, \tag{13}$$

respectively. X_j and Y_j represent operations on single qubits. The matrix expressions for the inverse of the rotations X_j and Y_j, denoted by $\overline{X}_j$ and $\overline{Y}_j$ respectively, are obtained by taking the Hermitian conjugates of the matrices in (13). With our convention $\langle 0|\overline{Y}_j S_j^x Y_j|0\rangle = -1/2$ so that a positive angle corresponds to a rotation in the clockwise direction. For a two-qubit QC we have

$$X_1 \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & i & 0 \\ 0 & 1 & 0 & i \\ i & 0 & 1 & 0 \\ 0 & i & 0 & 1 \end{pmatrix}, \tag{14}$$

where $|b_1 b_2\rangle \equiv |b_1\rangle|b_2\rangle$ and $b_i = 0, 1$. For example $X_1|10\rangle = (|10\rangle + i|00\rangle)/\sqrt{2}$ and $\overline{X}_1|10\rangle = (|10\rangle - i|00\rangle)/\sqrt{2}$. Using the same labeling of states as in (14) we have

$$Y_2 \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix}, \tag{15}$$

e.g. $Y_2|10\rangle = (|10\rangle - |11\rangle)/\sqrt{2}$ and $\overline{Y}_2|10\rangle = (|10\rangle + |11\rangle)/\sqrt{2}$.

The rotations X_j, Y_j and $\overline{X}_j$, $\overline{Y}_j$ can be implemented as MIs on the QCE. For example, on a three-qubit QC, this can be done as follows: Start up the QCE and choose one of the MI sets. Create a new MI set by clicking on the button "MI" on the toolbar of the main window and name the new MI set "Manual". Click on the button with the two qubits (at the top of the MI set window) and choose the number of qubits to be equal to three. Create a new MI by clicking the "white page" button at the top of the MI set window. An MI window with the name New opens. Give the MI the name X1 and optionally type the text "rotation of pi/2 about x, qubit 1, clock-wise !" in the description field. The parameters need to be chosen such that $-i\tau H_{ideal} = i\pi S_1^x/2$, where H_{ideal} is given by (6). In practice we put $h_1^x = 1$ (field (Hx_0,1) in Fig. 2),

Table 1. Input and output states and the corresponding expectation values (Q_1^z,Q_2^z) of the qubits for the CNOT operation.

Input state	Q_1^z	Q_2^z	Output state	Q_1^z	Q_2^z
$\|00\rangle$	0	0	$\|00\rangle$	0	0
$\|01\rangle$	0	1	$\|01\rangle$	0	1
$\|10\rangle$	1	0	$\|11\rangle$	1	1
$\|11\rangle$	1	1	$\|10\rangle$	1	0

$\tau/2\pi = 1/4$ ("Time step" in Fig. 2), "Main steps" = "Intermediate steps = 1" and leave all other parameters equal to zero (see Fig. 2). The execution time τ of a MI is determined by four parameters. There is an option to select the unit of time: Here it is convenient to express time in units of 2π. The total execution time is given by (Main steps)*(Intermediate steps)*(Time step). "Main steps" controls the pace at which (visualization) output is generated. Taking a too large value of "Main steps" may have negative impact on the real-time performance of the QCE. The MIs X_j for $j = 2, 3$ can be copied from X_1 by using the button with the overlaying pages. MIs Y_j and $\overline{X}_j$, $\overline{Y}_j$ can be constructed in a similar way. MIs can be deleted by first selecting them and by then clicking on the delete button (cross) on the toolbar of the MI set window.

4.1.1. Hadamard gate

The Hadamard gate, a single-qubit gate, is one of the most useful quantum gates. The Hadamard operation rotates the state $|0\rangle$ into $(|0\rangle + |1\rangle)/\sqrt{2}$ (up to an irrelevant phase factor), i.e. the uniform superposition state. In terms of the elementary rotations X and Y the Hadamard operation on qubit j reads

$$W_j = X_j^2\overline{Y}_j = Y_jX_j^2 = -\overline{X}_j^2\overline{Y}_j = -Y_j\overline{X}_j^2 = \frac{i}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \qquad (16)$$

For example

$$W_j|0\rangle = W_j\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{i}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}. \qquad (17)$$

The Hadamard operation can be generalized to any number of qubits [1]. The generalized operation is known as the Hadamard transform or as the Walsh-Hadamard transform. This operation consists of N Hadamard gates acting in parallel on N qubits. For example for $N = 2$, $W_2W_1|00\rangle = -(|00\rangle + |01\rangle + |10\rangle + |11\rangle)/2$. The Walsh-Hadamard transform produces a uniform superposition of all basis states.

4.2. Communication between two qubits

Computation requires some form of communication between the qubits. Two qubits j and k "communicate" with each other through the controlled phase shift

$$I_{jk} \equiv \begin{pmatrix} e^{i\phi_{00}} & 0 & 0 & 0 \\ 0 & e^{i\phi_{01}} & 0 & 0 \\ 0 & 0 & e^{i\phi_{10}} & 0 \\ 0 & 0 & 0 & e^{i\phi_{11}} \end{pmatrix}. \qquad (18)$$

4.3. Two-qubit operations: CNOT gate

A fundamental two-qubit operation is provided by the CNOT gate. The CNOT gate flips the second qubit if the state of the first qubit is $|1\rangle$, i.e. the first qubit acts as a control qubit for the second one, see Table 1. An arbitrary QA can be written as a combination of the CNOT function and single-qubit operations (universal gates) [1]. On an ideal QC the CNOT gate can be implemented by a combination of single-qubit operations and the controlled phase shift.

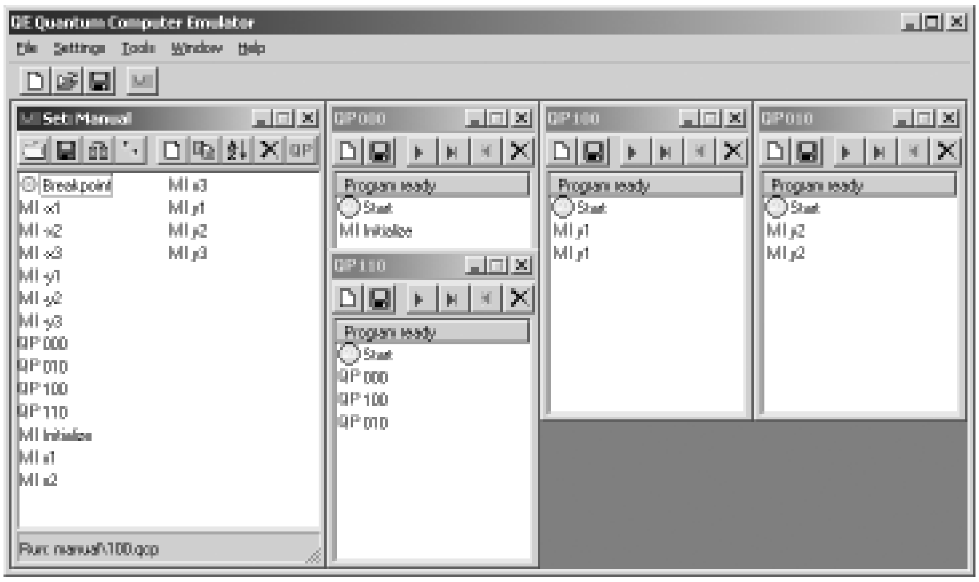


Figure 3. MI set window "Manual" together with the QP windows "000", "100", "010" and "110".

From Table 1 it can be seen that the CNOT gate can be written in matrix notation as

$$\mathrm{CNOT} \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} . \tag{19}$$

In order to bring the above matrix into a similar form as the controlled phase shift, the matrix needs to diagonalized. This may be done as follows

$$\mathrm{CNOT} = \begin{pmatrix} I_1 & 0 \\ 0 & 2S_2^x \end{pmatrix} = \begin{pmatrix} I_1 & 0 \\ 0 & 2\overline{Y}_2 S_2^z Y_2 \end{pmatrix} = \overline{Y}_2 \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} Y_2 = e^{-i\alpha} \overline{Y}_2 I_{12} Y_2 , \tag{20}$$

where I_1 denotes the 2×2 identity matrix for qubit 1, and $e^{-i\alpha}$ is a global phase factor. From Eq.(18) and Eq.(20) it follows that

$$I_{12} = \begin{pmatrix} e^{i\phi_{00}} & 0 & 0 & 0 \\ 0 & e^{i\phi_{01}} & 0 & 0 \\ 0 & 0 & e^{i\phi_{10}} & 0 \\ 0 & 0 & 0 & e^{i\phi_{11}} \end{pmatrix} = e^{i\alpha} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} , \tag{21}$$

so that $\phi_{00} = \phi_{01} = \phi_{10} = \alpha$ and $\phi_{11} = \alpha - \pi$.

We can implement the controlled phase shift, $I_{12} = e^{-i\tau H}$, by means of the Ising model

$$H = -J S_1^z S_2^z - h(S_1^z + S_2^z). \tag{22}$$

Then the values for ϕ_{jk} are $\phi_{00} = \tau(J/4 + h)$, $\phi_{01} = \phi_{10} = -\tau J/4$ and $\phi_{11} = \tau(J/4 - h)$. With these values for ϕ_{jk} and the condition $\phi_{00} = \phi_{01} = \phi_{10} = \alpha$, $\phi_{11} = \alpha - \pi$, we find that $h = -J/2$ and $J\tau = -\pi$. Hence the CNOT gate can be implemented as $\overline{Y}_2 I_{12} Y_2$ with the conditions $h = -J/2$ and $\tau J = -\pi$.

We now describe the main steps of how to program an ideal three-qubit QC so that it gives the outcome of a CNOT gate operating on the $|110\rangle$ state. The final result can be found in the MI set "gates" and the

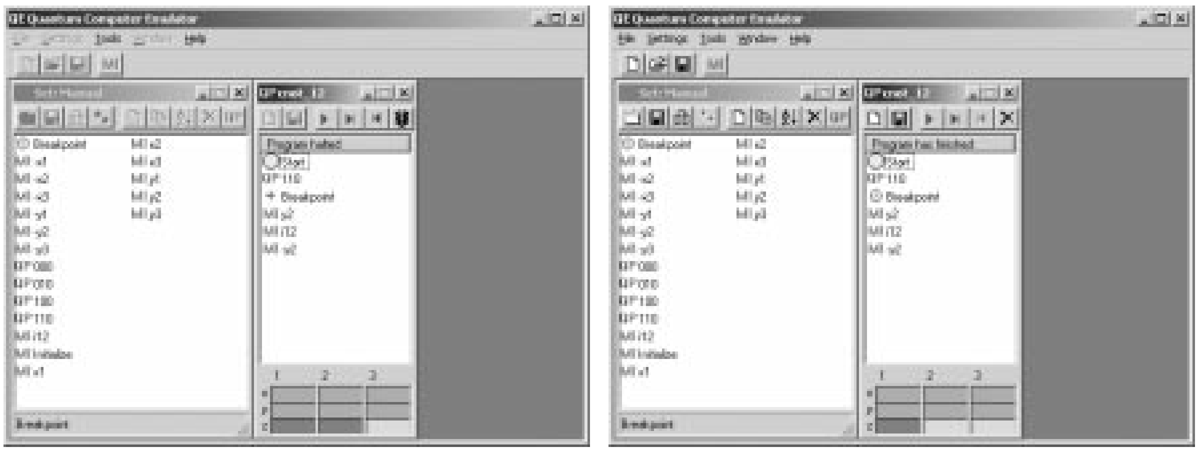


Figure 4. Left: MI set window "Manual" together with the QP windows "cnot_12" operating on the state $|110\rangle$. The QP "cnot_12" stops at the MI "Breakpoint". The three components of the three qubits are shown at the bottom of the QP window: $Q_1^x = Q_2^x = Q_3^x = Q_1^y = Q_2^y = Q_3^y = 0.5$ (greenisch brown or middle gray), $Q_1^z = Q_2^z = 1$ (dark gray or red) and $Q_3^z = 0$ (green or light gray), corresponding to the state $|110\rangle$. Right: MI set window "Manual" together with the QP window "cnot_12" operating on the state $|110\rangle$. The three components of the three qubits are shown at the bottom of the QP window: $Q_1^x = Q_2^x = Q_3^x = Q_1^y = Q_2^y = Q_3^y = 0.5$, $Q_1^z = 1$ and $Q_2^z = Q_3^z = 0$, corresponding to the state $|100\rangle$, the correct final state of the quantum computer.

QP "cnot_12" (see Fig. 1), included in the QCE software distribution. We start from the MI set "Manual" built in section 4.1. First the QC has to be initialized (setting each of the three qubits to $|0\rangle$) and then the first and second qubit have to be put in the state $|1\rangle$. We accomplish this on the QCE by creating the QP "110" which consists of the QPs "000", "100" and "010". We first make QP "000", which only contains the MI "initialize", a reserved MI in the QCE (see section 3.2), and puts the QC in the state $|000\rangle$. Starting from the MI set "Manual" open a new QP by clicking on the "white page" button or the menu item "File" on the top bar of the main window. A QP window named New opens. Drag the MI "initialize" from the MI set window "Manual" and drop it on the QP window (see right-hand side of Fig. 2). Save the QP by clicking on the save button (diskette) on the top bar of the QP window or by clicking the menu item "Tools" on the top bar of the main window and choose "Save program" or "Save program as". The "Save program as" window opens showing the QCE directory. If the subdirectory "Manual" does not yet exist, first create it using the standard Windows procedures and save the QP as "000.qcp" in "Manual". Add QP "000" to the MI set "Manual" by clicking on the "QP" button on the top bar of the MI set window "Manual". The "Add Quantum Program" opens, showing the subdirectory "Manual". Select "000.qcp" and click on open or press enter. Using the same procedure as described above we create QP "100" and QP "010". QP "100" (QP "010") consists of two MI's "Y_1" ("Y_2") and rotates clock-wise qubit 1 (qubit 2) about π around the y-axis. Finally we create QP "110" by a similar procedure but now we drag the QPs "000", "100" and "010" from the MI set window Manual on a new QP program window (see Fig. 3). The instructions in the program window can be rearranged by clicking on the instruction that you want to move and dragging it to the instruction below which you want to place it. The instruction is dropped below the instruction that becomes high-lighted during this movement. Instructions can be deleted by first selecting them and by then clicking on the delete button in the toolbar of the QP window. QP "110" puts the QC in the state $|110\rangle$, the starting position for our particular example. The QPs can now be closed by clicking on the cross in the top right corner of the QP windows.

The next task is to implement the CNOT gate itself. Create a new QP and call it "cnot_12". This QP needs to compute $\overline{Y}_2 I_{12} Y_2$ on the state $|110\rangle$. Since operations need to be read from right to left we first put the QC in the state $|110\rangle$ by dragging the QP "110" on the QP "cnot_12" window. For debugging reasons we drag MI "Breakpoint" on the QP window. This allows us to check whether we start from the correct input state. Second we rotate clock-wise qubit 2 about $\pi/2$ around the y-axis by means of MI "Y_2". This brings the QC in the state $(|100\rangle + |110\rangle)/\sqrt{2}$. Then we apply the controlled phase shift I_{12}. We construct MI "i_{12}" by taking the parameters such that $h = -J/2$ and $J\tau = -\pi$, i.e. $J^z_{12} = -1$, $h^z_1 = h^z_2 = 1/2$, $\tau/2\pi = 1/2$, "Main steps" = "Intermediate steps" = 1 and leaving all other model parameters equal to zero. Apart from some global phase factor this brings the QC in the state $(|100\rangle - |110\rangle)/\sqrt{2}$. Finally operating with $\overline{Y}_2$ on this state results in the final state $|100\rangle$ of the QC.

Run (step by step) the QP "cnot_12" by clicking on the corresponding play button in the QP window (or press Alt-F5). The program stops at the breakpoint. This is indicated by the arrow in front of the MI "Breakpoint". The values of the qubits appear when the mouse is moved over the bottom region of the QP window: $Q^x_1 = Q^x_2 = Q^x_3 = Q^y_1 = Q^y_2 = Q^y_3 = 0.5$ (greenisch brown or middle gray), $Q^z_1 = Q^z_2 = 1$ (dark gray or red) and $Q^z_3 = 0$ (green or light gray), corresponding to the state $|110\rangle$ (see Fig. 4). We can now stop the program by clicking at the stoplight button, continue or step back by clicking the appropriate buttons. We continue by clicking on the play button. The program finishes and the expectation values of the three qubit components become: $Q^x_1 = Q^x_2 = Q^x_3 = Q^y_1 = Q^y_2 = Q^y_3 = 0.5$, $Q^z_1 = 1$ and $Q^z_2 = Q^z_3 = 0$, corresponding to the state $|100\rangle$, the correct final state of the QC (see Fig. 4).

4.3.1. Operation of CNOT gate on singlet state

In the previous section we have demonstrated the operation of the CNOT gate on the state $|110\rangle$, a simple basis state. In what follows we demonstrate how the CNOT gate operates on an entangled state such as the singlet state, $|singlet\rangle = (|01\rangle - |10\rangle)/\sqrt{2}$.

We first have to bring the QC in the singlet state. Assume that the QC is in a state $|00\rangle$. Applying the rotation operator Y_1 yields $(|00\rangle - |10\rangle)/\sqrt{2}$. Next we apply to $Y_1|00\rangle$ the rotation operator $\overline{Y}_2$. This gives $(|00\rangle + |01\rangle - |10\rangle - |11\rangle)/2$. Application of the controlled phase shift I_{12}, as defined in Eq.(21) with $\alpha = 2\pi$ gives $I_{12}\overline{Y}_2 Y_1|00\rangle = (|00\rangle + |01\rangle - |10\rangle + |11\rangle)/2$. Finally, we again apply $\overline{Y}_2$ and find

$$\overline{Y}_2 I_{12}\overline{Y}_2 Y_1|00\rangle = (|01\rangle - |10\rangle)/\sqrt{2} \equiv |singlet\rangle. \tag{23}$$

Applying CNOT to a singlet state gives

$$\text{CNOT}|singlet\rangle = (|01\rangle - |11\rangle)/\sqrt{2}, \tag{24}$$

and hence $\langle singlet|\text{CNOT}\ Q^z_1\ CNOT|singlet\rangle = 1/2$. We can obtain a clear-cut answer in terms of expectation values of both qubits by applying additionally Y_1

$$Y_1\text{CNOT}|singlet\rangle = -|11\rangle. \tag{25}$$

Then, in the final state, $Q^z_1 = Q^z_2 = 1$.

On the QCE we first build the QPs "singlet" and "cnot_singlet" for qubits 1 and 2 and add them to the MI set "Manual" using the procedure described above (see Fig. 5). Run the QP "cnot_singlet". The program stops at the two breakpoints. After the first breakpoint the three-qubit QC is in the state $(|010\rangle - |100\rangle)/\sqrt{2}$, i.e. qubits 1 and 2 form a singlet and qubit 3 is in the state $|0\rangle$. We have: $Q^x_1 = Q^x_2 = Q^x_3 = Q^y_1 = Q^y_2 = Q^y_3 = 0.5$, $Q^z_1 = Q^z_2 = 0.5$ and $Q^z_3 = 0$. After the second breakpoint the CNOT gate on qubits 1 and 2 has been applied, resulting in the following values for the qubits: $Q^x_1 = 1$, $Q^y_1 = Q^z_1 = 0.5$, $Q^x_2 = Q^y_2 = 0.5$, $Q^z_2 = 1$ and $Q^x_3 = Q^y_3 = 0.5$, $Q^z_3 = 0$. At the end of the program the QC is in the state $-|110\rangle$ and $Q^x_1 = Q^x_2 = Q^x_3 = Q^y_1 = Q^y_2 = Q^y_3 = 0.5$, $Q^z_1 = Q^z_2 = 1$ and $Q^z_3 = 0$ (see Fig. 5).

4.4. Quantum Fourier Transform

The Quantum Fourier Transform (QFT) is an essential subroutine in e.g. Shor's factoring algorithm, the order finding algorithm and phase estimation in general [1, 9]. The QFT is not a new kind of Fourier transform: it is a particular application of the standard discrete Fourier transform. The latter is defined by $\mathbf{x} = U\mathbf{y}$ where $\mathbf{x} = (x_0, \ldots, x_{N-1})$, $\mathbf{y} = (y_0, \ldots, y_{N-1})$ and $U_{j,k} = e^{2\pi ijk/N}/\sqrt{N}$. Clearly U is a unitary matrix. In the context of quantum computation, the vectors $\mathbf{x}$ and $\mathbf{y}$ are just two different representations

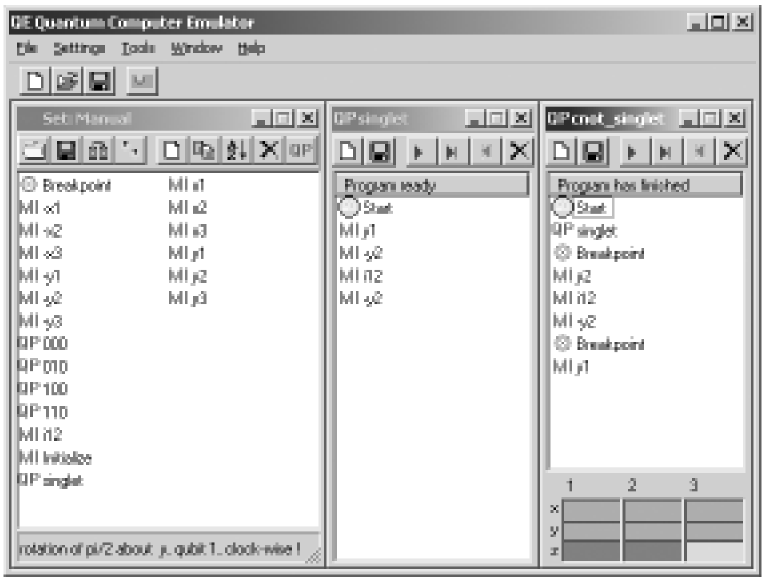


Figure 5. MI set "Manual" together with the QPs "singlet" and "cnot_singlet" operating on the state $|singlet\rangle$. The expectation values of the three qubit components are shown at the bottom of the QP window: $Q_1^x = Q_2^x = Q_3^x = Q_1^y = Q_2^y = Q_3^y = 0.5$, $Q_1^z = Q_2^z = 1$ and $Q_3^z = 0$ corresponding to the state $\pm|110\rangle$, the final state of the quantum computer.

of the basis vectors that span the Hilbert space of dimension N. The QFT relates the amplitudes in these two representations:

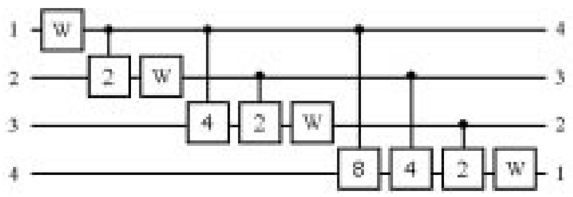
$$\sum_{j=0}^{N-1} a_j|x_j\rangle = \frac{1}{\sqrt{N}} \sum_{j,k=0}^{N-1} e^{2\pi ijk/N} a_j|y_k\rangle = \sum_{k=0}^{N-1} b_j|y_k\rangle. \tag{26}$$

Readers more familiar with traditional quantum mechanics recognize the similarity with the coordinate and momentum representation. For applications in QAs, it is expedient to label the states not by one integer but by the binary representation in terms of the states of the individual qubits. Then $N = 2^n$ where n is the number of qubits. Exploiting the massive parallelism of the ideal QC, the QFT can be carried out in $\mathcal{O}(n)$ quantum operations [1, 9].

In Fig. 6 we show a quantum network that performs a 4-qubit QFT [9]. The blocks labeled W perform a Walsh-Hadamard transform and the other blocks perform a controlled phase shift by the angle indicated. Not shown in the network is the series of SWAP-gates that interchanges the output qubits (1,4) and (2,3) [1, 9], hence the different labeling of input and output lines in Fig. 6. For the applications that we discuss below, these interchanges merely add to the operation count and can therefore be omitted.

The right-hand side of Fig. 6 shows the QCE with the MI-set "dft" and QPs that perform the 3 and 4-qubit QFT algorithm (included in the QCE software distribution). There is a clear one-to-one correspondence between the network and the QPs. The controlled phase shifts by $\pi/2$, $\pi/4$ and $\pi/8$ are performed by the MIs $b1(j,k)$, $b2(j,k)$, and $b3(j,k)$ respectively. For example, to define MI $b1(1,3)$, we proceed as in the case of the CNOT gate, i.e., we use Hamiltonian $H = -JS_1^zS_3^z - h(S_1^z + S_3^z)$ with $\tau J = \pi/2$, $\tau h = -\pi/4$.

In Fig. 6 dft3(4) performs a QFT on the initial state $|000\rangle$ ($|0000\rangle$), yielding the uniform superposition of the 3(4) qubits. As the uniform superposition can be written as a direct product of superpositions of each qubit, knowing the expectation values of the three components of each spin is sufficient to characterize the state. Thus we can easily check the result by reading off the three values per qubit from the bottom of the QP window.



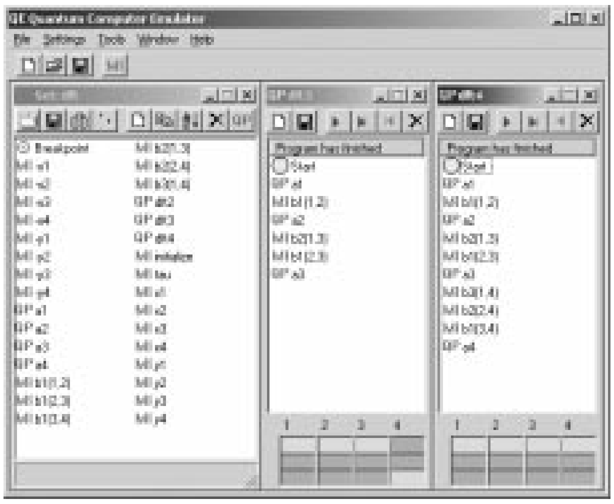


Figure 6. Left: Quantum network that performs a 4-qubit Quantum Fourier Transform. W and P denote the Walsh-Hadamard transform and permutation, respectively. The operations "2","4", and "8" perform controlled phase shifts with angles $\pi/2$, $\pi/4$ and $\pi/8$, respectively. Right: Ideal quantum computer implementation of a 3 and 4-qubit Quantum Fourier Transform on QCE.

4.5. Finding the period of a periodic function

Assume that we are given the function $f(n) = f(n+M)$ for $n = 0, \ldots, N-1$. On a QC we can determine the period M as follows. We use one register of qubits to store n and another one to store $f(n)$. As a first step, the QC is put in the state of uniform superposition over all n. The state of the QC can be written as

$$\begin{aligned}
\frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} |n\rangle |f(n)\rangle &= \frac{1}{\sqrt{N}} \left\{ \sum_{n=0}^{M-1} |n\rangle |f(n)\rangle + \sum_{n=M}^{2M-1} |n\rangle |f(n)\rangle + \ldots \right\} \\
&= \frac{1}{\sqrt{N}} \sum_{n=0}^{M-1} \left(|n\rangle + |n+M\rangle + \ldots \right) |f(n)\rangle, \qquad (27)
\end{aligned}$$

where, in the last step, we used the periodicity of $f(n)$. Using the Fourier representation of $|n\rangle$ we obtain

$$\frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} |n\rangle |f(n)\rangle = \frac{1}{N} \sum_{n=0}^{M-1} \sum_{k=0}^{N-1} \left(1 + e^{2\pi i k M/N} + e^{4\pi i k M/N} + \ldots \right) e^{2\pi i k n/N} |k\rangle |f(n)\rangle. \qquad (28)$$

Assuming that $f(n) = f(n')$ implies $n = n'$ for $0 \le n, n' < M$, the probability $p_k(M)$ to find the QC in the state $|k\rangle$ is given by

$$p_k(M) = \left| \frac{1}{N} \sum_{n=0}^{M-1} \left(1 + e^{2\pi i k M/N} + e^{4\pi i k M/N} + \ldots \right) \right|^2 . \qquad (29)$$

The results for $p_k(M)$ in the case $N = 8$ (3 qubits) are given in Table 2. From Table 2 it follows directly that the expectation values of the qubits are ($Q_1^z = Q_2^z = Q_3^z = 0$) if the period $M = 1$, ($Q_1^z = Q_2^z = 0$, $Q_3^z = 0.5$) if the period $M = 2$, ($Q_1^z = 0.5$, $Q_2^z = 0.375$, $Q_3^z = 0.34375$) if the period $M = 3$, and ($Q_1^z = 0$, $Q_2^z = Q_3^z = 0.5$) if the period $M = 4$. Thus, in this simple case the periodicity of $f(n)$ can be unambiguous determined from the expectation values of the individual qubits.

Table 2. Probability $p_k(M)$ to observe the state $|k\rangle$ after performing the Quantum Fourier Transform on the perodic function $f(n) = f(n+M)$ for $n = 0, \ldots, 7$.

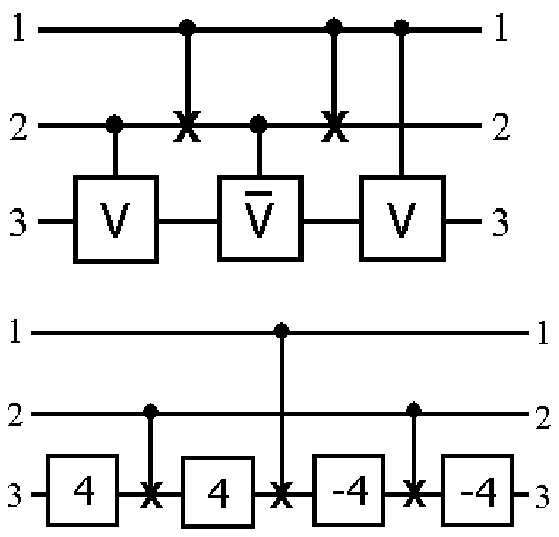
k	$p_k(M=1)$	$p_k(M=2)$	$p_k(M=3)$	$p_k(M=4)$
0	1	0.5	0.34375	0.25
1	0	0.0	0.01451	0.00
2	0	0.0	0.06250	0.25
3	0	0.0	0.23549	0.00
4	0	0.5	0.31250	0.25
5	0	0.0	0.23549	0.00
6	0	0.0	0.06250	0.25
7	0	0.0	0.01451	0.00

4.6. Examples of Quantum Algorithms

4.6.1. Toffoli gate

The Toffoli gate works with two control qubits: Only if for the first two qubits $Q_1^z = Q_2^z = 1$, the third qubit is flipped (see Table 3). We discuss two different implementations of the Toffoli gate.

A schematic representation of the first implementation is given in Fig. 7 [1, 47]. A dot (control bit) and a cross (target bit) connected by a vertical line represent a CNOT gate: If the qubit on the horizontal line with a dot equals one, the qubit on the horizontal line with a cross flips. A similar rule applies to the qubits connected by a vertical line with endpoints a dot and a square box (see Fig. 7). Only if the control qubit equals one, operation V or $\overline{V}$ is carried out.



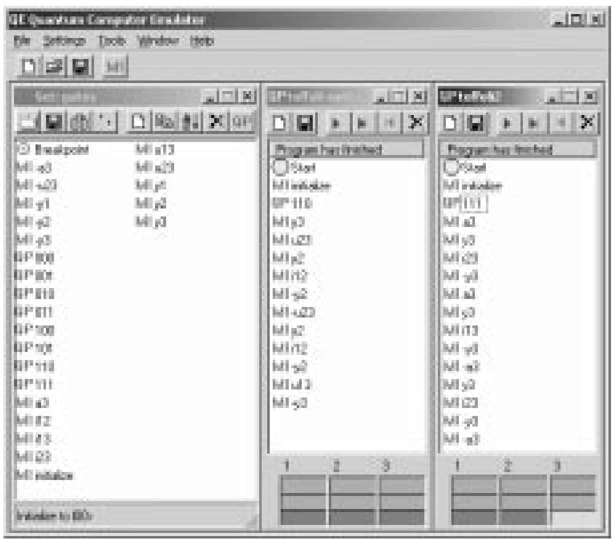

QE Quantum Computer Simulator
File Settings Tools Window Help
Breakpoint
Program has finished
Start

Figure 7. Left top: Quantum network for the Toffoli gate using CNOT gates and controlled phase shifts V and $\overline{V}$. Left bottom: Quantum network for the Toffoli gate using CNOT gates and single-qubit $\pm\pi/4$ rotations "± 4". Right: Ideal quantum computer implementation of the two Toffoli gate networks on QCE.

The internal operation of the Toffoli gate is shown in Table 3. The first two colums give all possible combinations of Q_1^z and Q_2^z. The next five columns show schematically the outcome of the various operations that build up the Toffoli gate (see Fig. 7). The colums labeled "CNOT" only show the value of the second qubit since the first qubit acts as a control bit and does not change. A "x" means that the operation is not carried out because the control bit is zero. The last column shows the full operation on the third qubit.

Table 3. Internal operation of the quantum network Fig. 7 (left top) of the Toffoli gate.

Q_1^z	Q_2^z	V	CNOT	$\overline{V}$	CNOT	V	Result
0	0	x	x	x	x	x	Q_3^z
0	1	V	x	$\overline{V}$	x	x	$V\overline{V}Q_3^z = Q_3^z$
1	0	x	1	$\overline{V}$	0	V	$\overline{V}VQ_3^z = Q_3^z$
1	1	V	0	x	1	V	$V^2Q_3^z = 1 - Q_3^z$

From Table 3 it follows that operation V has to be constructed such that V^2 flips Q_3 and that $\overline{V}V$ is equal to the identity matrix. These conditions are fullfilled by taking

$$V = \frac{e^{-i\pi/4}}{\sqrt{2}}\begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}. \tag{30}$$

For example,

$$V_{12} \equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{-i\pi/4}/\sqrt{2} & ie^{-i\pi/4}/\sqrt{2} \\ 0 & 0 & ie^{-i\pi/4}/\sqrt{2} & e^{-i\pi/4}/\sqrt{2} \end{pmatrix}. \tag{31}$$

In order to bring the above matrix into a similar form as the controlled phase shift, we first diagonalize this matrix. This may be done using the same procedure as in section 4.3:

$$\begin{aligned} V_{12} &= \begin{pmatrix} I_1 & 0 \\ 0 & 2^{-1/2}e^{-i\pi/4}(I_2 + 2iS_2^x) \end{pmatrix} = \begin{pmatrix} I_1 & 0 \\ 0 & 2^{-1/2}e^{-i\pi/4}(I_2 + 2i\overline{Y}_2 S_2^z Y_2) \end{pmatrix} \\ &= \overline{Y}_2 \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{-i\pi/2} \end{pmatrix} Y_2 \equiv e^{-i\beta}\overline{Y}_2 U_{12} Y_2, \end{aligned} \tag{32}$$

where I_1 and I_2 denote the 2×2 identity matrix for qubit 1 and 2 respectively, and $e^{-i\beta}$ is a global phase factor. From Eq.(18) and Eq.(32) it follows that

$$U_{12} = \begin{pmatrix} e^{i\phi_{00}} & 0 & 0 & 0 \\ 0 & e^{i\phi_{01}} & 0 & 0 \\ 0 & 0 & e^{i\phi_{10}} & 0 \\ 0 & 0 & 0 & e^{i\phi_{11}} \end{pmatrix} = e^{i\beta}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{-i\pi/2} \end{pmatrix}, \tag{33}$$

so that $\phi_{00} = \phi_{01} = \phi_{10} = \beta$ and $\phi_{11} = \beta - \pi/2$. Using the Ising model (22) to implement the controlled phase shift U_{12} we find that $h = -J/2$ and $\tau J = -\pi/2$ yields U_{jk}.

From the quantum network on the upper left-hand side of Fig. 7 and Eq.(32) it can be seen that the Toffoli gate can be implemented by the following sequence of operations

$$\overline{Y}_3 U_{13} Y_3 \overline{Y}_2 I_{12} Y_2 \overline{Y}_3 \overline{U}_{23} Y_3 \overline{Y}_2 I_{12} Y_2 \overline{Y}_3 U_{23} Y_3. \tag{34}$$

This sequence can be shortened by observing that in some cases a rotation is followed by its inverse. This leads to

$$\overline{Y}_3 U_{13} \overline{Y}_2 I_{12} Y_2 \overline{U}_{23} \overline{Y}_2 I_{12} Y_2 U_{23} Y_3. \tag{35}$$

The implementation of this version of the Toffoli gate on the QCE is very similar to the one of the CNOT gate. The QPs corresponding to the sequences (34) and (35) are "toffoli" and "toffoli-optimized", respectively. Both QPs are included in the QCE software distribution. We only discuss the latter. Open MI set "gates" by clicking on the "folder" button on the top bar of the MI set window and open QP "toffoli-optimized" in the directory "gates" by clicking on the "folder" button or on the menu item "File" on the top bar of the main window. QP "toffoli-optimized" acts on the state $|110\rangle$ (see Fig. 7). Execution of the program brings the QC in the state $|111\rangle$, the correct final state of the QC (see Fig. 7). It is easy to change QP "toffoli-optimized" so that it runs with other input states. In the QP "toffoli-optimized", first select the QP "110" and then delete it by pressing the delete button on the toolbar of the QP "toffoli-optimized" window. In the MI set "gates" window select another QP that creates another input state and drag it to the QP "toffoli-optimized" window and drop it on top of the MI "initialize".

A schematic representation of the second implementation of the Toffoli gate is also given in Fig. 7 and corresponds to the sequence of operations

$$\overline{A}_3\overline{Y}_3 I_{23} Y_3 \overline{A}_3 \overline{Y}_3 I_{13} Y_3 A_3 \overline{Y}_3 I_{23} Y_3 A_3, \tag{36}$$

where A_j denotes a rotation of $\pi/4$ around the y-axis. The QP "toffoli2" (see Fig. 7) executes this sequence of operations and can also be found on the directory "gates".

4.6.2. Grover's database search algorithm

As a more complicated example of a QA, we consider Grover's database search algorithm to find the needle in a haystack [4, 5]. On a conventional computer, finding an item out of N elements requires $\mathcal{O}(N)$ queries [48]. Grover has shown that a QC can find the item using only $\mathcal{O}(\sqrt{N})$ attempts [4, 5]. Assuming a uniform probability distribution for the needle, for $N = 4$ the average number of queries required by a conventional algorithm is 9/4 [17, 48]. With Grover's QA the correct answer can be found in a single query [15, 17] (this result only holds for a database with 4 items). Grover's algorithm for the four-item database can be implemented on a two-qubit QC.

The key ingredient of Grover's algorithm is an operation that replaces each amplitude of the basis states in the superposition by two times the average amplitude minus the amplitude itself. This operation is called "inversion about the mean" and amplifies the amplitude of the basis state that represents the searched-for item [4, 5]. To see how this works it is useful to consider an example. Consider a database containing four items and functions $g_j(x)$, $j = 0, \ldots, 3$ that upon query of the database return minus one if $x = j$ and plus one if $x \neq j$. Let us assume that the item to search for corresponds to e.g. number 2 ($g_2(0) = g_2(1) = g_2(3) = 1$ and $g_2(2) = -1$). Using the binary representation of integers, the QC is in the state (up to an irrelevant phase factor as usual)

$$|\Psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle - |10\rangle + |11\rangle). \tag{37}$$

The operator D that inverts states like (37) about their mean reads

$$D = \frac{1}{2}\begin{pmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{pmatrix}. \tag{38}$$

Applying D to $|\Psi\rangle$ results in $D|\Psi\rangle = |10\rangle$, i.e. the correct answer. In general, for more than two qubits, more than one application of D is required to get the correct answer [4, 5]. In this sense the two-qubit case is somewhat special.

Implementation of the above example on a two-qubit QC requires an expression of the preparation and query steps and of the operation of inversion about the mean in terms of elementary rotations. Initially we set the QC in the state $|00\rangle$ and then transform $|00\rangle$ to the state (37) by a two-step process. First we use the Walsh-Hadamard transform to bring the QC in the uniform superposition state: $W_2W_1|00\rangle = -(|00\rangle + |01\rangle + |10\rangle + |11\rangle)/2$, where W_j is given by Eq.(16). Next we apply a transformation F_2 that corresponds to the application of $g_2(x)$ to the uniform superposition state

$$F_2 = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{39}$$

This transformation can be implemented by first letting the system evolve in time

$$GW_2W_1|00\rangle = \frac{e^{-i\pi S_1^zS_2^z}}{2}(|00\rangle+|01\rangle+|10\rangle+|11\rangle) = \frac{1}{2}(e^{-i\pi/4}|00\rangle+e^{+i\pi/4}|01\rangle+e^{+i\pi/4}|10\rangle+e^{-i\pi/4}|11\rangle), \quad (40)$$

and applying a sequence of single-spin rotations to change the four phase factors such that we get the desired state. The two-qubit operation G is defined by

$$G = \begin{pmatrix} e^{-i\pi/4} & 0 & 0 & 0 \\ 0 & e^{+i\pi/4} & 0 & 0 \\ 0 & 0 & e^{+i\pi/4} & 0 \\ 0 & 0 & 0 & e^{-i\pi/4} \end{pmatrix}, \quad (41)$$

and performs a controlled phase shift. The two sequences $Y_jX_j\overline{Y}_j$ and $Y_j\overline{X}_j\overline{Y}_j$ operating on qubit j are particulary useful for this purpose since

$$Y_jX_j\overline{Y}_j|0\rangle = e^{+i\pi/4}|0\rangle,\ Y_jX_j\overline{Y}_j|1\rangle = e^{-i\pi/4}|1\rangle,\ Y_j\overline{X}_j\overline{Y}_j|0\rangle = e^{-i\pi/4}|0\rangle,\ Y_j\overline{X}_j\overline{Y}_j|1\rangle = e^{+i\pi/4}|1\rangle. \quad (42)$$

We find

$$\begin{aligned} &Y_1X_1\overline{Y}_1Y_2\overline{X}_2\overline{Y}_2\left[\frac{1}{2}(e^{-i\pi/4}|00\rangle+e^{+i\pi/4}|01\rangle+e^{+i\pi/4}|10\rangle+e^{-i\pi/4}|11\rangle)\right] \\ &= \frac{1}{2}(e^{-i\pi/4}|00\rangle+e^{-i\pi/4}|01\rangle+e^{+3i\pi/4}|10\rangle+e^{-i\pi/4}|11\rangle) = \frac{e^{-i\pi/4}}{2}(|00\rangle+|01\rangle-|10\rangle+|11\rangle). \quad (43)\end{aligned}$$

Thus we can construct the sequence F_j that transforms the uniform superposition state to the state that corresponds to $g_j(x)$:

$$F_0 = Y_1\overline{X}_1\overline{Y}_1Y_2\overline{X}_2\overline{Y}_2G,\ F_1 = Y_1\overline{X}_1\overline{Y}_1Y_2X_2\overline{Y}_2G,\ F_2 = Y_1X_1\overline{Y}_1Y_2\overline{X}_2\overline{Y}_2G,\ F_3 = Y_1X_1\overline{Y}_1Y_2X_2\overline{Y}_2G. \quad (44)$$

Finally we need to express the operation of inversion about the mean, i.e. the matrix D (see (38)), by a sequence of elementary operations. It is not difficult to see that D can be written as

$$D = W_1W_2\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}W_1W_2 \equiv W_1W_2PW_1W_2. \quad (45)$$

The same approach that was used to implement $g_2(x)$ also works for the controlled phase shift P $(= -F_0)$ and yields

$$P = Y_1\overline{X}_1\overline{Y}_1Y_2\overline{X}_2\overline{Y}_2G. \quad (46)$$

The complete sequence U_j operating on $|00\rangle$ reads

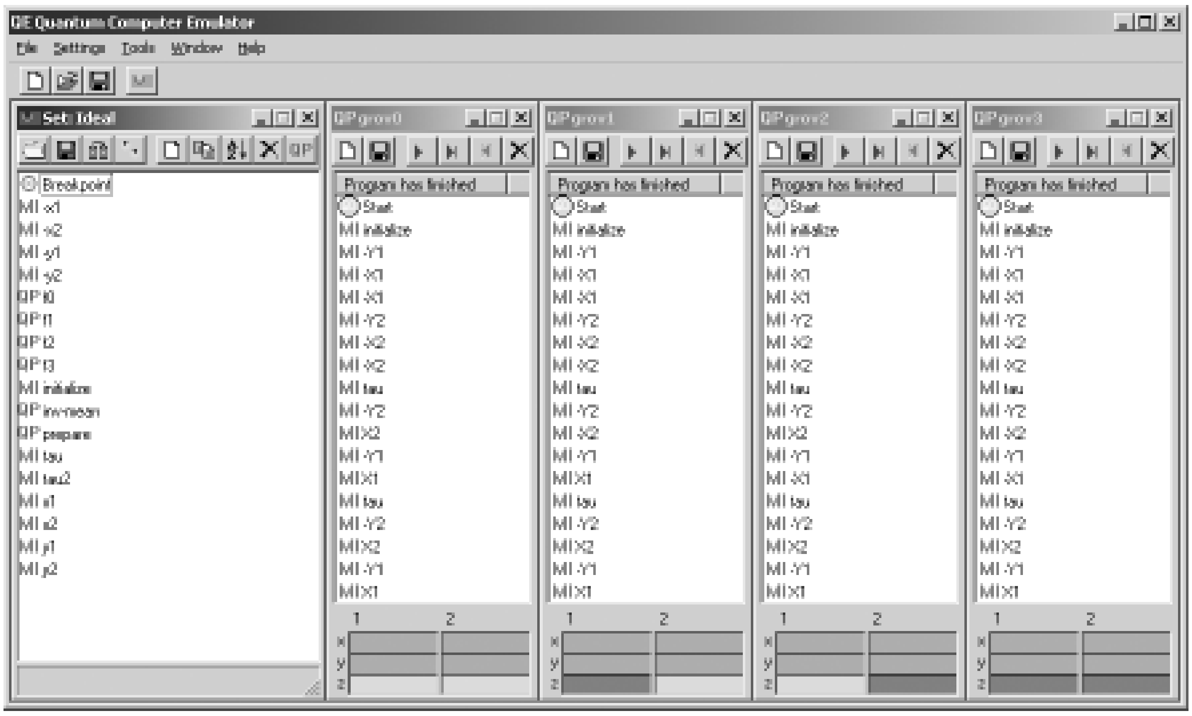
$$U_j = W_1W_2PW_1W_2F_jW_2W_1. \quad (47)$$

Each sequence U_j can be shortened by observing that in some cases a rotation is followed by its inverse. Making use of the alternative representations of the Walsh-Hadamard transform W_i (see Eq.(16)), the sequence for e.g. $j = 1$ can be written as

$$W_1W_2F_1 = -X_1X_1\overline{Y}_1\overline{X}_2\overline{X}_2\overline{Y}_2Y_1\overline{X}_1\overline{Y}_1Y_2X_2\overline{Y}_2G = -X_1\overline{Y}_1\overline{X}_2\overline{Y}_2G. \quad (48)$$

The sequences for the other cases can be shortened as well, yielding

$$\begin{aligned} U_0 &= X_1\overline{Y}_1X_2\overline{Y}_2GX_1\overline{Y}_1X_2\overline{Y}_2G\overline{X}_2\overline{X}_2\overline{Y}_2\overline{X}_1\overline{X}_1\overline{Y}_1, \\ U_1 &= X_1\overline{Y}_1X_2\overline{Y}_2GX_1\overline{Y}_1\overline{X}_2\overline{Y}_2G\overline{X}_2\overline{X}_2\overline{Y}_2\overline{X}_1\overline{X}_1\overline{Y}_1, \\ U_2 &= X_1\overline{Y}_1X_2\overline{Y}_2G\overline{X}_1\overline{Y}_1X_2\overline{Y}_2G\overline{X}_2\overline{X}_2\overline{Y}_2\overline{X}_1\overline{X}_1\overline{Y}_1, \\ U_3 &= X_1\overline{Y}_1X_2\overline{Y}_2G\overline{X}_1\overline{Y}_1\overline{X}_2\overline{Y}_2G\overline{X}_2\overline{X}_2\overline{Y}_2\overline{X}_1\overline{X}_1\overline{Y}_1. \quad (49)\end{aligned}$$



```
Breakpoint
MI -x1
MI -x2
MI -y1
MI -y2
QP t0
QP t1
QP t2
QP t3
MI initialize
QP inv-mean
QP prepare
MI tau
MI tau2
MI x1
MI x2
MI y1
MI y2
```

```
Program has finished
Start
MI initialize
MI -Y1
MI -X1
MI -X1
MI -Y2
MI -X2
MI -X2
MI tau
MI -Y2
MI X2
MI -Y1
MI X1
MI tau
MI -Y2
MI X2
MI -Y1
MI X1
```

```
Program has finished
Start
MI initialize
MI -Y1
MI -X1
MI -X1
MI -Y2
MI -X2
MI -X2
MI tau
MI -Y2
MI -X2
MI -Y1
MI X1
MI tau
MI -Y2
MI X2
MI -Y1
MI X1
```

```
Program has finished
Start
MI initialize
MI -Y1
MI -X1
MI -X1
MI -Y2
MI -X2
MI -X2
MI tau
MI -Y2
MI X2
MI -Y1
MI -X1
MI tau
MI -Y2
MI X2
MI -Y1
MI X1
```

```
Program has finished
Start
MI initialize
MI -Y1
MI -X1
MI -X1
MI -Y2
MI -X2
MI -X2
MI tau
MI -Y2
MI -X2
MI -Y1
MI -X1
MI tau
MI -Y2
MI X2
MI -Y1
MI X1
```

Figure 8. MI set window "Ideal" together with the QP windows "grov0", "grov1", "grov2" and "grov3", implementing Grover's database search algorithm on a two-qubit quantum computer for the four different cases $g_0(x), \ldots g_3(x)$.

Note that the QAs (49) are by no means unique: Various alternative expressions can be written down by using the algebraic properties of the X's and Y's. Sequences (49) return the correct answer, i.e. the position of the searched-for item. This is easily verified on the QCE. Open on the QCE the MI set "Ideal" and open the QPs "grov0", ..., "grov3" in the directory "grover" (see Fig. 8). In the MI set "Ideal", MI "tau" corresponds to the operation G. To implement G using (22) the parameters in MI "tau" are chosen such that $h = 0$ and $(\tau/2\pi)J = -1/2$. Running the four QPs results in

$$U_0|00\rangle = |00\rangle, \quad U_1|00\rangle = |10\rangle, \quad U_2|00\rangle = |01\rangle, \quad U_3|00\rangle = |11\rangle, \tag{50}$$

as can be seen from Fig. 8. Using the binary representation of integers with the order of the bits reversed, the final state of the QC gives the location of the item in the database. Hence QP "grov0", "grov1", "grov2" and "grov3" return 0, 1, 2 and 3, respectively, which are the correct answers.

4.6.3. Finding the order of a permutation

As an illustration of using the QCE to determine the period of a function we consider the problem of finding the order of a permutation [30]. An experimental realization of this QA on a 5-qubit NMR QC for the case of a permutation of 4 items is described in Ref. [30]. The theory in this section closely follows Ref. [30]. The problem is defined as follows: Given a permutation P of the integers $\{0, 1, ..., N-1\}$ and an integer $0 \le y \le N-1$, the order $r(y)$ is the smallest integer for which $P^r(y)y = y$. Thus, the purpose of the QA is to determine $r(y)$, for a given y and permutation P.

The basic idea of the QA to find the order of a permutation is to exploit the quantum parallelism to compute $P(y)$, $P^2(y)$, $P^3(y)$ and $P^4(y)$ simultaneously and filter out the power $r(y)$ that yields $P^r(y)y = y$. Denoting $f(n) = P^n$, finding $r(y)$ is the same as finding the period of $f(n)$, a problem that is solved by invoking the QFT.

First we consider the problem of generating a permutation of $N = 4$ items. We need two qubits to specify one of the 4 items. Using the binary representation of the integers $\{0, 1, 2, 3\}$ it is easy to see that the CNOT operation C_{12} (the first subscript denoting the control bit) corresponds to the permutation (in cycle notation) $P = (0)(1)(23)$ that interchanges items 2 and 3. Likewise, C_{21} generates the permutation $P = (0)(2)(13)$ and $C_{12}C_{21}C_{12}$ is equivalent to the permutation $P = (0)(1)(23)$. The remaining interchanges of two items can be generated by a combination of CNOT gates and NOT operations. Denoting the NOT operation on the j-th qubit by N_j, we find that $C_{12}N_1 = N_1C_{12}$ yields $P = (01)(2)(3)$, $C_{21}N_2 = N_2C_{21}$ yields $P = (1)(3)(02)$, and $C_{21}N_1C_{12}C_{21}$ yields $P = (1)(2)(03)$. Using these elementary interchanges we can construct any permutation.

The quantum network that carries out the QA to find the order of a permutation P is shown in Fig. 9. There is a ono-to-one mapping from this network onto the QA to find the period of the function $f(n)$ (see section 4.5). The first three qubits hold the integer n, qubits 4 and 5 hold $y = 2y_1 + y_0$. The three Walsh-Hadamard operations change the initial state $|000\rangle|y_0y_1\rangle$ into $|uuu\rangle|y_0y_1\rangle$ where we use the label "u" to denote the uniform superposition. Then, we apply the permutations $P^0y, P^1y, \ldots, P^7y$. In the actual implementation, the sequence of CNOT and NOT operations that implement the permutation P need to be replaced by Toffoli and CNOT gates respectively because (the power of) P is applied to the fourth and fifth qubit (representing the integer y), conditional on the state of the first three qubits. Finally, we perform a QFT on the first three qubits and consider the value of the first three qubits Q_1^z, Q_2^z, and Q_3^z. As explained above, in this simple case we can extract the period of the function, and hence the order of P acting on y, from the values of Q_1^z, Q_2^z, and Q_3^z.

The MI-set for this problem is called "order" and the QCE programs themselves can be found in the folder "order". In Fig. 10 we show the QCE window with the programs for $r = 1$, $r = 2$, $r = 3$, and $r = 4$ for the permutations $P = (0)(2)(13)$, $P = (01)(23)$, $P = (0)(132)$, $P = (3210)$ respectively. Also shown are the subprograms that perform these permutations. Executing these programs yields values of the three first qubits from which (by using Table 2) the correct value of $r(y)$ directly follows.

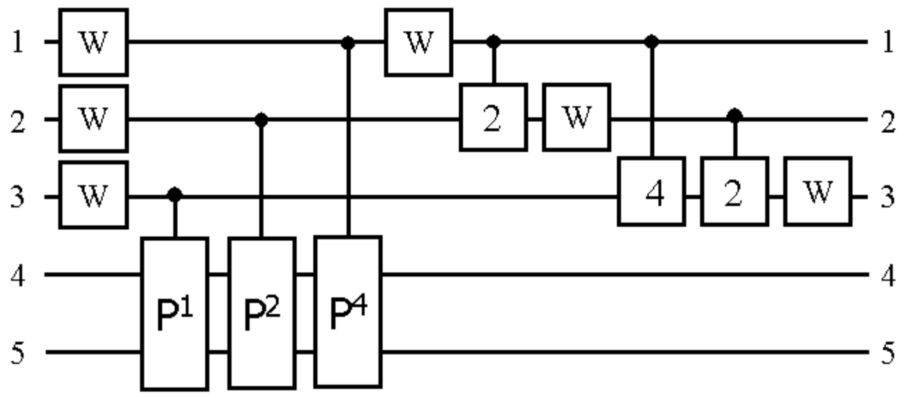


Figure 9. Quantum network of a quantum algorithm to find the order of a permutation of 4 items. W and P^k denote the Walsh-Hadamard transform and k applications of the permutation P, respectively. The operations "2" and "4" perform controlled phase shifts with angles $\pi/2$ and $\pi/4$, respectively.

4.6.4. Number factoring: Shor's algorithm

As another illustration of using the QFT to determine the period of a function we consider the problem of factoring integers, i.e. Shor's algorithm. For the case $N = 15$, an experimental realization of this QA on a 7-qubit NMR QC is given in Ref. [35]. The theory in this section closely follows Ref. [36].

The theory behind Shor's algorithm has been discussed at great length elsewhere [1, 3, 9]. Therefore we only recall the basic elements of Shor's algorithm and focus on the QCE implementation of the algorithm for the case $N = 15$. Shor's algorithm is based on the fact that the factors p and q of an integer $N = pq$ can be deduced from the period M of the function $f(j) = a^j \mathrm{mod} N$ for $j = 0, \ldots, 2^n - 1$ where $N \leq 2^n$. Here $a < N$ is a random number that has no common factors with N. Once M has been determined, at least one factor of N can be found by computing the greatest common divisor (g.c.d.) of N and $a^{M/2} \pm 1$.

Figure 10. Left: Ideal quantum computer implementation on QCE of a quantum algorithm to find the order of a permutation of 4 items. Right: Ideal QC implementation on QCE of Shor's algorithm to factor 15.

Compared to the example of the previous section, the new aspect is the modular exponentiation $a^n\text{mod}N$. For $N = 15$ this calculation is almost trivial. Using the binary represention of j we can write $a^n\text{mod}N = a^{2^{n-1}j_{n-1}} \ldots a^{2j_1}a^{j_0}\text{mod}N = (a^{2^{n-1}j_{n-1}}\text{mod}N)\ldots(a^{2j_1}\text{mod}N)(a^{j_0}\text{mod}N)\text{mod}N$, showing that we only need to implement $(a^{2^k j_k}\text{mod}N)$. For $N = 15$ the allowed values for a are $a = 2, 4, 7, 8, 11, 13, 14$. If we pick $a = 2, 7, 8, 13$ then $a^{2^k}\text{mod}N = 1$ for all $k > 1$ while for the remaining cases we have $a^{2^k}\text{mod}N = 1$ for all $k > 0$. Thus, for $N = 15$ only two (not four) qubits are sufficient to obtain the period of $f(j) = a^j\text{mod}N$ [36]. As a matter of fact, this analysis provides enough information to deduce the factors of $N = 15$ using Shor's procedure so that no further computation is necessary. Non-trivial quantum operations are required if we decide to use three (or more) qubits to determine the period of $f(j) = a^j\text{mod}N$ [36]. Following Ref. [36] we will consider a 7-qubit QC with 4 qubits to hold $f(j)$ and three qubits to perform the QFT.

In essence the quantum network for the Shor algorithm is the same as the one shown in Fig. 9 (and therefore not shown) with the permutations (two qubits) replaced by modular exponentiation (four qubits). The quantum networks to compute $a^j\text{mod}15$ for $j = 0, \ldots, 7$ and a fixed input a are easy to construct. For example, consider the case $a = 11 = |1011\rangle$. If j is odd then $11^j\text{mod}15 = 11$ and the network should leave $|1011\rangle$ unchanged. Otherwise $11^j\text{mod}15 = 1$ and hence it should return $|0001\rangle$. The network for this operation consists of two CNOT gates that have as control qubit, the same least-significant qubit of the three qubits that are input to QFT. The sequence of CNOT and Toffoli gates that performs similar operations for the other cases can be found in the same manner.

The results of this excercise are shown in the right-hand side of Fig.10 where we present a snapshot of the QCE implementation. The three QPs at the top show Shor's algorithm for the case $a = 2, 7, 11$. The examples $a = 7, 11$ are taken from Ref. [36]. The QPs for the corresponding modular exponentiation are shown in the lower part. In this simple case, we can determine the period of the function $f(j) = a^j\text{mod}N$ from the expectation values of the first three qubits. For $a = 2, 7$ we find ($Q_1^z = 0$, $Q_2^z = 0.5$, $Q_3^z = 0.5$) and hence the period $M = 4$, yielding the correct factors g.c.d.$(7^2 \pm 1, 15) = 3, 5$ of $N = 15$. Similarily, for $a = 11$ we find ($Q_1^z = 0$, $Q_2^z = 0$, $Q_3^z = 0.5$) corresponding to the period $M = 2$ and the factors g.c.d.$(11 \pm 1, 15) = 3, 5$.

4.6.5. A three-input adder

This subsection gives another illustration of the use of the QFT: A quantum algorithm to add the content of three 4-qubit registers. This example is taken from the PhD thesis of S. Bettelli [49]. The quantum network

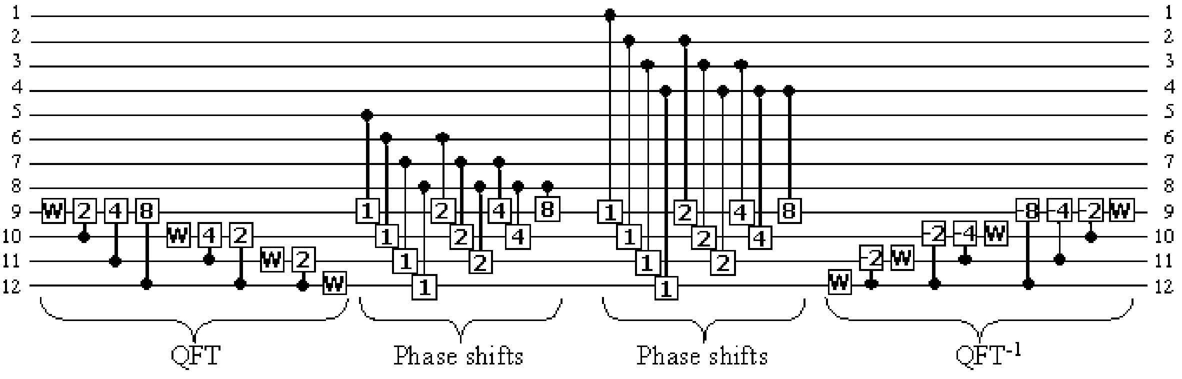


Figure 11. Quantum network of a three-input adder, as described in Ref. [49]. The algorithm performs a Quantum Fourier Transform (QFT) of register 3 (qubits 9 to 12), adds the content of register 2 (qubits 5 to 8) and the content of register 1 (qubits 1 to 4), followed by a QFT on register 3 to yield the final answer (register 1 + register 2 + register 3 mod16). W denotes the Walsh-Hadamard transform. The operations "±1", "±2", "±4", and "±8" perform conditional phase shifts with angles $\pm\pi$, $\pm\pi/2$, $\pm\pi/4$ and $\pm\pi/8$, respectively. Squares that touch each other indicate operations that can be done simultaneously.

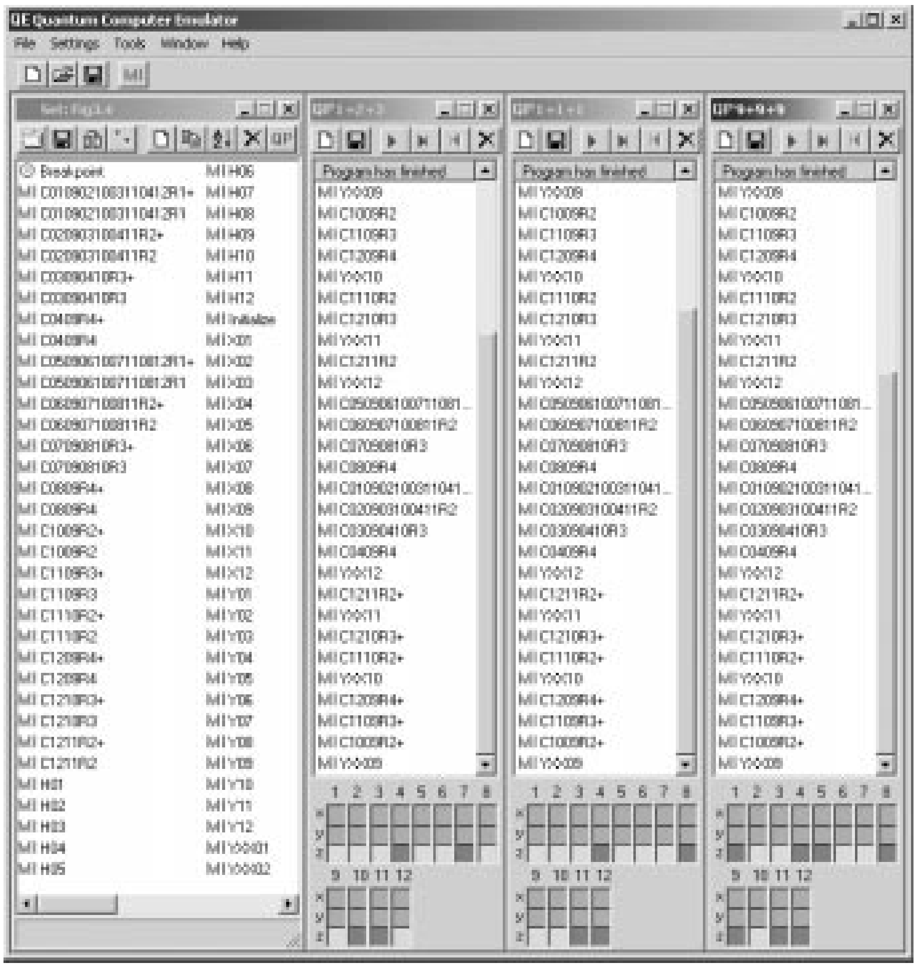
of the complete circuit is shown in Fig. 11. The modular structure of this approach is clear. Note that with respect to the QFT network of Fig. 6, both the labeling of qubits and the order of operations has been reversed. The former is merely a change of notation and the latter allowed because quantum algorithms are reversible (unitary transformations) by construction [1]. The basis idea of this algorithm is to use the QFT to first transfer the information in a register to the phase of the amplitudes, then use conditional phase shifts to add information from the two other registers and finally QFT back to the original representation. Note that this quantum network differs considerably from the one described in Ref. [47] and is also easier to implement.

The QCE implementation is shown in Fig. 12. The correct results of the three examples (1+1+1, 1+2+3, and 9+9+9) provided in the QCE software distribution can be read off from the bottom of the QPs. Not shown in Fig. 12 because of lack of space are many of the MIs and the parts of the QPs that set the initial state of the three four-qubit registers. Also clear from this example is that programming more complicated examples like this one should not be done by hand: the MIs and QPs for this examples have been generated by another program. Some of the MIs may look rather complicated but that is a little misleading: Whenever it is logically allowed to perform operations simultaneously, these operations have been put into one MI.

4.6.6. Number partitioning

As the last example, we discuss a QA to count the number of solutions of the number partitioning problem (NPP) [50]. The NPP is defined as follows: Does there exist a partitioning of the set $A = \{a_1, ..., a_n\}$ of n positive integers a_j into two disjoint sets A_1 and $A_2 = A - A_1$ such that the difference of the sum of the elements of A_1 and the sum of the elements of A_2 is either zero (if the sum of all elements of A is even) or one (if the sum of all elements of A is odd)? The simple examples included in the QCE software distribution may be useful to understand the problem. If $A = \{1, 2, 3, 4\}$ (see QP "npp1234") the answer to the NPP is yes because for $A_1 = \{1, 4\}$ and $A_2 = \{2, 3\}$, the sum of the elements of A_1 and A_2 are both equal to five. In this case there are two solutions because we can interchange A_1 and A_2. If $A = \{1, 1, 1, 4\}$ (see QP "npp1114") the answer is again yes as there is one solution namely $A_1 = \{1, 1, 1\}$ and $A_2 = \{4\}$. The difference of the sum of the elements of A_1 and A_2 is equal to one and that is OK because the sum of all elements of A is odd. If $A = \{2, 2, 2, 4\}$ (see QP "npp2224") there is no solution to the NPP.

The quantum network for the NPP QA is shown in Fig. 13. The basic idea behind this QA is a number of transformations that reduce the counting of the number of solutions of the NPP to finding the number of zero eigenvalues of a spin-1/2 Hamiltonian [50]. The largest part (in terms of the number of operations)



```
Breakpoint                      MI H06
MI C01090210031104120R1+        MI H07
MI C01090210031104120R1         MI H08
MI C02090310041 1R2+            MI H09
MI C02090310041 1R2             MI H10
MI C03090410R3+                 MI H11
MI C03090410R3                  MI H12
MI C0409R4+                     MI Initialize
MI C0409R4                      MI X01
MI C05090610071108120R1+        MI X02
MI C05090610071108120R1         MI X03
MI C06090710081 1R2+            MI X04
MI C06090710081 1R2             MI X05
MI C07090810R3+                 MI X06
MI C07090810R3                  MI X07
MI C0809R4+                     MI X08
MI C0809R4                      MI X09
MI C1009R2+                     MI X10
MI C1009R2                      MI X11
MI C1109R3+                     MI X12
MI C1109R3                      MI Y01
MI C1110R2+                     MI Y02
MI C1110R2                      MI Y03
MI C1209R4+                     MI Y04
MI C1209R4                      MI Y05
MI C1210R3+                     MI Y06
MI C1210R3                      MI Y07
MI C1211R2+                     MI Y08
MI C1211R2                      MI Y09
MI H01                          MI Y10
MI H02                          MI Y11
MI H03                          MI Y12
MI H04                          MI YXX01
MI H05                          MI YXX02
```

```
Program has finished
MIYXX09
MIC1009R2
MIC1109R3
MIC1209R4
MIYXX10
MIC1110R2
MIC1210R3
MIYXX11
MIC1211R2
MIYXX12
MIC050906100711081...
MIC06090710081 1R2
MIC07090810R3
MIC0809R4
MIC010902100311041...
MIC02090310041 1R2
MIC03090410R3
MIC0409R4
MIYXX12
MIC1211R2+
MIYXX11
MIC1210R3+
MIC1110R2+
MIYXX10
MIC1209R4+
MIC1109R3+
MIC1009R2+
MIYXX09
```

```
Program has finished
MIYXX09
MIC1009R2
MIC1109R3
MIC1209R4
MIYXX10
MIC1110R2
MIC1210R3
MIYXX11
MIC1211R2
MIYXX12
MIC050906100711081...
MIC06090710081 1R2
MIC07090810R3
MIC0809R4
MIC010902100311041...
MIC02090310041 1R2
MIC03090410R3
MIC0409R4
MIYXX12
MIC1211R2+
MIYXX11
MIC1210R3+
MIC1110R2+
MIYXX10
MIC1209R4+
MIC1109R3+
MIC1009R2+
MIYXX09
```

```
Program has finished
MIYXX09
MIC1009R2
MIC1109R3
MIC1209R4
MIYXX10
MIC1110R2
MIC1210R3
MIYXX11
MIC1211R2
MIYXX12
MIC050906100711081...
MIC06090710081 1R2
MIC07090810R3
MIC0809R4
MIC010902100311041...
MIC02090310041 1R2
MIC03090410R3
MIC0409R4
MIYXX12
MIC1211R2+
MIYXX11
MIC1210R3+
MIC1110R2+
MIYXX10
MIC1209R4+
MIC1109R3+
MIC1009R2+
MIYXX09
```

Figure 12. QCE implementation of the quantum network of Fig. 11 for adding three quantum registers of four qubits each. QPs (from left to right) compute 1+2+3, 1+1+1, and 9+9+9 mod16. Not shown are the MIs that set the initial values of the three four-qubit registers.

of the QA is a combination of gates that transforms the state of the QC such that the number of solutions n_s is a physically measureable quantity (the expectation values of the 15-th spin). For $n = 4$ we have $n_s = 16\sqrt{Q^z_{15}}$ [50].

In Fig. 13 we show the QCE window with (part of) the NPP MI set and the three examples discussed earlier. In the final state the values of Q^z_{15} are 0.015625, 0.00390625, and 0 respectively. The corresponding number of solutions is $n_s = 2$, $n_s = 1$ and $n_s = 0$. Clearly the QCE programs correctly solves NPP problems.

5. NMR-like quantum computer

Having described how QCE can be programmed to act as an ideal QC we now turn to the simulation of physical QCs. In this section we briefly discuss the main steps of going from an ideal, computer-science-type QC to a more realistic, physical model of QC hardware [37, 50]. We limit our presentation to the implementation of the CNOT gate and Grover's algorithm on a two-qubit, NMR-like QC. A more extensive discussion as well as many other examples can be found in Ref. [40] and in the QCE software distribution.

As a prototype QC model we will take the spin Hamiltonian for the two nuclear spins of the ^{1}H and ^{13}C atoms in a carbon-13 labeled chloroform molecule that has been used in the NMR-QC experiments [17, 18].

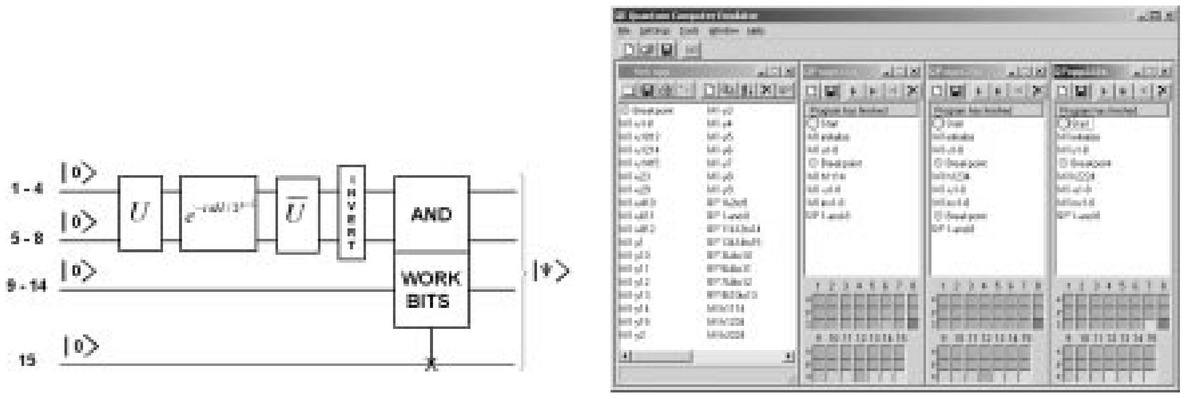


Figure 13. Left: Block diagram of the quantum algorithm that solves the number partitioning problem. The first four qubits are used to represent the integers to be partitioned. Qubits 5 to 8 are used to determine the number of solutions of the number partitioning problem. The remaining 7 qubits are used to relate n_s to a physically measurable quantity: The expectation value of the 15-th qubit. The unitary transformation U prepares the uniform superposition of the first 8 qubits, $\overline{U}$ is the inverse of U, and the combination of INVERT and AND gates sets the 15-th qubit to one if and only if the first eight qubits are all one. Right: QCE implementation of a quantum algorithm that solves the number partitioning problem.

The strong external magnetic field in the z-direction defines the computational basis. If the spin is aligned along the direction of the field the state of the qubit is $|0\rangle$, if the spin points in the other direction the state of the qubit is $|1\rangle$. In the absence of interactions with other degrees of freedom this spin-1/2 system can be modelled by the Hamiltonian

$$H_{NMR} = -J_{12}^z S_1^z S_2^z - h_1^z S_1^z - h_2^z S_2^z, \tag{51}$$

where $h_1^z/2\pi \approx 500$MHz, $h_2^z/2\pi \approx 125$MHz, and $J \equiv J_{12}^z/2\pi \approx -215$Hz [17]. In our numerical work we use the rescaled model parameters

$$J = -0.43 \times 10^{-6}, \quad h_1^z = 1, \quad h_2^z = 1/4. \tag{52}$$

The ratio $\gamma = h_2^z/h_1^z = 1/4$ expresses the difference in the gyromagnetic ratio of the nuclear spin of the ^{1}H and ^{13}C atom.

NMR uses radio frequency (RF) electromagnetic pulses to rotate the spins [42, 43]. By tuning the frequency of the RF-field to the precession frequency of a particular spin, the power of the applied pulse (= intensity times duration) controls how much the spin will rotate. The axis of the rotation is determined by the direction of the applied RF-field. By selecting the appropriate RF pulses, arbitrary single-spin rotation can be carried out. In other words, using RF pulses we can perfom any single-qubit operation. Communication between the qubits is accomplished through the spin-spin interaction in model (51).

5.1. Single-qubit operations

In NMR experiments it is impossible to shield a particular spin from the sinusoidal field. An application of a sinusoidal field not only affects the state of the resonant spin but also changes the state of the other spins (unless they are both perfectly aligned along the z-axis). An analytical, quantitative analysis of this simple-looking many-body problem is rather difficult. As the values of the model parameters (52) suggest, the interaction between the spins will have a negligible impact on the time evolution of the spins during application of the sinusoidal pulse if the duration of the pulse is much shorter than $1/J$ (we use units such that τJ is dimensionless). Thus, as far as the single-qubit operations are concerned, we may neglect the interaction between the two spins (this is also confirmed by numerical simulation of (51), see below). In this section we closely follow [40].

We consider the two-spin system described by the time-dependent Schrödinger equation

$$i\frac{\partial}{\partial t}|\Phi(t)\rangle = -\left[h_1^z S_1^z + h_2^z S_2^z + \tilde{h}_1^x(S_1^x \sin\omega t + S_1^y \cos\omega t) + \tilde{h}_2^x(S_2^x \sin\omega t + S_2^y \cos\omega t)\right]|\Phi(t)\rangle, \qquad (53)$$

for two interacting spins in a static and a rotating sinusoidal field. As usual, it is convenient to work in a rotating frame [42]. Substituting $|\Phi(t)\rangle = e^{it\omega(S_1^z+S_2^z)}|\Psi(t)\rangle$ we obtain

$$i\frac{\partial}{\partial t}|\Psi(t)\rangle = -\left[(h_1^z-\omega)S_1^z + (h_2^z-\omega)S_2^z + \tilde{h}_1^x S_1^y + \tilde{h}_2^x S_2^y\right]|\Psi(t)\rangle. \qquad (54)$$

Our aim is to determine the conditions under which we can rotate spin 1 about an angle φ_1 without affecting the state of spin 2. First we choose

$$\omega = h_1^z, \qquad (55)$$

i.e. the frequency of the sinusoidal field is tuned to the resonance frequency of spin 1. Then (54) can easily be integrated. The result is

$$|\Phi(t)\rangle = e^{ith_1^z(S_1^z+S_2^z)} e^{it\tilde{h}_1^x S_1^y} e^{it\mathbf{S}_2\cdot\mathbf{v}_{12}}|\Phi(0)\rangle, \qquad (56)$$

where $\mathbf{v}_{nm} \equiv (0, \tilde{h}_m^x, h_m^z - h_n^z)$. The third factor in (56) rotates spin 2 around the vector $\mathbf{v}_{12}$. This factor can be expressed as

$$e^{it\mathbf{S}_m\cdot\mathbf{v}_{nm}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\cos\frac{t|\mathbf{v}_{nm}|}{2} + i|\mathbf{v}_{nm}|^{-1}\begin{pmatrix} h_m^z - h_n^z & -ih_m^x \\ ih_m^x & h_n^z - h_m^z \end{pmatrix}\sin\frac{t|\mathbf{v}_{nm}|}{2}, \qquad (57)$$

and we see that the sinusoidal field will not change the state of spin 2 if and only if the duration t_1 of the pulse satisfies

$$t_1|\mathbf{v}_{12}| = t_1\sqrt{(h_1^z - h_2^z)^2 + (\tilde{h}_2^x)^2} = 4\pi n_1, \qquad (58)$$

where n_1 is a positive integer. The second factor in (56) is a special case of (57). Putting

$$t_1\tilde{h}_1^x = \varphi_1, \qquad (59)$$

the second factor in (56) will rotate spin 1 about φ_1 around the y-axis. Therefore, if conditions (55), (58), and (59) are satisfied we can rotate spin 1 about φ_1 without affecting the state of spin 2, independent of the physical realization of the QC. However, the first factor in (56) can still generate a phase shift. Although it drops out in the expression of the expectation value of the qubits, it has to be taken into account in a QC calculation because this phase shift depends on the state of the spins. Adding the condition

$$t_1 h_1^z = 4\pi k_1, \qquad (60)$$

where k_1 is a positive integer ($h_i^z > 0$ by definition), the first factor in (56) is always equal to one. A last constraint on the choice of the pulse parameters comes from the fact that

$$h_2^\alpha = \gamma h_1^\alpha \quad , \quad \tilde{h}_2^\alpha = \gamma\tilde{h}_1^\alpha \quad ; \quad \alpha = x,y,z. \qquad (61)$$

Without loss of generality we may assume that $0 < \gamma < 1$.

Using conditions (55), (58), (59), (60), and (61) and reversing the role of spin 1 and spin 2 we obtain

$$(1-\gamma)^2 k_1^2 + \frac{\gamma^2}{4}\left(\frac{\varphi_1}{2\pi}\right)^2 = n_1^2 \quad , \quad (1-\frac{1}{\gamma})^2 k_2^2 + \frac{1}{4\gamma^2}\left(\frac{\varphi_2}{2\pi}\right)^2 = n_2^2, \qquad (62)$$

where k_1, k_2, n_1, and n_2 are positive integers. The angles of rotation about the y-axis can be chosen such that $0 \le \varphi_1 \le 2\pi$ and $0 \le \varphi_2 \le 2\pi$. Of course, similar expressions hold for rotations about the x-axis.

In general (62) has no solution but a good approximate solution may be obtained if γ is a rational number and k_1 and k_2 are large. Let $\gamma = N/M$ where N and M are integers satisfying $0 < N < M$. It follows that the representation $k_1 = kMN^2$ and $k_2 = kNM^2$ will generate sufficiently accurate solutions of (62) if the integer k is chosen such that $2kNM(M-N) \gg 1$. In terms of k, N, and M, the relevant physical quantities are then given by

$$\frac{t_1 h_1^z}{2\pi} = 2kMN^2 \quad , \quad \frac{\tilde{h}_1^x}{h_1^z} = \frac{1}{2kMN^2}\frac{\varphi_1}{2\pi}, \quad , \quad \frac{t_2 h_1^z}{2\pi} = 2kM^3 \quad , \quad \frac{\tilde{h}_2^x}{h_1^z} = \frac{1}{2kM^3}\frac{\varphi_2}{2\pi}. \tag{63}$$

We have derived conditions (63) under the assumption of ideal sinusoidal RF pulses. In experiment there is no such limitation: the sinusoidal fields may be modulated by almost any waveform [43, 51]. However, the fact that in QC applications it is necessary to use single-spin pulses that do not change the state of the other spins remains. For general pulses, finding the form of the pulse that rotates spin 1 such that the state of spin 2 is not affected is a complicated non-linear optimization problem [40].

To summarize: If conditions (55), (58), (59), and (60) are satisfied we can rotate spin 1 about φ_1 without affecting the state of spin 2 and without introducing unwanted phase shifts. In our numerical experiments we use Eq.(63) to determine the duration of the sinusoidal pulses. These sinusoidal pulses will then be optimized in the sense that a pulse that rotates spin 1 (2) will change the state of spin 2 (1) only slightly if k satisfies $2kNM(M-N) \gg 1$.

5.2. Two-qubit operations

In section 4.3 we implemented the CNOT sequence (20) using the Ising model (22). Some modification is necessary to account for the fact that the two nuclear spins feel different static fields (see (51)). Comparison of (22) with (51) shows that the implemention of the CNOT operation requires additional rotations:

$$\text{CNOT} = \overline{Y}_2 e^{-i\tau(h_1^z-h)S_1^z} e^{-i\tau(h_2^z-h)S_2^z} e^{-i\tau H_{NMR}} Y_2 = \overline{Y}_2 e^{-i\tau(h_1^z-h)S_1^z} e^{-i\tau(h_2^z-h)S_2^z} Y_2 \overline{Y}_2 e^{-i\tau H_{NMR}} Y_2, \tag{64}$$

where we used the fact that $Y_2\overline{Y}_2 = 1$. The extra phase shifts in (64) can be expressed in terms of single-qubit operations. The identities

$$e^{-i\tau(h_1^z-h)S_1^z} = Y_1 X_1' \overline{Y}_1 = \overline{X}_1 Y_1' X_1, \quad e^{-i\tau(h_2^z-h)S_2^z} = Y_2 X_2' \overline{Y}_2, \tag{65}$$

define the single-spin rotations X_1', Y_1', and X_2'.

In the case of Grover's database search algorithm, the representation of G in terms of the time evolution of (51) reads

$$G = e^{-i\pi S_1 S_2} = e^{-i\tau h_1^z S_1^z} e^{-i\tau h_2^z S_2^z} e^{-i\tau H_{NMR}} = Y_2 X_2'' \overline{Y}_2 Y_1 X_1'' \overline{Y}_1 e^{-i\tau H_{NMR}}, \tag{66}$$

where $\tau = -\pi/J$. This choice of τ also fixes the angles of the rotations, and also all parameters of the operations X_1'' and X_2''.

As (65) suggests, there are many different, logically equivalent sequences that implement the CNOT gate on an NMR-like QC. We have chosen to limit ourselves to the respresentations

$$\text{CNOT}_1 = Y_1 X_1' \overline{Y}_1 X_2' \overline{Y}_2 I' Y_2, \quad \text{CNOT}_2 = Y_1 X_1' X_2' \overline{Y}_1 \overline{Y}_2 I' Y_2, \quad \text{CNOT}_3 = \overline{X}_1 Y_1' X_2' \overline{Y}_2 X_1 I' Y_2, \tag{67}$$

where we introduced the symbol I' to represent the time evolution $e^{-i\tau H_{NMR}}$ with $\tau = -\pi/J$.

On an ideal QC there is no difference between the logical and physical computer and the sequences (67) give identical results. However, on a physical QC such as the NMR-like QC (51) this is not the case. On a physically realizable NMR-like QC $X_1X_2 \neq X_2X_1$ unless $2kNM(M-N) \gg 1$ and (63) are satisfied *exactly*. We will use sequences (67) to demonstrate this unpleasant feature of realistic QCs.

5.3. Model parameters

The model parameters for the rotating sinusoidal fields are determined according to the theory outlined above. We use the integer k to compute all free parameters and label the results of the QC calculation by

Table 4. Model parameters of single-qubit operations on an NMR-like quantum computer using rotating sinusoidal fields for the case ($k=1$, $N=1$, $M=4$), see (63). Parameters of model (53) that do not appear in this table are zero, except for the interaction $J=-0.43\times10^{-6}$ and the constant magnetic fields $h_1^z=1$ and $h_2^z=0.25$. The time-dependent Schrödinger equation is solved using a time step $\delta/2\pi=0.01$.

	$\tau/2\pi$	ω	$\tilde{h}_1^x$	$\tilde{h}_2^x$	φ_x	$\tilde{h}_1^y$	$\tilde{h}_2^y$	φ_y
X_1	8	1.00	-0.0312500	-0.0078125	$-\pi/2$	-0.0312500	-0.0078125	0
X_2	128	0.25	-0.0078125	-0.0019531	$-\pi/2$	-0.0078125	-0.0019531	0
Y_1	8	1.00	0.0312500	0.0078125	0	0.0312500	0.0078125	$\pi/2$
Y_2	128	0.25	0.0078125	0.0019531	0	0.0078125	0.0019531	$\pi/2$
X_1'	8	1.00	0.0559593	0.0139898	$-\pi/2$	0.0559593	0.0139898	0
X_2'	128	0.25	0.0445131	0.0111283	$-\pi/2$	0.0445131	0.0111283	0
Y_1'	8	1.00	-0.0559593	-0.0139898	0	-0.0559593	-0.0139898	$\pi/2$
X_1''	8	1.00	0.0872093	0.0218023	$-\pi/2$	0.0872093	0.0218023	0
X_2''	128	0.25	0.0523256	0.0130914	$-\pi/2$	0.0523256	0.0130914	0

the subscript $s=2kMN^2$. For reference we present the set of parameters corresponding to $s=8$ ($k=1$) in Table 4. Multiplying s (the duration of the sinusoidal pulse) with the unit of time (2 ns for the case at hand) shows that in our simulations, single-qubit operations are implemented by using short pulses that are, in NMR terminology, non-selective and hard. In contrast to the analytical treatment given in section 5.1, in all our simulations the interaction J is non-zero.

The two-qubit operation I' can be implemented by letting the system evolve in time according to Hamiltonian H_{NMR}, given by (51). I' is the same for both an ideal or NMR-like QC. Note that the condition $\tau J=-\pi$ yields $\tau/2\pi=1162790.6977$, a fairly large number (compared to $h_1^z=1$, see (51)).

5.4. Results

As a first check we execute all sequences on a QCE implementation of the ideal QC and confirm that they give the exact answers (results not shown). It is also necessary to rule out that the numerical results depend on the time step δ used to solve the time-dependent Schrödinger equation. The numerical error of the product formula used by QCE is proportional to δ^2 [52]. It goes down by a factor of about one hundred if we reduce the time step by a factor of 10. Within the two-digit accuracy used to present our data, there is no difference between the results for $\delta=0.01$ and $\delta=0.001$.

In Table 5 we present simulation results for CNOT^5 acting on one of the basis states and Y_1CNOT^5 acting on a singlet state, using the three logically equivalent but physically different implementations CNOT$_1$, CNOT$_2$ and CNOT$_3$ (see Eq.(67)). It is clear that some of the least accurate implementations ($s=8$) do not reproduce the correct answers if the input corresponds to one of the four basis states. Moreover, if the operations act on the exact singlet state, the results strongly depend on the CNOT implementation if $s\le32$. In agreement with the theoretical analysis given above, the exact results are recovered if s is sufficiently large. The pulses used in these simulations are so short that the presence of a non-zero J has a negligible effect on the single-qubit pulses. These simulations convincingly demonstrate that in order for a QA to work properly, it is not sufficient to show that it correctly operates on the basis states.

In contrast to computation in the classical framework, quantum computation can make use of entangled states. At the point where the QA actually uses an entangled state, the QA is most sensitive to (accumulated) phase errors. As another illustration of this phenomenon, we present in Table 6 some typical results obtained by executing Grover's database search algorithm. We used the same NMR-like QC as for the CNOT calculations. We conclude that reasonably good answers are obtained if $s\ge32$, in concert with our observations for the CNOT operation.

5.5. Physical QCs: generic features

In contrast to a conventional computer, a QC accepts as input linear superpositions of basis states and can return superpositions as well. If a quantum gate correctly operates on each of the basis states, it will

Table 5. Expectation values of the two qubits as obtained by performing a sequence of five CNOT operations on an NMR-like quantum computer. The initial states $|10\rangle$, $|01\rangle$, $|11\rangle$, and $|singlet\rangle = (|01\rangle - |10\rangle)/\sqrt{2}$ have been prepared by starting from the state $|00\rangle$ and performing exact rotations of the spins. The CNOT operations on the singlet state are followed by a $\pi/2$ rotation of spin 1 to yield a non-zero value of qubit 1. The time $s = \tau/2\pi = 2kMN^2$ determines the duration and strength of the sinusoidal pulses through relations (63), see Table 4 for the example of the case $s = 8$. The CNOT operation itself was implemented by applying the CNOT sequence given by (67). On an ideal quantum computer, CNOT^4 is the identity operation. For $s = 256$ all results are exact within an error of 0.01.

	Ideal QC		$s=8$		$s=16$		$s=32$		$s=64$	
Operation	Q_1^z	Q_2^z	Q_1^z	Q_2^z	Q_1^z	Q_2^z	Q_1^z	Q_2^z	Q_1^z	Q_2^z
$(\text{CNOT}_1)^5\|00\rangle$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$(\text{CNOT}_2)^5\|00\rangle$	0.00	0.00	0.24	0.76	0.50	0.26	0.20	0.07	0.06	0.02
$(\text{CNOT}_3)^5\|00\rangle$	0.00	0.00	0.23	0.76	0.50	0.26	0.20	0.07	0.06	0.02
$(\text{CNOT}_1)^5\|10\rangle$	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$(\text{CNOT}_2)^5\|10\rangle$	1.00	1.00	0.76	0.24	0.50	0.74	0.80	0.93	0.95	0.98
$(\text{CNOT}_3)^5\|10\rangle$	1.00	1.00	0.77	0.24	0.50	0.74	0.80	0.93	0.95	0.98
$(\text{CNOT}_1)^5\|01\rangle$	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00
$(\text{CNOT}_2)^5\|01\rangle$	0.00	1.00	0.24	0.24	0.51	0.74	0.20	0.93	0.06	0.98
$(\text{CNOT}_3)^5\|01\rangle$	0.00	1.00	0.23	0.24	0.51	0.74	0.20	0.93	0.06	0.98
$(\text{CNOT}_1)^5\|11\rangle$	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
$(\text{CNOT}_2)^5\|11\rangle$	1.00	0.00	0.76	0.76	0.50	0.26	0.80	0.07	0.95	0.02
$(\text{CNOT}_3)^5\|11\rangle$	1.00	0.00	0.77	0.76	0.50	0.26	0.80	0.07	0.95	0.02
$Y_1(\text{CNOT}_1)^5\|singlet\rangle$	1.00	1.00	0.90	1.00	0.03	1.00	0.58	1.00	0.88	1.00
$Y_1(\text{CNOT}_2)^5\|singlet\rangle$	1.00	1.00	0.98	0.24	0.95	0.74	0.98	0.93	0.99	0.98
$Y_1(\text{CNOT}_3)^5\|singlet\rangle$	1.00	1.00	0.79	0.24	0.55	0.74	0.82	0.93	0.95	0.98

also do so for any general linear superposition unless the operation generates additional phase factors that depend on the input state. Of course this does not happen on an ideal QC but, as we have seen in the simple case of a 2-qubit NMR-like QC, on a realistic QC it is bound to occur.

For each realization of QC hardware, there is a one-to-one correspondence between the QA and the unitary matrix that transforms the superposition accordingly. A QA will operate correctly under *all* circumstances if the *whole* unitary matrix representing the QA is a good approximation to the ideal one. In other words, the magnitude and the phase of *all* matrix elements should be close to their ideal values. It is not sufficient to have, for instance, two different CNOT gates that operate correctly by themselves: Also the relative phases that they generate should match. For n qubits there are $2^n(2^n - 1)$ complex numbers that specify the unitary matrix corresponding to a QA. All these numbers should be close to their ideal values, otherwise the QA is bound to produce wrong answers for some inputs.

6. Summary and outlook

In this paper we have shown how the Quantum Computer Emulator (QCE) can be programmed to execute a variety of quantum algorithms on different types of quantum computer hardware. An important topic that we did not touch upon at all is the effect of the interaction of the quantum computer with its environment (dissipation, decoherence).

Dissipation cannot be treated within the context of the models that the QCE can solve: Instead of solving the time-dependent Schrödinger equation we need to solve the equations of motion of the full density matrix. Although still feasible for a small number of qubits N, the computation time now scales with 2^{2N} instead of with 2^N. The attractive features of QCE, interactivity and real-time performance, are then rapidly lost if the number of qubits increases. On the other hand, important aspects of decoherence can be studied within the general model (2) with (10). Hence QCE can be used for this purpose as well. Exploring these interesting aspects is left for future research.

Table 6. Expectation values of the two qubits as obtained by running Grover's database search algoritm on an NMR-like quantum computer. The time $s = \tau/2\pi = 2kMN^2$ determines the duration and strength of the sinusoidal pulses through relations (63), see Table 4 for the example of the case $s = 8$. Within two-digit accuracy all results for $s = 256$ are exact.

	Ideal QC		$s=8$		$s=16$		$s=32$		$s=64$	
Item position	Q_1^z	Q_2^z	Q_1^z	Q_2^z	Q_1^z	Q_2^z	Q_1^z	Q_2^z	Q_1^z	Q_2^z
0	0.00	0.00	0.48	0.53	0.15	0.16	0.04	0.04	0.01	0.01
1	1.00	0.00	0.52	0.50	0.85	0.15	0.96	0.04	0.99	0.01
2	0.00	1.00	0.55	0.48	0.15	0.84	0.04	0.96	0.01	0.99
3	1.00	1.00	0.45	0.50	0.85	0.85	0.96	0.96	0.99	0.99

Acknowledgement

Support from the "Nederlandse Stichting Nationale Computer Faciliteiten (NCF)" is gratefully acknowledged.

References

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

[2] P. Shor, *Proc. 35th Annu. Symp. Foundations of Computer Science*, ed. S. Goldwasser (IEEE Computer Soc., Los Alamitos CA, 1994) p. 124.

[3] P.W. Shor, *SIAM Review*, **41**, (1999), 303.

[4] L.K. Grover, *Proc. 28th Annual ACM Symposium of Theory of Computing* (ACM, Philadelphia, 1996).

[5] L.K. Grover, *Phys. Rev. Lett.*, **79**, (1997), 325.

[6] R.P. Feynman, *Int. J. Theor. Phys.*, **21**, (1982), 467.

[7] S. Lloyd, *Science*, **261**, (1993), 1569.

[8] D.P. DiVincenzo, *Science*, **270**, (1995), 255.

[9] A. Ekert, R. Jozsa, *Rev. Mod. Phys.*, **68**, (1996), 733.

[10] V. Vedral, M. Plenio, *Progress in Quantum Electronics*, **22** (1998) 1.

[11] J.J. Cirac, P. Zoller, *Phys. Rev. Lett.*, **74**, (1995), 4091.

[12] C. Monroe, D.M. Meekhof, B.E. King, W.M. Itano, D.J. Wineland, *Phys. Rev. Lett.*, **75**, (1995), 4714.

[13] T. Sleator, H. Weinfurther, *Phys. Rev. Lett.*, **74**, (1995), 4087.

[14] P. Domokos, J.M. Raimond, M. Brune, S. Haroche, *Phys. Rev.*, **A52**, (1995) 3554.

[15] J.A. Jones, M. Mosca, *J. Chem. Phys.*, **109**, (1998), 1648.

[16] J.A. Jones, M. Mosca, R.H. Hansen, *Nature*, **393**, (1998), 344.

[17] I.L. Chuang, L.M.K. Vandersypen, Xinlan Zhou, D.W. Leung, S. Lloyd, *Nature*, **393**, (1998), 143.

[18] I.L. Chuang, N. Gershenfeld, M. Kubinec, *Phys. Rev. Lett.*, **80**, (1998), 3408.

[19] B.E. Kane, *Nature*, **393**, (1998), 133.

[20] Y. Makhlin, G. Schön, A. Shnirman, *Nature*, **398**, (1999), 305.

[21] Y. Nakamura, Yu. A. Pashkin, J.S. Tsai, *Nature*, **398**, (1999), 786.

[22] G. Nogues, A. Rauschenbeutel, S. Osnaghi, M. Brune, J.M. Raimond, S. Haroche, *Nature*, **400**, (1999), 239.

[23] K. Mølmer, A. Sørensen, *Phys. Rev. Lett.*, **82**, (1999), 1835.

[24] A. Sørensen, K. Mølmer, *Phys. Rev. Lett.*, **82**, (1999), 1971.

[25] R. Fazio, G.M. Palma, J. Siewert, *Phys. Rev. Lett.*, **83**, (1999), 5385.

[26] T.P. Orlando, J.E. Mooij, Lin Tian, C.H. van der Wal, L.S. Levitov, S. Lloyd, J.J. Mazo, *Phys. Rev.*, **B60**, (1999), 15398.

[27] A. Blias, A. Zagoskin, *Phys. Rev.*, **A61**, (2000), 042308.

[28] M.C. de Oliveira, W.J. Munro, *Phys. Rev.*, **A61**, (2000), 042309.

[29] G. Toth, S. Lent, *Phys. Rev.*, **A63**, (2001), 052315.

[30] L.M.K. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, R. Cleve, I.L. Chuang, *Phys. Rev. Lett.*, **85**, (2000), 5452.

[31] R. Marx, A.F. Fahmy, J.M. Meyers, W. Bernel, S.J. Glaser, *Phys. Rev.*, **A62**, (2000), 012310.

[32] E. Knill, R. Laflamme, R. Martinez, C.-H. Tseng, *Nature*, **404**, (2000), 368.

[33] D.G. Cory, R. Laflamme, E. Knill, L. Viola, T.F. Havel, N. Boulant, G. Boutis, E. Fortunato, S. Lloyd, R. Martinez, C. Negrevergne, M. Pravia, Y. Sharf, G. Teklemariam, Y.S. Weinstein, W.H. Zurek, *Fortschr. Phys.*, **48**, (2000), 875.

[34] J.A. Jones, *Fortschr. Phys.*, **48**, (2000), 909.

[35] L.M.K. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, M.H. Sherwood, I.L. Chuang, *Nature*, **414**, (2001), 883.

[36] L.M.K. Vandersypen, Ph.D. Thesis, Dept. of Electrical Engineering, Stanford University, 2001.

[37] H. De Raedt, A.H. Hams, K. Michielsen, K. De Raedt, *Comp. Phys. Comm.*, **132**, (2000), 1.

[38] QCE can be downloaded from http:/www.compphys.org/qce.htm

[39] K. Michielsen, H. De Raedt, K. De Raedt, *Nanotechnology*, **13**, (2002), 23.

[40] H. De Raedt, K. Michielsen, A. Hams, S. Miyashita, K. Saito, *Eur. Phys. J.*, **B27**, (2002), 15.

[41] G.P. Berman, G.D. Doolen, D.D. Holm, V.I. Tsifrinovich, *Phys. Lett.*, **A193**, (1994), 444.

[42] C.P. Slichter, *Principles of Magnetic Resonance* (Springer, Berlin, 1990).

[43] R. Ernst, G. Bodenhausen, A. Wokaun, *Principles of Nuclear Magnetic Resonance in One and Two Dimensions* (Oxford University Press, New York, 1987).

[44] M. Suzuki, S. Miyashita, A. Kuroda, *Prog. Theor. Phys.*, **58**, (1977), 1377.

[45] M. Suzuki, *Proc. Japan Acad.*, **69** *Ser. B*, (1993), 161.

[46] http://www.compphys.org/QCE/help.htm

[47] A. Barenco, C.H. Bennett, R. Cleve, D.P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J.A. Smolin, H. Weinfurter, *Phys. Rev.*, **A52**, (1995), 3457.

[48] T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Introduction to Algorithms* (MIT Press, Cambridge, 1994).

[49] S. Bettelli, PhD Thesis, University of Trento, 2002.

[50] H. De Raedt, K. Michielsen, K. De Raedt, S. Miyashita *Phys. Lett.*, **A200**, (2001), 227.

[51] R. Freeman, *Spin Choreography* (Spektrum, Oxford, 1997).

[52] H. De Raedt, *Comp. Phys. Rep.*, **7**, (1987), 1.