Quantum Computer Simulator Based on the Circuit Model of Quantum Computation

Ioannis G. Karafyllidis, *Member, IEEE*

***Abstract*—A quantum computer simulator is presented. This simulator is an engineering work and no deep understanding of quantum mechanics is required from the user. The simulator is based on the circuit model of quantum computation in which quantum gates act on quantum registers which comprise a number of quantum bits (qubits). The inputs to the simulator are the initial states of the qubits that form a quantum register and the quantum gates applied at each computation step. The inputs are entered through a graphical user interface. The outputs of the simulator are the matrices that represent the quantum register state at each quantum computation step and graphical outputs that show the probability of measuring each one of the possible quantum register base states and the phase of each state at each computation step. The well-known Deutsch's algorithm and the quantum Fourier transform, which is the base of many quantum algorithms, are presented using this simulator. Furthermore, the generation and variation of entanglement during quantum computations can be calculated using this simulator. The quantum computer simulator is a useful tool for the study of quantum computer circuits, quantum computing, and the development of new quantum algorithms.**

***Index Terms*—Entanglement, quantum alogorithms, quantum computer, quantum computing, simulation.**

I. Introduction

QUANTUM computation is one of the major challenges for computer science and engineering in the twenty-first century [1], [2]. Quantum computers are far more efficient that their classical counterparts in factoring large numbers [3], searching databases [4]–[6], and simulating physical systems [7]. All the aspects of quantum computing are currently under intensive study and, as a result, the first quantum algorithms appeared before the actual fabrication of the first quantum gate [8], [9].

Current technology does not allow yet the implementation of quantum computers that can manipulate a sufficient number of qubits. In cases where physical system implementation is very difficult, expensive or time consuming, simulation can fill the gap by producing models of the physical system and by mimicking its operation based on the mathematical and physical description of the system. Simulation has been extensively used in the study of the emerging nanoelectronic circuits [10]–[12], and will probably become an indispensable tool for studying quantum computer circuits and developing new quantum algorithms.

Manuscript received December 20, 2004; revised March 9, 2005. This paper was recommended by Associate Editor A. I. Csurgay.

The author is with the Department of Electrical and Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece (e-mail: ykar@ee.duth.gr).

Digital Object Identifier 10.1109/TCSI.2005.851999

The quantum computer simulator presented here is based on the circuit model of quantum computation [13]. In this model quantum computations and quantum algorithms are represented by circuits, which comprise quantum gates and quantum registers. Earlier versions of this simulator have been used for the visualization of the quantum Fourier transform [14], the study of cellular quantum computer architecture [15], the simulation of entanglement generation and variation in quantum computations [16] and the definition and evolution of quantum cellular automata [17]. The complete simulator presented here comprises a graphical user interface used for entering the inputs, which are the initial states of the quantum bits (qubits) that form a quantum register and the quantum gates applied at each computation step. The quantum computation is visualized using two graphical outputs showing probability of measuring each one of the possible quantum register base states and the phase of each state at each computation step.

The simulator presented here is an engineering work and has been developed for the study of quantum computation, quantum circuits and for the development of new quantum algorithms by users with no deep understanding of quantum mechanics. The simulator is based on the circuit model of quantum computation and, unlike other simulators, is independent of the quantum computer fabrication technology. Unitary evolutions and tensor product calculations in Hilbert space are completely hidden form the user, allowing her/him to concentrate on the computation and its possible applications. Furthermore, the novelty of this simulator is the effective visualization of quantum computations by producing graphs of the probability to measure each quantum register base state and its phase at each quantum computation step. This visualization of quantum computation is an advantage of this simulator, since all the information that can be extracted from a quantum computation, which is the variation of probability distribution in base states and the phase variation, is represented in two graphs. The generation and variation of entanglement (which is considered as a natural resource) during quantum computations can be calculated using this simulator. This simulator can also be used to explore the emerging fields of quantum computer engineering applications such as modeling and simulation of physical systems and processes [18]–[20] and pattern recognition and image processing using quantum computers [21]–[23]. The simulator is free and can be obtained by sending an e-mail to the author.

This paper is organized as follows. The quantum computer simulator is described in Section II, simulated quantum computations and quantum algorithms are presented in Section III and the conclusions are presented in Section IV.

(1). *Start*

(2). *Read* the number of qubits in the quantum register and the number of computation steps.

(3). *Read* the initial state of the quantum register.

(4). *Read* the quantum gate configuration at each computation step.

(5). *Calculate* the tensor product of the initial state of the quantum register qubits.

(6). *Take* the first computation step (*i=1*).

(7). *Calculate* the tensor product of the quantum gate matrices applied at the *ith* computation step.

(8). *Calculate* the new state of the quantum register.

(9). *Calculate* the phase of the quantum register state.

(10). *If* the number of computation steps is less that or equal to the number defined by the user set *i=i+1* and *go to* step (7) *else,* continue.

(11). *Write* the probability amplitudes and probability distribution at each computation step.

(12) *Write* the final quantum register state.

(12). *Produce* graphical output.

Fig. 1. Pseudocode of the basic structure of the quantum computer simulator.

II. Description of the Quantum Computer Simulator

The pseudocode of the basic structure of the quantum computer simulator is shown in Fig. 1. The inputs to the simulator are the number of the qubits in the quantum register, the number of computation steps, the initial state of the qubits that form a quantum register and the gates applied at each computation step.

All inputs are entered through a graphical user interface, which is shown in Fig. 2. The leftmost column is the quantum register and each column represents the quantum gates applied at each computation steps. The user can enter a variety of quantum gates such as, Hadamard, phase shift, controlled-phase shift, controlled-NOT, controlled-controlled-NOT, and Fredkin gates. In the case of phase gates, the user enters the value of the phase in a box. If more that 6 qubits are to be used or the computation comprises more than 13 steps the input data are entered as text files.

The user enters the number of qubits and the number of computation steps, which in the case of Fig. 2, are 2 and 4, respectively. After that the user enters the initial state of the quantum register. In the case of Fig. 2 the initial state is: $|1\rangle|1\rangle = |11\rangle$.

The quantum computation shown in Fig. 2 comprises two qubits and five steps. The computation data are entered in the two last rows and in the first five columns of the graphical user interface. The first step is the initial state of the quantum register. In the second step a Hadamard gate (H) is applied to qubit 1 and no gate (I) to qubit 0. In the third step the controlled-NOT gate is applied with qubit 1 as control (Cntr) and qubit 0 as target (CN). The same gate is applied in the fourth step. In the fifth step a Hadamard gate (H) is applied to qubit 1 and no gate (I) to qubit 0. The circuit that describes this quantum computation is shown in Fig. 3(a). This circuit entangles and disentangles the two qubits in the quantum register [8], [13].

In Fig. 3(b) a matrix is shown. The columns of this matrix represent the computation steps and the rows the possible base states of the quantum register, which are also written in decimal form. The first computation step is the initial state of the quantum register, which in this case is $|11\rangle$. A measurement of the quantum register at this step will give the state $|11\rangle$ with probability equal to 1. All matrix elements in the first column are zero except the element that corresponds to state 3 (in decimal), which is 1. At the second step a Hadamard gate is applied to the qubit 1 and sets its state in base-state superposition. A measurement of the quantum register state at the end of this step will give states $|01\rangle$ and $|11\rangle$ with probability 0.5 and the matrix elements that correspond to states 1 and 3 (in decimal) are 0.5. A controlled-NOT gate is applied at the third step. A measurement at the end of this step will give states $|01\rangle$ and $|10\rangle$ (1 and 2 in decimal) with probability 0.5 each, and so on. A graphical representation of this matrix is shown in Fig. 3(c). The x-axis represents the computation steps (matrix columns) and the y-axis the possible quantum register base states in decimal form (matrix rows).

Each matrix element corresponds to a rectangle and its value, i.e., the probability to measure this state at this computation step, is represented as a gray-scale image in which probability 1 is represented by black and probability 0 by white. Fig. 3(d) shows the phase of each quantum register state at each computation step. An arrow with its initial point located at the point (i, j) corresponds to the phase of the $j\ th$ quantum register state (in decimal) at the $i\ th$ computation step. The direction of the arrow indicates the value of the phase, which can take any value between 0° and 360°. An arrow parallel to the x-axis pointing to the right corresponds to a phase angle of 0°. If the state j has zero probability to be measured at step i, there exist no arrow with initial point at (i, j).

According to the hidden variable theorem no classical system can simulate completely a quantum system. Therefore, the complete visualization of the quantum computer operation using a classical computer is impossible. The visualization shown in Fig. 3(c) and (d) is an effective visualization that depicts the most important aspects of quantum computations.

It is well known that the computational complexity of any algorithm simulating quantum systems with two base states is $O(2^n)$, where n is the number of quantum systems. This simulator is no exception to that and suffers from exponential slowdown as the number of qbits increases. Matrices representing qbits and quantum gates contain a large number of zeros and the computation complexity was somewhat improved using sparse matrix techniques.

III. Application of the Quantum Computer Simulator

A. *Simulation of Deutsch's Quantum Algorithm*

The problem posed by Deutsch is: "Given a binary function $f(x)$ find out if the function is constant (i.e., $f(0) = f(1)$) or balanced (i.e., $f(0) \neq f(1)$) by performing the minimum number of computations." On a classical computer one has to compute the value of $f(x)$ twice and compare the results, but, as Deutsch showed, on a quantum computer one computation is enough [24].

The key element in Deutsch's algorithm is the quantum circuit shown in Fig. 4, which acts on a two-qubit quantum register $|x\rangle|a\rangle$ and leaves the $|x\rangle$ qubit in its previous state, but takes the other qubit to a state which is the modulo 2 sum of a and $f(x)$. The action of this circuit is given by

$$|x\rangle|a\rangle \xrightarrow{U} |x\rangle|a \oplus f(x)\rangle. \qquad (1)$$

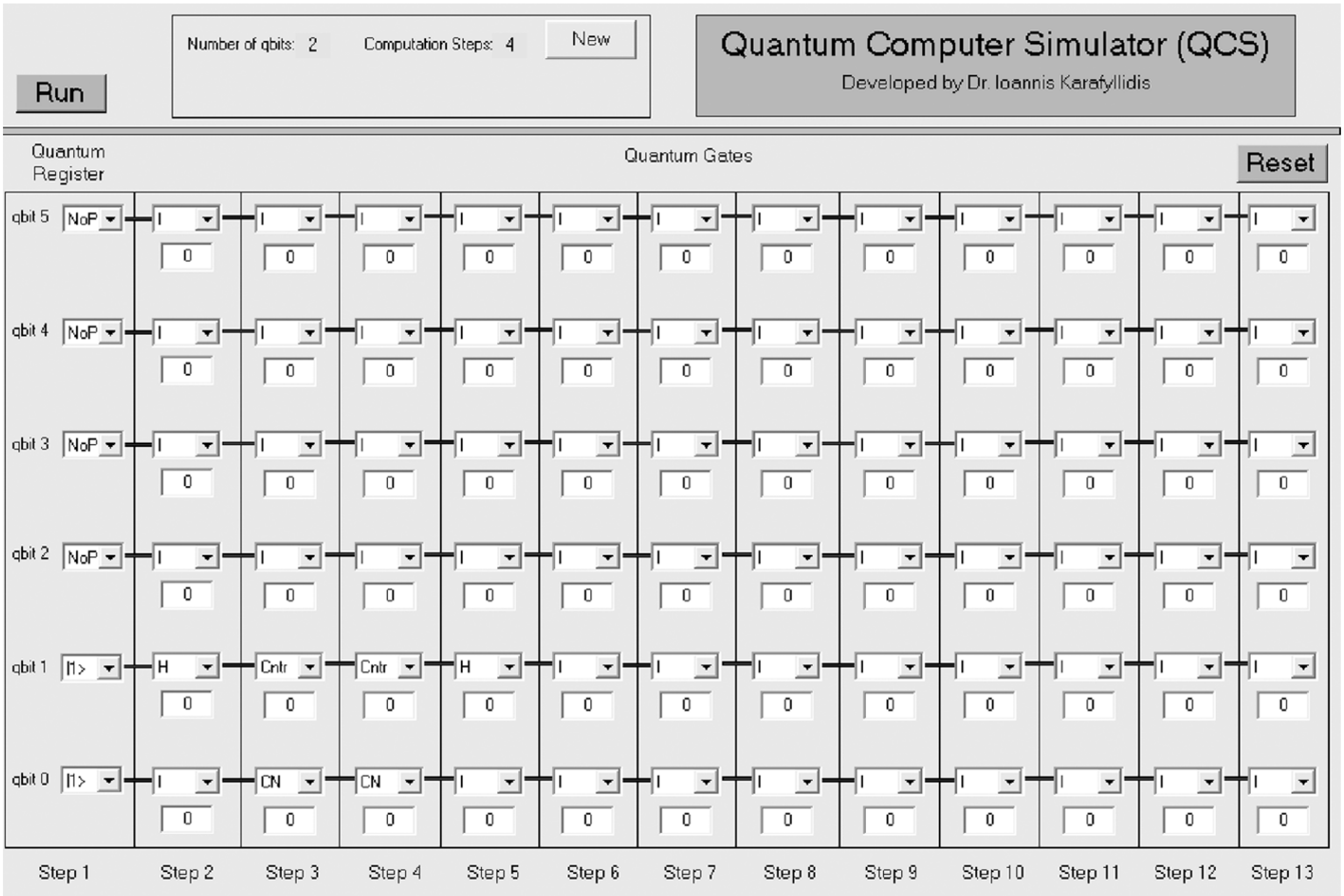

Fig. 2. Graphical user interface of the quantum computer simulator.

The quantum circuit of Deutsch's algorithm is shown in Fig. 5(a). The initial state of the two qubits are $|0\rangle|1\rangle$. This is considered as the first computation step. At the second step two Hadamard (H) quantum gates are applied to the qubits changing their states according to

$$|0\rangle|1\rangle \xrightarrow{H} \left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]. \tag{2}$$

At the third step the quantum circuit of Fig. 4 is applied and the states of the two qubits are taken to [12]

$$\left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \xrightarrow{H} \begin{cases} \pm\left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] & \text{if } f(0)=f(1) \\ \pm\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] & \text{if } f(0)\neq f(1). \end{cases} \tag{3}$$

At the fourth step an H gate is applied to the upper qubit and the states of the qubits are taken to

$$\begin{aligned} &\pm\left[\tfrac{|0\rangle+|1\rangle}{\sqrt{2}}\right]\left[\tfrac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \xrightarrow{H} \pm|0\rangle\left[\tfrac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \quad \text{if } f(0)=f(1) \\ &\pm\left[\tfrac{|0\rangle-|1\rangle}{\sqrt{2}}\right]\left[\tfrac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \xrightarrow{H} \pm|1\rangle\left[\tfrac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \quad \text{if } f(0)\neq f(1). \end{aligned} \tag{4}$$

A measurement of the upper qubit will result in $|0\rangle$ if $f(0)=f(1)$ and in $|1\rangle$ if $f(0)\neq f(1)$.

Fig. 5(b) and 5(c) shows the simulation of Deutsch's algorithm in the case where $f(x)=x$. In this case the circuit of Fig. 4 is a controlled-NOT gate. We have $f(0)=0$ and $f(1)=1$, therefore $f(0)\neq f(1)$ and the upper qubit should be $|1\rangle$. Fig. 5(b) shows that at the end of the fourth step two quantum register base states are possible each with probability 0.5: the state $|1\rangle|0\rangle$ (decimal 2) and the state $|1\rangle|1\rangle$ (decimal 3). In both states the upper qubit is in state $|1\rangle$, as expected. Fig. 5(c) shows the phase variation during the quantum computation. Note Fig. 5(b) shows the same data for steps two and three, in which all four quantum register base states are equally probable, but Fig. 5(c) shows differed phases for these two steps, which is very important in Deutsch's algorithm.

B. Simulation of the Quantum Fourier Transform

The quantum Fourier transform (QFT) is a unitary operator which acts on the Hilbert space and is used to force interference between qbits. QFT is a key subroutine of quantum algorithms for factoring and simulation and is the heart of the hidden- subgroup problem, the solution of which is expected to lead to the development of new quantum algorithms [8], [9], [13]. QFT acts on a base state of the N-dimensional Hilbert space as follows:

$$\mathrm{QFT}_N : |a\rangle \mapsto \frac{1}{\sqrt{N}}\sum_{c=0}^{N-1} e^{2\pi i \frac{ac}{N}}|c\rangle \tag{5}$$

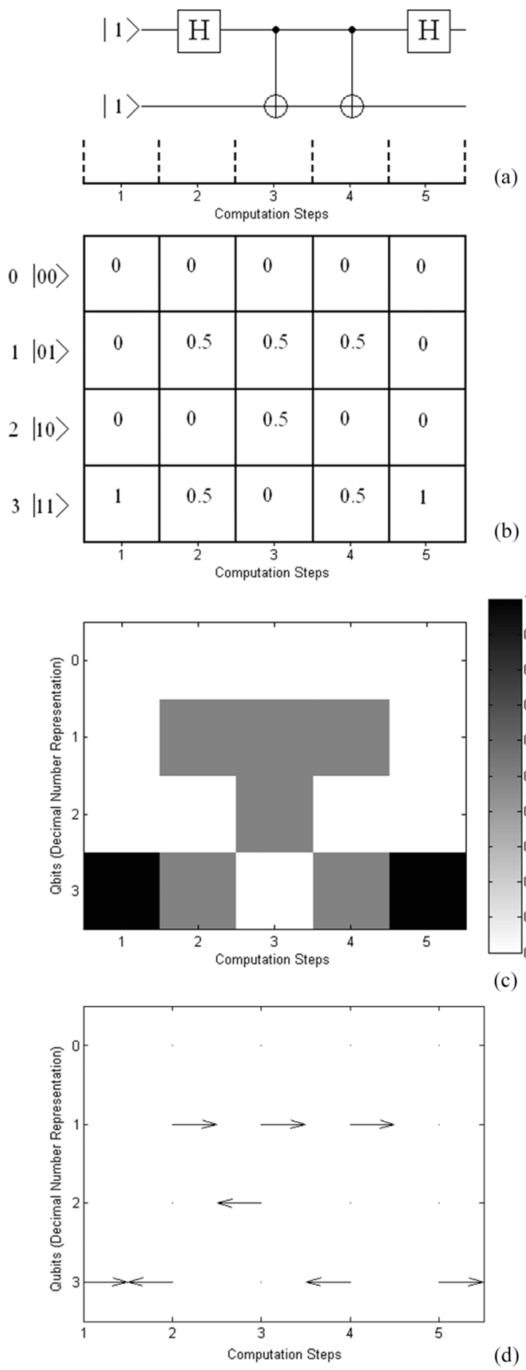

Fig. 3. Quantum computation and its simulation. (a) The circuit describing the quantum computation. (b) The matrix showing the probability of measuring each base state of the quantum register at each computation step. (c) The graphical output of the quantum computer simulator showing the probability distribution at each computation step. (d) The graphical output of the quantum computer simulator showing the phase variation during the quantum computation.

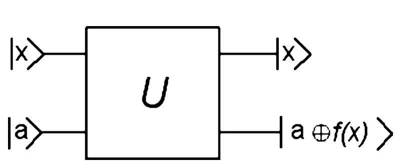

Fig. 4. Quantum circuit that takes $|x\rangle|a\rangle$ to $|x\rangle|a \oplus f(x)\rangle$.

where QFT_N is the symbol of the QFT on the N-dimensional Hilbert space, $|a\rangle$ and $|c\rangle$ are base states in decimal representa-

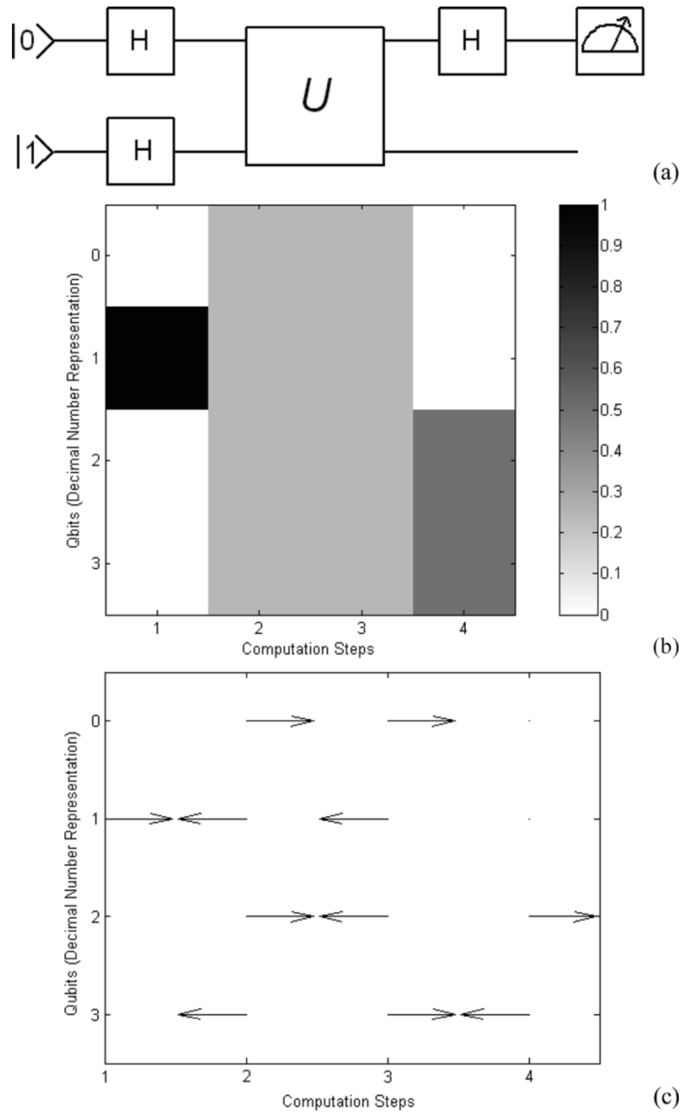

Fig. 5. (a) The quantum circuit of Deutsch's algorithm. (b) and (c) simulation of Deutsch's algorithm in the case of a controlled-NOT gate. (b)The probability distribution at each computation step. (c) The phase variation during the computation.

tion, and $(a\, c)$ is the ordinary multiplication of the two decimal numbers a and c. Any state vector in the N-dimensional Hilbert space can be expressed as a superposition of base states and QFT transforms it according to

$$\mathrm{QFT}_N : \sum_{a=0}^{N-1} x_a |a\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{c=0}^{N-1} \sum_{a=0}^{N-1} x_a e^{2\pi i \frac{ac}{N}} |c\rangle = \frac{1}{\sqrt{N}} \sum_{c=0}^{N-1} y_c |c\rangle \tag{6}$$

where y_c is the classical discrete Fourier transform (DFT) of x_a

$$\mathrm{DFT}_N : y_c \mapsto \sum_{a=0}^{N-1} e^{2\pi i \frac{ac}{N}} x_a. \tag{7}$$

The quantum circuits for the QFT comprise only two types of quantum gates, the Hadamard (H) and the controlled phase shift (Φ). The quantum circuit for the four qubit QFT is shown in Fig. 6(a). The initial state of the four qubit quantum register is $|0\rangle|1\rangle|0\rangle|1\rangle$ (decimal 5). Fig. 6(b) and (c) shows the simulation

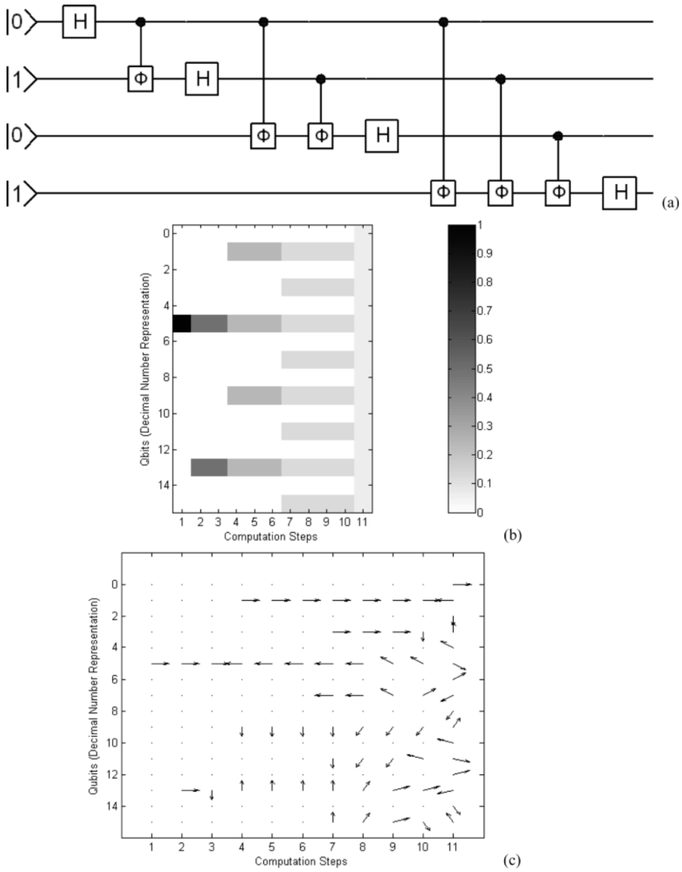

Fig. 6. (a) Quantum circuit of the four qubit QFT. (b) and (c) Simulation of the four qubit QFT.

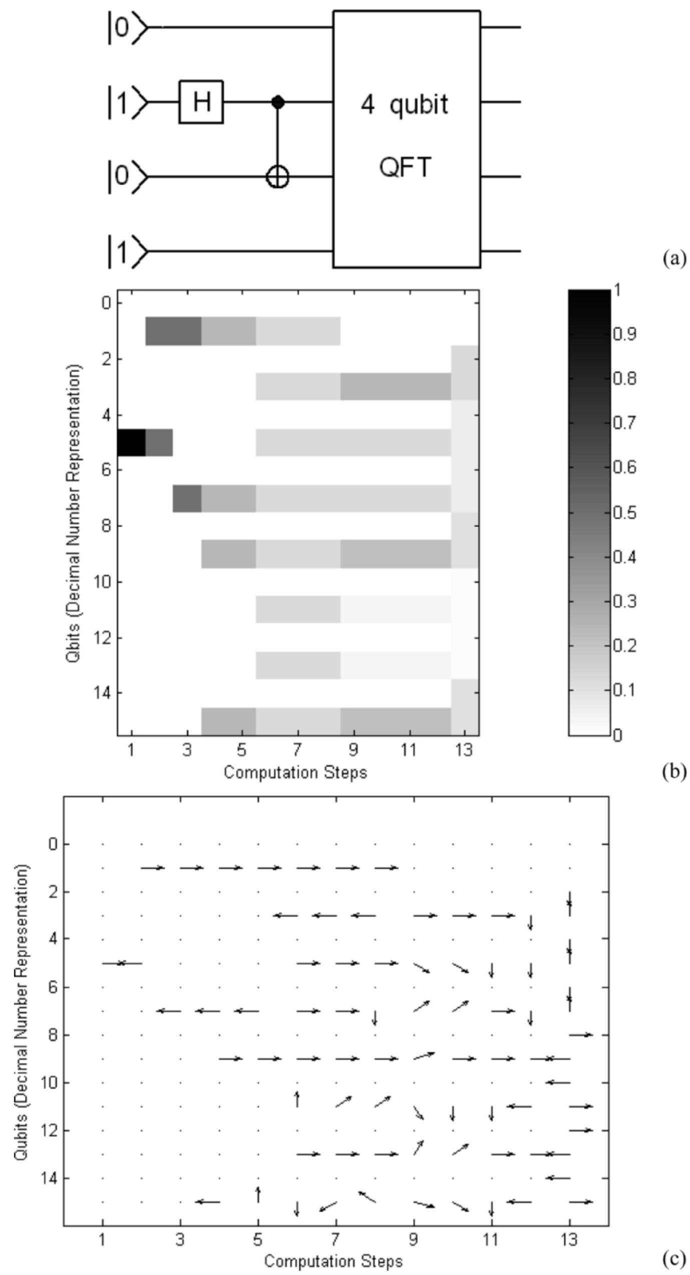

Fig. 7. (a) Quantum circuit of the four qubit QFT in which the second and third qubits are entangled. The "4 qubit QFT" block is the circuit of Fig. 6(a). (b) and (c) Simulation of the four qubit QFT with two entangled qubits.

of the four qubit QFT. Note that at the end of the computation all base states of the quantum register have the same probability to be measured, but their phases are different. The phase variation during the computation is very important in QFT.

In the quantum computation shown in Fig. 7(a) a four qubit quantum register with the same initial state as in Fig. 6(a) is used. The second and the third qubits are entangled using a Hadamard gate followed by a controlled-NOT gate and then a QFT follows. The block "4 qubit QFT" represents the circuit of Fig. 6(a). Fig. 7(b) and (c) shows the simulation of this quantum computation. Note that the base states $|0\rangle|0\rangle|0\rangle|0\rangle$ (decimal 0) and $|0\rangle|0\rangle|0\rangle|1\rangle$ (decimal 1) are not reached at the end of the computation, because of the entanglement. Simulations showed that different initial states and entanglement of different qubits followed by QFT, may bring the quantum register to desirable states. One of the important problems one faces in quantum computing is to prepare the quantum register, i.e., to take it to a desirable state which will be used as initial for a quantum computation to follow. The ability to prepare quantum registers using entanglement followed by QFT has been revealed using this simulator.

C. Calculation of Entanglement Generation and Variation During Quantum Computations

One of the major challenges in quantum computation is the characterization of entanglement. This characterization has three aspects: the qualification of entanglement (i.e., to determine if a given state is entangled or not), the quantification of entanglement (i.e., to calculate the amount of entanglement of an entangled state) and the manipulation of entanglement (i.e., to predict and control the generation and variation of entanglement in quantum computations).

A pure state is entangled if and only if its state vector $|\psi\rangle$ cannot be expressed as a tensor product of pure states of its parts $|\psi_A\rangle$ and $|\psi_B\rangle$

$$|\psi\rangle \neq |\psi_A\rangle \otimes |\psi_B\rangle. \tag{8}$$

A pure state's entanglement is measured by its entropy of entanglement

$$E(\psi) = S(\rho_A) = S(\rho_B) \tag{9}$$

i.e., the apparent entropy of either subsystem [13]. In (9) $S(\rho)$ is the von Neumann entropy given by

$$\begin{aligned} S(\rho_A) &= -Tr(\rho_A \log_2 \rho_A) \\ S(\rho_B) &= -Tr(\rho_B \log_2 \rho_B) \end{aligned} \tag{10}$$

where ρ_A and ρ_B are the reduced density matrices obtained as partial traces of the whole system's pure density matrix $\rho = |\psi\rangle\langle\psi|$ over the two subsystems

$$\begin{aligned} \rho_A &= Tr_B\left(|\psi\rangle\langle\psi|\right) \\ \rho_B &= Tr_A\left(|\psi\rangle\langle\psi|\right). \end{aligned} \tag{11}$$

In terms of the eigenvalues $\{\lambda_i\}$ of the density matrix ρ the von Neumann entropy is given by [8], [13]

$$S(\rho) = -Tr(\rho \log_2 \rho) = \sum_i -\lambda_i \log_2 \lambda_i. \tag{12}$$

The entanglement E is measured in entangled bits (ebits), which are defined as the amount of entanglement in a maximally entangled state of any pure bipartite system for which $E = 1$. In the case of a quantum register that comprises n qubits the system can be divided into two parts: the qubit j and the rest of the quantum register now comprising $n - 1$ qubits. In this case the entanglement is obtained as described earlier. A more indicative measure of entanglement is the sum of the entanglements calculated by considering each qubit as one of the parts and the rest qubits as the other part

$$E = \sum_{j=1}^{n} S(\rho_j) = \sum_{j=1}^{n} -Tr(\rho_j \log_2 \rho_j). \tag{13}$$

Using (13) the entanglement of a quantum register with n qubits, four of which are fully entangled is 4. Equations (9)–(13) are used for the calculation of entanglement in the quantum computer simulator presented here.

The quantum circuit of Fig. 8(a) represents a computation where the same configuration of quantum gates is applied three times. The repeated configuration starts with an H gate applied to the first qubit followed by six CN gates and ends with an H gate applied again to the first qubit. Fig. 8(b) shows the simulation of this quantum computation. The initial state is represented by decimal 0 and is shown in black. Fig. 8(c) shows the entanglement variation for this computation in ebits. The y-axis is the calculated entanglement measure at each computation step according to (9)–(13) and the x-axis numbers the computation steps.

Entanglement is generated after the application of the first CN gate, increases by one ebit each time a CN gate is applied and reaches its maximum value (4 ebits) after the application of the third CN gate. The application of the fifth and sixth CN gates followed by an H gate applied to the first qubit, fully disentangles the QR and the entanglement is now zero ebits. The application of the same quantum gate configuration two more times results in the same entanglement variation.

A similar quantum computation is shown in Fig. 9(a) where, after the application of an H gate, which sets in base-state superposition the first qubit, the quantum gate configuration enclosed in the dashed rectangle is applied nine times. The simulation of this computation is shown in Fig. 9(b) and the entanglement variation in Fig. 9(c). As shown in Fig. 9(c) entanglement is generated, the state of the QR becomes fully entangled, then partially disentangled and then fully entangled again. The same variation is repeated nine times. Quantum computa-

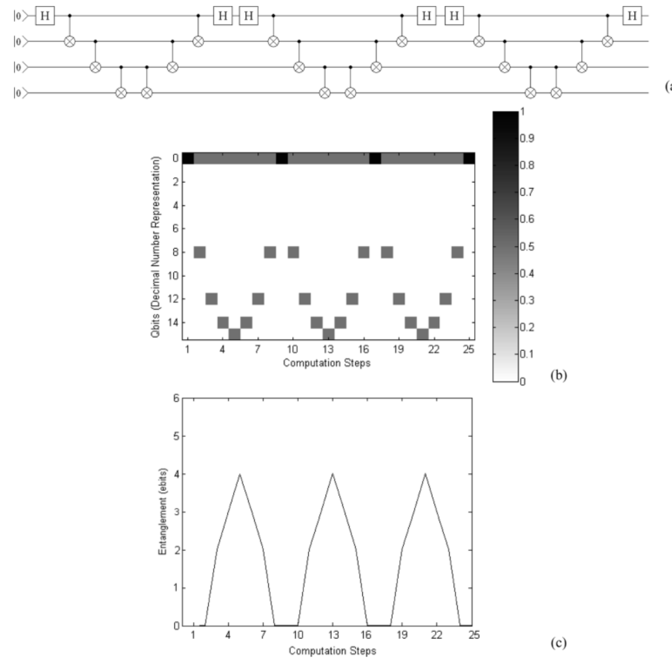

Fig. 8. (a) Quantum computation with a configuration of quantum gates applied three times. The configuration starts with an H gate applied to the first qubit followed by six CN gates and ends with an H gate applied again to the first qubit. (b) Simulation of this quantum computation. (c) Entanglement variation for this quantum computation.

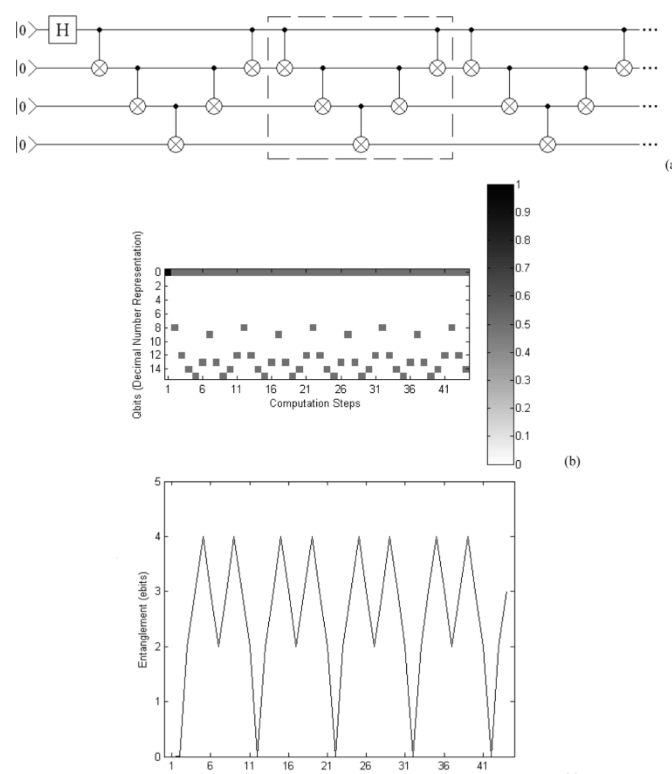

Fig. 9. (a) A quantum computation where after the application of an H gate to the first qubit, the quantum gate configuration enclosed in the dashed rectangle is applied nine times. (b) The simulation of this quantum computation. (c) Entanglement variation for this quantum computation.

tions of Figs. 8 and 9 indicate that it may be possible to produce periodic entanglement variation by applying repeatedly the same quantum gate configuration. This fact may lead to useful quantum computations and even in new quantum algorithms. An entangled chain is formed when each qubit in the QR is entangled with its neighboring qubits. Entangled chains are important in modeling physical phenomena using quantum computers and the simulation results show that manipulation of entanglement in these chains is possible [16].

IV. Conclusion

A quantum computer simulator based on the circuit model of quantum computation was presented. The novelty of this simulator is the effective visualization of quantum computations by producing graphs of the probability to measure each quantum register base state and its phase at each quantum computation step. Furthermore, this simulator calculates the generation and variation of entanglement during quantum computations. The ability to prepare quantum registers using entanglement followed by QFT and the production of periodic entanglement variation by applying repeatedly the same quantum gate configuration have been revealed using this simulator. No deep understanding of quantum mechanics is required from the user. This simulator can be used to explore new quantum computations, such as quantum register preparation and can be a useful tool for the study of quantum computing and the development of new quantum algorithms.

References

[1] R. J. Hughes and C. P. Williams, "Quantum computing: the final frontier?," *IEEE Intell. Syst.*, vol. 15, no. 5, pp. 10–18, 2000.
[2] A. M. Steane and E. G. Rieffel, "Beyond bits: the future of quantum information processing," *IEEE Computer*, vol. 33, no. 1, pp. 38–45, 2000.
[3] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.
[4] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.*, vol. 79, no. 2, pp. 325–328, 1997.
[5] ——, "Quantum computers can search rapidly by using almost any transformation," *Phys. Rev. Lett.*, vol. 80, no. 19, pp. 4329–4332, 1998.
[6] C. P. Williams, "Quantum search algorithms in science and engineering," *IEEE Comput. Sci. Eng.*, vol. 3, no. 2, pp. 44–51, 2001.
[7] S. Lloyd, "Universal quantum simulators," *Science*, vol. 273, no. 5278, pp. 1073–1078, 1996.
[8] C. P. Williams and S. H. Clearwater, *Explorations in Quantum Computing*. New York: Springer-Verlag, Telos, 1998.
[9] G. P. Berman, G. D. Doolen, R. Mainieri, and V. I. Tsifrinovich, *Introduction to Quantum Computers*. London, U.K.: World Scientific, 1998.
[10] I. Karafyllidis, "A simulator for single-electron devices and circuits based on simulated annealing," *Superlattices and Microstructures*, vol. 25, no. 4, pp. 567–572, 1999.
[11] ——, "Design and simulation of a single-electron random-access memory array," *IEEE Trans. Circuits Syst. I*, vol. 49, no. 9, pp. 1370–1375, 2002.
[12] G. T. Zardalidis and I. Karafyllidis, "Design and simulation of a nanoelectronic single-electron universal Fredkin gate," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 12, pp. 2395–2403, Dec. 2004.
[13] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge University Press, 2000.
[14] I. Karafyllidis, "Visualization of the quantum Fourier transform using a quantum computer simulator," *Quantum Inform. Processing*, vol. 2, no. 4, pp. 271–288, 2003.
[15] ——, "Cellular quantum computer architecture," *Phys. Lett. A*, vol. 320, no. 1, pp. 35–38, 2003.
[16] ——, "Simulation of entanglement generation and variation in quantum computation," *J. Computat. Phys.*, vol. 200, no. 1, pp. 383–397, 2004.
[17] ——, "Definition and evolution of quantum cellular automata with two qubits per cell," *Phys. Rev. A*, vol. 70, pp. 0 044 301-1–0 044 301-4, 2004.
[18] D. A. Meyer, "Physical quantum algorithms," *Computer Phys. Commun.*, vol. 146, no. 3, pp. 295–301, 2002.
[19] G. Ortiz, E. Knill, and J. E. Gubernatis, "The challenge of quantum computer simulations of physical phenomena," *Nuclear Phys. B (Proceedings Supplements)*, vol. 106–107, no. 1, pp. 151–158, 2002.
[20] J. Yepez, "Quantum computation for physical modeling," *Computer Phys. Commun.*, vol. 146, no. 3, pp. 277–279, 2002.
[21] R. Schützhold, "Pattern recognition on a quantum computer," *Phys. Rev. A*, vol. 67, no. 6, pp. 062 311-1–062 311-6, 2003.
[22] M. Sasaki, A. Carlini, and R. Jozsa, "Quantum template matching," *Phys. Rev. A*, vol. 64, no. 2, pp. 022 317-1–022 317-11, 2001.
[23] C. A. Trungenberger, "Phase transforms in quantum pattern recognition," *Phys. Rev. Lett.*, vol. 89, no. 27, pp. 277 903-1–277 903-4, 2002.
[24] D. Deutsch, "Quantum theory, the Church-Turing principle and the universal quantum computer," in *Proc. Royal Soc. Lond.*, vol. A 400, 1985, pp. 97–117.

Ioannis G. Karafyllidis (M'93) received the Dipl. Eng. and Ph.D. degrees in electrical engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1983 and 1991, respectively.

In 1992, he joined the Department of Electrical and Computer Engineering, Democritus University of Thrace, Thrace, Greece, as a faculty member, where he is currently an Associate Professor. His current research emphasis is on quantum computing and on modeling and simulation of nanoelectronic devices and circuits.

Dr. Karafyllidis is a Fellow of the Institute of Nanotechnology and a member of the Technical Chamber of Greece (TEE).