P- ISSN 1991-8941 E-ISSN 2706-6703 *Journal of University of Anbar for Pure Science (JUAPS)* Open Access
2008,(2), (1) :139-144

PROBABILISTIC QUANTUM COMPUTER SIMULATOR

Yusra Mahmood Humadi

College of Science, University of Anbar

ARTICLE INFO

Received: 1 / 4 /2008
Accepted: 24 / 4 /2008
Available online: 30/4/2008
DOI: 10.37652/juaps.2008.15527

Keywords:
Quantum Computation,
Quabit,
Quabit Toffoli Gate,
Quabit Pauli Z,
Hilbert Space.

ABSTRACT

ABSTRACT:

Quantum Computation is a new field in science. It relates to quantum physics and computer science. The concept of quantum computers regarded new theoretical stage due to the fact that working and real quantum computers not exist and in development stage at the moment. Quantum computers are completely different from the classical computers. Quantum computer building blocks are several quantum objects such as atoms, ions or photons, interpreting their states, or polarizations as ones, zeros, and superposition's of ones and zeros. Each quantum bit or qubit can exist in a superposition of states with different probabilities of seeing a one or a zero when one observes the qubit. In this paper, a Probabilistic Quantum Computer Simulator (PQCS) was build to be used as interesting tools to study the various quantum algorithms that intended to be executed on the real quantum computer. The toolbox contains the basic Quantum Units such as NOR Gates, Quabit Toffoli Gate and Quabit Pauli Z that represented in Hilbert Space (HS). The designer of quantum algorithm uses the Drag and Drop facility to accomplish the overall design. The proposed system implemented in Visual Basic and has interactive Graphical User Interface (GUI) that offer flexible design tools for building quantum algorithms in various fields.

Introduction:

Quantum information processing is the result of using the physical reality that quantum theory tells us about for the purposes of performing tasks that were previously thought impossible or infeasible. Devices that perform quantum information processing are known as quantum computers. In this paper we examine how quantum computers can be made by using the simulation system that was built in using Visual Basic, and also how this can be done reliably even when there is a possibility for errors to occur.

* Corresponding author at: College of Science, University of Anbar, Iraq.E-mail address: Yusraalobaidy@yahoo.com

Quantum devices rely on the ability to control and manipulate binary data stored in the phase information of quantum wave functions that describe the electronic states of individual atoms or the polarization states of photons. While existing quantum technologies are in their infancy, we shall see that it is not too early to consider scalability and reliability. In fact, such considerations are a critical link in the development chain of viable device technologies capable of orchestrating reliable control of tens of millions quantum bits in a large-scale system. The theory of quantum information processing (QIP) uses quantum mechanical two-level systems such as the two spin states of spin 1/2 atoms, or the horizontal and vertical polarization states of a single photon to store and

manipulate binary information. Such systems are used to describe the single unit of quantum data known as a qubit [3] whose two states are distinguished as the binary states "0" and "1." One of the distinguished features of QIP from classical computational theory is that the permitted states of a single qubit fill a two-dimensional vector space and can be written as the superposition of the two binary states "0" and "1." In this manner, the state of an n-qubit quantum register spans a 2n-dimensional vector space as the superposition of all of the possible 2n binary bitsring states. An n-bit logic operation is permitted to act on one or all possible bitstring states of the register in a single clockstep. Thus, an exponential increase in the processed information at each clockstep is paralleled by a polynomial increase in the data size manipulated. The purpose of simulation is to be able to build quantum circuits and see what their output is. Also it can be a useful tool to implement the few available quantum algorithms such as Shor's factoring algorithm [1,3] and Grover's search algorithm [2].

Quantum Basics:

Qubits:

A qbit has two computational basis vectors $|0\rangle$ and $|1\rangle$ of the Hilbert space corresponding to the classical bit values 0 and 1 and an arbitrary state $\varphi\rangle$ of a qbit is

$$|\varphi\rangle = a|0\rangle + b|1\rangle = a\begin{bmatrix}1\\0\end{bmatrix} + b\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}a\\b\end{bmatrix} \quad |a|^2 + |b|^2 = 1.$$

- Definition

A collection of n qbits is called a qregister of size n. It may contain any of the N = 2n-dimensional computational basis vectors, n qbit of size N, or arbitrary superposition of these vectors. If the content of the qbits of a qregister is known then the state of the qregister can be computed by means of a tensor product as below[4]:

$$|\varphi\rangle = |qbit_{N-1}\rangle \otimes |qbit_{N-2}\rangle \otimes \cdots \otimes |qbit_1\rangle \otimes |qbit_0\rangle.$$

Example: consider the following two qubits:
When we join the two qbits we yield a four-dimensional register

|00> +|01> |10> +|11>/2

Quantum Qubit Gates:

These gates are correspond the classical computer gates that are used to build the universal quantum computer. Quantum Qubit Gates (QQG) can summarize as bellow:

1. Qubit Pauli Z Gate. This gate flip the sign of the state $|1\rangle$ using the following operator

2. If the quantum state represents by $\alpha 0|0\rangle + \alpha 1|1\rangle$ or in vector notation $\begin{pmatrix}\alpha 0\\ \alpha 1\end{pmatrix}$ then the corresponding output from the quantum Pauli Z gate is

$$\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}\alpha 0\\ \alpha 1\end{pmatrix} = \begin{pmatrix}\alpha 0\\ -\alpha 1\end{pmatrix}$$

Figure (1) shows the representation of this gate.

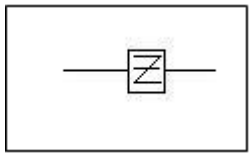


Figure (1) Qubit Pauli Z Gate.

2-Pauli X gate. This gate uses the operator $\sigma_1 = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$ the output of this gate is obtained by:

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

So this is the NOT gate in quantum state. The diagram of NOT gate (Pauli X gate) is shown in figure(2).

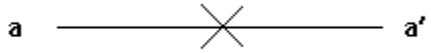


Figure (2) Pauli X gate.

3- Hadamard Gate. The Hadamard gate is one of the most useful quantum gates. This gate {see figure (3)} is sometimes described as being like a square root of NOT gate. The Hadamared Operator is

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

The output is

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} \alpha 0 \\ \alpha 1 \end{pmatrix} = \begin{pmatrix} \alpha 0 + \alpha 1 \\ \alpha 0 - \alpha 1 \end{pmatrix}$$

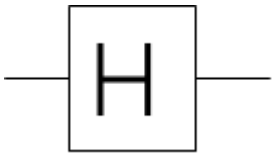


Figure (3) Hadamard Gate.

4-Qubit Controlled Not Gate. Also called (Controlled Not Gate CNOT), it has two inputs, first called controlled input, the other called target input.

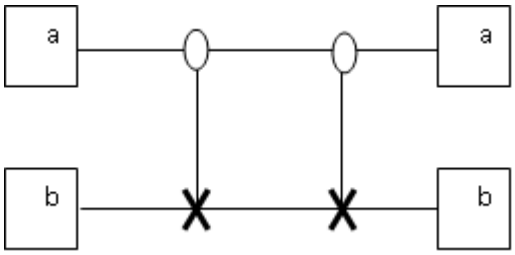
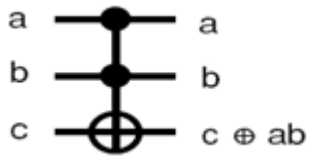


Figure (3) Qubit Controlled Not Gate.

5-Qubit Toffoli Gate (CCNOT). The Universal Toffoli Gate has three inputs. The first two inputs are copied to the first two output pins and the third output is the Exclusive OR of the third input and the AND of the first two inputs.



The unitary matrix is defined as :

$$U_{toffoli} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Figure (3) Qubit Toffoli Gate

Quantum Computer Simulator

A quantum computer is a set of *n* qubits in which the following operations are feasible

- ☒ Each qubit can be prepared in some known state A|0>+B|1>.
- ☒ Each qubit can be measured in the basis |0>, |1>.
- ☒ A universal quantum gate (or set of gates) can be applied at will to any fixed-sizesubset of gates.
- ☒ Quantum circuits are built by interconnecting quantum gates. Several limitations are imposed in the realization of quantum circuits. First, the circuits are acyclic; feedback from one part of the circuit to another is not allowed, there are no loops.
- ☒ We cannot copy qubits.

The simulator basically built using Visual Basic language and has the following characteristics:

- ☒ Simplicity in design and implementing various circuits.
- ☒ Interactive design through using Drag and Drop facilities in selecting the Quantum Gates.
- ☒ The possibilities to check the design manually.
- ☒ The simulator offers flexible design tools that can be changed if not desired.

P- ISSN 1991-8941 E-ISSN 2706-6703 *Journal of University of Anbar for Pure Science (JUAPS)* Open Access
2008,(2), (1) :139-144

Simulator Frameworks

The basic part of our simulator is the Design Grid (DG). DG is (M,N) dimension which form the basic part of the simulator where the Quantum gates can be dropped according to the design philosophy.

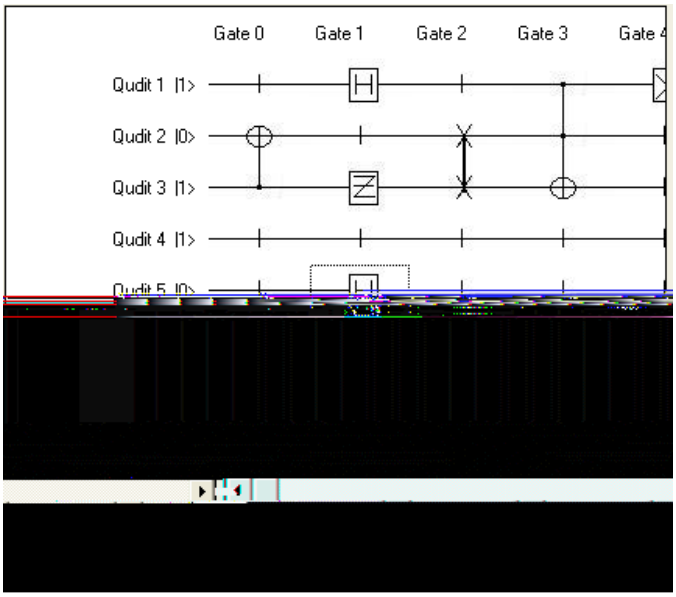


Figure (4): DG has (500 x 500) Row-Column dimensions

Figure (5) Quantum Gates tools.

DG as seen in figure (4) has (500 x 500) Row-Column dimensions. The implementer of the Quantum algorithm can choose any form of quantum gates form the Quantum tools and drag it to specified location and drop it there.

Seconded part of simulator is the Quantum Gates tools, which contains the quantum gates intended to be added.

The simulator functionality can be accessed using the control panel that contains the functions of simulator as seen in figure (6).

As seen in the figure these operations can be classified into the following:

Figure (6) Simulator Control panel.

1. Add bit. To add Quantum bits to the DG.
2. Remove bit.
3. Add Gate. To add a Gate cell to the DG.
4. Remove Gate.
5. Start Simulation. To initiates the simulation.
6. Show output vector.
7. Clear the circuits. For erasing the circuit.

Experimental Results:

After completing the design, the implementer can start the simulator to simulate the circuit implemented. For example the following circuit was implemented in the simulator that represents the Toffoli Gates and Hadamard Gate.

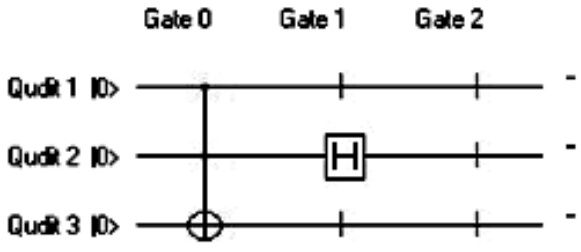


Figure (7) Simulator Grid.

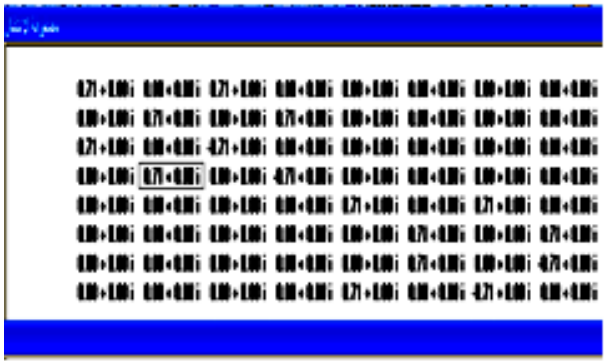


Figure (8) Transfer Matrix.

The implementer press the Start simulation and the simulator perform the overall process and as seen in the figure(6) the input is (|000>) . The simulator offers to the implementer the choice to see the transfer matrix to verify the circuit. The second example of Quantum circuit shown in figure (8) this circuit consists of (2) Hadamared Gate and (2) CNOT gate and one gate for pauli X and pauli Z gate. The input is (|10110>) the implementation was very easy by dragging and dropping the gates into the framework grid.

Conclusions:

The field of quantum computing has made remarkable progress. The field of quantum circuit simulation is rich in possible heuristics that may reduce the resource requirements. The ultimate quantum computer simulator would be one that uses a quantum circuit to simulate other circuits, thereby reducing exponential overheads. But quantum computation is a wonderful problem to squeeze the difficulty of quantum mechanics into a smaller and smaller place. Since quantum computation relies so heavily on the non-local aspect of quantum theory we can extend and stress the theory in new and exciting ways. In this paper, an exciting simulator was built and can be used for testing various quantum algorithms as it implemented in real quantum computer. The simulator was very flexible to initiates and has very nice facility to compare the results and to design in place manner which mean that the quantum implementer can verify the design through trail and error manner.

References:

1. A collection of papers on quantum computing – Stefano Bettelli http://arxiv.org/ (2002).
2. Ethan Bernstein and Umesh Vazirani. ‘Quantum Complexity Theory’. SIAM Journal on Computing, 26(5):1411–1473, October 1997.
3. L. Adleman, J, Demarrais, and M.D. Huang. ‘Quantum Computability’. SIAM Journal on Computing, 26(5):1524–1540, 1997.
4. .W Shor, "Algorithms for quantum computation: Discrete Logarithms and Factoring" www.citeseer.ist.psu.edu/14533.html (1986).
5. S. Aaronson. "Quantum Computing, Postselection, and Probabilistic Polynomial-Time".
6. Proceedings of the Royal Society of London A, 461:3473– 3482, 2005.

P- ISSN 1991-8941 E-ISSN 2706-6703 *Journal of University of Anbar for Pure Science (JUAPS)* Open Access
2008,(2), (1) :139-144

المحاكي الحاسوبي الكمي الاحتمالي

يسرى محمود حمادي

Email: Yusraalobaidy@yahoo.com

الخلاصة

الاحتساب الكمي هو حقل جديد من حقول العلوم. يتعلق بفيزياء الكم وعلوم الحاسبات ويعتبر مرحلة نظرية جديدة نتيجة لكون الحاسبات الكمية الحقيقية العاملة هي في مرحلة التطوير حاليا. الحاسبات الكمية تختلف تماما عن الحاسبات الاعتيادية حيث ان الحاسبات الكمية تتكون من مجموعة من المكونات الكمية مثل الذرات والأيونات والفوتونات وترجمة حالاتها أو أستقطاباتها على أنها تمثل حالا الصفر أو الواحد أو حالة التموضع الفائق الكمي من الحالات وبأحتماليات مختلفة اعتمادا على المراقب الذي يرى الحالة على أنها حالة الصفر أو الواحد. في هذا البحث تم تطويرحاسبة محاكية كمية احتمالية تستخدم لغرض دراسة تنفيذ الخوارزميات الكمية والتي يراد لها أن تنفذ على الحاسبات الكمية الحقيقية. حيث يوجد في واجهة البرنامج صندوق للأدوات يحتوي على البوابات الكمية الاساسية مثل بوابة (NOR) وبوابة (Quabit Toffoli) وبوابة (Quabit Pauli Z) التي تمثل في فضاء هلبرت. ان مصمم الخوارزمية لديه امكانية السحب والافلات لتنفيذ التصميم النهائي. ان النظام المعد نفذ باستخدام اللغة (Visual Basic 6) ويحتوي على واجهة رسومية تفاعلية التي تقدم ادوات تصميم مرنة لبناء خوارزميات كمية في حقول مختلفة.