HOW QUANTUM ALGORITHMS WORK

Marcel POPESCU[1], Constantin P. CRISTESCU [2]

Se prezintă informaţie esenţială asupra algoritmilor cuantici şi demonstrează cu ajutorul a două exemple că mediul computaţional MATLAB este foarte potrivit pentru simularea pe calculatoare clasice în scopul studierii capabilităţii acestora.

The paper presents essential information on quantum algorithms and based on two examples demonstrates that the computational environment MATLAB is highly appropriate for classical implementation in order to test their capability.

Key words: qbit**,** quantum information, quantum algorithm, quantum gates.

1. Introduction

In recent decades, starting with the ideas of P. Benioff (1980) and R. Feynman (1982) [1], quantum theory of information – information processing using quantum systems, has been developed. Besides the possibility to implement computing machines which can simulate in a precise and efficient manner quantum systems, this theory offers much simpler solutions for certain applications of modern communications and informatics. Until now, certain algorithms based on the principles of quantum physics, which elegantly solve tasks such as cryptography, searching in databases, finding the period of a function (Fourier transform), large numbers factorization, have been proven [2].

Superposition, entanglement and ***interference*** are the main characteristics of quantum world that make possible faster solving of certain problems that require exponential computing time with a classical computer [3].

Using superposition and interference a function ***f(x)*** can be evaluated simultaneously for all values of ***x***. This property of quantum computers is called quantum parallelism.

"The important property of an entangled pair is that as soon as the state of one particle is known, by the projection resulting from a measurement, the state of the other particle is known instantly, no matter the distance between the particles at the moment of the measurement" [3].

[1] PhD student, Department of Physics 1, University POLITEHNICA of Bucharest, Romania
[2] Professor, Department of Physics 1, University POLITEHNICA of Bucharest, Romania, e-mail:cpcris@physics.pub.ro

Another important feature of quantum information is the measurement process which, in general, changes the state of the system being measured. Any quantum computation process ends with system measurement (Fig.1) [4].

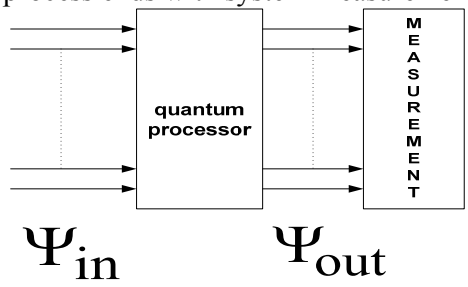

Fig.1. Quantum Computation

Before implementing quantum algorithms, the natural step is to simulate them on a classical computer. The two main reasons for this are finding design faults prior to manufacturing and better understanding of designs [5]. In this work we analyze 2 quantum algorithms by making their simulations using Matlab.

The remaining part of the paper is organized as follows. Section 2 provides the necessary background on quantum algorithms. In section 3 we describe how Matlab could be used to simulate quantum algorithms. Simulation results together with an analysis of Deutsch and Groover's algorithms are given in sections 4 and 5. Finally, conclusions are presented.

2. Qbits and quantum elementary operations

The bit is the fundamental unit of classical information. It assumes two distinct values, "0" and "1" represented by two physical quantities such as two voltage values, two current values, etc. Quantum theory uses to represent information, quantum systems with two levels such as: two polarization states of photons, two energy levels of atoms, etc. The equivalent of a bit - a qbit can be defined as a quantum system in which the Boolean states are represented by two normalized and orthogonal states, denoted {| 0 >, | 1 >}. The two states form a basis in the complex 2-dimensional Hilbert space and any other pure state of a qbit can be written as a superposition:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle . \quad (1)$$

Since the two numbers, $\{\alpha, \beta\}$ are subject to only one constraint –the sum of their absolute squares must be 1, the amount of information that can be represented seems to be infinite. However, nature allows extracting only a single value – a bit. The probability of the value being "1" is $|\beta|^2$ and being "0" $|\alpha|^2$ [4].

A quantum memory register is a physical system composed of ***n*** qbits. Any state vector of this system can be expressed as a superposition of the states that make up a base in the 2^n - dimensional complex Hilbert space [6].

In the classical computer, information processing is done by logic gates. A logic gate changes the input bit value in accordance with a truth table. Unlike the classical logic gate, a quantum gate is a unitary transformation applied to the state vector of a qbit. This operation can be implemented by applying an external field on the system for a given period of time. A quantum algorithm is specified as a sequence of unitary transformations U_1, U_2, U_3... which act on one or more qbits.

$$|\Psi\rangle = \sum_{i_{k=\{0;1\}}} \alpha_{i_1 i_2 \ldots i_n} |i_1\rangle \otimes |i_2\rangle \ldots \otimes |i_n\rangle \overset{\text{notation}}{=} \sum_i \alpha_i |i\rangle$$
$$\text{where } i_1, i_2 \ldots i_n \in \{0,1\};\ i = \sum_{k=1}^{n} 2^{n-k} \cdot i_k;\ \alpha_i \in C \tag{2}$$

The most used quantum logic gates (illustrated in Fig.2) are:

- ***NOT*** gate is the same gate as in classical computation with the additional characteristic that it respects the superposition:

$$X|0\rangle = |1\rangle;\ X|1\rangle = |0\rangle;\ X(\alpha|0\rangle + \beta|1\rangle) = \alpha|1\rangle + \beta|0\rangle \tag{3}$$

- ***PHASE FLIP*** gate changes the phase of the qbit conditional on its value:

$$Z|0\rangle = |0\rangle;\ Z|1\rangle = -|1\rangle;\ Z(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle - \beta|1\rangle \tag{4}$$

- ***HADAMARD*** gate maps the | 0 > and | 1 > in to a superposition:

$$H|0\rangle = \frac{1}{\sqrt{2}} \cdot \left[|0\rangle + |1\rangle\right];\ H|1\rangle = \frac{1}{\sqrt{2}} \cdot \left[|0\rangle - |1\rangle\right]; \tag{5}$$

- ***CNOT*** **(controlled - NOT)** is a 2-qbits gate that applies the NOT gate to the second bit – the target bit, if the first bit - the control bit is"1":

$$U_{CNOT}|00\rangle = |00\rangle;\ U_{CNOT}|01\rangle = |01\rangle;$$
$$U_{CNOT}|10\rangle = |11\rangle;\ U_{CNOT}|11\rangle = |10\rangle; \tag{6}$$

Any unitary transformation applied to a set of qbits can be obtained the 1 qbit gate and the CNOT gate [7].

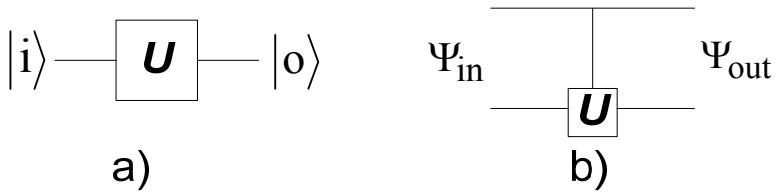

Fig.2. a) One qbit logic gate; b) two qbits logic gate

Before concluding this paragraph, we mention that in the quantum theory of information, making a copy of an unknown state is impossible. This is specified by the **No-Cloning Theorem** [2].

3. Simulating quantum algorithms with Matlab

Matlab is a computing environment that is based on operations with matrices, which makes it a useful tool in simulating the matrix formalism of quantum processes.

The representation of one qbit state in the simulation can be done using the standard base of a 2-dimensional Hilbert space:

$$\left\{|0\rangle \to [1 \quad 0]^T;\ |1\rangle \to [0 \quad 1]^T\right\} \Rightarrow |\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \to [\alpha \quad \beta]^T; \tag{7}$$

The state of a quantum register may be given, either specifying each qbit state, then making the tensor product using Matlab function ***kron***, or specifying the 2^n – dimensional column vector in which the element ***i+1*** is the coefficient of the ***i*** state. (See equation 2):

$$|\Psi\rangle = \sum_i \alpha_i |i\rangle \to [\alpha_0 \quad \alpha_1 \quad \ldots \quad \alpha_{2^n-1}]^T; \tag{8}$$

An algorithm is given as a series of unitary transformations applied to the quantum register. These transformations are represented in a matrix form. Transformation matrices corresponding to he gates specified in section 2 are:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix};\ Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix};\ H = \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}; U_{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{9}$$

Starting with this representation of the quantum physics formalism, we simulate the algorithms that work with up to 12 qbits. For Groover algorithm the simulation time is in the order of tens of seconds. Limitation to 12 qbits is imposed by the fact that a transformation of ***n*** qbits is implemented with a matrix having [2^n x 2^n] complex elements. Further improvement has been made by dynamically calculating the matrix of operators. This approach allows performing simulations on up to 24 qbits, but the simulation time grows exponentially.

4. Deutsch algorithm

A useful example for illustrating how quantum algorithms work is Deutsch algorithm, which determine if ***f:{0,1}→{0,1}*** is constant [2]. In spite of its simplicity, full use of the superposition and interference has been made here in order to characterize ***f*** with only one evaluation.

For any function $f:\{0,1\}^n \rightarrow \{0,1\}$, a quantum circuit described by unitary operator can be build as shown by eq. (10).

$$U_f|xy\rangle = |x\rangle \otimes |f(x) \oplus y\rangle = \begin{cases} |x\rangle \otimes |f(x)\rangle; \ y = 0 \\ |x\rangle \otimes |\overline{f(x)}\rangle; \ y = 1 \end{cases} \tag{10}$$

where $x \in \{0,1\}^n$ and top bar means bit negation .Such a circuit is called the ***oracle*** of function f.

Consider an oracle for an unknown function $f:\{0,1\} \rightarrow \{0,1\}$, which is a black box that inside can calculate a complex problem, and after a fixed period of time to give a Boolean answer. In order to determine if this function is constant, two evaluations are required using classical logic. It may be proven that using quantum information theory one evaluation is enough. The circuit which implements this quantum algorithm is given in Fig.3.

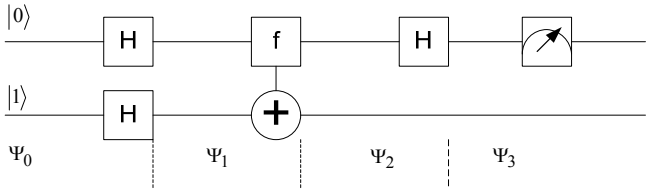

Fig.3. Quantum circuit for Deutsch algorithm [2]

The equations describing the circuit are given below; H is Hadamard transform (eq. 5):

$$|\Psi_0\rangle = |01\rangle$$

$$|\Psi_1\rangle = H \otimes H|01\rangle = \frac{1}{2}\left(|00\rangle - |01\rangle + |10\rangle - |11\rangle\right) \tag{11}$$

$$|\Psi_2\rangle = \frac{1}{2} \cdot \left(|0\rangle \otimes |f(0)\rangle - |0\rangle \otimes |\overline{f(0)}\rangle + |1\rangle \otimes |f(1)\rangle - |1\rangle \otimes |\overline{f(1)}\rangle\right) \tag{12}$$

$$\begin{aligned} |\Psi_3\rangle = H \otimes I \ \ \Psi_2 = \frac{1}{2 \cdot \sqrt{2}} \cdot [&|0\rangle \cdot (|f(0)\rangle - |\overline{f(0)}\rangle + |f(1)\rangle - |\overline{f(1)}\rangle) + \\ &+ |1\rangle \cdot (|f(0)\rangle - |\overline{f(0)}\rangle - |f(1)\rangle + |\overline{f(1)}\rangle)] \end{aligned} \tag{13}$$

$$\begin{cases} f(0) = f(1) \Rightarrow |\Psi_3\rangle = \frac{1}{\sqrt{2}} \cdot |0\rangle \cdot (|f(0)\rangle - |\overline{f(0)}\rangle) \\ f(0) \neq f(1) \Rightarrow |\Psi_3\rangle = \frac{1}{\sqrt{2}} \cdot |1\rangle \cdot (|f(0)\rangle - |\overline{f(0)}\rangle) \end{cases} \tag{14}$$

If the first qbit is found „0” after measurement, then f is constant, otherwise it is not constant.

The simulation of this algorithm in Matlab may look like this:

- initialization of the quantum register containing the two qbits ($|\Psi_0\rangle \rightarrow$ psi_0):

q1 = [1;0]; q2 = [0;1]; psi_0 = kron(q1,q2);

- create the superposition of states by applying the Hadamard gate ($|\Psi_1\rangle \rightarrow$ psi_1):

*psi_1 = kron(H,H)*psi_0;*

- in order to simulate the oracle, its matrix representation is necessary. Let this matrix be U_f (corresponding to the operator from equation 10), then apply this ($|\Psi_2\rangle \rightarrow$ psi_2):

*psi_2 = Uf*psi_1;*

- apply again the Hadamard gate to the first qbit while the second qbit is left unchanged (identity transform). In this way, interference is used to reduce the superposition of the states of the first qbit ($|\Psi_3\rangle \rightarrow$ psi_3):

*psi_3 = kron(H, eye(2))*psi_2;*

- the measurement process of the first qbit is simulated as the projection on one of the two base states. In this way one determines the probability amplitude of finding the qbit in one of the two states, and then the probability:

[prob_0] = measure(psi_3,1);

For a given oracle, the simulation results are represented in Fig. 4. On the *x*-axis is the value expressed in the decimal system (eq. 2) possible to be stored in the register, and on the *y*-axis the probability of finding that value. Each graph corresponds to the algorithm steps. It can be observed that psi_3 is a superposition of |2 > = |10> and |3 > = | 11> with equal probability , 0.5, then the first qbit is "1" and for this simulation the oracle considered, implement a function which is not constant.

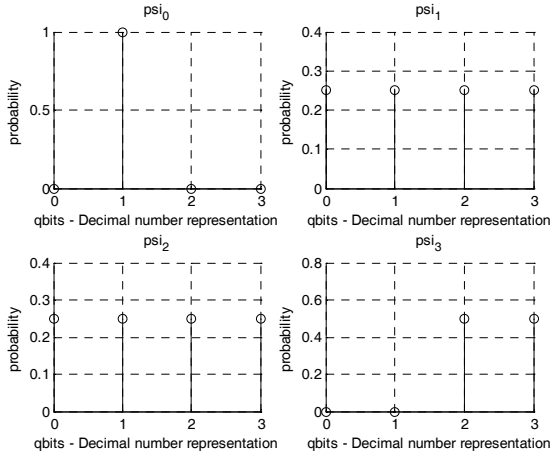

Fig.4. Probability distribution of states on each step of Deutsch algorithm

5. Groover search algorithm

Searching in a database with length N of one or more elements is a problem that a classical computer can solve in N/2 steps on average. The following algorithm can do this search in $\sqrt{N}$ steps.

Consider a quantum circuit – oracle, which implements a function $f:\{0,1\}^n \rightarrow \{0,1\}$, $f(x) = 1$ for $x = t$; and $f(x) = 0$ for $x \neq t$. The scope of the algorithm is to find t. Classically, this can be achieved after 2^n - 1 trials, in the worst case scenario. Finding t with a quantum algorithm uses three subroutines:

- **H** Hadamard transformation applied to n qbits: $\mathbf{H} = H \otimes H \otimes H.... \otimes H$

- **M** (marking subroutine), which invoke oracle. This subroutine changes the sign of the coefficient of state $|t>$, leaving all other states unchanged. The operator that implements this subroutine is: $\mathbf{M} = \mathbf{I} - 2 \cdot |t><t|$, where **I** is the identity transform.

- **B**, changes the sign of the coefficients of all states except the „blank state" |0000...0>.

The steps of the algorithm are:

- register initialization to state |000...00>;
- apply Hadamard gate to all qbits in order to achieve the superposition.
- apply Groover iteration : **M H B H** for $m = floor(\pi/4*2^{n/2})$ times;
- with a probability very close to 1 (but not exactly 1), measuring the register after the quantum algorithm is finished we find the state $|t>$.

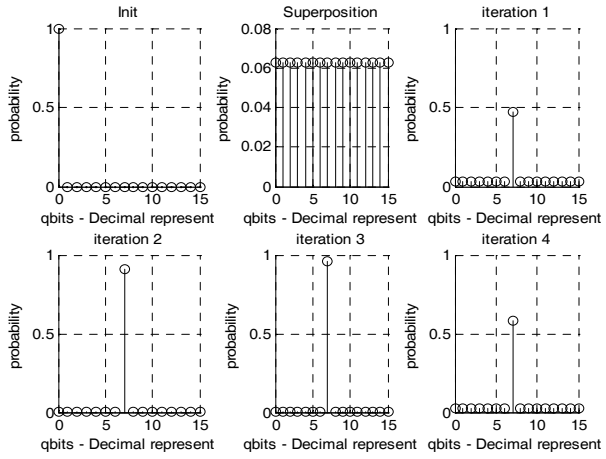

Fig.5. Probability distribution of states on each Groover algorithm step (4 qbits)

In Fig.5 the probability distribution of states after each step of the Groover search algorithm for n = 4 qbits is represented in the same way as in Fig.4 . The number of iterations required is $m = floor(\pi/4*2^{n/2}) = 3$ which is optimal. After 3

iterations the probability of finding $|t>$ by measuring the register, searched value is ***0.9613***. An additional iteration decreases this probability to: ***0.5817***.

The explanation of how this algorithm works can be done by using a geometrical interpretation. In this way, each Groover iteration is equivalent to a rotation of the quantum register state vector in the plane defined by $|t>$ and $|\mu\rangle = \frac{1}{\sqrt{N}} \sum_{i}^{N-1} |i\rangle$. Starting with $|\mu\rangle$, created after the Hadamard transform, minimum angle between the register vector state and $|t>$ is obtained after $m = floor(\pi/4*2^{n/2})$ iterations. Making one more iteration leads to a decrease of probability of finding the state $|t>$ after measurement.

Applying Groover algorithm with 12 qbits, the optimal number of iterations is m = 50. After 50 iterations, the probability to find state $|t>$ is ***0.999945***. The probability then decreases and after ***2m*** iterations the quantum register is in the state close to $|\mu>$.

7. Conclusion

The paper presents basic information on quantum algorithms and, using the powerful matrix formalism of the Matlab environment the Deutsch algorithm and the Groover search algorithm are simulated on a classical computer. Such simulation of quantum algorithms is useful both for better understanding of designs and for identifying accidental design faults prior to manufacturing.

R E F E R E N C E S

[1] *R. Feynamn*, Simulating Physics with Computers, International Journal of Theoretical Physics, Vol. 21, Nos. 6/7, 1982

[2] *M. Nakahara, T. Ohmi*, Quantum Computing From *Linear Algebra to Physical Realizations*, Taylor & Francis Group, 2008

[3] *C. P. Cristescu, M. Popescu*, Entangled quantum states, Quantum Teleportation and quantum information, UPB Sci. Bul., to be published

[4] *G. Chen, D. Church, B. G. Englert, C. Henkel, B. Rohwedder, M. Scully, S. Zubairy*, Quantum Computing Devices, Principles, Design and Analysis, Chapman & Hall/CRC, 2007

[5] *I. G. Karafyllidis*, Quantum Computer Simulator Based on the Circuit Model of Quantum Computation, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 52, 8, AUGUST 2005

[6] *Wim van Dam*, Nonlocality & Communication Complexity, PhD thesis

[7] A. *Barenco, et. All*., Elementary gates for quantum computations, Phys. Rev. A 52, 3457 - 3467 (1995)