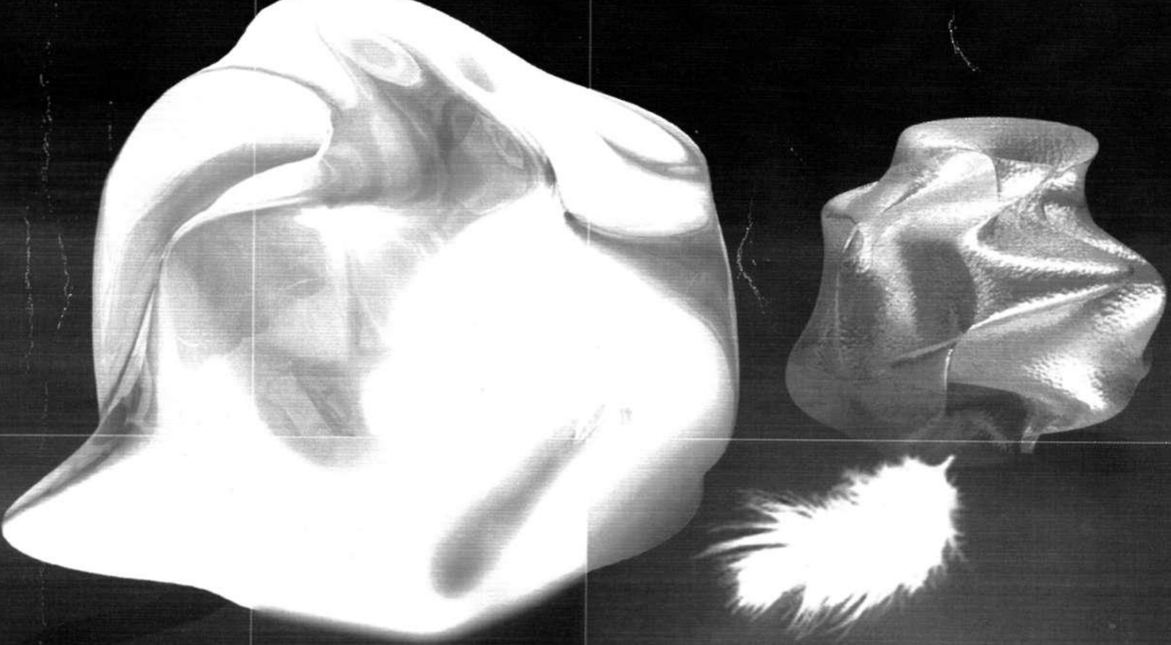
Том 3, № 3, Сентябрь 2008

ISSN 1819-4362

НЕЧЕТКИЕ СИСТЕМЫ и мягкие вычисления



Научный журнал Российской ассоциации нечетких систем и мягких вычислений

УДК 658.012.011.56

FAST ALGORITHM AND HW DESIGN FOR EFFECIENT INTELLIGENT COMPUTATION OF MAIN QUANTUM ALGORITHM OPERATORS ON CLASSICAL COMPUTER

Ulyanov I.S., Porto M., Ulyanov S.V.
International University of Nature, Society, and Man "Dubna", Moscow, Russia
ST Microelectronics, Milan, Italy

Поступила в редакцию 15.03.2008.

Представляется общий подход к моделированию квантовых вычислений на классическом компьютере. Предлагается быстрый алгоритм для симуляции квантового алгоритма Гровера на большой неотсортированной базе данных. Демонстрируется сравнение со стандартным способом симуляции квантовых вычислений. Описывается дизайн аппаратного обеспечения для реализации основных квантовых операторов. В качестве теста эффективности дизайна предлагается алгоритм Гровера. Демонстрируется возможность эффективного моделирования квантовых вычислений.

The general approach for quantum algorithm (QA) simulation on classical computer is introduced. Efficient fast algorithm and corresponding SW for simulation of Grover's quantum search algorithm (QSA) in large unsorted database and fuzzy simulation is presented. Comparison with common QA simulation approach is demonstrated. Hardware (HW) design method of main quantum operators that are used in simulation of QA and fuzzy operators is described. Grover's QSA as Benchmark of HW design method application is presented. This approach demonstrates the possibility of classical efficient simulation of quantum algorithm gates (QAG) and in general fuzzy simulation approaches.

Ключевые слова: квантовый алгоритм, эффективное моделирование, быстрый алгоритм, аппаратная реализация.
Keywords: quantum algorithm gate, efficient simulation, fast algorithm, hardware implementation.

1. Introduction

Quantum algorithms (QA) demonstrate great efficiency in many practical tasks such as factorization of large integer numbers, where classical algorithms are failing or dramatically ineffective [1]. Practical application is still away due to lack of the physical HW implementation of quantum computers. We describe design method of main quantum operators and hardware implementation of QAG for fast search in large database and related topics concerning the control of a process, including search-of-minima *intelligent operations.* This method is very useful for minimum efforts of searching among a set of values and in particular is the first step for the realization of a HW control systems exploiting artificial intelligence in order to fuzzy control in a robust way a non-linear process or in order to efficient search in a database. The presented HW performs all the functional steps of a Grover QSA (This algorithm and its modifications are described in [2] and briefly in Appendix). By suitable changes of traditional matricial approach, a modular n-qubit-hybrid structure is realized in order to prove the usefulness of iterations of the gate, which provide a higher probability of exact solution finding. A minimum-entropy based method is adopted as a termination condition criterion and realized in a digital part together with display output.

The possibility of providing an external clock signal for iteration management allows to implement a very fast Grover's QSA, many times (for comparison in details see Table 2 below) faster than the corresponding software (SW) realization [3], and less sensitive to qubits improvement. The difference between classical and QAs is following: problem solved by QA is coded in the structure of the quantum operators. Input to QA in this case is always the same. Output of QA says which problem was coded. In some sense you give a function to QA to analyze and QA returns its property as an answer.

Formally, the problems solved by QAs could be stated as follows:

Input	*A function* $f\colon \{0,1\}^n \to \{0,1\}^m$
Problem	*Find a certain property of f*

QA studies qualitative properties of the functions.

The core of any QA is a set of unitary quantum operators or quantum gates. In practical representation quantum gate is a unitary matrix with particular structure. The size of this matrix grows exponentially with the number of inputs, making it impossible to simulate QA's with more than 30-35 inputs [3] on classical computer with von Neumann architecture. In this report we present a practical approach to simulate most of known QA's on classical computers. We present the results of the classical efficient simulation of the Grover's QSA as a Benchmark of this approach and background for quantum soft computing and fuzzy control based on quantum genetic (evolutionary) algorithms and quantum neural network. The role of this approach in quantum soft computing and in fuzzy simulation is discussed in [4 - 9].

2. Structure of QA gate system design

The background of QA simulation is a generalized representation of QA as a set of sequentially applied smaller quantum gates as it is presented on the Figure 1a. From the structural point of view each QA requires a particular set of quantum gates, but generally each particular set can be divided into three main subsets with same function for all QA's: *Superposition* operators, *Entanglement* operators and *Interference* operators. This division permits to generalize the approach of QA simulation and to create a classical tool to simulate any type of known QA. Furthermore, local optimization of QA components according to specific hardware realization makes it possible to develop appropriate hardware accelerator of QA simulation using classical gates [4, 5].

2.1. Generalized approach in QA simulation

In general, any QA can be represented as a circuit of smaller quantum gates as it is demonstrated on the Figure 1 [4]. The circuit presented in the Figure 1 is divided into five general steps:

Step 1: *Input.* Quantum state vector is set up to an initial value for this concrete algorithm.

For example, input for Grover's QSA is a quantum state$|\phi_0\rangle$described as a superposition of the quantum states as

$$|\phi_0\rangle = |0\rangle \otimes \cdots \otimes |0\rangle \otimes |1\rangle . \tag{1}$$

where$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$; $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$; $\otimes$ denotes Kronecker tensor product operation.

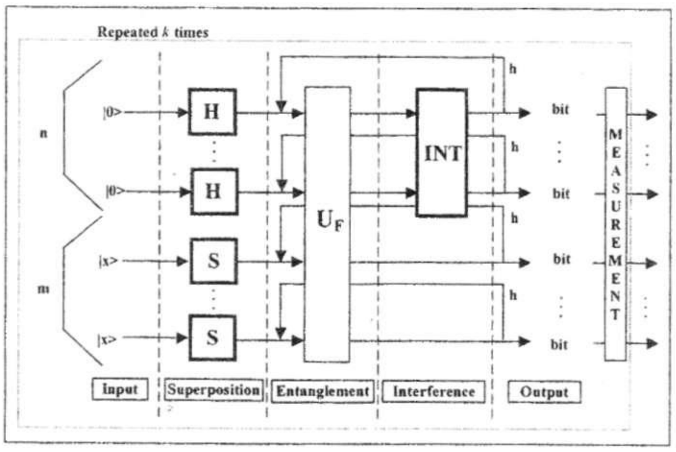


Figure 1a: Circuit representation of QA

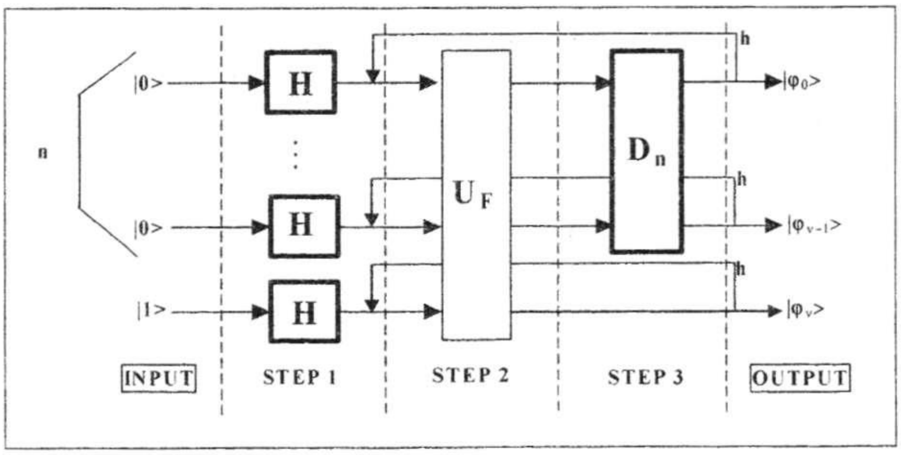


Figure 1b: Quantum circuit of Grover's QSA

Such a quantum state can be presented as it is shown on the Figure 2a. The coefficients a_i are called probability amplitudes [2, 4]. Probability amplitudes may take negative or even complex values. The only constrain on the values of the probability amplitudes is

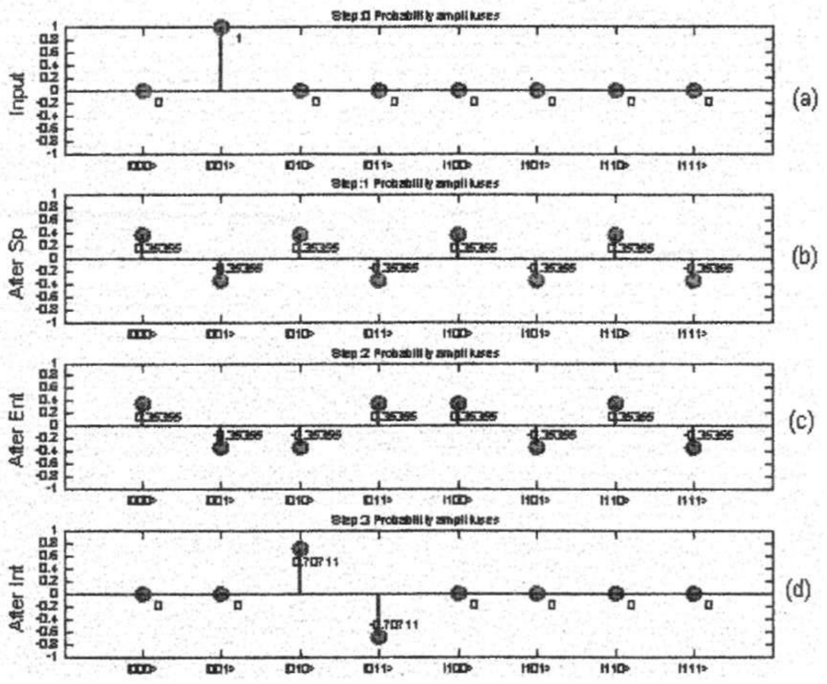


Figure 2: Dynamics of Grover's QSA probability amplitudes of state vector on each algorithm step

$$\sum_i a_i^2 = 1 \tag{2}$$

Step 2: *Superposition.* The state of the quantum state vector is transformed in the way that probabilities are distributed uniformly among all basis states. The result of the superposition step of Grover's QSA is presented on the Figure 2b in probability amplitude representation and in the Figure 3b in probability representation.

Step 3: *Entanglement.* Probability amplitudes of the basis vector corresponding to the current problem are flipped while rest basis vectors left unchanged. Entanglement is done via controlled-NOT-operation. Result of entanglement operation application to the state vector after superposition operation is shown on the Figure 2c and in the Figure 3c. Note that, an entanglement operation does not affect the probability of state vector to be measured. Actually entanglement prepares a state, which cannot be represented as a tensor product of simpler state vectors. For example, consider state $\phi_1$presented on the Figure 2b and state ϕ_2 presented on the Figure 2c:

$$\begin{aligned}\phi_1 = 0.35355\left(|000\rangle - |001\rangle + |010\rangle - |011\rangle + |100\rangle - |101\rangle + |110\rangle - |111\rangle\right) \\ = 0.35355\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right)\left(|0\rangle - |1\rangle\right),\end{aligned}$$

$$\begin{aligned}\phi_2 &= 0.35355\left(|000\rangle - |001\rangle - |010\rangle + |011\rangle + |100\rangle - |101\rangle + |110\rangle - |111\rangle\right)\\ &= 0.35355\left(|00\rangle - |01\rangle + |10\rangle + |11\rangle\right)|0\rangle \text{-} 0.35355\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right)|1\rangle\end{aligned}$$

As it was shown above, described state ϕ_1 can be presented as tensor product of simpler states, while state ϕ_2 cannot.

Step 4: *Interference.* Probability amplitudes are inverted about the average value. As a result the probability amplitude of states "marked" by entanglement operation will increase. Result of interference operator application is presented on the Figure 2d in a probability amplitude representation and in the Figure 3d in a probability representation.

Step 5: *Output* On this step performed measurement operation (extraction of the state with maximum probability), and following interpretation of the result. For example, in case of Grover's QSA required index is coded in first n bits of the measured basis vector.

Steps of QA's realized by unitary quantum operators. Simulation of quantum operators is a key point in general QA simulation. In order to accelerate QA's basic quantum operators must be studied.

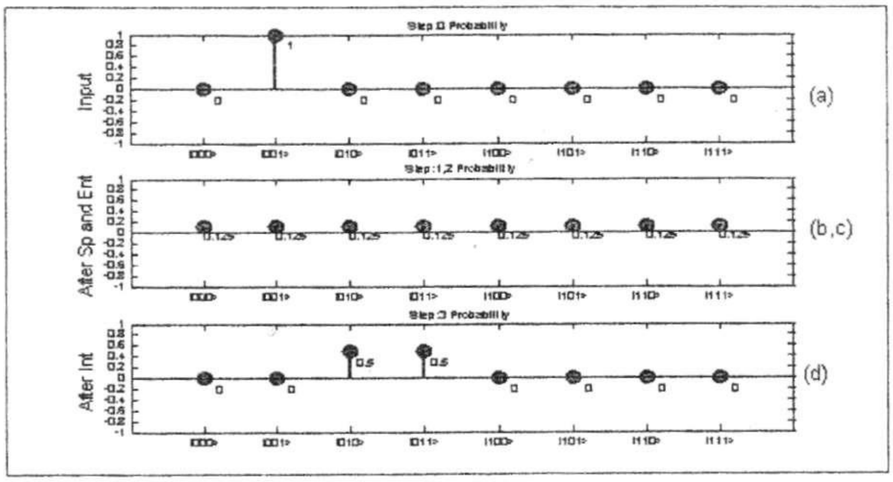


Figure 3: Dynamics of Grover's QSA probabilities of state vector on each algorithm step

2.2. Main QA operators

We consider superposition, entanglement and interference operators from simulation viewpoint. In this case superposition and interference have more complicated structure and differ from algorithm to algorithm. And then we consider entanglement operators, since they have similar structure for all QA's, and differ only by function being analyzed.

2.2.1. Superposition operators of QA's

In general, the superposition operator consists of the combination of the tensor products Hadamard Hoperators with identity operator$I$:

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The superposition operator of most QA's can be expressed as (see, Figure 1a)

$$Sp = \left(\overset{n}{\underset{i=1}{\otimes}} H\right) \otimes \left(\overset{m}{\underset{i=1}{\otimes}} S\right) \tag{3}$$

where n and m are the numbers of inputs and of outputs respectively. Operator. S may be or Hadamard operator H or identity operator I depending on the algorithm. Numbers of outputs $m$as well as structures of corresponding superposition and interference operators are presented in the Table 1 for different QA's.

Table 1: Parameters of superposition and interference operators of main quantum algorithms

Algorithm	**Superposition**	m	**Interference**
Deutsch's	$H\otimes I$	1	$H\otimes H$
Deutsch-Jozsa's	$^nH\otimes H$	1	$^nH\otimes I$
Grover's	$^nH\otimes H$	1	$D_n\otimes I$
Simon's	$^nH\otimes {}^nI$	n	$^nH\otimes {}^nI$
Shor's	$^nH\otimes {}^nI$	n	$QFT_n\otimes {}^nI$

Note that superposition and interference operators are often contain tensor power of Hadamard operator which is called Walsh-Hadamard operator. It is known [2,4] that elements of the Walsh-Hadamard operator could be obtained as:

$$[{}^nH]_{i,j} = \frac{(-1)^{i*j}}{2^{n/2}} = \frac{1}{2^{n/2}}\begin{cases} 1, \text{ if } i*j \text{ is even} \\ \text{-1, if } i*j \text{ is odd} \end{cases} \tag{4}$$

where $i = 0, 1, ..., 2^n, j = 0, 1, ..., 2^n$.

This approach improves greatly performance of classical simulation of the Walsh–Hadamard operators, since its elements could be obtained by the simple replication according to the rule presented in Eq. (4).

Example 1: Consider superposition operator of Deutsch's algorithm, $n = 1$, $m = 1$, $S = I$:

$$[Sp]_{i,j}^{Deutsch} = \frac{(-1)^{i+j}}{2^{1/2}} \otimes I = \frac{1}{\sqrt{2}}\begin{pmatrix} (-1)^{0*0}I & (-1)^{0*1}I \\ (-1)^{1*0}I & (-1)^{1*1}I \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} I & I \\ I & -I \end{bmatrix} \tag{5}$$

Example 2: Consider superposition operator of Deutsch-Jozsa's and of Grover's algorithm, for the case $n = 2, \quad m = 1, \quad S = H$:

$$[Sp]_{i,j}^{D.-J.'s,\, Grover's} = \frac{(-1)^{i+j}}{2^{2/2}} \otimes H = \frac{1}{2}\begin{pmatrix} (-1)^{0*0}H & (-1)^{0*1}H & (-1)^{0*2}H & (-1)^{0*3}H \\ (-1)^{1*0}H & (-1)^{1*1}H & (-1)^{1*2}H & (-1)^{1*3}H \\ (-1)^{2*0}H & (-1)^{2*1}H & (-1)^{2*2}H & (-1)^{2*3}H \\ (-1)^{3*0}H & (-1)^{3*1}H & (-1)^{3*2}H & (-1)^{3*3}H \end{pmatrix}$$
$$= \frac{1}{2}\begin{pmatrix} H & H & H & H \\ H & -H & H & -H \\ H & H & H & H \\ H & -H & H & -H \end{pmatrix} \tag{6}$$

Example 3: Superposition operator of Simon's and of Shor's algorithms, $n = 2, m = 2, S = I$:

$$[Sp]_{i,j}^{Simon,\, Shor} = \frac{(-1)^{i+j}}{2^{2/2}} \otimes^2 I = \frac{1}{2}\begin{pmatrix} (-1)^{0*0}({}^2I) & (-1)^{0*1}({}^2I) & (-1)^{0*2}({}^2I) & (-1)^{0*3}({}^2I) \\ (-1)^{1*0}({}^2I) & (-1)^{1*1}({}^2I) & (-1)^{1*2}({}^2I) & (-1)^{1*3}({}^2I) \\ (-1)^{2*0}({}^2I) & (-1)^{2*1}({}^2I) & (-1)^{2*2}({}^2I) & (-1)^{2*3}({}^2I) \\ (-1)^{3*0}({}^2I) & (-1)^{3*1}({}^2I) & (-1)^{3*2}({}^2I) & (-1)^{3*3}({}^2I) \end{pmatrix}$$
$$= \frac{1}{2}\begin{pmatrix} {}^2I & {}^2I & {}^2I & {}^2I \\ {}^2I & -{}^2I & {}^2I & -{}^2I \\ {}^2I & {}^2I & {}^2I & {}^2I \\ {}^2I & -{}^2I & {}^2I & -{}^2I \end{pmatrix} \tag{7}$$

2.2.2. Interference operators of main QA's

Interference operators must be selected for each algorithm individually according to the parameters presented in the Table 1. Consider some particular parts of interference operators. Interference operator consists of interference part, which is different for all algorithms, and from measurement part, which is the same for most of algorithms and consists of m tensor power of identity operator. Consider interference operator of each algorithm.

Example 1: *Interference operator of Deutsch' algorithm.* Interference operator of Deutsch's algorithm consists of tensor product of two Hadamard transformations, and can be calculated using Eq. (4) with$n = 2$:

$$\left[Int^{\ Deutsch}\right]_{i,j} = {}^2H = \frac{(-1)^{\ i*j}}{2^{2/2}} = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \tag{8}$$

Note that in Deutsch's algorithm, Walsh-Hadamard transformation in interference operator is used also for the measurement basis.

Example 2: *Interference operator of Deutsch-Jozsa's algorithm.* Interference operator of Deutsch-Jozsa's algorithm consists of tensor product of n power of Walsh-Hadamard operator with an identity operator. In general form the block matrix of the interference operator of Deutsch-Jozsa's algorithm can be written as:

$$\left[Int^{\ Deutsch-Jozsa's}\right]_{i,j} = \frac{(-1)^{i*j}}{2^{\frac{n}{2}}} \otimes I, \tag{9}$$

where $i = 0, ..., 2^n - 1, j = 0, ..., 2^n - 1$.

Example 3: *Interference operator of Deutsch-Jozsa's algorithm*, $n = 3, k_1 = 2, k_2 = 1$:

$$\left[Int^{\,Deutsch-Jozsa's}\right]_{i,j} = \frac{(-1)^{i*j}}{2^{\frac{2}{2}}} \otimes I = \frac{1}{2}\begin{pmatrix} I & I & I & I \\ I & -I & I & -I \\ I & I & I & I \\ I & -I & I & -I \end{pmatrix} \tag{10}$$

Example 4: *Interference operator of Grover's algorithm.* Interference operator of Grover's algorithm can be written as a block matrix of the following form:

$$\left[Int^{\,Grover}\right]_{i,j} = D_n \otimes I = \left(\frac{1}{2^{n/2}} - {}^{n}I\right) \otimes I =$$

$$\left(-1+\frac{1}{2^{n/2}}\right)\otimes I\Big|_{i=j}, \left(\frac{1}{2^{n/2}}\right)\otimes I\Big|_{i\neq j} = \frac{1}{2^{n/2}}\left\{\begin{array}{l} -I, i=j \\ I, i\neq j \end{array}\right., \tag{11}$$

where $i = 0, ..., 2^n - 1, j = 0, ..., 2^n - 1$, D_n refers to diffusion operator:

$$[D_n]_{i,j} = \frac{(-1)^{1\,AND\,(i=j)}}{2^{n/2}}.$$

Example 5: *Interference operator of Grover's QSA*, $n = 2, m = 1$:

$$\left[Int^{\,Grover}\right]_{i,j} = D_2 \otimes I = \left(\frac{1}{2^{2/2}} - {}^{2}I\right) \otimes I = \left(-1+\frac{1}{2}\right)\otimes I\Big|_{i=j}, \left(\frac{1}{2}\right)\otimes I\Big|_{i\neq j}$$

$$= \begin{pmatrix} \left(-1+\frac{1}{2}\right)I & \frac{1}{2}I & \frac{1}{2}I & \frac{1}{2}I \\ \frac{1}{2}I & \left(-1+\frac{1}{2}\right)I & \frac{1}{2}I & \frac{1}{2}I \\ \frac{1}{2}I & \frac{1}{2}I & \left(-1+\frac{1}{2}\right)I & \frac{1}{2}I \\ \frac{1}{2}I & \frac{1}{2}I & \frac{1}{2}I & \left(-1+\frac{1}{2}\right)I \end{pmatrix} = \frac{1}{2}\begin{pmatrix} -I & I & I & I \\ I & -I & I & I \\ I & I & -I & I \\ I & I & I & -I \end{pmatrix} \tag{12}$$

Note that with bigger number of qubits, gain coefficient will become smaller. Dimension of the matrix increases according to 2^n, but each element can be extracted using Eq. (11), without allocation of entire operator matrix.

Remark. Since$D_n D_n^* = I$, D_n is unitary and is therefore a possible quantum state transformation. While the matrix D_n is clearly unitary it can to have the decomposition form$D_n = -H_n R_n^1 H_n$, where$R_n^1[i,j] = 0$, if$i \neq j$, $R_1^1[1,1] = -1$ and$R_n^1[i,i] = +1$, if $1 < i \leqslant N$.

In concrete form the operator D_n (*diffusion – inversion about average*) in Grover algorithm is decomposed as

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

and can be accomplished with $O(n) = O(\log(N))$quantum gates [2,4]. It means that from the viewpoint of efficient computation the form as in Eq. (11) is more preferable.

Example 6: *Interference operator of Simon's algorithm.* Interference operator of Simon's algorithm is prepared in the same manner as superposition as well as superposition operators of Shor's algorithms and can be described as following Eq. (3) and Eq. (7)

$$\left[Int^{\,Simon}\right]_{i,j} = {}^{n}H \otimes {}^{m}I = \frac{(-1)^{i*j}}{2^{n/2}} \otimes {}^{m}I =$$

$$\frac{1}{2^{n/2}}\begin{pmatrix} (-1)^{0*0}\cdot {}^mI & \cdots & (-1)^{0*j}\cdot {}^mI & \cdots & (-1)^{0*(2^n-1)}\cdot {}^mI \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (-1)^{i*0}\cdot {}^mI & \cdots & (-1)^{i*j}\cdot {}^mI & \cdots & (-1)^{i*(2^n-1)}\cdot {}^mI \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (-1)^{(2^n-1)*0}\cdot {}^mI & \cdots & (-1)^{(2^n-1)*j}\cdot {}^mI & \cdots & (-1)^{(2^n-1)*(2^n-1)}\cdot {}^mI \end{pmatrix} \quad (13)$$

Remark. In general, interference operator of Simon's algorithm coincides with interference operator of Deutsch-Jozsa's algorithm Eq. (9), but each block of the operator matrix Eq. (13) consists of m tensor products of identity operator.

Remark. Each odd block (when product of the indexes is an odd number) of the Simon's interference operator Eq. (13), has a negative sign. Actually if $i = 0, 2, 4, \ldots 2^n - 2$ or $j = 0, 2, 4, \ldots 2^n - 2$ the block sign is positive, else block sign is negative. This rule is applicable also for Eq. (9) of Deutsch-Jozsa's algorithm interference operator. Then it is convenient to check if one of the indexes is an even number instead of calculating their product. Then Eq. (13) can be reduced as:

$$\left[Int^{Simon}\right]_{i,j} = {}^nH \otimes {}^mI = \frac{(-1)^{i*j}}{2^{n/2}} \otimes {}^mI = \frac{1}{2^{n/2}}\begin{cases} {}^mI, \text{if } i \text{ is odd or if } j \text{ is odd} \\ -{}^mI, \text{if } i \text{ is even and } j \text{ is even} \end{cases} \quad (14)$$

Example 7: Interference operator of Shor's algorithm. Interference operator of Shor's algorithm uses Quantum Fourier Transformation (QFT) operator, calculated as:

$$[QFT_n]_{i,j} = \frac{1}{2^{n/2}} e^{J(i*j)\frac{2\pi}{2^n}} \quad (15)$$

where: J - imaginary unit, $i = 0, ..., 2^n - 1$ and, $j = 0, ..., 2^n - 1$. With $n = 1$ we can observe the following relation:

$$QFT_{k_1}|_{k_1=1} = \frac{1}{2^{\frac{1}{2}}}\begin{pmatrix} e^{J*(0*0)2\pi/2^1} & e^{J*(0*1)2\pi/2^1} \\ e^{J*(1*0)2\pi/2^1} & e^{J*(1*1)2\pi/2^1} \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = H \quad (16)$$

Eq. (16) can be also presented in harmonic form using Euler formula:

$$[QFT_{k_1}]_{i,j} = \frac{1}{2^{k_1/2}}\left(\cos\left((i*j)\frac{2\pi}{2^{k_1}}\right) + J\sin\left((i*j)\frac{2\pi}{2^{k_1}}\right)\right)$$

2.2.3. Entanglement operators of main QA's

In general entanglement operators are part of QA where the information about the function being analyzed is coded as input-output relation. Let's discuss the general approach for coding binary functions into corresponding entanglement gates. Consider arbitrary binary function: $f : \{0,1\}^n \to \{0,1\}^m$, such that $f(x_0, ..., x_{n-1}) = (y_0, ..., y_{m-1})$. In order to create unitary quantum operator, which performs the same transformation, first we transfer irreversible function f into reversible function F, as following: $F : \{0,1\}^{m+n} \to \{0,1\}^{m+n}$, such that

$$F(x_0, ..., x_{n-1}, y_0, ..., y_{m-1}) = (x_0, ..., x_{n-1}, f(x_0, ..., x_{n-1}) \oplus (y_0, ..., y_{m-1}))$$

where $\oplus$ denotes addition modulo 2. Having reversible function F we can design an entanglement operator matrix using the following rule:

$$[U_F]_{i^B,\, j^B} = 1 \text{ iff } F(j^B) = i^B, \quad i, j \in \left[\underbrace{0,..,0}_{n+m}; \underbrace{1,..,1}_{n+m};\right] \quad B \text{ denotes binary coding}$$

Actually resulted entanglement operator is a block diagonal matrix, of the form:

$$U_F = \begin{pmatrix} M_0 & & 0 \\ & \ddots & \\ 0 & & M_{2^n-1} \end{pmatrix}$$

Each block $M_i, i = 0, ..., 2^n - 1$ consists of m tensor products of I or of C operators, and can be obtained as following:

$$M_i = \overset{m-1}{\underset{k=0}{\otimes}} \begin{cases} I, \text{iff } F(i,k) = 0 \\ C, \text{iff } F(i,k) = 1 \end{cases} , \tag{17}$$

where C stays for NOT operator, defined as:$C = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. It is clear that entanglement operator is a sparse matrix. Using sparse matrix operations it is possible to accelerate the simulation of entanglement operation.

*Example*1. Entanglement operator for binary function: $f : \{0,1\}^2 \to \{0,1\}^1$ such that:$f(x) = 1|_{x=01}\ 0|_{x\neq 01}$. Reversible function F in this case will be: $F : \{0,1\}^3 \to \{0,1\}^3$, such that:

(x, y)	$(x, f(x) \oplus y)$
$00, 0$	$00, 0 \oplus 0 = 0$
$00, 1$	$00, 0 \oplus 1 = 1$
$01, 0$	$01, 1 \oplus 0 = 1$
$01, 1$	$01, 1 \oplus 1 = 0$
$10, 0$	$10, 0 \oplus 0 = 0$
$10, 1$	$10, 1 \oplus 0 = 1$
$11, 0$	$11, 0 \oplus 0 = 0$
$11, 1$	$11, 1 \oplus 0 = 1$

And corresponding entanglement block matrix can be written as:

$$U_F = \begin{array}{c} \\ |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{array} \begin{array}{c} \begin{array}{cccc} \langle 00| & \langle 01| & \langle 10| & \langle 11| \end{array} \\ \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & \boxed{C} & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix} \end{array}$$

Figure 2c demonstrates the result of the application of this operator in Grover's QSA. Entanglement operators of Deutsch and of Deutsch-Jozsa's algorithms have the same form.

Example 2. Entanglement operator for binary function:$f : \{0,1\}^2 \to \{0,1\}^2$, such that $f(x) = 10|_{x=01,11}\ 00|_{x\neq 01,11}$ and

$$U_F = \begin{array}{c} \\ |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{array} \begin{array}{c} \begin{array}{cccc} \langle 00| & \langle 01| & \langle 10| & \langle 11| \end{array} \\ \begin{pmatrix} I \otimes I & 0 & 0 & 0 \\ 0 & \boxed{C \otimes I} & 0 & 0 \\ 0 & 0 & I \otimes I & 0 \\ 0 & 0 & 0 & \boxed{C \otimes I} \end{pmatrix} \end{array}$$

Entanglement operators of Shor's and of Simon's algorithms have the same form.

3. Grover's quantum search algorithm: Case study

From Item 1. Grover's search problem [4] is so stated:

Input	*A function* f: $\{0,1\}^n \rightarrow \{0,1\}$ *such that* $\exists x \in \{0,1\}^n$: $(f(x) = 1 \wedge \forall y \in \{0,1\}^n:\ x \neq y \Rightarrow f(y) = 0)$
Problem	*Find* x

The problem is to decide what class the input function belonged to. It is harder because we are dealing with 2^n classes of input functions (each function of the kind described constitutes a class). Description of the Grover's algorithm is given in Appendix. The whole algorithm is depicted in the scheme of Figure 1b, where U_F is an unitary operator that describe entanglement, and D_n is Grover's diffusion operator that describe interference. Differently from other QAs, in Grover's algorithm it is possible to iterate h times step 2 and step 3 (Entanglement and Interference blocks) until the best solution is reached. Moreover interference part is governed by the matrix D_n, in Eq. (11), whose elements are:

$$d_{ij} = \begin{cases} 1/2^{n-1} - 1 & i = j \\ 1/2^{n-1} & i \neq j \end{cases}$$

The final output vector is given by the formula

$$V = [(D_n \otimes I) \cdot U_F]^h \cdot {}^{n+1}H \tag{18}$$

An example of evolution of Grover's algorithm (18) with n =2 is given in Figure 2. Figures 2, a-d are reported in order basis vector and superposition, entanglement and interference output vectors respectively. Each couple of elements having opposite sign represents the probability amplitude of a certain element of the database. This fact will be very useful for the circuit design because it allows to storing only half-size vectors. The result, obtained after h iteration, minimize the following Shannon entropy:

$$S(h) = -\sum_{k=1}^{2^{n+1}} \|\varphi_k(h)\|^2 \log \|\varphi_k(h)\|^2, \tag{19}$$

where $\varphi_k(h)$is the k^{th} element of output vector taken after h iterations.

4. Results of classical QA gate simulation

Analyzing quantum operators presented in the section 2 we can do the following simplification for increasing performance of classical QA simulations: a) All quantum operators are symmetrical around main diagonal matrices; b) State vector is allocated as a sparse matrix; c) Elements of the quantum operators are not stored, but calculated when necessary using Eqs. (6), (11), and (17); and d) as a termination condition we consider minimum of Shannon entropy of the quantum state, calculated as:

$$S^{Sh} = -\sum_{i=0}^{2^{m+n}} p_i \log p_i \tag{20}$$

Calculation of the Shannon entropy is applied to the quantum state after interference operation [6]. Minimum of Shannon entropy Eq. (20) corresponds to the state when there are few state vectors with high probability (states with minimum uncertainty). Selection of appropriate termination condition is important since QA's are periodical.

Figure 4 shows results of the Shannon information entropy calculation for the Grover's algorithm with 5-inputs. From Figure 4 follows that for five inputs of Grover's QSA an optimal number of iteration for successful result is four.

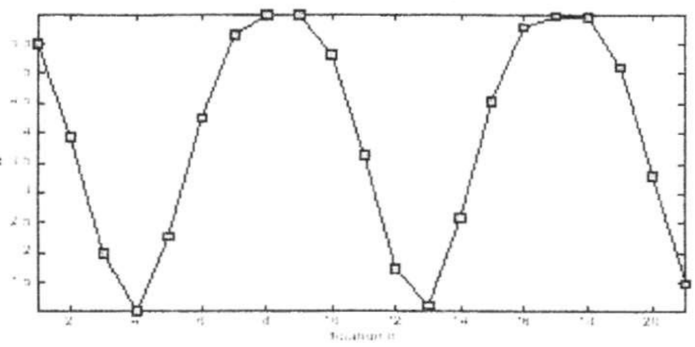


Figure 4: Shannon entropy analysis of Grover's QSA dynamics with five inputs

Table 2: Temporal complexity of Grover's QSA simulation on 1.2GHz computer with two CPUs

n	Number of iterations h	Temporal complexity, seconds	
		Approach 1 (*one iteration*)	Approach 2 (h *iterations*)
10	25	0.28	~0
12	50	5.44	~0
14	100	99.42	~0
15	142	489.05	~0
16	201	2060.63	~0
20	804	-	~0
30	25.375	-	0.016
40	853.549	-	4.263
50	26.353.589	-	12.425

After that probability of correct answer will decrease and algorithm may fail to produce correct answer. Simulation results of fast Grover QSA are summarized in Table 2.

Numbers of iterations for fast algorithm were estimated according to termination condition as minimum of Shannon entropy of quantum state vector.

The following approaches were used in simulation.

Approach 1: Quantum operators are applied as matrices, elements of quantum operator matrices are calculated dynamically according to Eqs. (6), (11), and (17).

Classical Hardware limit of this approach is around 20 qubits, caused by exponential temporal complexity.

Approach 2: Quantum operators are replaced with classical gates.

Product operations are removed from simulation according to [7]. State vector of probability amplitudes is stored in compressed form (only different probability amplitudes are allocated in memory).

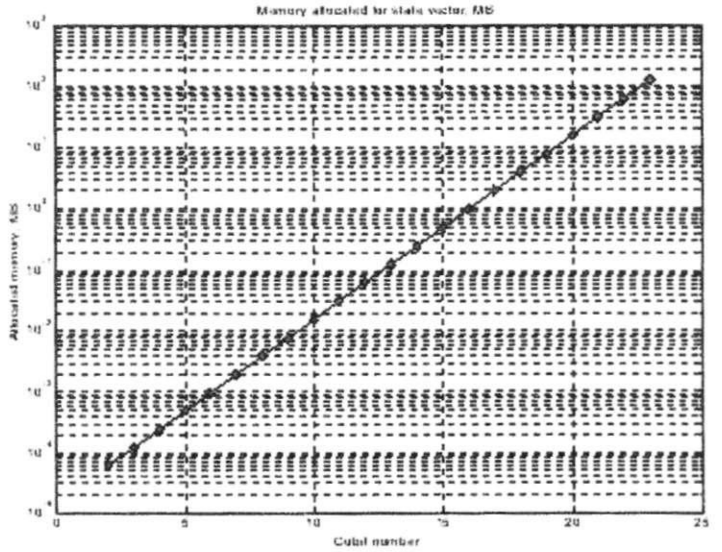


Figure 5: Spatial complexity of Grover QA simulation

With second approach it is possible to perform classical efficient simulation of Grover's QSA with arbitrary large number of inputs (50 qubits and more).

With allocation of the state vector in computer memory, this approach permits to simulation 26 qubits on PC with 1GB of RAM. Figure 5 shows memory required for Grover algorithm simulation, when whole state vector is allocated in memory: Adding one qubit require double of the computer memory needed for simulation of Grover's QSA in case when state vector is allocated completely in memory.

Temporal complexity of Grover's QSA is presented in Figure 6. In this case state vector is allocated in memory, and quantum operators are replaced with classical gates according to [4, 7].

Fastest case is when we compress state vector and replace quantum operator matrices with corresponding classical gates according with [4, 8, 9]. In this case we obtain speed-up according to Approach 2.

Let us consider a new design method of HW architecture and implementation for main quantum operators using as Benchmark Grover's QSA [5] and fast algorithm simulation of main quantum operators described above in item 2 of present article.

5. HW implementation of main quantum algorithm operators

It has been found [5, 10, 11] a new method and circuit that implements the operations performed in second and third step of a quantum algorithm (the so-called

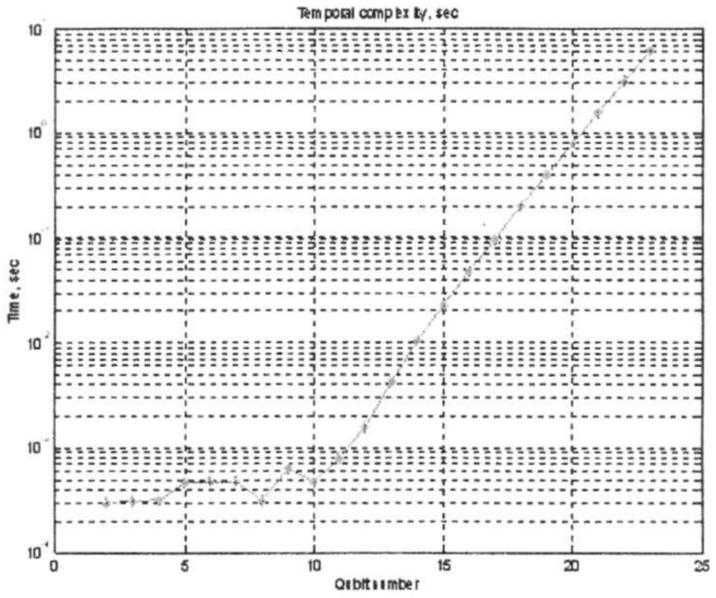


Figure 6: Temporal complexity of Grover's QSA

entanglement and interference operators), able to perform Grover interference without products.

The proposed circuit, which is one of the first hardware realizations of QA, is also the first one not based on matrices products but on functional relation between input and output vectors. A general form of the entanglement output vector $U_F = G$in Eq. (18) can be the following:

$$G = [g_1, g_2, \ldots, g_i, \ldots, g_{2^{n+1}}] \tag{21}$$

where $g_i = y_i \oplus f_{1+\frac{INT(i-1)}{2}}$ and y_i is the general term of superposition transformed in a suitable binary value.

The so-called superposition vector is fixed if we choose as input the canonical base. In order to find a suitable input-output relation, some particular properties of matrix $D_n \otimes I$ have to be taken in consideration.

The generic element v_i of V(that is our output in Eq. (18)) can be written as follows in function of g_i:

$$v_i = \begin{cases} \frac{1}{2^{n-1}} \sum\limits_{j=1}^{2^n} g_{2j-1} - g_i \ , \ for\ i\ odd \\ \frac{1}{2^{n-1}} \sum\limits_{j=1}^{2^n} g_{2j} - g_i \ , \quad for\ i\ even \end{cases} \tag{22}$$

This fact allows a great reduction of the number of operation (and therefore of electronics components) and consequently a significant increase of computational speed.

According to the proposed high-level scheme [4. 5], our circuit realization can be divided into two main parts:

Part I: *Base module.* It implements a 3-qubits system and it performs step-by-step calculation of output values. This part is divided in the following subparts:

a: *Entanglement*	**c**: *Interference*
b: *Pre-Interference*	**d**: *Modular interface*

Part II: *Control module.* It performs entropy evaluation in Eqs (19) and (20), vector storing for iterations and output visualization. This part also provides initial superposition of basis vectors |0> and |1>.

Entanglement operator composed by eight driven switches (see, Figure 7).

Referring to Figure 9, the switches (MAX394) present in the circuit are only the odd ones. They receive the elements of initial superposed vector in couples (Vout1 and VN1, Vout2 and VN2...) and perform the exchange according to the signal coming from the encoder.

The output signals (O1,...,O8) are the odd values of the entangled vector (even values are correspondent opposite value). These values are summed and scaled (the scaling factor is $1/4$ in the case of three qubits) by the OPAMP (see Figure 8), which constitutes the pre-interference step.

The differences among this sum and each one of the elements are performed by the Interference block, whose structure is reported in Figure 9.

6. Modular system

In order to realize modular system, some devices has been introduced. First and more important is operational amplifier (see, Figure 10). Labels M1, M2, M3 in Figure 10 are joined with corresponding others of different modules, performing parallel configuration. By this way output of two modules was summed and divided by 2, which is the result we wish to obtain in order to realize Grover algorithm [4,5] for $n+1$ qubits. In fact each module performs three qubits QSA and by adding a second module we can realize four qubits QSA. Each module must be unequivocally identified through his address (a selector assign this address on each one).

So the control module can send information to a specified module. The control module [5] send bit stream containing

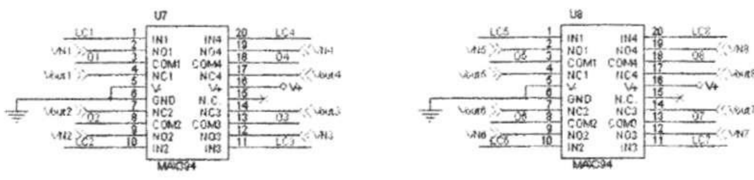


Figure 7: Entanglement circuit

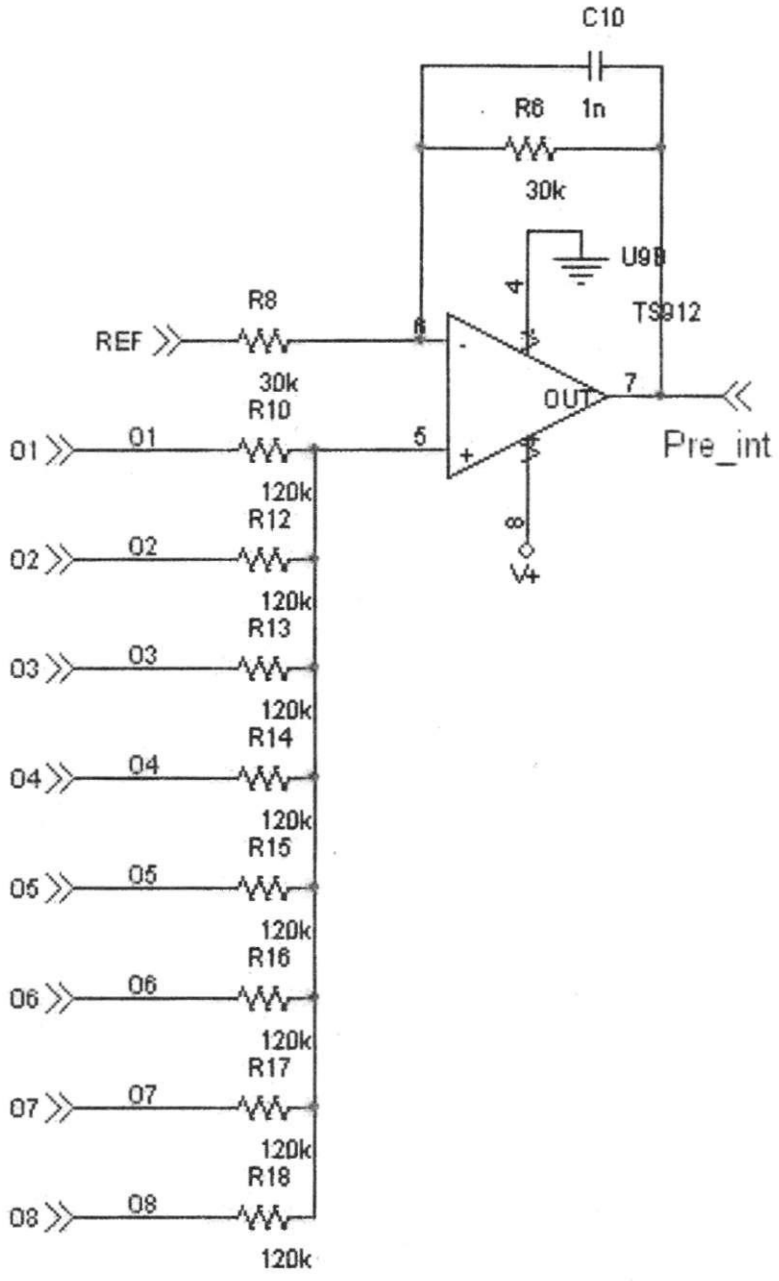


Figure 8: Pre-interference circuit

address and data to bus. On each module there is a device (74HC85A in Figure 11) that compares address sent by CPLD with the label of module and, if these are equal, allows it to process data.

Therefore these address indicate which modules must be in third state or which other must communicate through D/A and A/D converters with CPLD (see, Figure 11).

As previously reported the Control module performs entropy evaluation, vector storing for iterations and output visualization; however its main aim is to manage algorithm iterations.

Control module, that has been realized in digital way (CPLD programmable logic), is able to communicate with Base Modules through addressing system previously described (see in Figure 11 comparator '74HC85A') and D/A, A/D converters.

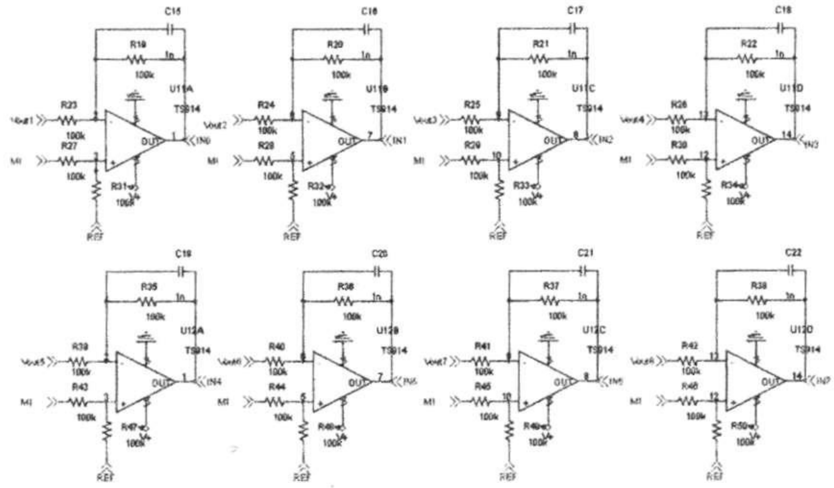


Figure 9: Interference circuit

Figure 13 shows the main board ($n = 3$) for Grover's QSA that realized the modular structure.

Figure 14 shows Pre-prototype board implementation of 3-qubit version of Grover's QSA. With this pre-prototype successful experimental simulation result of Grover's quantum gate Eq. (18) is achieved.

Information criteria as minimum Shannon entropy defined in Eq. (19) and $x_0 = |01\rangle \equiv 1$ as searching element are used. Analysis of these experimental results in detail is developed in [9].

In order to provide the target value (element to be find) each base module has a latch able to store it (see, Figure 12).

General methodology of quantum algorithm gate design and SW simulation results are described in [8 - 11].

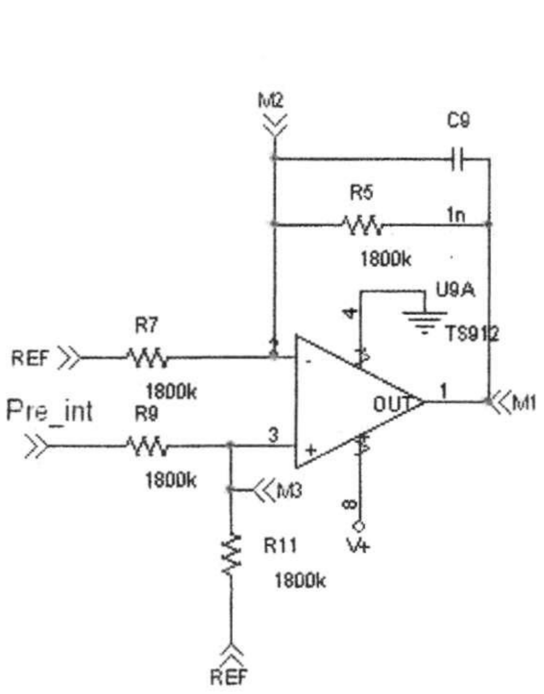


Figure 10: Modular Interface, Increasing qubit

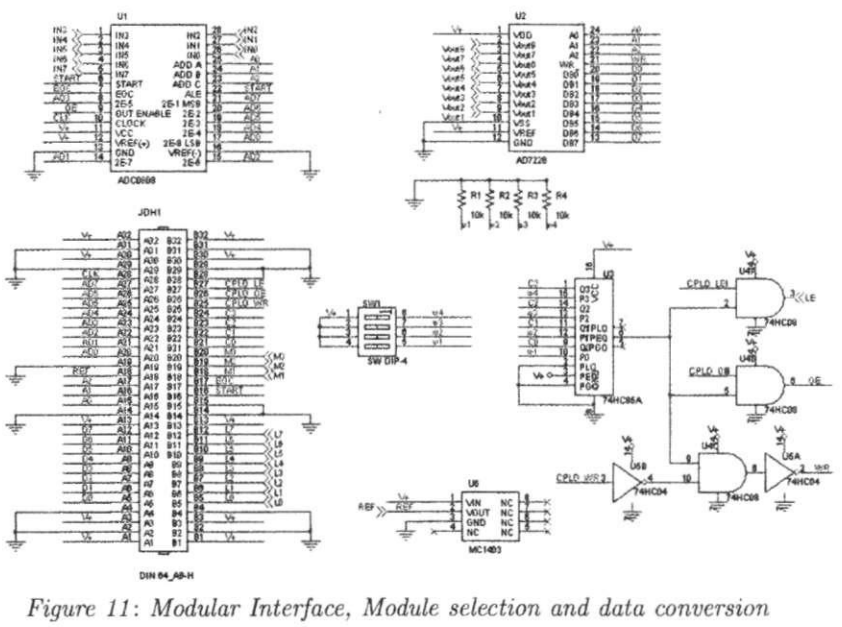


Figure 11: Modular Interface, Module selection and data conversion

7. Conclusions

1. Efficient simulation of QA's on classical computer with large number of inputs is difficult problem. For example, to operate only with 50 qubits state vector directly, it is necessary to have at least 128TB of memory (for the moment largest supercomputer has only 10TB [12]). In present report, for concrete important example as Grover's QSA, it is demonstrated the possibility to override spatio-temporal complexity, and to perform efficient simulations of QA on classical computers.

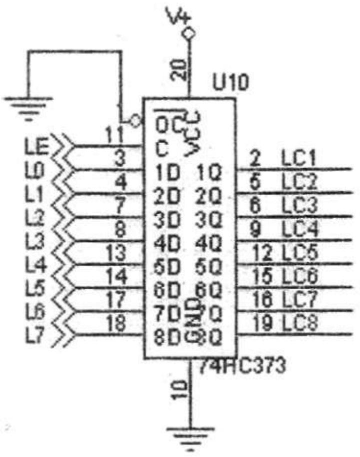


Figure 12: Modular Interface, Latch

2. Design method and hardware implementation of modular system for realization of Grover's Quantum Search Algorithm are presented. Hardware design of main quantum operators for quantum algorithm gates simulation on classical computer is developed. Hardware implementation for realization of information criteria as minimum Shannon entropy for quantum algorithm termination is demonstrated.

3. These results are the background for efficient simulation on classical computer the quantum soft computing algorithms, robust fuzzy control based on quantum genetic (evolutionary) algorithms and quantum fuzzy neural networks (that can realized as modified Grover's QSA), AI-problems as quantum game's gate simulation approaches and quantum learning, quantum associative memory, quantum optimization, etc. [5, 8, 9, 14 - 24].

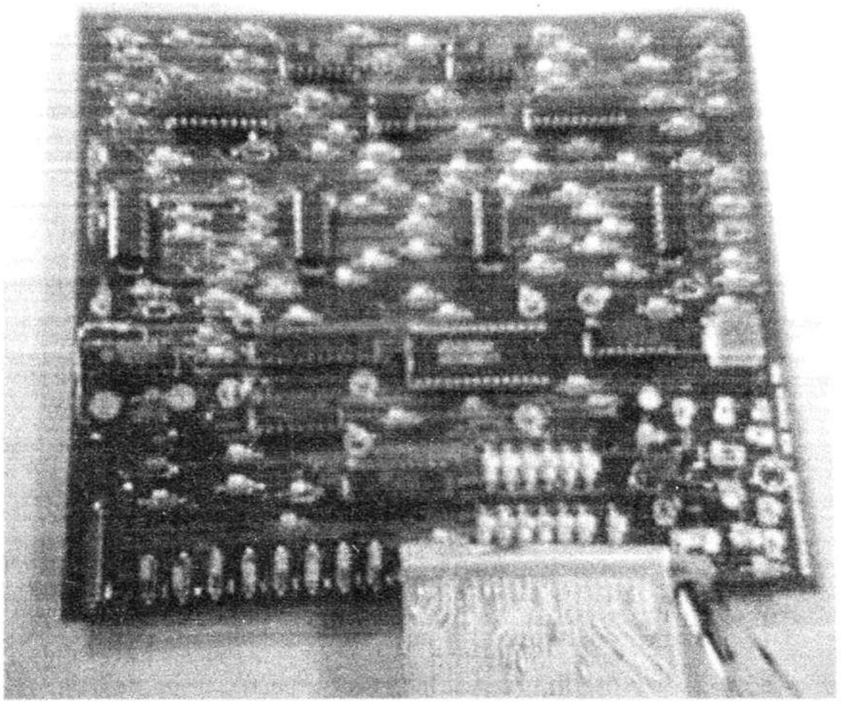


Figure 13: Main 3-qubits board with modular structure

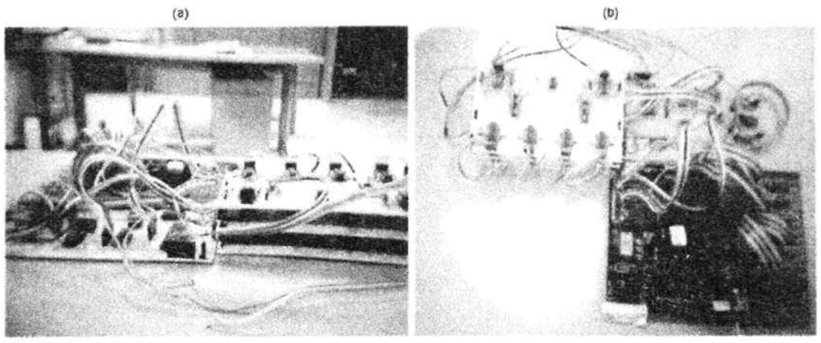


Figure 14: Pre-prototype of Grover's QSA gate

References

[1] Shor P. Why haven't more quantum algorithms been found? // J. ACM. 2003. V. 50. No 1.

[2] Nielsen M.A., Chuang I.L. Quantum Computation and Quantum Information. Cambridge University Press, UK. 2000.

[3] Niwa J., Matsumoto K., Imai H. General-purpose parallel simulator for quantum computing // Physical Review A. 2002 .V. A66. No 6.

[4] Ghisi F., S.V. Ulyanov The information role of entanglement and interference operators in Shor's quantum algorithm gate dynamics // J. Modern Optics. 2000. V .47. No 12. P. 2079-2090.

[5] Ulyanov S.V., Porto D. Method and hardware architecture for controlling a process or for processing data based on quantum soft computing // PCT patent WO 01/67186 A1, 2000.

[6] Ulyanov S.V., Kurawaki I. Information analysis of quantum gates for simulation of quantum algorithms on classical computers // Proc. QCM&C 2000. Kluwer Publ., 2001. P. 207 - 214.

[7] Ulyanov I., L. Litvintseva, A. Yazenin Fast algorithm for efficient simulation of quantum algorithm gates on classical computer // Proc. SCI'2003, Orlando, USA. V. III, 2003. P. 416 - 421.

[8] Ulyanov S.V., Takahashi K., Rizzotto G.G. Quantum soft computing: Quantum global optimization and quantum learning processes – Benchmarks of application in AI, informatics and intelligent control systems // Proc. SCI' 2003, Orlando, USA. Vol. III. 2003. P. 422 – 427.

[9] Ulyanov S.V., Takahashi K, Ulyanov I.S. Quantum soft computing via robust control: From Structure and HW implementation of quantum algorithm gates to Classical efficient simulation of Quantum Algorithm Gates and Robust Control // Proc. ICSCCW'2003, Turkey, 2003.

[10] Rizzotto G.G, Amato P., Porto D. M. Design of an analogue circuit implementing the superposition operation in a quantum search algorithm // PCT patent n. 01830383.4, 2001,

[11] Rizzotto G.G, Amato P., Porto D. A new method and circuit for implementing entanglement and interference operations in a quantum search algorithm // PCT patent n. 02425447.6, 2002.

[12] http://www.es.jamstec.go.jp

[13] Galindo A., Martin-Delgado M.A. Information and computation: Classical and quantum aspects // Review of Modern Physics. 2002. V. 74. No 2. P. 347 - 377.

[14] Giraldi G.A., Porugal R., Thess R.N. Genetic algorithms and quantum computation // arXiv: cs.NE/043003v1 4 Mar 2004. 27p.

[15] Martinez M., Longpré L., Kreinovich V. Fast quantum algorithms for handling probabilistic, interval, and fuzzy uncertainty // Proc. IEEE Conference, 2003. P. 395 – 400.

[16] Rigatos G.G., Tzafestas S.G. Parallelization of a fuzzy control algorithm using quantum computation // IEEE Trans. Fuzzy Systems. 2002. V. 10. No 4. P. 451 – 460.

[17] Han K.-H., Kim J.-H. Quantum-inspired evolutionary algorithms with a new termination criterion, H_ε gate, and two-phase scheme // IEEE Trans. Evolutionary Computation. 2004. V. 8. No 2. P. 156 – 169.

[18] Kouda N., Matsui N., Nishimura H. Control for swing-up of an inverted pendulum using qubit neural network // Proc. SICE'2002, Osaka. 2002. P. 765 – 770.

[19] Zhang G., Jin W., Hu L. Quantum evolutionary algorithm for multi-objective optimization problems // Proc. IEEE Intern. Symp. on Intelligent Control, Texas, USA. 2003. P. 703 – 708.

[20] Ulyanov S.V., Litvintseva L.V., Ulyanov S.S. Computational intelligence with quantum game's approach and robust decision-making in communication information uncertainty // Proc. Intern. Conf. Computational Intelligence (*ICCI'2004*). North Cyprus, 2004. P. 172 – 187.

[21] Andrecut M., Ali M.K. Quantum associative memory // Inter. J. Modern Physics. 2003. V. 17, No 12. P. 2447 – 2472.

[22] Hunziker M., Meyer D.A., Park J. The geometry of quantum learning // arXiv: quant-ph/0309059v1 5 Sep 2003. 20p.

[23] Pykacz J., D'Hooghe B., Zapatrin R. Quantum computers as fuzzy computers // Fuzzy Days 2001. LNCS. V. 2206. 2001. P. 526 – 535.

[24] Ulyanov S. V. System and method for control using quantum soft computing // US patent 6,578,018 B1. 2003.

Appendix: interpretation of main quantum operators in grover's QSA

We present the steps in Grover's algorithm [13] with the quantum circuit shown in Figure 1b.

A1. Grover's quantum search algorithm.

Step 1. Initialize the quantum registers to the state

$$|\psi_1\rangle := |00...0\rangle\,|1\rangle\,.$$

Step 2. Apply bit-wise the Hadamard one-qubit gate to the source register, so as to produce a uniform superposition of basis states in the source register, and also to the target register:

$$|\psi_2\rangle := U_H^{\otimes(n+1)}\,|\psi_1\rangle = \frac{1}{2^{(n+1)/2}}\sum_{x=0}^{2^n-1}|x\rangle\boxed{\sum_{y=0,1}(-1)^y\,|y\rangle}.$$

Step 3. Apply the operator

$$U_{f_{x_0}} : |\psi_3\rangle := U_{f_{x_0}}\,|\psi_2\rangle = \frac{1}{2^{(n+1)/2}}\sum_{x=0}^{2^n-1}(-1)^{f_{x_0}(x)}\,|x\rangle\boxed{\sum_{y=0,1}(-1)^y\,|y\rangle}.$$

Let U_{x_0}be the operator by $U_{x_0}|x\rangle := (1-2|x_0\rangle\langle x_0|)|x\rangle = \left\{\begin{array}{ll} -|x_0\rangle & \text{if } x = x_0, \\ +|x\rangle & \text{if } x \neq x_0 \end{array}\right.$, that is, it flips the amplitude of the marked state leaving the remaining source basis states unchanged.

Grover presents this operator graphically with a sort of "quantum comb" where the spikes denote the uniform amplitudes of state (Step 2) and the action of U_{x_0} is to flip over the spike corresponding to the marked item.

We realize that the state in the source register of Step 3 equals precisely the result of the action of U_{x_0}, i.e. $|\psi_3\rangle = ([1-2|x_0\rangle\langle x_0|]\otimes 1)|\psi_2\rangle$.

Step 4. Apply next the operation D known as *inversion about the average.* This operator is defined as follows$D_n := -\left(U_H^{\otimes n}\otimes I\right)U_{f_0}\left(U_H^{\otimes n}\otimes I\right)$, where $U_{f_0}$is the operator in Step 3 (for$x_0 = 0$). The effect of this operator on the source is to transform$\sum_x \alpha_x|x\rangle \mapsto \sum_x(-\alpha_x + \langle\alpha\rangle)|x\rangle$, where $\langle\alpha\rangle := 2^{-n}\sum_x \alpha_x$is the mean of the amplitudes, so its net effect is to amplify the amplitude of $|x_0\rangle$ over the rest.

Step 5. Iterate Steps 3 and 4 a number of times m.

Step 6. Measure the source qubits (in the computational basis). The number m is determined such that the probability of finding the searched item x_0 is maximal.

Remark. The basic component of the algorithm is the quantum operation encoded in Steps 3 and 4, which is repeatedly applied to the uniform state $|\psi_2\rangle$ in order to find the marked element.

Remark. Although this procedure resembles the classical strategy, Grover's neatly designed operation enhances by constructive interference of quantum amplitudes (see Table 1) the presence of the marked state one looks for.

It is possible to give a more general formulation to the operators entering Steps 3 and 4 of the algorithm [13]. To this end it is sufficient to focus on the source qubits and introduce the following definitions.

A2. Grover's QA main operators and its properties. A *Grover operator* G is any unitary operator with at most two different eigenvalues, i.e., G is a linear superposition of two orthogonal projectors P and Q:

$$G = \alpha P + \beta Q \qquad P^2 = P \qquad Q^2 = Q \qquad P + Q = 1$$

where $\alpha, \beta \in \mathbb{C}$ are complex numbers of unit norm.

A *Grover kernel* K is the product of two Grover operators: $K = G_2G_1$. Some elementary properties follow immediately from these definitions: 1) Any Grover kernel K is a unitary operator; 2). Let the Grover operators G_1, G_2 be chosen such that$G_1 = \alpha P_{x_0} + \beta Q_{x_0}, P_{x_0} + Q_{x_0} = 1$, $G_2 = \gamma\bar{P} + \delta\bar{Q}$, $\bar{P} + \bar{Q} = 1$, with $P_{x_0} = |x_0\rangle\langle x_0|$, and $\bar{P}$ given by the rank 1 matrix

$$\bar{P} := \frac{1}{N}\begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}.$$

This is clearly a projector $\bar{P} = |k_0\rangle\langle k_0|$ on the subspace spanned by the state$|k_0\rangle = \frac{1}{\sqrt{N}}(1, ..., 1)^t$, where the superscript denotes the transpose.

Then, if we take the following parameters, $\alpha = -1, \beta = 1, \gamma = -1, \delta = 1$, the Grover kernel K reproduces the original Grover's choice. This property follows immediately by construction. In fact, we have in this case $G_1 = 1 - 2P_{x_0} =: G_{x_0}$whilst the operator $G_2 = 1 - 2\bar{P}$ coincides (up to a sign) with the diffusion operator D introduced by Grover to implement the inversion about the average of Step 4. The iterative part of the algorithm in Step 5 corresponds to applying m times the Grover kernel K to the initial state $|x_{in}\rangle := 2^{-n/2}\sum\limits_x |x\rangle$, which describes the source qubits after Step 2, searching for a final state $|x_f\rangle$ of the form $|x_f\rangle := K^m |x_{in}\rangle$ such that the probability $p(x_0)$ of finding the marked state is above a given threshold value. We may take this value to be $^1/_2$, meaning that we choose a probability of success of 50% or larger. Thus, we are seeking under which circumstances the following condition $p(x_0) = |\langle x_0| K^m |x_{in}\rangle|^2 \geqslant 1/2$ holds true. The analysis of this probability gets simplified if we realize that the evolution associated to the searching problem can be mapped onto a reduced 2D-space spanned by vectors$\left\{ |x_0\rangle, |x_\perp\rangle := \frac{1}{\sqrt{N-1}} \sum\limits_{x \neq x_0} |x\rangle \right\}$. Then we can easily compute the projections of the Grover operators G_1, G_2 in the reduced basis with the result

$$G_1 = \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix}, \quad G_2 = \begin{pmatrix} \delta & 0 \\ 0 & \gamma \end{pmatrix} + (\gamma - \delta) \begin{pmatrix} \frac{1}{N} & \frac{\sqrt{N-1}}{N} \\ \frac{\sqrt{N-1}}{N} & -\frac{1}{N} \end{pmatrix}.$$

Form now on, we shall fix two of the phase parameters using the freedom we have to define each Grover factor in K up to an overall phase. Then we decide to fix them as follows: $\alpha = \gamma = -1$. With this choice, the Grover kernel K takes the following form in this basis

$$K = \frac{1}{N} \begin{pmatrix} 1 + \delta(1-N) & -\beta(1+\delta)\sqrt{N-1} \\ (1+\delta)\sqrt{N-1} & \beta(1+\delta-N) \end{pmatrix}.$$

The source state $|x_n\rangle$ has the following components in the reduced basis

$$|x_n\rangle \quad = \quad \frac{1}{\sqrt{N}} |x_0\rangle + \sqrt{\frac{N-1}{N}} |x_\perp\rangle .$$

In order to compute the probability amplitude in$p(x_0)$, we introduce the spectral decomposition of the Grover kernel K in terms of its eigenvectors $\{|k_1\rangle, |k_2\rangle\}$ with eigenvalues$e^{i\omega_1}, e^{i\omega_2}$. Thus we have

$$a(x_0) := \langle x_0| K^m |x_{in}\rangle = \frac{1}{\sqrt{N}} \sum_{j=1}^{2} \left\{ |\langle x_0 \mid k_j\rangle|^2 + \sqrt{N-1} \langle x_0 \mid k_j\rangle \langle k_j \mid x_\perp\rangle \right\} e^{im\omega_j}.$$

This in turn can be cast into the following closed form:

$\langle x_0| K^m |x_{in}\rangle \quad = \quad e^{im\omega_1} \left(\frac{1}{\sqrt{N}} + (e^{im\Delta\omega} - 1) \langle x_0 \mid k_2\rangle \langle k_2 \mid x_{in}\rangle \right)$, (A1)

with $\Delta\omega = \omega_2 - \omega_1$. In terms of the matrix invariants$DetK = \beta\delta, Tr = -(\beta + \delta) + (1+\beta)(1+\delta)\frac{1}{N}$, the eigenvalues $\varsigma_{1,2} := e^{i\omega_{1,2}}$are given by

$\varsigma_{1,2} = \frac{1}{2} TrK \mp \sqrt{-DetK + \frac{1}{4}(Trk)^2}$. (A2)

The corresponding unnormalized eigenvectors are

$$|k_{1,2}\rangle \propto \begin{pmatrix} \frac{A \mp \sqrt{-4(DetK)N^2 + A^2}}{2(1+\delta)\sqrt{N-1}} \\ 1 \end{pmatrix},$$

with$A := (\beta - \delta)N + (1-\beta)(1+\delta)$. Although we could work out all the expressions for a generic value N of elements in the list, we shall restrict our analysis to the case of a large number of elements, $N \to \infty$. Thus, in this asymptotic limit we need to know the behavior for $N \geqslant 1$of the eigenvector$|k_2\rangle$, which turns out to be

$$|k_2\rangle \propto \begin{pmatrix} \frac{\beta-\delta}{1+\delta}\sqrt{N} + O(\frac{1}{\sqrt{N}}) \\ 1 \end{pmatrix}.$$

Thus, for generic values of β, δ we observe that the first component of the eigenvector dominates over the second one, meaning that asymptotically $|k_2\rangle \sim |x_0\rangle$ and then

$$\langle x_0 \mid k_2\rangle \langle k_2 \mid x_{in}\rangle = O(\frac{1}{\sqrt{N}}).$$

This implies that the probability success will never reach the threshold value. Then we are forced to tune the values of the two parameters in order to have a well-defined and nontrivial algorithm, and we demand$\beta = \delta \neq -1$. Now the asymptotic behavior of the eigenvector changes and is given by a balanced superposition of marked and unmarked states, as follows$|k_2\rangle \sim \frac{1}{\sqrt{2}} \begin{pmatrix} i\delta^{1/2} \\ 1 \end{pmatrix}$.

This is normalized and we see that none of the component dominates. When we insert this expression into (1) we find

$$|\langle x_0| K^m |x_{in}\rangle| \sim \frac{1}{2} |\delta| \left|e^{im\Delta\omega} - 1\right| \sim \left|\sin(\frac{1}{2}m\Delta\omega)\right|.$$

This result means that we have succeeded in finding a class of algorithms, which are appropriate for solving the quantum search problem. Now we need to find out how efficient they are. To do this let us denote by M the smallest value of the time step m at which the probability becomes maximum; then, asymptotically, $M \sim [| \pi/\Delta\omega |]$.

As it happens, we are interested in the asymptotic behavior of this optimal period of time M. From the Eq. (A2) we find the following behavior as$N \to \infty$:$\Delta\omega \sim \frac{4}{\sqrt{N}} Re\sqrt{\delta}$. Thus, if we parameterize$\delta = e^{i\phi}$, then we finally obtain the expression$M \sim \left[\frac{\pi}{4\cos\phi/2}\sqrt{N}\right]$.

Therefore, we conclude that the Grover algorithm of the class parameterized by ϕ as a well-defined quantum search algorithm with an efficiency of order$O(\sqrt{N})$.

Remark. There have been many applications of Grover's work to quantum searching: finding the mean and median of a given set of values; searching the maximum/minimum; searching more than one marked item, quantum counting, i.e., finding the number of marked items without caring about their location, etc. There is also a nice geometrical interpretation of the Grover kernel $K = -G_2G_1$in terms of two reflections G_1 and $-G_2$, one about $|x_\perp\rangle$ and the other about $|x_{in}\rangle$, producing a simple rotation of the initial state by an angle $\theta = 2\arcsin\frac{1}{\sqrt{N}}$ in the plane spanned by $|x_0\rangle$and $|x_\perp\rangle$. With this construction it is straightforward to arrive at the following exact condition for the optimal value m of iterations:

$$m = \left[\frac{1}{2}\left(\frac{\pi}{2arcsin\frac{1}{\sqrt{N}}} - 1\right)\right].$$

Finally, it has been shown that Grover's algorithm is optimal, that is, its quadratic speed-up cannot be improved for unstructured lists.

Let us point out the following interpretation of the Grover's operators. Let us think of the computational basis $\{|x\rangle\}$ as a coordinate basis in Quantum Mechanics and introduce the quantum discrete Fourier transform in the standard fashion

$$|\hat{x}\rangle := U_{DFT}\,|x\rangle = \frac{1}{\sqrt{N}}\sum_{y=0}^{N-1} e^{2\pi i x\cdot y/N}\,|y\rangle.$$

The transformed basis $\{\,|\hat{x}\rangle\,\}$can then be seen as the dual momentum basis. Then, it is easy to see that in such a basis the projector operator $\bar{P}$ takes the following form: $U_{DFT}^{-1}\bar{P}U_{DFT} = |0\rangle\langle 0| =: P_0$. This means that the Grover operator G_2 takes the same matrix form in the momentum basis as the Grover operator G_1 in the coordinate basis. They are somehow dual of each other. The original Grover kernel takes then the form$K = U_{DFT}G_{x=0}U_{DFT}^{-1}G_{x_0}$, which shows that a Grover kernel has a part local in coordinate space and another part, which is local in momentum space.

A3. About the stability of Grover's algorithm. The expression for optimal number of iterations M can also be given another meaning regarding the stability of the Grover's case $\phi = 0$. It is plain that under a small perturbation $\delta\phi$ around this value, its optimal nature is not spoiled in first order for we find a behavior, which is quadratic in the perturbation, namely, $M \approx \frac{\pi}{4}(1 + 0.125(\delta\phi)^2)\sqrt{N}$. This stability considered here is with respect to perturbations in eigenvalues (or eigenvectors) in the reduced 2-dimensional subspace specified by the quantum search problem. We also require these types of perturbations to hold in all iterations. However, if we happen to choose a Grover kernel with a ϕ far from 0 we may end up with a searching algorithm for which the leading behavior order $O(\sqrt{N})$ is masqueraded by the big value of the coefficient and the time to achieve a succeeding probability becomes very large. For instance, we may have a Grover kernel with a behavior $M \approx 10^3\sqrt{N}$ and for a value of it would turn out as efficient as a classical algorithm of order$O(N) = 10^6$. Thus, the limit $\phi \to \pi$ behaves as a sort of classical limit where the quantum properties disappear.

Remark. We find this behavior as reminiscent of a quantum phase transition where the transition is driven by quantum fluctuations instead of standard thermal fluctuations. In this type of transition each quantum phase is characterized by a ground state, which is different in each phase. It is the variation of a coupling constant in the Hamiltonian of the quantum many-body problem which controls the occurrence of one quantum phase or another in the same manner as the temperature does the job in thermal transition. In our case we may consider the two different asymptotic behaviors of the eigenvector $|k_2\rangle$ as playing the role of two ground states. Following this analogy, we may see our family of algorithms parameterized by a torusT $= S^1 \times S^1$, where the parameters β and δ take their values and the difference $g := \beta - \delta$ is a sort of coupling constant which governs in which of the two phases we are. When $g \neq 0$ we fall into a sort of disordered phase where the efficiency of this class of Grover's algorithms is spoiled. However, when g $= 0$ we are located precisely at one equal superposition of the principal cycles of the torus which defines a one-parameter family of efficient algorithms.

A4. Robustness of Grover's Algorithm: The influence of initial conditions. Next we shall address the issue of to what extent this one-parameter family of algorithms depends on the choice of initial conditions for the initial state$|x_{in}\rangle$. We would like to check that the stable behavior we have found is not disturbed under perturbations of initial conditions. Let us consider a more general initial state$|x_{in}\rangle$, which is not the precise one used in the original Grover's algorithm but instead it is chosen as

$$|x_{in}\rangle \quad = \quad \frac{a}{\sqrt{N}}\,|x_0\rangle + b\sqrt{\frac{N-1}{N}}\,|x_\perp\rangle,$$

where a and b are chosen to satisfy a normalization condition. Then, it is possible to go over the previous analysis and find that the probability amplitude is now given by

$$\langle x_0|\,K^m\,|x_{in}\rangle \quad = \quad e^{im\omega_1}\left(\frac{a}{\sqrt{N}} + (e^{im\Delta\omega}-1)\,\langle x_0 \mid k_2\rangle\,\langle k_2 \mid x_{in}\rangle\right),$$

where now the new initial state is $|x_{in}\rangle$. We have to distinguish two cases: (i) the coefficient a of the marked state is order 1; and (ii) it is order bigger than 1, say of order$O(\sqrt{N})$. In the latter case (ii), it means that the initial state is so peaked around the marked state that we do not even need to resort to a searching algorithm, but instead measure directly on the initial state to find successfully the marked state. Therefore, we shall restrict to case (i) in the following. Now, the key point is to realize that all the previous asymptotic analysis is dominated by the behavior of the eigenvector $|k_2\rangle$ given by expression for$|k_2\rangle$, which is something intrinsic to the Grover kernel and independent of the initial conditions.

Thus, if condition $\beta = \delta \neq -1$ is not satisfied, then as we are in case (i) the first term in the RHS of last equations is not relevant and we are led again to the conclusion that the algorithm is not efficient. On the contrary, if condition $\beta = \delta \neq -1$ is satisfied, the same mechanism operates again and the algorithm has a probability of success measured by $|\langle x_0|\,K^m\,|x_{\,in}\rangle| \approx |\,b\,|\sin\left(\frac{m\Delta\omega}{2}\right)$ with $\Delta\omega$ also given by above expression. Then we may conclude that the class of algorithms is stable under perturbations of the initial conditions.