Fast algorithm for efficient simulation of quantum algorithm gates on classical computer

Sergey A. PANFILOV, Sergey V. ULYANOV, Ludmila V. LITVINTSEVA
Yamaha Motor Europe N.V. R&D Office, Via Bramante, 65, 26013, Crema (CR), Italy
Alexander V. YAZENIN
Dept. of Informatics, Tver State University, Ul. Zhelyabova 33, 170000, Tver, Russian Federation

ABSTRACT

The general approach for quantum algorithm simulation on classical computer is introduced. Efficient fast algorithm for simulation of Grover's quantum search algorithm in unsorted database is presented. Comparison with common quantum algorithm simulation approach is demonstrated.

Keywords: Quantum algorithm, efficient simulation, fast algorithms

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 hardware implementation of quantum computers.
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:

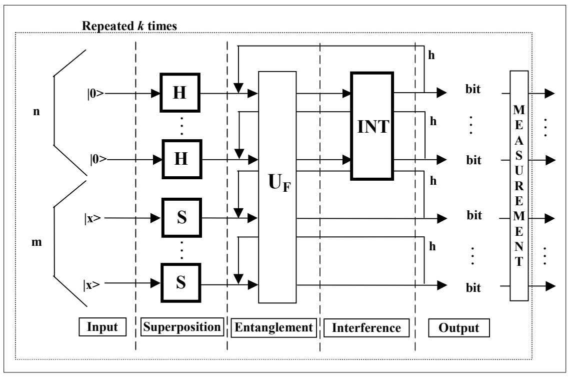
Input	A function f: $\{0,1\}^n \rightarrow \{0,1\}^m$
Problem	Find a certain property of f

Thus 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 QAs with more than 30-35 inputs [2] on classical computer with von Neumann architecture. In this report we present a practical approach to simulate most of known QAs on classical computers. We present the results of the classical efficient simulation of the Grover's quantum search algorithm (QSA) as a benchmark of this approach.

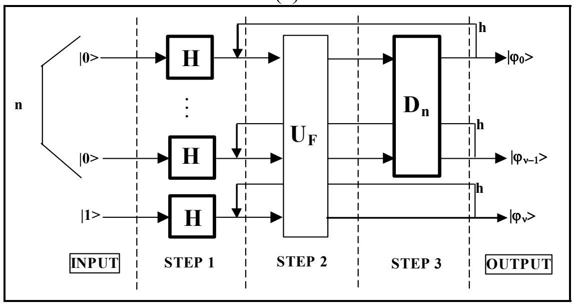
2. STRUCTURE OF QA GATE SYSTEM DESIGN

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 QAs: 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. Further more, local optimization of QA components according to specific hardware realization makes it possible to develop appropriate hardware accelerator of QA simulation using classical gates [3, 4].



(a)



(b)

Figure 1: a) Circuit representation of QA; b) Quantum circuit of Grover's QSA

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 [3].
The circuit presented in the Figure 1 is divided on 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 tensor product

$$|\phi_0\rangle = a_0|0...01\rangle = a_0|0\rangle \otimes \cdots \otimes |0\rangle \otimes |1\rangle, \quad (1)$$

where $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$; $|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$; $\otimes$ denotes Kronecker tensor product operation [1]. Such a quantum state can be presented as it is shown on the Figure 2a.

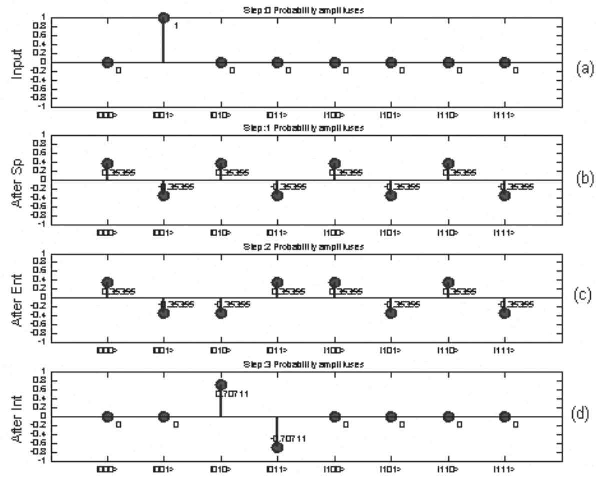


Figure 2: Dynamics of Grover's QSA probability amplitudes of state vector on each algorithm step

The coefficients in the Eq. (1) are called probability amplitudes [3]. Probability amplitudes may take negative or even complex values. The only one constraint on the values of the probability amplitudes is

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

The actual probability of the arbitrary quantum state $a_i\left|i\right\rangle$ to be measured is calculated as a square of its probability amplitude value $p_i = a^2_i$.

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 can not be represented as a tensor product of simpler state vectors. For example, consider state ϕ_1 presented on the Figure 2b and state ϕ_2 presented on the Figure 2c:

$$\begin{aligned}\phi_1 &= 0.35355\left(|000\rangle - |001\rangle + |010\rangle - |011\rangle + |100\rangle - |101\rangle + |110\rangle - |111\rangle\right) = \\ &= 0.35355\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right)\left(|0\rangle - |1\rangle\right)\\ \phi_2 &= 0.35355\left(|000\rangle - |001\rangle - |010\rangle + |011\rangle + |100\rangle - |101\rangle + |110\rangle - |111\rangle\right) = \\ &= 0.35355\left(|00\rangle - |01\rangle + |10\rangle + |11\rangle\right)|0\rangle - 0.35355\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right)|1\rangle\end{aligned}$$

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

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 2a in a probability amplitude representation and in the Figure 3a 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 QAs are realized by unitary quantum operators. Simulation of quantum operators is a key point in general QA simulation. In order to accelerate QAs basic quantum operators must be studied.

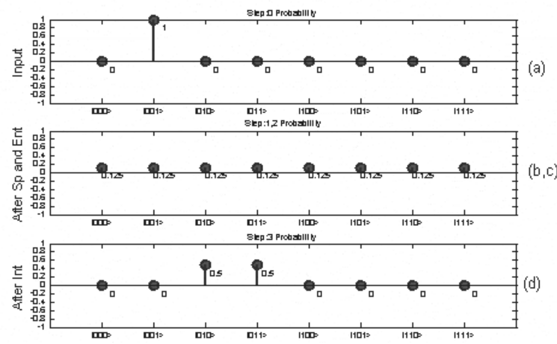


Figure 3: Dynamics of Grover's QSA probabilities of state vector on each algorithm step

2.2. Main QA operators

We consider superposition, entanglement and interference operators from simulation view point. 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 QAs, and differ only by function being analyzed.

Superposition operators of QAs In general, the superposition operator consists of the combination of the tensor products Hadamard H operators with identity operator I :

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

For most QAs the superposition operator can be expressed as

$$Sp = \left(\bigotimes_{i=1}^{n} H\right) \otimes \left(\bigotimes_{i=1}^{m} S\right) = {}^{n}H \otimes {}^{m}S, \tag{3}$$

where n and m are the numbers of inputs and of outputs respectively, left side power operation means tensor power. Operator, S depending on the algorithm may be or Hadamard operator H or identity operator I . Numbers of outputs m as well as structures of corresponding superposition and interference operators are presented in the Table 1 for different QAs.

Note that superposition and interference operators are often contain tensor power of Hadamard operator (H) which is called Walsh-Hadamard operator (W). It is known [3] that elements of the Walsh-Hadamard operator could be obtained as

$$\left[{}^{n+1}H\right]_{i,j} = \frac{(-1)^{i*j}}{2^{n/2}}\left[{}^{n}H\right], \tag{4}$$

where $i = 0,1,\ j = 0,1$.

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	${}^{n}H \otimes H$	1	${}^{n}H \otimes I$
Grover's	${}^{n}H \otimes H$	1	$D_n \otimes I$
Simon's	${}^{n}H \otimes {}^{n}I$	n	${}^{n}H \otimes {}^{n}I$
Shor's	${}^{n}H \otimes {}^{n}I$	n	$QFT_n \otimes {}^{n}I$

This approach improves greatly speedup 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, $n=1,\ m=1,\ S=I$:

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

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

$$[Sp]^{Deutsch-Jozsa's,Grover's}_{i,j} = \frac{(-1)^{i*j}}{2^{2/2}} \otimes H$$
$$= \frac{1}{2}\begin{pmatrix} (-1)^{0*0}H & (-1)^{0*0}H & (-1)^{0*1}H & (-1)^{0*1}H \\ (-1)^{0*0}H & (-1)^{0*0}H & (-1)^{0*1}H & (-1)^{0*1}H \\ (-1)^{1*0}H & (-1)^{1*1}H & (-1)^{1*1}H & (-1)^{1*1}H \\ (-1)^{1*0}H & (-1)^{1*1}H & (-1)^{1*1}H & (-1)^{1*1}H \end{pmatrix} \quad (6)$$
$$= \frac{1}{2}\begin{pmatrix} H & H & H & H \\ H & -H & H & -H \\ H & H & -H & -H \\ H & -H & -H & H \end{pmatrix}$$

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

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

Interference operators of main QAs 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.

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 $n=2$:

$$\left[Int^{Deutsch}\right]_{i,j} = {}^{2}H = \frac{(-1)^{i*j}}{2^{2/2}}H = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \quad (7)$$

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

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:

$$\left[Int^{Deutsch-Jozsa's}\right]_{i,j} = \frac{(-1)^{i*j}}{2^{\frac{n}{2}}} \otimes I, \quad (8)$$

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

Example 4: Interference operator of Deutsch-Jozsa's algorithm, $n=2, m=1$:

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

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

$$\left[Int^{Grover}\right]_{i,j} = D_n \otimes I = \left(\frac{1}{2^{n/2}} - {}^{n}I\right) \otimes I = \left(-1+\frac{1}{2^{n/2}}\right)\otimes I\Big|_{i=j}, \left(\frac{1}{2^{n/2}}\right)\otimes I\Big|_{i\neq j} = \frac{1}{2^{n/2}}\begin{cases} -I, i=j \\ I, i \neq j \end{cases}, \quad (10)$$

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

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

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

$$\left[Int^{Grover}\right]_{i,j} = D_2 \otimes I = \left(\frac{1}{2^{2/2}} - {}^2I\right)\otimes I$$
$$= \left(-1+\frac{1}{2}\right)\otimes I\Big|_{i=j}, \left(\frac{1}{2}\right)\otimes I\Big|_{i\neq j} \qquad (11)$$
$$= \frac{1}{2}\begin{pmatrix} -I & I & I & I \\ I & -I & I & I \\ I & I & -I & I \\ I & I & I & -I \end{pmatrix}.$$

Note that with growing 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. (10), without allocation of entire operator matrix.

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 algorithm) and can be described as following Eq. (12) and Eq.(12a):

$$\left[Int^{\ Simon}\right]_{i,j} = {}^nH \otimes^m I = \frac{(-1)^{i*j}}{2^{n/2}} \otimes {}^mI \qquad (12)$$
$$= \frac{1}{2^{n/2}}\begin{pmatrix} (-1)^{0*0}\cdot{}^mI & \cdots & (-1)^{0*j}\cdot{}^mI & \cdots & (-1)^{0*(2^n-1)}\cdot{}^mI \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (-1)^{i*0}\cdot{}^mI & \cdots & (-1)^{i*j}\cdot{}^mI & \cdots & (-1)^{i*(2^n-1)}\cdot{}^mI \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (-1)^{(2^n-1)*0}\cdot{}^mI & \cdots & (-1)^{(2^n-1)*j}\cdot{}^mI & \cdots & (-1)^{(2^n-1)*(2^n-1)}\cdot{}^mI \end{pmatrix}$$

Remark. In general, interference operator of Simon's algorithm coincides with interference operator of Deutsch-Jozsa's algorithm Eq. (8), but each block of the operator matrix Eq. (12) 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. (12), 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. (8) 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. (12) can be reduced as:

$$\left[Int^{Simon}\right]_{i,j} = {}^nH \otimes^m I = \frac{(-1)^{i*j}}{2^{n/2}} \otimes {}^mI$$
$$= \frac{1}{2^{n/2}}\begin{cases} {}^mI, \text{if } i \text{ is odd or if } j \text{ is odd} \\ -{}^mI, \text{if } i \text{ is even and } j \text{ is even} \end{cases} \qquad (12a)$$

Interference operator of Shor's algorithm. Interference operator of Shor's algorithm uses Quantum Fourier Transformation operator (QFT) [1], calculated as:

$$\left[QFT_n\right]_{i,j} = \frac{1}{2^{n/2}} e^{J(i*j)\frac{2\pi}{2^n}}, \qquad (13)$$

where: J - imaginary unit, $i = 0,\ldots,2^n-1$ and, $j = 0,\ldots,2^n-1$.

With $n=1$ we can observe the following relation:

$$QFT_{k_1}\Big|_{k_1=1} = \frac{1}{2^{\frac{1}{2}}}\begin{pmatrix} e^{J*(0*0)2\pi/2^1} & e^{J*(0*1)2\pi/2^1} \\ e^{J*(1*0)2\pi/2^1} & e^{J*(1*1)2\pi/2^1} \end{pmatrix} \qquad (14)$$
$$= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = H$$

Eq. (13) can be also presented in harmonic form using Euler formula:

$$\left[QFT_{k_1}\right]_{i,j} = \frac{1}{2^{k_1/2}}\left(\cos\left((i*j)\frac{2\pi}{2^{k_1}}\right) + J\sin\left((i*j)\frac{2\pi}{2^{k_1}}\right)\right)$$
(15)

Entanglement operators of main QAs 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:

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

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:

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

where $\oplus$ denotes addition modulo 2.

Having reversible function F we can design an entanglement operator matrix using the following rule:

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

B denotes binary coding. Actually resulted entanglement operator is a block diagonal matrix, of the form:

$$U_F = \begin{pmatrix} M_0 & & 0 \\ & \ddots & \\ 0 & & M_{2^n-1} \end{pmatrix} \qquad (16)$$

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

$$M_i = \bigotimes_{k=0}^{m-1}\begin{cases} I, \text{iff } F(i,k)=0 \\ C, \text{iff } F(i,k)=1 \end{cases}, \qquad (17)$$

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 property of sparse matrix operations it is possible to accelerate the simulation of the entanglement.

Example 6: Entanglement operator for binary function:

$$f:\{0,1\}^2 \to \{0,1\}^1,$$

such that: $f(x)=1\big|_{x=01}\ 0\big|_{x\neq 01}$

Reversible function F in this case will be:

$$F:\{0,1\}^3 \to \{0,1\}^3,$$

such that:

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

The corresponding entanglement block matrix can be written as:

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

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 7: Entanglement operator for binary function:

$$f:\{0,1\}^2 \to \{0,1\}^2,$$

such that: $f(x)=10\big|_{x=01,11}\ 00\big|_{x\neq 01,11}$

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

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

3. 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), (10), (16) and (17);

d) As termination condition we consider minimum of Shannon entropy of the quantum state, calculated as:

$$H^{Sh} = -\sum_{i=0}^{2^{m+n}} p_i \log p_i \tag{18}$$

Calculation of the Shannon entropy is applied to the quantum state after interference operation [5].

Minimum of Shannon entropy Eq. (18) 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 QAs are periodical. Figure 4 shows results of the Shannon information entropy calculation for the Grover's algorithm with 5 inputs.

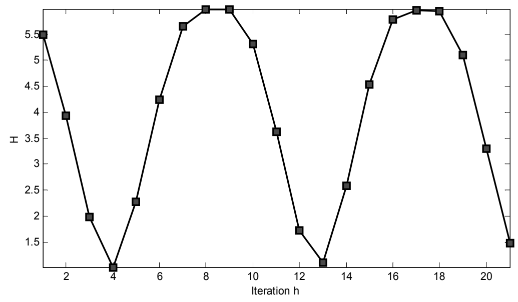


Figure 4: Shannon entropy analysis of Grover's QSA dynamics with five inputs

Figure 4 shows that for five inputs of Grover's QSA an optimal number of iterations, according to minimum of the Shannon entropy criteria for successful result, is exactly four. After that probability of correct answer will decrease and algorithm may fail to produce correct answer. Note that theoretical estimation for 5 inputs gives $\frac{\pi}{4}\sqrt{2^5}=4.44$ iterations.

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), (10), 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 [4]. State vector of probability amplitudes is stored in compressed form (only different probability amplitudes are allocated in memory). 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.

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

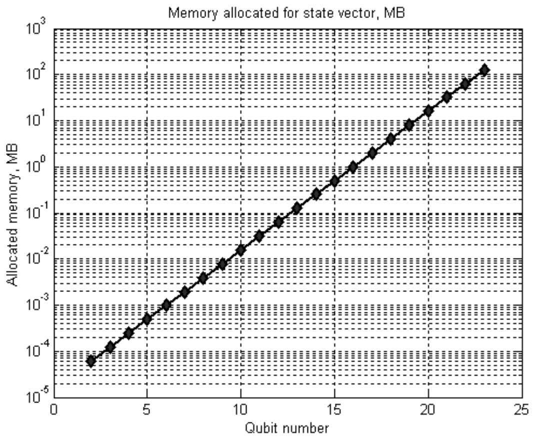


Figure 5: Spatial complexity of Grover QA simulation

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 [3, 4]. Fastest case is when we compress state vector and replace quantum operator matrices with corresponding classical gates according with [3,4]. In this case we obtain speedup according to Approach 2.

4. CONCLUSIONS

Efficient simulation of QAs 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 [6]). In present report, for concrete important example as Grover's QSA [1], it is demonstrated the possibility to override spatio-temporal complexity, and to perform efficient simulations of QA on classical computers. Comparison with sparse matrix based approaches [2] give us new effective possibility for simulation of quantum control algorithms using classical computers.

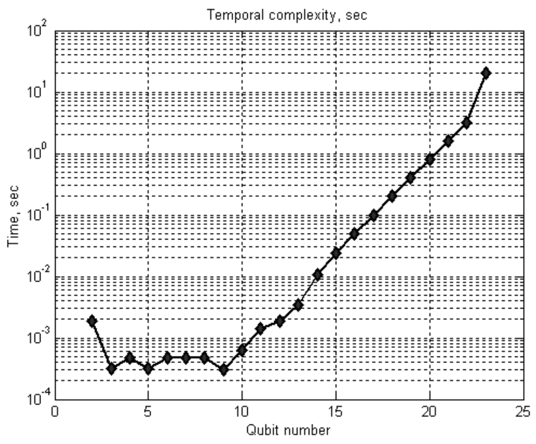


Figure 6: Temporal complexity of Grover's QSA

5. REFERENCES

[1] P. Shor, **Why haven't more quantum algorithms been found?**, Journal of the ACM (JACM)**,** Vol. 50, Issue 1, pp. 87-90, 2003; L. G. Valiant, **Quantum circuits that can be simulated classically in polynomial time**, SIAM J. of Computing**,** Vol. 31, No 4, pp. 1229-1254, 2002; L. Grover, **Quantum mechanics helps in searching of the needle in a haystack**, Phys. Rev. Lett., Vol. 79, No 2, pp. 325-328, 1997; M. Nielsen and I. Chuang, **Quantum Computation and Quantum Information**, Cambridge Univ. Press, 2002

[2] J. Niwa, K. Matsumoto and H. Imai, **General-purpose parallel simulator for quantum computing,** Phys. Rev. A, Vol. 66, 062317, 2002

[3] S.V. Ulyanov, F. Ghisi, S. Panfilov, I. Kurawaki and L.V. Litvintseva, **Simulation of quantum algorithms on classical computers**, Università degli Studi di Milano, Polo Didattico e di Ricerca di Crema Note del Polo Vol. 32, Crema, 1999

[4] P. Amato, S. Ulyanov, D. Porto, S.A. Panfilov and G. Rizzotto, **Hardware architecture system design of quantum algorithm gates for efficient simulation on classical computers,** in Proc. SCI 2003, Vol. 3, pp. 398-403, Orlando, 2003

[5] S.V. Ulyanov, S.A. Panfilov, I. Kurawaki, A.V. Yazenin, **Information analysis of quantum gates for simulation of quantum algorithms on classical computers**, in Proc. QCM&C2000, Kluwer Academic/Plenum Publ., pp.207-214, 2001

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