A Quantum Processing Framework for Quantum Algorithms

Mohammad Reza Soltan Aghaei[1], Zuriati Ahmad Zukarnain[2]
1- Department of Computer Engineering, Khorasgan Branch, Islamic Azad University, Isfahan, Iran.
Email: soltan@khuisf.ac.ir, msoltanaghaii@yahoo.com(Corresponding author)
2-Department of Communication Technology and Network, Faculty of Computer Science,UPM, Malaysia.
Email:zuriati@fsktm.upm.edu.my

Received: March 2012 Revised: June 2012 Accepted: July 2012

ABSTRACT:
The focus of this study is on developing a framework for a Quantum Algorithm Processing Unit (QAPU) with the hybrid architecture for classical-quantum algorithms. The framework is used to increase the implementation performance of quantum algorithms and design Quantum Processing Units (QPU). The framework shows a general plan for the architecture of quantum processors who is capable of run the quantum algorithms. In particular, the QAPU can be used as a quantum node to design a quantum multicomputer. At first, the hybrid architecture is designed for the quantum algorithms. Then, the relationship between the classical and the quantum part of hybrid algorithms is extracted, and main stages of the hybrid algorithm are developed. Next, the framework of the QAPU is designed. Furthermore, the framework is implemented and simulated for the existing quantum algorithms on a classic computer. It is shown that the framework is appropriate for quantum algorithms.

KEYWORDS: Quantum Processing Unit, Quantum Circuit, Quantum Algorithm, Quantum computing, Hybrid System.

1. INTRODUCTION

A quantum computer is a device that takes advantage of quantum mechanical effects to perform certain computations faster than a purely classical machine. It operates by manipulating those quantum bits or qubits with a fixed sequence of logic gates and performs it exponentially. The theory of quantum complexity determines when quantum computers may offer a computational speed-up over classical computers. At present, there are only a few generals well-known techniques in the field of quantum computing; furthermore, finding the problems that are amenable to quantum speedup is a high priority.

The Quantum Processing Unit (QPU) is the processor of a quantum computer that can do quantum computations. A typical component in QPU is a quantum device that runs quantum algorithms; namely, Quantum Algorithm Processing Unit (QAPU). This device can also be applied as a quantum node in quantum multi-computers. A quantum multicomputer is a distributed system that is composed of quantum computers connected through a quantum network. The nodes can be connected by creating an entangled state among them. Van Meter described the architecture of a quantum multicomputer for running Shor's factoring algorithm, but this architecture does not comprise other quantum algorithms [1]-[2]-[3]. Moreover; the quantum algorithms are also known as the hybrid (classical-quantum) algorithms, and thus, a general plan of hybrid algorithms is required to show the interaction between quantum and classical part of hybrid algorithms.

The focus of this study is on developing a framework of the QAPU and hybrid architecture for classical-quantum algorithms. Firstly, it was necessary to design the hybrid architecture for the quantum algorithms. This, however, required the analysis of the existing quantum algorithms as presented in the next section. Meanwhile, the relationship between the classical and quantum parts of the hybrid algorithms, and the main stages of the hybrid algorithm were determined. Furthermore, the framework was implemented and simulated for the existing quantum algorithms on a classic computer.

2. RELATED WORKS

Nowadays, many researchers focus on designing a quantum computer by implementing quantum algorithms. David Deutsch and Richard Jozsa showed an algorithm that could be run in poly-log time on a quantum computer, but required linear time on a deterministic Turing machine [4]. This may have been the first example of a quantum computer being shown to be exponentially faster than a deterministic Turing machine. In 1994, Peter W. Shor showed that the quantum factoring algorithm is a polynomial time algorithm for prime factorization and discrete logarithms on a quantum computer [5]. Lov Grover in 1996 discovered the quantum search algorithm. It is used to solve NP-hard problem [6]. It is sometimes referred to as amplitude amplification and has been found to be useful for quantum counting, and as a wrapper for other algorithms [7]-[8].Bennett and co-workers [9] and Zalka [10] showed that Grover's algorithm is optimal. No classical or quantum algorithm can solve this problem faster than a time of order. Furthermore, other algorithms are Simon's algorithm which finds the hidden string [11], Hallgren's algorithm which solves the Pell's equation [12], or the topic of quantum random walks [13]-[14]. Nonetheless, efficient quantum algorithms are limited in number and scope; no real breakthrough has yet been achieved in physical implementations. The most important point is that these algorithms are not still matured adequately to be applied in real quantum computations.

The motivation of this study was to continue the research of Van Meter [1] in designing multi-computers. He proposed the architecture of a quantum multicomputer optimized for running Shor's factoring algorithm [1]-[3]. The creation of the quantum multicomputer began with the optimization of the quantum modular exponentiation for Shor's factoring algorithm. The algorithm presented by Van meter reduces wall-clock time by a factor of one million for a six-thousand bit number [3]. Van Meter described it in terms of calculating factoring a large number, and this architecture does not comprise other quantum algorithms [1]-[2]-[3].Similarly, he described a linear topology for quantum networks, but left research on the hardware requirements of nodes which are capable to execute quantum algorithms and interconnection networks to other researchers. This study has been carried out to design the framework of nodes for a multicomputer that runs the available quantum algorithms. The quantum circuit of the node can be applied as a QAPU in the processor of a quantum computer.

So far, a few efficient quantum algorithms have been introduced. Although there has been no quantum computer to run these algorithms yet, each quantum algorithm has been done mathematically. The execution of the quantum algorithms is performed by simulation on a classical computer. There is no uniform platform that includes all of them. Therefore, a quantum processor unit that executes all the quantum algorithms is needed. Moreover, the quantum algorithms are also known as the hybrid (classical-quantum) algorithms, and thus, a general plan of hybrid algorithms is required to show the interaction between quantum and classical parts of hybrid algorithms. It is natural to have feedback loops for the interaction between classical and quantum parts of the hybrid architecture. On the other hand, in the theory of quantum computation, only feed forward quantum circuits are investigated, because a quantum circuit represents a sequence of applications of time evolution operators. Several researchers have studied quantum feedback control, [15]-[16] however, it is unclear whether those mechanisms can be physically implemented. These quantum circuit feedback loops are not usually presented to lead reversible quantum circuits [17]. On the other hand, Tetsuro Nishino argues that quantum circuit feedback loops should not be avoided, and a quantum recurrent circuit mode, which is a new convenient mathematical notation for ordinary quantum circuits, with some feedback mechanisms is introduced [18]. As such, it is natural to describe quantum circuits as ones with feedback loops if we want to visualize the total amount of the necessary hardware.

In this study, a general plan of hybrid algorithms and a framework of the QAPU which can execute quantum algorithms were developed. This processor can also be applied as a quantum node in a quantum multicomputer or a quantum co-processor in the classical computer which executes quantum algorithms. In achieving the main objective, the following steps are therefore adopted: At first, the quantum algorithms were analyzed to find the hybrid architecture and a framework for the quantum algorithms. Meanwhile, the hybrid architecture was designed for the quantum algorithms. Furthermore, the relationship between the classical and quantum parts of the hybrid algorithms, and the main stages of the hybrid algorithm were determined as presented in the next section. Finally the designed framework was implemented and simulated for the existing quantum algorithms on a classic computer.

Wang A.M. described a quantum network model mathematically [19]. This mathematical network model consists of some quantum sub-networks. Each sub-network is constructed of a set of elementary quantum gates. The quantum network in this paper corresponds to a unitary transformation. He improved the method of Barenco et al [20] to design a quantum network in terms of the elementary gates. The Wang's mathematical model [21] applies to the quantum part of the hybrid architecture for implementation and simulation.

3. HYBRID ARCHITECTURE

The QAPU is a quantum device that runs quantum algorithms. They require a hybrid architecture in order to execute quantum and classical operations. The classical operations can be run on a classical computer, and the quantum operations on a quantum computer. Fig. **1** illustrates the hybrid architecture between a classical computer with the CPU and a quantum computer with the QPU.

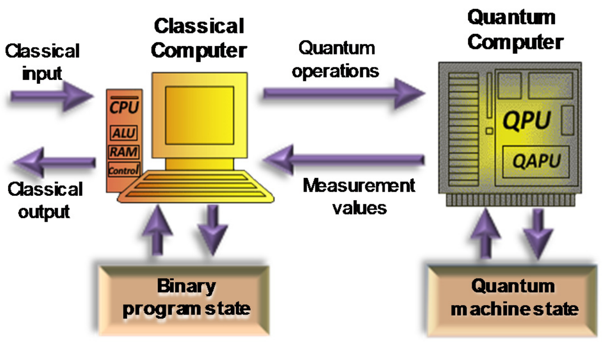


Fig. 1. The hybrid architecture of classical and quantum computers.

It is important to highlight that a QAPU can be used as a quantum node in a quantum multicomputer. a quantum multicomputer consists of a large number of small nodes and a qubus interconnection [1]. Any single quantum computer will have an ultimate limit to its storage capacity and performance. The nodes perform the actual computation. Each node consists of two halves, the quantum part which holds the quantum data, and the classical part which contains the real-time measurement and control circuitry for the quantum device. The classical operations are executed by the Classical node; therefore, it is replaced with the CPU. Meanwhile, the quantum operations are executed by the Quantum node, and it is replaced with the QAPU. The block diagram of the quantum multicomputer with these replacements is shown in Fig.2.

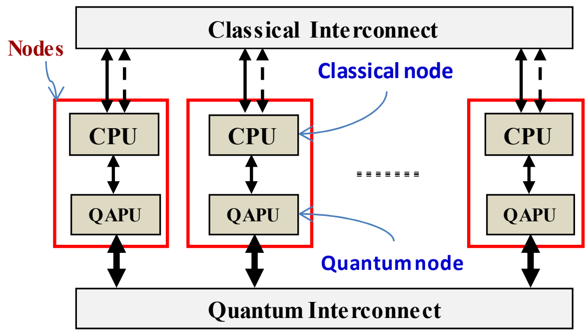


Fig.2. A block diagram of the quantum multicomputer.

In Fig.2, the nodes are shown to perform the actual computation. According to this diagram, the two halves are the QAPU and the CPU. Two real-time interconnections-classical and quantum are shown in the above figure. The quantum interconnection is based on the qubus approach for its link technology. The dashed lines are the non-real-time communication, while the solid lines are the real-time communication, either classical (thin lines) or quantum (thick lines).

In its simplest form, a quantum algorithm consists of a unitary transformation and a subsequent measurement of the resulting state. For the traditional computational tasks which include searching or mathematical calculations, efficient quantum implementations often have the form of probabilistic algorithms.

The quantum algorithms such as Shor's algorithm consist of two parts. The first part is a classical algorithm which can be run on a classical computer, while the second part is a quantum algorithm that can be run on a quantum computer or simulated on a classical computer. Fig.3 shows the relationship between the classical part and the quantum part of the hybrid architecture [22]-[23].

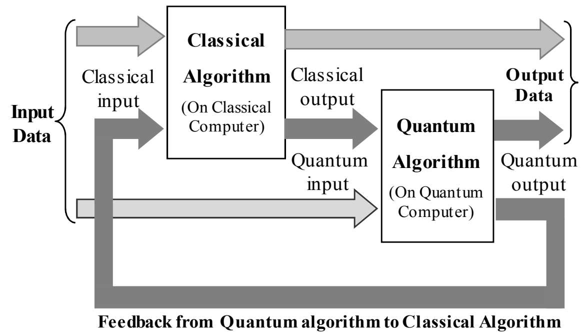


Fig.3. The relationship between the classical and quantum parts of the hybrid architecture.

The quantum algorithms are also known as hybrid algorithms that consist of both the classical and quantum components. Moreover, the quantum portion of many algorithms is probabilistic; often needs multiple runs to get the desired result. The main stages of the hybrid architecture can be done as follows:

1. Pre-calculate certain classical factors (initialize and run the classical part of the algorithm).
2. Run the quantum algorithm on the quantum circuit:
   a) Initialize the quantum node (Initialize the quantum circuit and define all gates, switches and unitary functions).
   b) Prepare inputs state (store inputs on target and control registers).
   c) Execute the quantum portion of the algorithm (apply gates and the unitary transformation on the input data).
   d) Measure the output of the machine state (measure the output registers of the quantum circuit).
   e) Evaluate Measurement (If the desired result is retrieved, then the post-processing in step 3 is done).
   f) Exit if the desired result is obtained (If a solution is found, then exit the quantum circuit, or else step 2 is repeated).
3. Finish post-processing (run the second classical part of the algorithm).

Steps 1 and 3 can be executed on a classical computer, while step 2 can be executed on a quantum computer using the quantum circuit. Measuring and evaluating the quantum circuit can be done on a classical computer through a simulation on a classical computer. If measuring and evaluating of the quantum circuit is a stage in the quantum part, then it is natural to have feedback loops in the quantum part of the hybrid architecture to visualize the total amount of the necessary hardware. In the theory of quantum computation, only feed forward quantum circuits are investigated. These quantum circuit feedback loops are not usually presented to lead reversible quantum circuits. On the other hand, in the present study, quantum circuit feedback loops are included based on the research by Tetsuro Nishino [18]. The diagram illustrated in Fig.4 indicates the development of a general plan in [23]for the hybrid algorithms simulated on a classical computer.

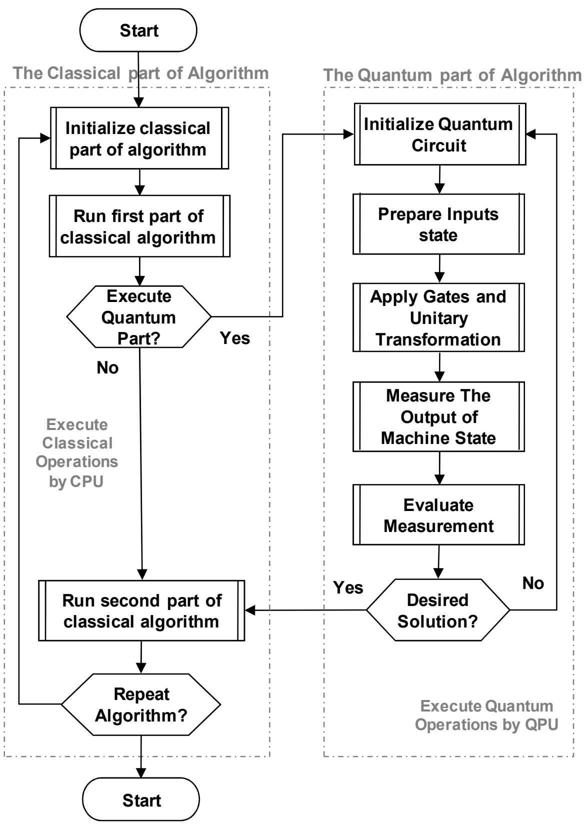


Fig.4. The classical and quantum parts of the hybrid architecture.

4. THE QAPU FRAMEWORK

The designed framework of QAPU is shown in Figure. The framework is setup, implemented and simulated for the existing quantum algorithms on a classic computer.

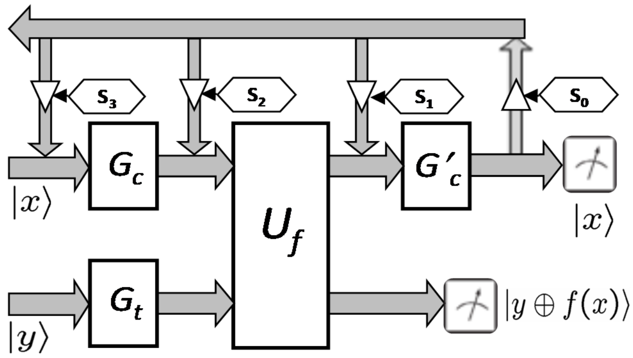


Fig.5. The QAPU framework

In this quantum circuit, there are two inputs $|x\rangle$ and $|y\rangle$. The input data require two registers. The first one named the *control register* is used to store $|x\rangle$ and the second register named *the target register* is used to store $|y\rangle$named. The first register $|x\rangle$ is applied on the inputs of gate G_c and the second one$|y\rangle$is applied on the inputs of gate G_t. An *n*-qubit quantum register can exist in a superposition of all possible 2^nstates $|0\rangle$ to $|2^n - 1\rangle$at the same time. This effect allows a quantum computer to calculate a function on all possible inputs at the same time, in a single pass. Therefore, the regular functions for G_c and G_t are Hadamard Transform and Quantum Fourier Transform (QFT). Most quantum algorithms are used in same functions for gate G'_c, therefore if G_c is QFT, then G'_c is QFT^{-1}.

The common part in all quantum algorithms is the black box or the oracle function U_f that is often used to model a subroutine of calculations and is reversible. Classically, a black-box function can be simply thought of as a box that evaluates an unknown function *f*. The input is some *n*-bit string $|x\rangle$and the output is given by an *m*-bit string$f(x)$. In quantum, such a box can only exist if it is reversible. To create a reversible box, the input $|x\rangle$ is output together with $f(x)$. To make the box reversible, an additional *m*-bit input $|y\rangle$is added and the output of the result is $|y \oplus f(x)\rangle$where $\oplus$denotes bitwise addition modulo 2. In particular, if $|y\rangle$ is fixed to be $y = 0 \dots 0$, the output is $f(x)$. Note that U_f now induces a transformation on *n*+*m*-bit strings that can be described by a permutation of the 2^{n+m}possible strings; in particular it is unitary.

In some quantum algorithms, the execution of some functions or operators is repeated and there is a feedback that makes iteration. For example, in the Grover's algorithm, the operator $G = HU_{0^\perp}HU_f$that can be applied by the following sequence of transformations and the feedback is iterated. In the framework, this feedback is implemented with switches S_0, S_1, S_2 and S_3. In the Grover's algorithm, the switches S_0 and S_2 are closed for $O(\sqrt{N})$ time, while S_1 and S_3 are always opened.

As mentioned above, Wang A.M. proposed a quantum network model mathematically[21]. This model can be applied to implement and simulate the quantum part (QAPU) of the hybrid algorithms.

5. RESULT

In this section, the results gathered from the implementation and simulations of the framework for the existing quantum algorithms are presented. The simulation was done on a classical computer with CPU Intel Core i7, 2.13 GHz and 8 GB RAM. The simulation of the quantum algorithms requires a huge amount of memory for the quantum computation. There are several limitations in choosing the maximum number of qubits for the input data, and we execute the quantum algorithms with maximum 20 qubits for the above computer. Matlab and C++ were used to implement the hybrid algorithms. The hybrid architecture and the framework of QAPU were implemented and tested using the existing quantum algorithms such as Deutsch's, Deutsch-Jozsa, Grover's, Simon's, Shor's algorithms, and also quantum version of Dijkestra's algorithm[22] and Prim's algorithm[24] which the quantum search can be replaced with the classical search.

In the quantum algorithm circuit, two n-qubit inputs $|x\rangle$ and $|y\rangle$ are defined. The gates G_c , G'_c , and G_t are defined for each algorithm. First, the register $|x\rangle$ is applied on the inputs of gate $G_c$and the second register $|y\rangle$is applied on the inputs of gate G_t. The usual functions for G_c and G_t are Hadamard Transform and Quantum Furrier Transform (QFT).

In some of the quantum algorithms, the execution of some functions or operators is iterated and there is a feedback that makes this iteration. This feedback is implemented with switches S_0, S_1, S_2 and S_3. These switches apply the iteration of some quantum functions in the known quantum algorithm such as Simon's and Grover's algorithms, and it is not such as a feedback loop. But it is natural to have feedback loops in the quantum part of hybrid architecture to visualize the total amount of the necessary hardware and for future quantum algorithms. As mentioned before, we may have feedback loop in quantum part of hybrid architecture, but in the theory of quantum computation, only feed forward quantum circuits are investigated. We include Tetsuro Nishino argues that quantum circuit feedback loops should not be avoided, and the QAPU framework included a feedback loop.

The initialization and setup of the existing quantum algorithms have been demonstrated on the proposed framework. In the next, the setting up of the framework for the Grover's search algorithm is explained.

5.1. Setting up the framework for Quantum Search Algorithm:

The Grover's algorithm includes two different inputs, |x⟩ and |y⟩. The n-qubit input |x⟩ is stored in the control register and the 1-qubit |y⟩ in the target register. As the quantum circuit of this algorithm, the n-qubit control register is initialized to $|\mathbf{0}\rangle^{\otimes n}$ and the 1-qubit target register to |1⟩. The gate Gc is replaced with the n-qubit Hadamard gate, while Gt is replaced with the 1-qubit Hadamard gate. The gate G′c is replaced with a sequence of operators$\boldsymbol{H^{\otimes n} U_{0\perp} H^{\otimes n}}$. Meanwhile, the Grover's algorithm iterates the operator $\boldsymbol{G = G'_c U_f = H^{\otimes n} U_{0\perp} H^{\otimes n} U_f}$ that is defined by the following sequence of transformations and known as the Grover Iterate. A feedback is needed to ensure the iteration of these sequence transformations of G for $\mathbf{O}(\sqrt{\boldsymbol{N}})$ times. This feedback can easily be implemented using switches S_0, S_1, S_2 and S_3. This can be done by closing S_0 and S_3, while leaving S_1 and S_2 open. The feedback is connected for $\mathbf{O}(\sqrt{\boldsymbol{N}})$ times that the algorithm is executed. The initialized framework for the Grover's algorithm is shown in Fig.6.

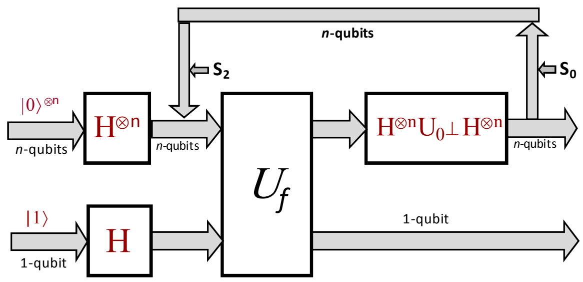


Fig.6. Setting up the framework for Grover's algorithm.

Classically, a deterministic algorithm needs to make $N = 2^n$ - *1* queries to identify marked item *w* in the worst case and a probabilistic algorithm still needs $O(2^n)$ queries, but the quantum search algorithm solves this problem with $O\left(\sqrt{2^n}\right)$ queries and this is known to be the best possible. The quantum search algorithm is executed with maximum 20 qubits as the input data by classical computer with CPU of Pentium Intel Core i7 and 8 GB RAM. The simulation result is described inFig.7shows n=20 qubits as a data index. The number of possible inputs is N=2^{20}=1,048,576 which is the number of records in, while the recorded number of 1,000,000 is the desired record. The amplitude value of the desired record reaches one, while this is zero for the other records after $(\pi/4)\sqrt{N}$ =803 iterations. In this figure, the execution of the algorithm is continued until O($\sqrt{N}$)=1024 iterations.

6. CONCLUSION

This study proposed the framework for a device which executes the quantum algorithms. This device can be applied as a unit in the quantum processing unit (QPU), namely QAPU. It can also be applied as the quantum node in the quantum multicomputer. The quantum algorithms are known as hybrid algorithms that consist of classical and quantum components. Moreover, the quantum portion of many algorithms is probabilistic; often need multiple runs to get the desired result. Therefore, at first, the hybrid architecture was designed for the quantum algorithms. The relationship between the classical and quantum parts of the hybrid algorithms was then extracted. After that the main stages of the hybrid architecture algorithm were determined as shown in Fig.4 Next, the QAPU framework was designed and developed as shown in Fig.5. Furthermore, the framework was setup, implemented and simulated for the existing quantum algorithms on a classic computer. It is shown that the framework is appropriate for the quantum algorithms. The framework is useful to design of quantum processor for the quantum computer.

REFERENCE

[1] R. Van Meter, K. Nemoto, W. J. Munro, and K. M. Itoh, **"Distributed arithmetic on a quantum multicomputer,"**ISCA '06 Proceedings of the 33rd annual international symposium on Computer Architecture2006, pp. 354-365.

[2] R. Van Meter and M. Oskin, **"Architectural implications of quantum computing technologies,"***ACM Journal on Emerging Technologies in Computing Systems (JETC),* vol. 2, pp. 31-63, 2006.

[3] R. Van Meter, K. M. Itoh, and T. D. Ladd, **"Architecture-dependent execution time of Shor's algorithm,"***Proc. Int. Symp. on Mesoscopic Superconductivity and Spintronics, Also Arxiv preprint quant-ph/0507023,* 2006.

[4] D. Deutsch and R. Jozsa, **"Rapid Solution of Problems by Quantum Computation,"***Proc. Royal Soc. London,* vol. 439, pp. 553-558, 1992.

[5] P. W. Shor, **"Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer,"** in *35th Ann. Symp. Foundations of Computer ScienceLos Alamitos, Calif.: IEEE Computer Society Press*, 1994, pp. 124-134.

[6] L. K. Grover, **"A Fast Quantum Mechanical Algorithm for Database Search,"** in *28th Annual ACM Symposium on the Theory of Computation* New York: ACM Press, 1996, pp. 212–219.

[7] L. K. Grover, **"Fixed-point quantum search,"***Physical Review Letters,* vol. 95, 2005.

[8] G. Brassard, P. Høyer, and A. Tapp, **"Quantum counting,"** in *25th Int. Colloquium on Automata, Languages and Programming (ICALP'98)*: Springer, 1998, pp. 820–831.

[9] C. H. Bennett and others, **"Strengths and Weaknesses of Quantum Computing,"***SIAM J. Computing,* vol. 26, pp. 1510-1523, 2001.

[10] C. Zalka, **"Grover's quantum searching algorithm is optimal,"***Physical Review A,* vol. 60, pp. 2746-2751, 1999.

[11] D. Simon, **"On the power of quantum computation,"** in *35th Ann. Symp. on Foundations Computer Science*: ACM, 1994, pp. 116–124.

[12] S. Hallgren, **"Polynomial-time quantum algorithms for pell's equation and the principal ideal problem,"** in *34th ACM Symp. on Theory of Computing (STOC)*, 2002, pp. 653-658.

[13] Y. Aharonov, L. Davidovich, and N. Zagury, **"Quantum random walks,"***Physical Review A,* vol. 48, pp. 1687–1690, 1993.

[14] J. Kempe, **"Quantum random walks: an introductory overview,"***Contemporary Physics,* vol. 44, pp. 307-327, 2003.

[15] S. Lloyd and J. J. E. Slotine, **"Quantum feedback with weak measurements,"***Physical Review A,* vol. 62, p. 12307, 2000.

[16] R. Ruskov and A. N. Korotkov, **"Quantum feedback control of a solid-state two-level system,"***Arxiv preprint cond-mat/0107280,* 2001.

[17] M. A. Nielsen and I. L. Chuang, ***Quantum Computation and Quantum Information*****:** Cambridge University Press, 2000.

[18] T. Nishino, **"Mathematical models of quantum computation,"***New Generation Computing,* vol. 20, pp. 317-337, 2002.

[19] A. M. Wang,**"A Universal Quantum Network-Quantum Central Processing Unit",***Chinese Physics Letters,* vol. 18, pp. 620-622, 2001.

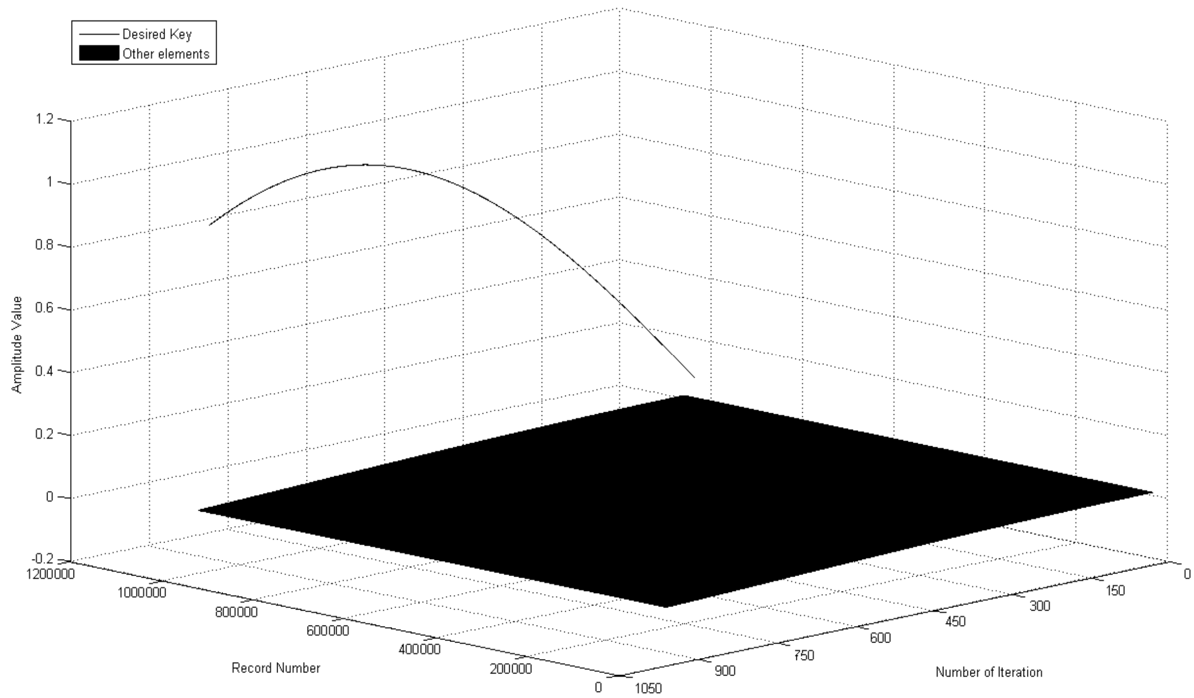
[20] A. Barenco, D. Deutsch, A. Ekert, and R. Jozsa, **"Conditional quantum dynamics and logic gates,"***Physical Review Letters,* vol. 74, pp. 4083-4086, 1995.

[21] A. M. Wang **"Quantum Central Processing Unit and Quantum Algorithm,"***Chinese Physics Letters,* vol. 19, pp. 620-622, 2002.

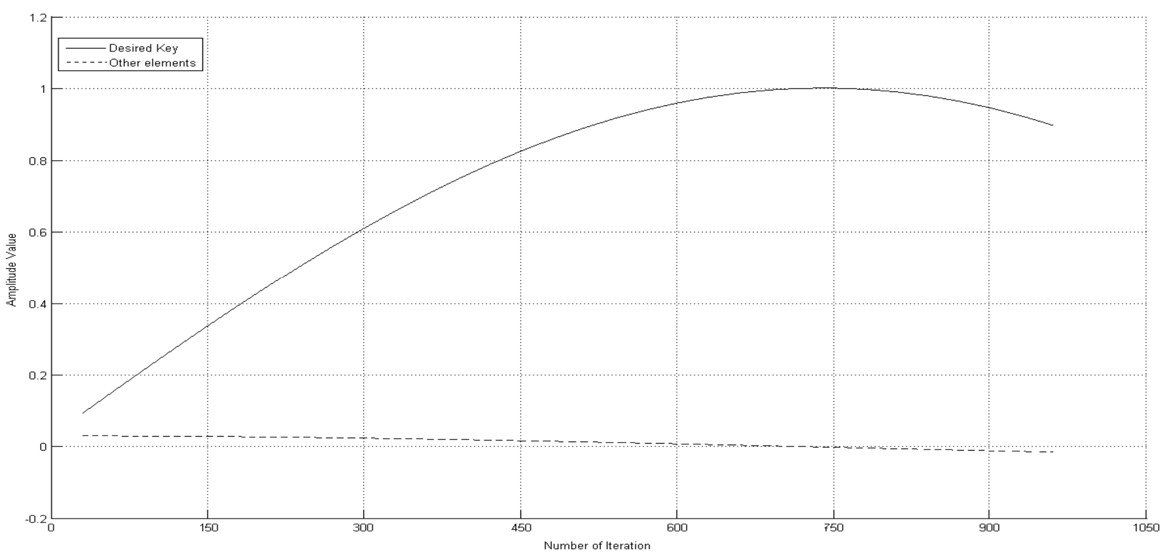
[22] M. R. Soltan Aghaei, Zuriati Ahmad Zukarnain, Ali Mamat, and H. Zainuddin, **"A Hybrid Algorithm for Finding Shortest Path in Network Routing,"***Journal of Theoretical and Applied Information Technology,* vol. 5, 2009.

[23] M. R. Soltan Aghaei, Zuriati Ahmad Zukarnain, Ali Mamat, and H. Zainuddin, **"A Hybrid Architecture Approach for Quantum Algorithms,"***Journal of Computer Science,* vol. 5, pp. 725-731, 2009.

[24] M. R. Soltan Aghaei, Zuriati Ahmad Zukarnain, Ali Mamat, and H. Zainuddin, **"A Quantum Algorithm for Minimal Spanning Tree,"** in *3rd Int. Sym. on Information Technology (ITsim08)* Malaysia: Proc. IEEE, 2008.



(a) The amplitude of 1,048,576 records and the record of 1,000,000 is the solution key which find with 803 iterations.



(b) Comparison the amplitude of key and other elements with 20 qubits input data and 1,048,576 records and 1024 iterations.

Fig.7. The simulation result of quantum search algorithm with 20 qubits.