Shor's Algorithm For Quantum Numbers Using MATLAB Simulator

Shweta Nagaich
Dept. of Computer science
ITM University
Gwalior, India
Shweta11nagaich@gmail.com

Y.C.Goswami
School of Engineering and Technology
ITM University
Gwalior, India
ycgoswami@yahoo.co.in

Abstract— In the field of Quantum computing, the Peter Shor gave an important algorithm known as Shor's factorization algorithm. With the existing coding it has an issue of finding order and periods finding. Search of new quantum numbers has been already on to strengthen the quantum computing. The development of quantum computer is also slow due to limited availability of simulation. In this paper, we have reported new quantum numbers using Shor's algorithm and simulation. The graphical representation of numbers are shown in this paper.

Keywords— Quantum Computing; Shor's algorithm; order finding; period finding; simulators

I. INTRODUCTION

A quantum computer is a computation system that makes direct use of quantum mechanical phenomena. To perform operation on data superposition and entanglement is used [1]. Quantum computers are different from digital computers based on transistors. Digital computers require data to be encoded into binary digits (bits), each of which is always in one of two definite states (0 or 1), quantum computation uses qubits (quantum bits), which can be in superposition's of states. A quantum computer with spins as quantum bits was also formulated for use as a quantum space time in 1969 [2]. Even today quantum computing is still in its infancy, but experiments have been carried out in which quantum computational operations were executed on a very small number of qubits [3]. Large-scale quantum computers will be able to solve certain problems much quicker than any classical computer using the best currently known algorithms, like integer factorization using Shor's algorithm or the simulation of quantum many-body systems. There exist quantum algorithms, such as Simon's algorithm, which run faster than any possible probabilistic classical algorithm [4]. Using computational resource a classical computer could be made to simulate any quantum algorithm; quantum computation [5].

Quantum computers require very different algorithms and the grand challenge for simulation is to develop simulation algorithm adapted to the coming era of quantum computers. In this paper we have reported simulation for shor's algorithm using MATLAB. Simulation result using quantum bits verify the possible eavesdropper's presence, changing the state of the system and hence disrupting the whole process. We have given new Quantum numbers that increase the speed of task and time complexity is decreases of computers.

II. SERVERY OF QUANTUM SIMULATION

In 1994 Peter Shor's, at AT&T's Bell Labs in New Jersey, discovers an important algorithm. It allowed a quantum computer to factor large integers quickly. It solved both the factoring problem and the discrete log problem. Shor's algorithm could theoretically get around many of the cryptosystems in use today. Its invention sparked a tremendous interest in quantum computers [6]. The Simulations of Shor's algorithm have been reported with aimed at mathematical applications and modeling discrete systems. When quantum computers become practical, these types of simulations should find use in control and AI applications.

Obenland and Despain [1998][7] describe a parallel simulator, which accesses the feasibility of quantum information processing systems. He validates an analytical example of execution time for the simulator, which indicates that parallel quantum computer simulation is really scalable. Tucci [2000][8] proposed a novel family of quantum computing algorithms, which generalized the Deutsch-Jozsa, Simon and Shor ones. The destination of his algorithms was to calculate conditional probability distributions. Such ideas are useful in applications of Decision Theory and Artificial Intelligence, where inferences are drawn based on uncertain knowledge. The family of algorithms that he projects is based on a structure method that generalizes a Fredkin-Toffoli (FT) construction method used, in the sphere of classical reversible computing. FT showed how, given any binary deterministic circuit, one can construct another binary deterministic circuit which performs the same computations in a reversible way. Tucci shows how, given any classical stochastic network (classical Bayesian net), one can make a quantum network (quantum Bayesian net) which can do the same calculations as the definitive one, but in a (piecewise) reversible manner. Therefore, he holds out the FT construction method so that it can be given to any stochastic circuit, not just binary deterministic ones.

Carlini and Hosoya [2000][9, 10] demonstrated a quantum version of the classical probabilistic algorithms a la Rabin. The Quantum algorithm is based along the essential use of Grover's operator in the quantum search of a database and of Shor's Fourier transform for extracting the

2327-0659/15 $31.00 © 2015 IEEE
DOI 10.1109/ACCT.2015.16

periodicity of a purpose, and their combined use in the counting algorithm originally introduced by Brassard et al. One of the principal characteristics of their quantum probabilistic algorithm is its full integrity and reversibility, which would cause its use possible as part of larger and more complicated networks in quantum information processing systems. As an example of this, they describe polynomial time algorithms for reading some important problems in number theory, such as the examination of the primality of an integer, the so called 'prime number theorem' and Hardy and Little wood's conjecture about the asymptotic number of representations of an even integer as a core of two flowers.

III. PROBLEM FORMULATION & METHODOLOGY

In this report, The Shor’s algorithm suffers a problem of Space Complexity as there is less space in Shor’s Quantum Computing to store the data. In the area of Quantum computing there are demands for new quantum number and Qubits. We provide reduced all these troubles in our report.

In this report, we design an algorithm and also provide simulation for it. We represented the Qubits in a graph. Here we provide new Quantum factoring numbers. The code was compiled on MATLAB v 8.2.0.701 (R2013b) 64-bit (win64) on a Intel® core™ i3-370M processor 2.40 GHz running window® 7 Home Basic (64-bit). A preliminary design of the system is made. A prototype of from this design is created which usually lacks many complex characteristics. By evaluation the prototype and comparing it to the requirements a second prototype is planned. The existing prototype is evaluated in the same way as was the previous prototype, and, if necessary, some other prototype is produced from it. Once the organization has satisfied the requirements it is rectified and the final scheme is designed. The system is evaluated and tried out with maintenance carried out on a continuing basis.

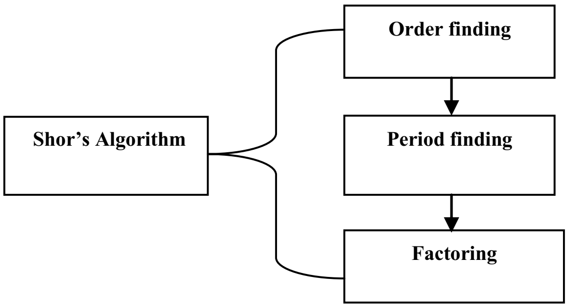
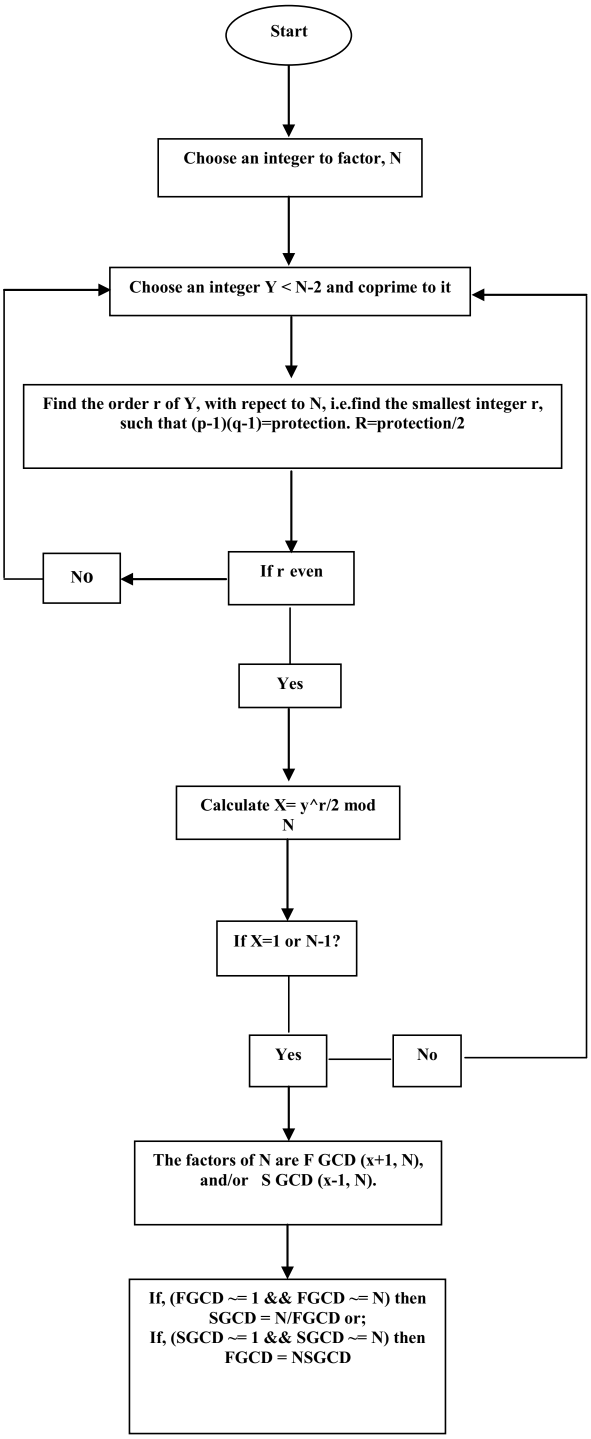


Fig. 1 Shows the Concatenation of three algorithms

Fig. 2 Flow Chart of Algorithm

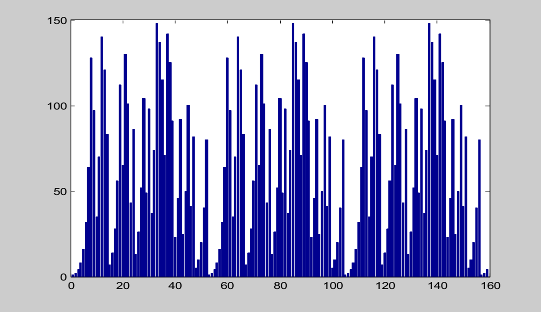

Start
Choose an integer to factor, N
Choose an integer Y < N-2 and coprime to it
Find the order r of Y, with repect to N, i.e.find the smallest integer r, such that (p-1)(q-1)=protection. R=protection/2
No
If r even
Yes
Calculate X= y^r/2 mod N
If X=1 or N-1?
Yes
No
The factors of N are F GCD (x+1, N), and/or S GCD (x-1, N).
If, (FGCD ~= 1 && FGCD ~= N) then SGCD = N/FGCD or;
If, (SGCD ~= 1 && SGCD ~= N) then FGCD = NSGCD

IV. Result

Example 1for a quantum number , N=159

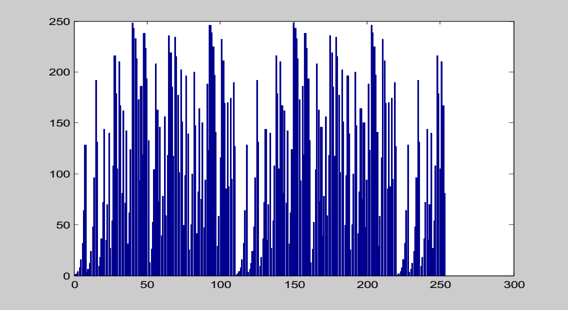
```
Enter Number That U Want To Check: 159
3    53
Product Is: 104

Repeated Term Is: 52
Your Remainder Repetion Is Shown As
a 3
GCD Of Random Number 3
Value 105
First GCD:: 3
Second GCD:: 53
Your Remainder Repetition Is Shown As Show 51
```



Example 2 for quantum number N= 253

```
Enter Number That U Want To Check: 253
```



Example 3 for non quantum number N=215

```
Enter Number That U Want To Check: 215
  43
Product Is: 168
Repeated Term Is: 84
Your Remainder Repetition Is Shown As
a 5
GCD Of Random Number 5
Value 166
First GCD:: 5
Second GCD:: 43
Your Remainder Repetition Is Shown As
Show 27
 "Sorry This is Not a Quantum Number".
```

In this report we discover raw and different quantum numbers shown in the upshot. This algorithm factoring the prime number, but in the case same numbers is prime but they are not a quantum number its show a different resolution. We factor two and three digit quantum number. Following are the numbers shown quantum numbers. 15,21,33,35,39,55,57,69,77,87,95,111,115,141,143,177,187, 01,203,235,237,249,253,295,299,319,335,355,371,395,407, 413,415,437,517,551,581,649,667,737,767,851,869,893,913 ,923.

V. Conclusions

Quantum computing is a big and new searching area of inquiry. Thither are many algorithms and methods such as Shor's algorithm. In this report, we solve many complicated mathematical problems in Shor's algorithm and reduced the problem of order finding or period finding. The outcomes indicate that the quantum computer will be capable to execute any test that a classical computer can. These quantum numbers can be used in quantum computers, the quantum computers are essentially buitup of the same characters. These quantum numbers can be employed for a memory which hold the current information about the organization and a processor which perform operations on the current state of the computers or same kind of input-output part where data can be put into computers.

Acknowledgment

The authors are thankful to Department of Computer science for Providing LAB Facilities.

References

[1] Neil Gershenfeld and Isaac L. Chuang."Quantum Computing with Molecules" article in Scientific American

[2] Finkelstein, David (1969). "Space-Time Structure in High Energy Interactions".

[3] In Gudehus, T.; Kaiser, G. Fundamental Interactions at High Energy. "New qubit control bodes well for future of quantum computing", New York: Gordon & Breach.

[4] Nielsen, Michael A.; Chuang, Isaac L. Quantum Computation and Quantum Information. p. 202.

[5] Nielsen, Michael A.; Chuang, Isaac L. (2010). "Quantum Computation and Quantum Information" Cambridge University Press. p. 13. ISBN 978-1-107-00217-3.

[6] J. Kempe and A. Shalev. "The hidden subgroup problem and permutation group theory". In Proc. 16th ACM-SIAM Symp. on Discrete Algorithms (SODA), pages 1118-1125, 2005.

[7] Obenland, K.M. and Despain, A.M. (1998) "A Parallel Quantum Computer Simulator" Presented at High Performance Computing 1998.

[8] Tucci, R.R. (2000) "Quantum Computer as an Inference Engine"

[9] Lomonaco, S.J. Jr (2000) "Shor's Quantum Factoring Algorithm"

[10] Meyer, David A. (2000) "Quantum games and quantum algorithms" AMS Contemporary Mathematics volume: Quantum Computation and Quantum Information.

[11] Shor, P.W. (1994) "Algorithms for Quantum Computation: Discrete Logarithms and Factoring" 35th Annual Symposium on Foundations of Computer Science, IEEE, pp.124-134.

[12] N.Gershenfeld and Isaac L. Chuang "Quantum Computing with Molecules" article in Scientific American 10th edition 2010.

[13] D. DEUTSCH, Quantum theory, the Church-Turing principle and the universal quantum computer Proceedings of the Royal Society of London A 400, pp. 97-117 (1985)

[14] V. Kumar, “Quantum Computing: Quantum Key Distribution”, IOSR Journal of Computer Engineering (IOSR-JCE) Volume 16, Issue 2, Ver. XII (Mar-Apr. 2014), PP 122-125

[15] R.P. Feynman , “ Simulation Physic With Computers , ” Int. J. Theor . 21,467-88,1982.

[16] R. Jozsa. “Quantum factoring, discrete logarithms and the hidden subgroup problem,” IEEE, 3(2), 2000

[17] Quantiki, Quantum wiki page (2011), List of QC simulators http://www.quantiki.org/wiki/List of QC simulators.

[18] Eqcs (2012), Eqcs-0.0.8 http://home.snafu.de/pbelkner/eqcs/index.html.

[19] LibQuantum (2012), Quantum Error Correction http://www.libquantum.de/api/1.1/Quantum-Error-Correction.html.

[20] M. Nielsen and I. Chuang, “Introductory to the Tenth Anniversary Addition”, Cambridge University, ISBN 978- 1- 107- 00217- 3 Hardback Published in 2010

[21] R. James Bevington “COM3401, Quantum Simulation and the Further Examination of Shor's Algorithm”, Department of Computer ScienceUniversity of Exeter April 2005, p-60.