2012 International Conference on Networks and Information (ICNI 2012)
IPCSIT vol. 57 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V57.11

Log-mod-finding: A New Idea for Implementation of Shor's Algorithm

Mehdi Masoudi [1], Ahmad Pour Monjezy [2]

[1] IAU branch of Izeh, Izeh, Iran (masoudi737@gmail.com)

[2] IAU science and research branch of Tehran Tehran, Iran (a.pourmonjezy@srbiau.ac.ir)

Abstract. One of the most important challenges in the Shor's algorithm is about minimizing the number of elements which is useful to construct an effective quantum computer. The best proposed architecture for implementation of Shor's algorithm is order-finding approach. In this approach for a quantum register such as $|x\rangle$, it requires 2n qubits and 3 control qubit. However, using simple mathematical conversation which has been applied in this paper, in addition to easier implementation, we could decrease the required qubit for register $|x\rangle$ to n qubit. This makes a significant reduction in the number of implementation elements. Our experiments on 1000 RSA numbers shows that the proposed method could find prime factor all of the RSA numbers just with one decimal number. As a result, using this implementation, we can have a more precise method for cracking RSA numbers.

Keywords. Shor's algorithm, quantum computing

1. Introduction

Since Shor's discovered a polynomial time algorithm for factorization on a quantum computer, a lot of effort has been directed towards building a working quantum computer. Despite all these efforts, it is still extremely difficult to control even a few qubits. It is thus of great interest to study exactly how few qubits are needed to factor an n-bit number.

Quantum factorization consists of classical pre-processing, a quantum algorithm for log-mod-finding and classical postprocessing. We will concentrate on the quantum part of factorization and consider classical parts as being free as long as they are computable in polynomial time. The only use of quantum computation in Shor's algorithm is to find the order of $a^{|x\rangle}$ mod N, where N is an n-qubit integer that we want to factor. The order r of $a^{|x\rangle}$ mod N is the least positive integer such that $a^r \equiv 1 \pmod N$.

For completeness, we now give the full algorithm for factoring N as given in [3]:

1. Pick a pseudo-random number a <N.
2. Compute gcd (a, N).this may be done using the Euclidean algorithm.
3. If the greatest common divisor (a, N) ≠ 1, then there is a nontrivial factor of N, so we are done.
4. Otherwise, use the log-mod-finding to find r, the period of the relation (1) function:

$$f(x) = a^{|x\rangle} \bmod N = 2^{|x\rangle \log_2^a} \bmod N$$
$$f(x + r) = f(x) \qquad (1)$$

5. If r is odd go to back step 1.
6. If $a^{r/2} \equiv -1 \pmod N$ go back to step 1.
7. The factor of N is gcd ($a^{r/2} \pm 1$, N). We are done.

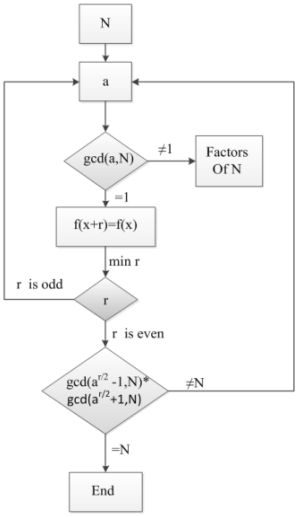

Fig. 1: Shor's algorithm flow chart

2. The Circuit

In log-mod-finding approach instead of calculating the amount of $a^{|x\rangle}$ mod N, $2^{|x\rangle \log_2^a}$ mod N is calculated. In order to understand easily, four stages have shown in Figure 2. In the first stage, quantum register |x⟩ is multiplied by the integer, and then two power of the result be calculated, after this, mod N is calculated and then using the quantum furrier transform, The frequency of repetition was obtained and delivered to the classical process to do the rest algorithm.

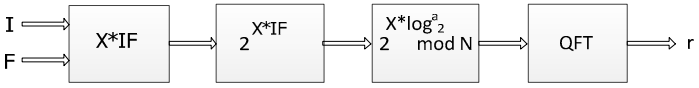

Fig. 2: quantum factorization stages

2.1. The multiplier gate

Considering the expression (2) the multiplier, integer and decimal part of Log_2^a are considered as an integer and multiplied by the quantum register |x⟩ and then the number of decimal qubits for the intended destination has to be removed low significant Results qubits. Whatever, the number of qubit which allocated to the decimal number be more, circuit has more precision in response. Also simulation for 1000 RSA number showed this circuit funded RSA factors successfully just with one decimal number.

$$a^{|x\rangle} = 2^{|x\rangle \log_2^a} = 2^{|x\rangle * I.F} \simeq 2^{[x * I.F]} \quad (2)$$

For implementing the multiplier circuit of reversible HGN and PG gate (Figure 3) is used. A multiplier circuit by these gates has been implemented by multiplying the minimum components gates for multiplying operation.

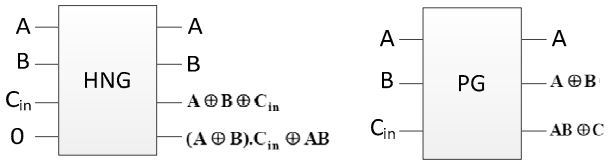

Fig. 3: Reversible HNG and PG gate

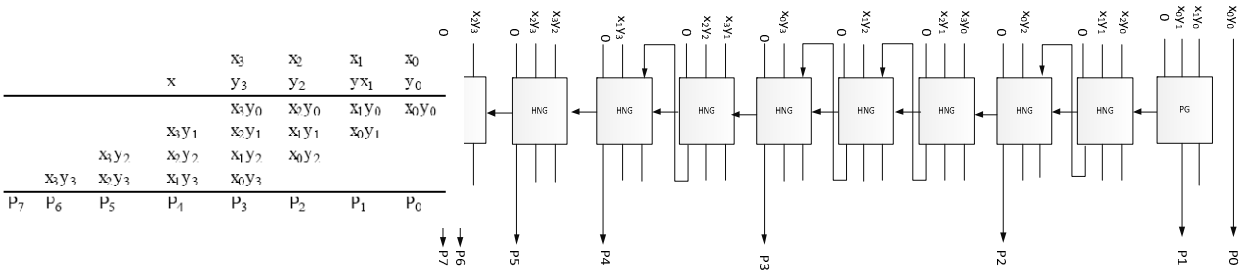

Fig. 4: Partial products in a 4*4 multiplication

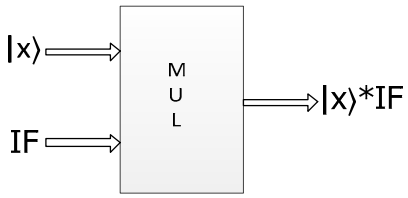

Fig. 5: multiplier gate

The Figure 3 is a four qubits multiplier. These components as MUL (Figure 5) and introduce it used to multiplied by the quantum register |x⟩ and the IF (integer and decimal part of $\mathrm{Log}_2^{\mathrm{a}}$), the number of qubits which has been consider in for decimal, it is necessary to remove the given low significant result of multiplying.

2.2. The decoder gate

After multiplying |x⟩*IF, the two power the result must be calculated. We know that when the two power a number such as z, (z +1)th result qubit will be one. This behaviour is similar to the output of a decoder; using a quantum decoder these components can easily be implemented. Figure 6 shows an example of decoder 2 to 4 which has been implemented by CNOT and Toffoli gate.

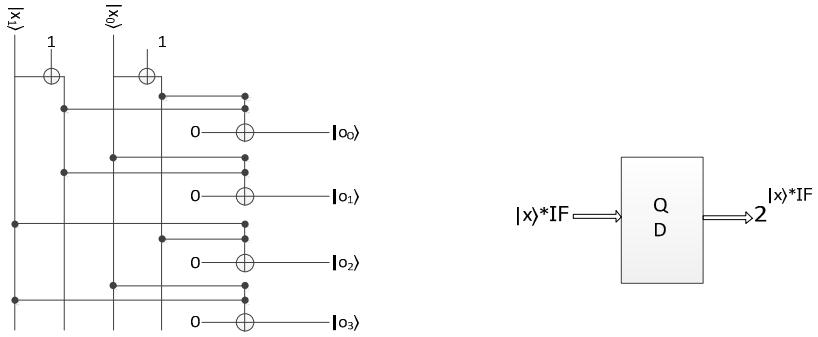

Fig. 6: partial products in a 2 to 4 quantum decoder

2.3. The mod gate

Now, a circuit is needed to calculate mod N after result of decoder. In this circuit conditional phase shifter has been used, this gate if the control and target qubit equal 1 phase change equals to $e^{\frac{2\pi i}{2^k}}$ (k is equivalent to phase shift amount which has shown on conditional phase shifter) occurred on target qubit.

If the controlled gate apply for quantum number |a⟩ and Fourier |b⟩ the result will be equal to Fourier | a⟩ + | b⟩ of the two numbers, and if this action reverse on number |a⟩ and Fourier |b⟩ the result will be Fourier | a⟩ - | b⟩. so these component can be used for next operation according to the bars located at the left and right of these, any one can distinguish a component from other.

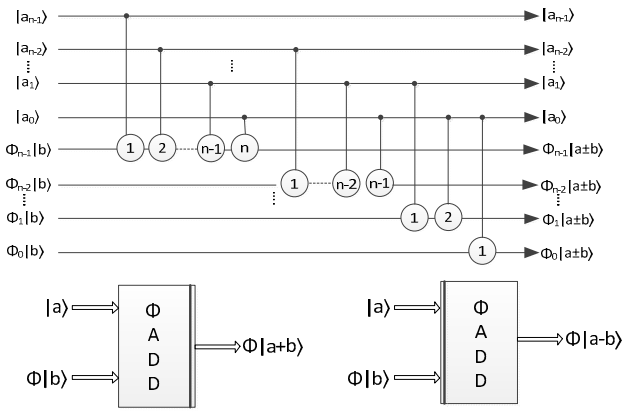

Fig.7: adder and subtracter gate

To obtain the Mod components ΦMOD (Figure 8) has been introduce, this component can subtract the first number from the second, due to controlling of valuable qubits if the result will be negative, the subtraction number will be added the first number again. Using this component to calculate the mod amount of first stage amount of $2^{n} * N$ will be subtracted from $2^{|x\rangle * \log_2^a}$ and again if the result will be negative in sign the same amount would be added to it. This action will be done till the amount of $2^{0} * N$ is acquired, then at the last stage the output would be $2^{|x\rangle * \log_2^a}$ mod N.

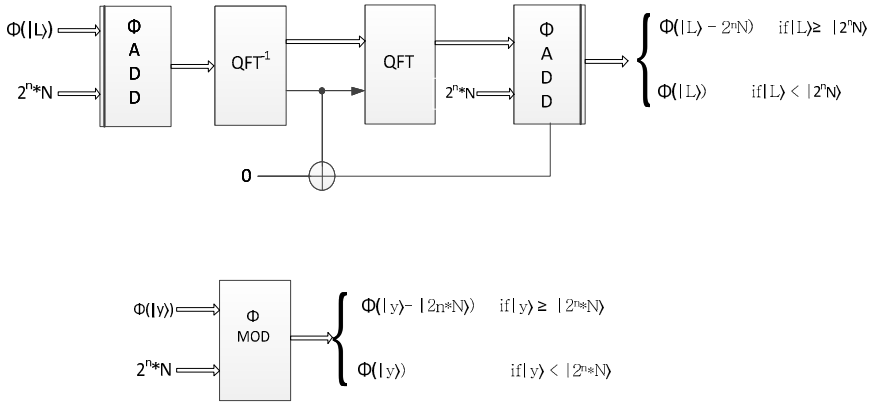

Fig. 8: Implementation ΦMOD gate

Using this component, as is shown in Figure 9 together, will be placed besides the obtained result from the mod series will be taking the quantum Fourier transform, r come from it.

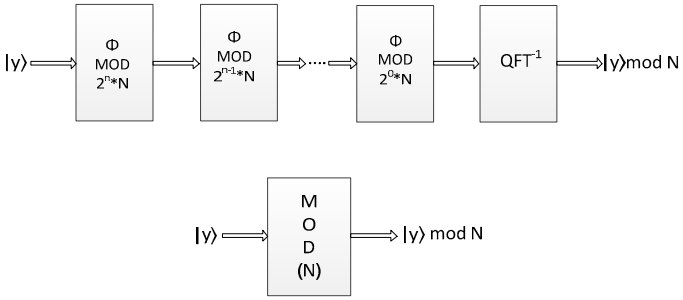

Fig. 9: mod gate

3. The log-mod-finding circuit

According to the previous components the quantum part can be implement as figure 10, this circuit has there input, the first of them is quantum register of |x⟩ which intern is zero modes that has been transfer from Hadamard gates till a mode of superposition include all of numbers can be seen. The second input is equal to the amount of I, which intern is equal to integer part of Log_2^a. This input is obtained by classical process and

delivered to quantum part. The third input is F which in turn is equal to integer part of Log_2^a , if the circuit output subjected to quantum Fourier transform the result will be equalled to $2^{|x\rangle * \log_2^a}$ mod N period or r, this result is a classical number. Then classical part of this number can be received and according to the algorithm $\gcd(a^{\frac{r}{2}} \pm 1, N)$ will be calculated. If the obtained numbers is the factors N, the algorithm will be ended otherwise, classical part can continue using new amount of algorithm until the prime number will be found.

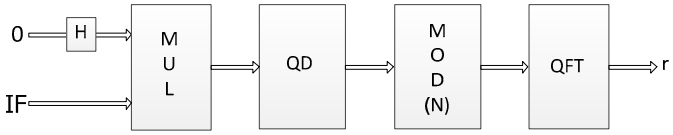

Fig. 10: log- mod- finding circuit

4. Conclusion

In this paper, we proposed utilizing $2^{|x\rangle * \log_2^a}$ expression instead of $a^{|x\rangle}$ in the fourth step of Shor's algorithm. This simple change could reduce the number of qubits into n for a determined register x. Also it could outperform existing methods in terms of number of gates significantly. The simulations assert that the proposed circuit cracked 1000 of RSA numbers successfully just with one decimal number.

5. References

[1] S. Beauregard, "Circuit for Shor's algorithm using 2n+3 qubits, Quantum Information and Computation," quant-ph/0205095v3, 2003 pp 175–185.

[2] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM J. Comp., 26, 1997 pp. 1484-1509.

[3] P. Barrera, A. Calabro,L. Fortuna, D. Porto, "A new method for implemention gate operation a quantum factoring algorithm, " Circuits and Systems, ISCAS. 2003, pp 777- 780.

[4] M. Haghparast, S. j. Jassbi, K. Navi, Design of a Nover reversible multiplier circuit using HNG gate in nanotechnology, World Appl. Sci. j, 3(6), 2008, pp 974-978.

[5] R. zhou, Y. Shi,J. Cao, H. Wang, "Comment on Design of a Novel reversible multiplier circuit usin HNG Gate in nanotechnology, " World Appl. Sci. j,10(2) , 2010,pp 161-165.

[6] G. Draper, "Addition on a quantum computer, " quant-ph/0008033v1, 2000.

[7] M. Nakhara, T. ohmi, "quantum computing from linear algebra to physical realizations, " Taylor & Francis book,ISBN-13:978-0-7503-0983-7,2008.

[8] X. Fu, W. Bao, C. Zhou, "Design and Implementation of Quantum Factorization Algorithm, " Proceding 3th of the IEEE conference Intelligent Information Technology and Security Informatics(IITSI), April 2010, pp. 748 – 752.

[9] R. Cleve and J. Watrous (2000), Fast parallel circuits for the quantum Fourier transform, Proceedings 41st Annual Symposium on Foundations of Computer Science (FOCS'00), pp. 526-536.

[10] M. Mosca and A. Ekert (1999), the hidden subgroup problem and eigenvalue estimation on a quantum computer, Lecture Notes in Computer Science, 1509, pp. 174-188.

[11] S. Parker and M.B. Plenio (2000), efficient factorization with a single pure qubit and logN mixed qubits, Phys. Rev. Lett., 85, pp. 3049-3052.

[12] A. Barenco, C. Bennett, R. Cleve, D.P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J.A. Smolin, and H. Weifurter (1995), Elementary gates for quantum computation, Phys. Rev. A, 52, pp. 3457-3467.

[13] D. Beckman, A.N. Chari, S. Devabhaktuni, and J. Preskill (1996), efficient networks for quantum factoring, Phys. Rev. A, 54, pp. 1034- 1063.

[14] L.M.K. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, M.H. Sherwood, and I.L. Chuang (2001), Experimental realization of Shor's quantum factoring algorithm using magnetic resonance, Nature, 414, pp. 883-887.