Three-Qubit Implementation of Quantum Fourier Transform for Shor's Algorithm

Deepanshu Trivedi, Ankur Saharia, Kamalkishor Choure, Manish Tiwari, Ravi Kumar Maddila, and Ghanshyam Singh

Abstract Quantum computers are capable of very fast computation as compared to the classical counterpart. Problems impossible for the classical computer are efficiently solved on a quantum computer. Shor's factoring algorithm (SFA) calculates the prime factors of a given number exponentially quicker than the available classical algorithm. The paper deals with a vivid explanation of the methodology and various other future possibilities related to the development of the SFA. The paper also emphasizes the three-qubit realization of the quantum Fourier transform on the IBM Q experience.

Keywords Shor's factoring algorithm · Quantum computing · Quantum Fourier transform

1 Introduction

In the early 1980s, quantum prototype for the turing machine was proposed by physicist Paul Benioff that lead to the beginning of quantum computing [1]. Later, Richard Feymann and Yuri Manin suggested that quantum computers can outperform classical computers [2]. Then, Shor came up with a quantum algorithm for factoring integer with the capacity to decrypt all possible secured systems [3]. The classical computer works on the classical bits- 0 and 1, while the quantum computer makes use of qubits—$|0\rangle$ and $|1\rangle$. Various objects used as a qubit (electrons, protons, and nucleus). Researchers are using outermost electrons in phosphorus as qubits.

Quantum computing makes use of two basic phenomena of quantum mechanics—quantum superposition and quantum entanglement [4].

D. Trivedi (✉) · A. Saharia · K. Choure · R. K. Maddila · G. Singh
Department of Electronics and Communication Engineering, MNIT Jaipur, Jaipur, India
e-mail: 2018pwc5108@mnit.ac.in

M. Tiwari
Department of Electronics and Communication Engineering, Manipal University Jaipur, Jaipur, India

M. Tiwari et al. (eds.), *Optical and Wireless Technologies*, Lecture Notes in Electrical Engineering 771, https://doi.org/10.1007/978-981-16-2818-4_12

Quantum superposition refers to the uncertainty of the particle to be in several states at once. For example, an electron can be either in ground state or in the excited state. By the principle of superposition, the electron is in the state which is a linear combination of both the states $b_0|0\rangle + b_1|1\rangle$ where b_0 and b_1 are the coefficients can be complex numbers which are adding to 1.

Quantum entanglement is interpreted as the exchange of quantum information between two particles at a distance. It means that when the particles are separated, the quantum states of each particle is dependent of the state of other particle and cannot be defined independently.

SFA is one of the well-known application of the quantum computers. It takes $O\big((\log N)^3\big)$ time to factor a number faster than its classical equivalent. The error introduced due to the use of physical qubits and large number of gates, and the algorithm is still far away from the real-time implementation.

The most difficult case of factorization is when a number is the product of two odd primes which are equal in length. This is the outline of RSA cryptosystem, which uses a public key N, the product two large odd primes. RSA cryptography is based on the fact that it is difficult to factor a very large number. In order to crack the RSA cryptosystem, Shor proposed a quantum factoring algorithm which is polynomial in time.

2 Basic Concepts of Shor's Algorithm

Kitaev replaced a fully coherent QFT by the semi-classical quantum Fourier transform (sc-QFT) in the Shor's algorithm. In sc-QFT, each time one of the qubits of the period register is measured [5]. The measurement on the second qubit is determined by the result of measurement on the first qubit. So the $2\log_2 N$ qubits required for the period register can be replaced by a single qubit. Hence, the number of qubits is now reduced to execute the Shor's algorithm. For example, for $N = 15, 21, 35$, the number of qubits required is $n = 5, 6, 7$. The scalable algorithm has been realized with an ion-trap quantum computer that provides success probabilities above 90%.

In 2017, WANG Yahui et al. proposed a quantum algorithm capable of breaking the public key cryptosystem like RSA [6]. It has some essential outlines like—(1) without factoring a number the plaintext can be recovered from the ciphertext, (2) even order of the elements to be avoided, (3) with better probability of success than Shor's algorithm, (4) equal complexity compared to Shor's algorithm.

The algorithm proposed by Peter Shor works iff (if and only if) the period is even. If the period is odd, the factors cannot be found, and the algorithm is relaunched using distinct a values for the function $a^x \mod N = 1$. For the square coprimes, the factors can be found using odd orders [7]. This somehow increases the possibility of success by considering odd orders. The rate of success of the algorithm can be improved by avoiding square coprimes rather than to consider the odd orders. Earlier author considered factoring 21 with the coprime four giving order three. In spite of odd order, the factors are successfully calculated by the algorithm, $3 = \gcd\left(4^{\frac{3}{2}} + 1, 21\right)$, and $7 = \gcd\left(4^{\frac{3}{2}} - 1, 21\right)$. As coprime is square, the factors o are integers, but coprimes do not always serve the purpose. For example, factoring 21 with coprime 16. The paper analyzes the role of odd orders in factoring a number, and it should not be ignored directly.

The recent research in SFA, Amico et al. [8] discussed the implementation of the compiled form of SFA for the specific case of $N = 15$, 21, and 35 on the ibmqx5 superconducting chip. Using the sc-QFT, the algorithm is implemented using small number of physical bits as compared to a large qubits required using the coherent quantum Fourier transform. The similarity between the theoretically obtained values (distribution of phase) and experimentally calculated values gives the quantitative measure for which square of statistical overlap is used.

Nene et al. [9] presented the simulation of the algorithm on MATLAB using quantum computing function tool box. Development and commercialization of quantum computer are still far away; so, the paper produced a standardized method for the implementation of SFA on a classic computer. The analysis of the periodicity of the function upto 3-digit of N is presented where the result of simulations are collaborated with the theoretical results.

Vivid description about the methodology and mathematical analysis of the different parts of Shor's algorithm is clearly explained by Loceff [10]. It also deals with the basic concepts of quantum mechanics and explanation of other quantum algorithm like Simon's algorithm, Deutsch's algorithm, and quantum teleportation that leads to clear understanding of the SFA.

3 Methodology

The algorithm composed of two parts:

- Classically processing the problem by changing the factoring problem to period finding problem.
- A quantum algorithm to find the period of the function responsible of quantum speedup.

Steps involved in the process of SFA are [11]:

Step 1 Choose a random integer 'a' such that ($a < N$)
Step 2 Compute the gcd(a, N), the greatest common divisor of N. This can be done using Euclidean algorithm.
Step 3 If gcd(a, N) $\neq 1$, signifies there is non-trivial factors of N. Then the function $f(x) = a^x \bmod (N)$ is used to find the unknown number 'r' which gives the period.
Step 4 If the period 'r' found to be odd, go back to Step 1;
Step 5 If the period 'r' found to be even, then go to Step 6

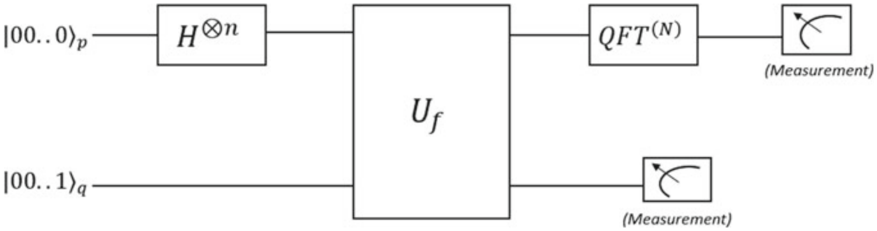

Fig.1 General circuit of Shor's factoring algorithm. Reproduced from [10]

Step 6 $\gcd\left(\frac{a^r}{2}+1, N\right)$ and $\gcd\left(\frac{a^r}{2}-1, N\right)$ are the non-trivial factors of N. The process is now over.

In order to find the factor of a number, we need to find the power x of integer a for which the function $a^x \bmod N = 1$ where a is some random number which is less than N and exponent x is the order or period of a (Fig. 1).

Two quantum registers are required for the implementation of the algorithm. First register, known as the period register or A-register for storing period values. Second register, known as the computational register or B-register, is used to store the result of the modular exponentiation function (MEF) given by $a^x \bmod N$ [10]. Depending on the number N, the size of the registers varies. The qubits in a period register should be in the range $\log_2\left(N^2\right) \leq n_p \leq \log_2\left(2N^2\right)$ while $n_q = \log_2 N$ qubits in the computational register.

Two separable states are prepared $|00..0\rangle_p|00..1\rangle_q$ where the notation p and q signifies the period register and computational register [8]. All possible x values are stored in the period register which gives the approximate value of the period. When $|00..0\rangle_p$ passes through the multi-dimensional Hadamard gate it results in $\frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1}|x\rangle_p$ which is the equal superposition of all qubits, where $Q = 2^{n_p}$.

As soon as the n-qubit passes through the Hadamard gate, the concept of quantum parallelism comes into effect which suggests that if we apply the unitary transformation to all the possible 2^{n_p} inputs, it will produce the superposition of the results of applying f to them in parallel [10].

After the first step, the qubits are now in the tensor product which when passed through the uniform transformation function (U_f) result in $\frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1}|x\rangle_p|a^x \bmod N\rangle_q$. Now, to find the period, QFT is used. As a result of QFT, interference between different possible states occurs, and it produces different superposition states as the output [12]. This interference either makes the signal stronger or weaker depending upon the type of interference depending upon their phase and amplitude.

Now both the registers are measured, but the order of measurement is the trick. How? Let us see….

If we measure the A-register, then the B-register would collapse into its normalized partner, $|f(x)\rangle^n$.

If we measure the B-register first, then the A-register would also disintegrate due to the property of entanglement, but in this case that would be $\frac{N}{r} = m$, not one, pre-images x for every $f(x)$ values.

Now we choose to measure the B-register first, and later on, the A-register is measured. The measurement of A-register after the B-register would result in one of the m values, $x_0 + jr$, but we have no way to extract r (period) from the measurement, so we do not measure A-register yet. Now, apply QFT to the period register or the A-register and measure the qubits later [10].

The measurement obtained in the quantum part is classically processed contributing to the final part of the algorithm. The period value r can be found using the continued fraction algorithm [13] or get direct estimation of the period value by running the algorithm several times [8].

The algorithm proposed by Shor's in 1994 known as the factorization algorithm can be implemented using $n_q = \log_2 N$ qubits in the computational registers that are used for the MEF and $n_p = 2\log_2 N$ qubits in the period registers for QFT. Thus, the entire algorithm would require a total of $3\log_2 N$ qubits which is still a challenge for present quantum computer is N is large [5].

4 Three-Qubit Implementation of QFT

In order to find the period of the function f, the function values are calculated at every interval or points $(x_1, x_{2,} \ldots x_n)$ simultaneously. When measured, it will give one of the possible values and neglect all others by the property of entanglement. QFT differs as it operates on superposition state and produces different superposition state as the output [12]. The component interferes constructively or destructively depending on their amplitude and phase.

The three-qubit implementation of QFT is implemented on IBM Q experience as shown in Fig. 3. If the QFT operates on any basis state alone, the output is the superposition of all the possible states. Change of phase of various states can be seen using QFT. QFT makes use of two gates-Hadamard gate (single qubit) and the controlled rotation gate (two-qubit) as shown in Fig. 2. QFT is an essential part of Shor's algorithm. On the other hand, the MEF can also be implemented using the IBM Q experience platform for the complete implementation of the Shor's factoring algorithm.

5 Conclusion

Due to various properties of quantum mechanics like quantum superposition, entanglement, and parallelism, the various public-key cryptographic systems (RSA, ECC, etc.) are no longer secure and can be breached shortly. The new cryptographic system needs to be introduced for cyberspace security shortly. Since Shor proposed a

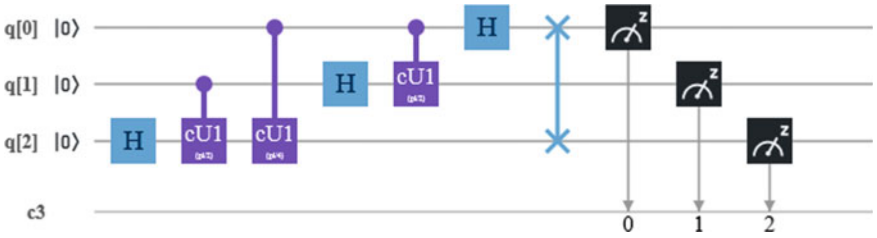

Fig. 2 Three-qubit QFT implementation using Hadamard gate and phase controlled gate. Reproduced from [10]

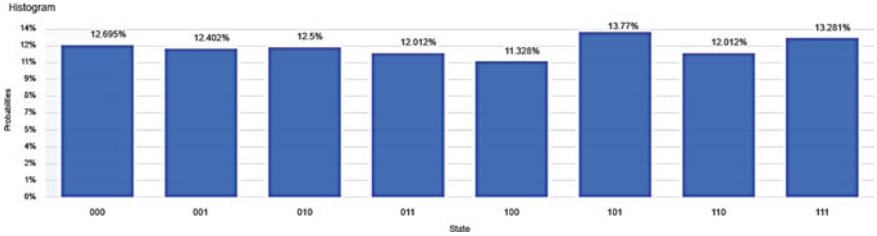

Fig. 3 Superposition of all states of three-qubit input as the result of QFT; *x*-axis-state and *y*-axis-probability

factoring quantum algorithm in 1994, various techniques of implementing different versions of the algorithm have been suggested. Also, various other facts like the consideration of odd periods are taken into use and cannot be denied straight away. New quantum algorithm breaking cryptography without factoring is also studied. Though, reducing number of qubits is still a challenge in the field of quantum.

Acknowledgements We acknowledge the open-source platform of IBM.

References

1. Benioff P (1980) The computer as a physical system: a microscopic quantum mechanical hamiltonian model of computers as represented by Turing machines. J Stat Phys 22(5):563–591
2. Feynman RP (1982) Simulating physics with computers. Int J Theor Phys 21(6/7):467–488
3. Shor PW (1997) Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J Comput 26(5):1484–1509
4. Sharma AK, Ghunawat A (2019) A review on linear optics quantum computing. IEEE Conf Proc
5. Monz T, Nigg D, Martinez EA, Brandl MF, Schindler P, Rines R, Wang SX, Chuang IL, Blatt R (2016) Realization of scalable Shor's algorithm. Science 351(6277):1068–1070
6. Yahui W, Songyuan Y, Huanguo Z (2017) A new computing algorithm for computing RSA ciphertext period. Wuhan Univ J Nat Sci 22(1):068–072
7. Lawson T (2015) Odd orders in Shor's factoring algorithm. Quantum Inf Process 14(3):831–838

8. Amico M, Saleem ZH, Kumph M (2019) Experimental study of Shor's algorithm using IBM Q experience. Phys Rev A 100(1)
9. Nene MJ, Upadhyay G (2016) Shor's algorithm for quantum factoring. Adv Comput Commun Technol:325–331
10. Loceff M (2015) A course in quantum computing 1:610–650
11. Zhang W, Xu C, Li F, Feng J (2007) A period-finding method for Shor's algorithm. Int Conf Comput Intell Secur:778–780
12. Ekert A, Jozsa R (1996) Quantum computing and Shor's factoring algorithm. Rev Mod Phys 68(3):733–753
13. Vazirani U (2004) Shor's factoring algorithm. CS 294-2, Fall 2004. http://www.cs.berkeley.edu/~vazirani/f04quantum/notes/lec9.pdf