ISSN 2278-3091
Volume 10, No.2, March - April 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse971022021.pdf
https://doi.org/10.30534/ijatcse/2021/971022021

Implementation of Shor's Algorithm in QISKIT by Quantum Computing using IBMQ

Poornashree S J[1], Prameela Kumari N[2]
[1]School of ECE, REVA UNIVERSITY, Bangalore,poornashreesj@gmail.com
[2] School of ECE, REVA UNIVERSITY, Bangalore, prameela.n@reva.edu.in

ABSTRACT

Quantum machine learning is the combination of quantum computing and classical machine learning. It helps in solving the problems of one field to another field. Shor's algorithm is used for factoring the integers in polynomial time. Since the best-known classical algorithm requires super polynomial time to factor the product of two primes, the widely used cryptosystem, RSA, relies on factoring being impossible for large enough integers. In this paper we will focus on the quantum part of Shor's algorithm, which actually solves the problem of period finding. In polynomial time factoring problem can be turned into a period finding problem so an efficient period finding algorithm can be used to factor integers efficiently.

Key words: Quantum, IBMQ

I. INTRODUCTION

Shor's algorithm is used for integer factorization and it is a polynomial-time quantum computer algorithm. Informally, it solves the following problem: Given an integer find its prime factors. It was invented in 1994 by the American mathematician Peter Shor.On a quantum computer, to factor an integer N, Shor's algorithm runs in polynomial time (the time taken is polynomial in, the size of the integer given as input).If a quantum computer with a sufficient number of qubits could operate without succumbing to quantum noise and other quantum-decoherence phenomena, then Shor's algorithm could be used to break public-key cryptography schemes, such as the widely used RSA scheme. RSA is based on the assumption that factoring large integers is computationally intractable. As far as is known, this assumption is valid for classical (non- quantum) computers; no classical algorithm is known that can factor integers in polynomial time. Shor's algorithm is efficient on an ideal quantum computer for integers factorization, so it is feasible to defeat RSA by constructing a large quantum computer. It helps in design and construction of quantum computers, and for the study of new quantum-computer algorithms. It has also helps in research on new cryptosystems that are secure from quantum computers, collectively called post- quantum cryptography.

In 2001 Shor's algorithm was explained by a group at IBM, who factored 15 into 3*5, using an NMR implementation of a quantum computer with 7 qubits. Two independent groups were implemented Shor's algorithm using photonic qubits, emphasizing that multi-qubit entanglement was observed when running the Shor's algorithm circuits. In 2012, the factorization of 15 was performed with solid- state qubits. Also, in 2012, the factorization of 21 was achieved, setting the record for the largest integer factored with Shor's algorithm. In 2019, the factorization of 35 has been done with the help of Shor's algorithm on an IBM Q System One, due to cumulating errors algorithms was failed. However, much larger numbers have been factored by quantum computers using other algorithms, specifically quantum annealing.

II. LITERATURE SURVEY

To contribute in any field it is very important to be aware of the works that are currently in progress. In this regard during this mini project, a thorough review of various works carried out in the field of quantum mechanics was carried out. A brief description of the same is presented here.

In "Near term implementation of Shor's Algorithm using Qiskit" [1] the authors Casimer DeCusatis et al. have presented preliminary work on quantum computing, Although the fundamental principles of quantum computing have been known for decades, it is only within the past few years that practical quantum computers have become available. These systems are limited to a small number of qubits, they cannot demonstrate quantum advantage for many practical problems. Near term implementation of Shor's Algorithm using the Qiskiton an IBM Q System One quantum computer was explained in this paper. We present an implementation capable of factoring small two-digit prime numbers, and discuss the limitations of noise when using real quantum computers vs. simulations.

The authors HarashtaTatimmaLarasatiet al. of "Simulation of Modular Exponentiation Circuit for Shor's Algorithm in Qiskit" This paper discusses and demonstrates the construction of a quantum modular exponentiation circuit in the Qiskit simulator for use in Shor's Algorithm for integer factorization problem (IFP), which is deemed to be able to crack RSA cryptosystems when a large-qubit quantum computer exists. We base our implementation on Vedral, Barenco, and Ekert (VBE) proposal of quantum modular exponentiation, one of the firsts to explicitly provide the aforementioned circuit. Furthermore, we present an example simulation of how to construct a 7 x mod 15 circuit in a step-by-step manner, giving clear and detailed information and consideration that currently not provided in the existing literature, and present the whole circuit for use in Shor's Algorithm. Our present simulation shows that the 4-bit VBE quantum modular exponentiation circuit can be constructed, simulated, and measured in Qiskit.

III. BACKGROUND ON SHOR'S ALGORITHM

Consider a number N which is the product of two primes. To determine the prime factors, we first take an initial guess at some number, g, which is either a factor of N or which shares a factor with N. The fact that we can use a guess that shares factors with N derives from Euclid's Theorem [4, 14], a2,000-year-old method from discrete mathematics which allows us to find the greatest common divisor (GCD), and therefore the factors of interest. This makes the problem significantly easier to solve. However, for a reasonably large N it's highly unlikely that our initial guess g will turn out to either be a factor of N or share a factor with N. It can be shown [13] that for any pair of whole numbers A and B which do not share a factor, multiplying A by itself enough times eventually results in an integer multiple of Bplus 1,

$$i.e. Ap = mB + 1$$

For some integers p and m. This means that

$$gp = mN + 1$$

Rearranging terms and factoring yields the following:

$$gp - 1 = mN$$

$$(gp/2 - 1)(gp/2 + 1) = mN$$

The two terms on the left side of are factors of mN. Of course, we are interested only in factors on N, not m. Further, we're interested only in integer factors of N, so this will only work if p is an even number (if p is odd, then p/2 is a fraction, not a whole number). If we encounter either of these conditions when attempting to solve we simply start over with a new value of g. It can be shown that we're 99% likely to find a useful guess within 10 attempts. At this point, solving for p would take exponential time on a conventional computer, but can be done in quadratic time on a quantum computer. To see this, we will state without proof the following theorem for integer values m, m2 and r.

$$\text{If } g\,x = mN + r, \text{ then } g\,x+p = m2\,N + r$$

From this, we can see that p repeats with some period r, or in other words g x, g x+p, g x+2p, g x-p and so on are all separated by some constant value r. We can therefore reduce the problem of factoring N to the problem of finding the period r. This can be achieved using a Quantum Fourier Transform(QFT). When using the QFT, we input a sequence of values and the output is a superposition of all other numbers weighted in a particular manner (the weights correspond to the frequency of the input value). If we input a super position,then the output is also a superposition of all possible states, which add or subtract either constructively or destructively. Quantum physics tells us that if we input a superposition and the resulting output could have come from more than one element in the superposition, then we'll be left with a superposition of just those elements. In our algorithm, if we take an input superposition of all possible exponents (i.e. x,x+p, x-p, x+2p, etc.) then the output contains just those possibilities that would result in the same value of r, spaced apart with a constant period, p (or equivalently with a frequency which is the reciprocal of the period). We now have a quantum superposition of values that repeats with a period p;if we can find the frequency of these repeating values, we can determine p. Expressing this in standard bra-ket notation yields the relationship

$$|x\rangle + |x+p\rangle + |x+2p\rangle + \ldots \xrightarrow{\text{QFT}} |1/p\rangle$$

Figure 1: Bra-ket notation for period finding

Finding p means we can compute (g p/2 +/- 1), which is an improved guess that shares factors with N. The period finding algorithm is shown schematically in figure 2, for various size input registers of qubits initialized to zero. Note that this is aversion of Simon's Algorithm [9] using the Simon oracle Qf .

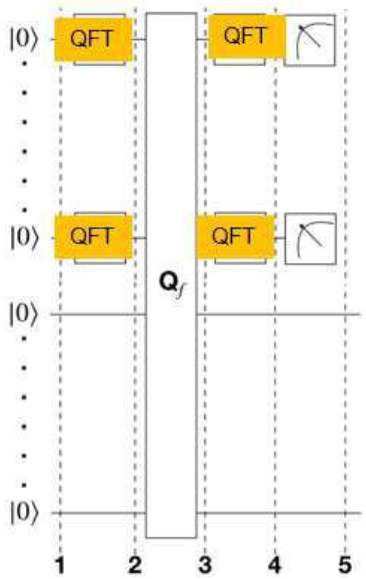

Figure 2: High level circuit for period finding

IV. METHODOLOGY

The problem we are trying to solve is that, given an integer N, we try to find another integer p between 1 and N that divides N.Shor's algorithm consists of two parts:

1. A reduction, which can be done on a classical computer.
2. A quantum algorithm to solve the order-finding problem.

Classical part

1. Pick a pseudo-random number a < N
2. Compute gcd(a, N). This may be done using the Euclidean algorithm.
3. If gcd(a, N) ≠ 1, then there is a nontrivial factor of N, so we are done. Otherwise, use the period-finding subroutine (below) to find r, the period of the following function:
   f(x) = ax mod N, i.e. the smallest integer r for which f(x + r) = f(x).
4. If r is odd, go back to step 1.
5. If a r/2 ≡ -1 (mod N), go back to step 1.

The factors of N are gcd(ar/2 ± 1, N). We are done.

For example: Given $N = 15$, $a = 7$, and $r = 4$, we have $\gcd(7^2 \pm 1, 15) = \gcd(49 \pm 1, 15)$, where $\gcd(48, 15) = 3$ and $\gcd(50, 15) = 5$. For N that is a product of two distinct primes, p and q, the value of $\varphi(N)$ is just $N - p - q + 1$, which for $N = 15$ is 8, and r divides 8

The Problem: Period Finding

Let's look at the periodic function:

$$f(x) = a^x \bmod N$$

where aand N are positive integers, aa is less than N, and they have no common factors. The period, or order (r), is the smallest (non-zero) integer such that:

$$a^r \bmod N = 1$$

The example of periodic function was shown below

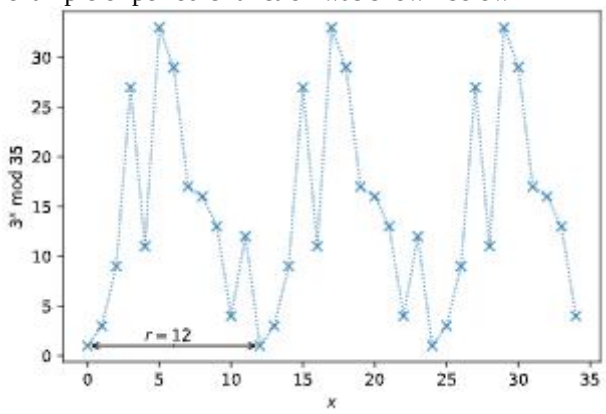

Figure 3: Example of the periodic function

Quantum phase estimation on the unitary operator can be solved by shor's solution

$$U|y\rangle \equiv |ay \bmod N\rangle$$

To see how this is helpful, let's work out what an eigenstate of U might look like. If we started in the state $|1\rangle$, we can see that each successive application of U will multiply the state of our register by a(modN), and after rr applications we will arrive at the state$|1\rangle$ again. For example with a=3 and N=35.

$$\begin{aligned} U|1\rangle &= |3\rangle \\ U^2|1\rangle &= |9\rangle \\ U^3|1\rangle &= |27\rangle \\ &\vdots \\ U^{(r-1)}|1\rangle &= |12\rangle \\ U^r|1\rangle &= |1\rangle \end{aligned}$$

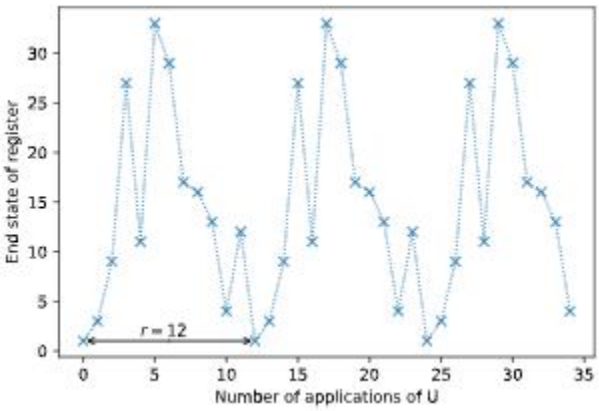

Figure 4: effect of successive application of u

So a superposition of the states in this cycle ($|u0\rangle$) would be an eigenstate of U:

$$|u_0\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |a^k \bmod N\rangle$$

This eigenstate has an eigenvalue of 1, eigenstate can be one in which the phase is different for each of these computational basis states. Similarly the case in which the phase of the kth state is proportional to k:

$$|u_1\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2\pi i k}{r}} |a^k \bmod N\rangle$$

$$U|u_1\rangle = e^{\frac{2\pi i}{r}} |u_1\rangle$$

This is a particularly interesting eigenvalue as it contains r. In fact, rr has to be included to make sure the phase differences between the r computational basis states are equal. This is not the only eigenstate with this behaviour; to generalise this further, we can multiply an integer s, to this phase difference, which will show up in our eigenvalue:

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2\pi i s k}{r}} |a^k \bmod N\rangle$$

$$U|u_s\rangle = e^{\frac{2\pi i s}{r}} |u_s\rangle$$

We now have a unique eigenstate for each integer value of ss where $0 \leq s \leq r - 1$

Very conveniently, if we sum up all these eigenstates, the different phases cancel out all computational basis states except $|1\rangle$:

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle$$

Since the computational basis state $|1\rangle$ is a superposition of these eigenstates, which means if we do QPE on UU using the state $|1\rangle$, we will measure a phase:

$$\phi = \frac{s}{r}$$

Where ss is a random integer between 0 and r−1. We finally use the continued fractions algorithm on ϕ to find r. The circuit diagram looks like this

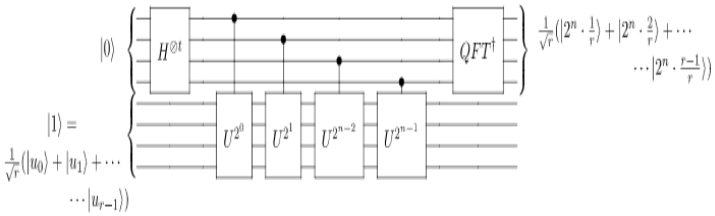

Figure 5: Block Diagram

3. Qiskit Implementation

In this example we will solve the period finding problem for a=7 and N=15. We provide the circuits for U where:

$$U|y\rangle=|ay \bmod 15\rangle$$

without explanation. To create Ux, we will simply repeat the circuit x times.

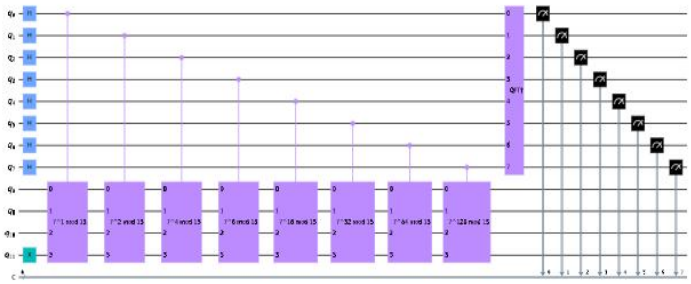

Figure 6: Quiskit Implementation

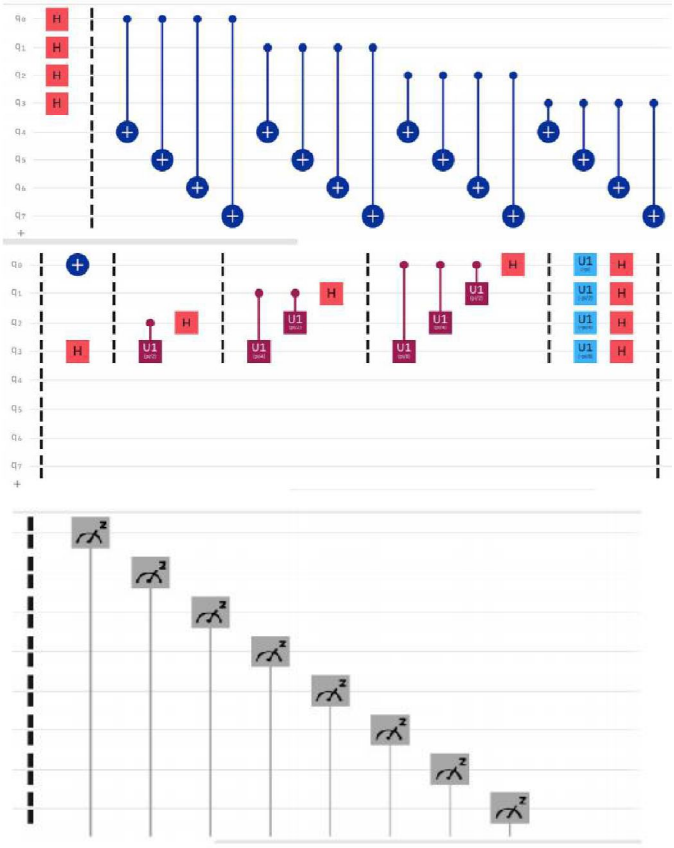

Figure 7: Circuit Composer realization

V. RESULTS

Shor's algorithm is designed and implemented using Qiskit and results are verified with respect to probabilities.

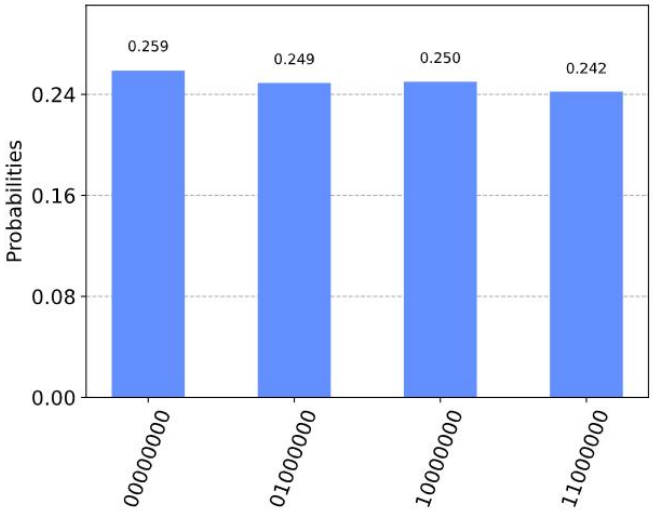

Fig 8: Result of shor's algorithm

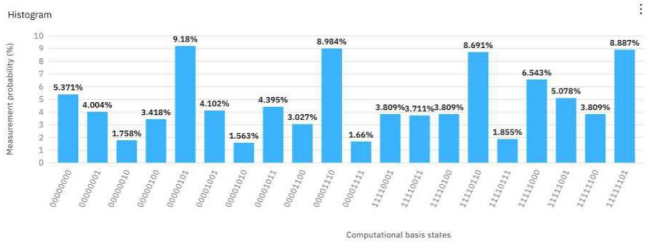

Figure 9: Real quantum computer result of period finding using Shor's Algorithm

VI. CONCLUSIONS

Quantum systems produces typical patterns which simplifies and enhances the computational power that can be advantageous in handling huge data at a faster rate. Hence the whole motivation in this work was in understanding and analysing the circuit design using Quantum mechanics. First a thorough study was carried out to understand the basics of quantum mechanics.Two circuits i.e. half adder and full adder were designed and implemented on IBMQ and thoroughly analysed for the results. It was observed that, in quantum computing results are based on probabilities for all the possible input combination at one go. Due to this there as an enormous reduction in the time to obtain the outputs when compared to classical computation wherein we need to force all the input combinations one by one.

Acknowledgment

The authors are gratefully acknowledge the facilities and support provided by the director of the school of Electronics And Communication Engineering of REVA UNIVERSITY, We also extend thanks to all teaching and non-teaching staff who had helped directly or indirectly to make this project successfully.

REFERENCES

[1]. 1. Shor, P.W. (1994). "Algorithms for quantum computation: discrete logarithms and factoring".Proceedings 35th Annual Symposium on Foundations of Computer Science. IEEEComput. Soc. Press: 124–134. doi:10.1109/sfcs.1994.365700 (https://doi.org/10.1109%2Fsfcs.1994.365 700). ISBN 0818665807.
2. See also pseudo-polynomial time.
3. Beckman, David; Chari, Amalavoyal N.; Devabhaktuni, Srikrishna; Preskill, John (1996). "Efficient Networks for Quantum Factoring" (https://authors.library.caltech.edu/2179/1/BECpra9 6.pdf) (PDF). Physical Review A. 54 (2): 1034–1063. arXiv:quant-ph/9602016 (https://arxiv.org/ abs/quant-ph/9602016). Bibcode:1996PhRvA..54.1034B (https://ui.adsabs.harvard.edu/abs/19 96PhRvA..54.1034B). doi:10.1103/PhysRevA.54.1034 (https://doi.org/10.1103%2FPhysRevA. 54.1034). PMID 9913575 (https://pubmed.ncbi.nlm.nih.gov/9913575).
4. "Number Field Sieve" (http://mathworld.wolfram.com/NumberFieldSieve.html). wolfram.com. Retrieved 23 October 2015.
5. Gidney, Craig. "Shor's Quantum Factoring Algorithm" (https://algassert.com/post/1718). Algorithmic Assertions. Retrieved 28 November 2020.
6. Vandersypen, Lieven M. K.; Steffen, Matthias; Breyta, Gregory; Yannoni, Costantino S.; Sherwood, Mark H. & Chuang, Isaac L. (2001), "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance" (http://cryptome.org/shor-nature.pdf) (PDF), Nature, 414 (6866): 883–887, arXiv:quant-ph/0112176 (https://arxiv.org/abs/quant-ph/01 12176), Bibcode:2001Natur.414..883V (https://ui.adsabs.harvard.edu/abs/2001Natur.414..883 V), CiteSeerX 10.1.1.251.8799 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.251. 8799), doi:10.1038/414883a (https://doi.org/10.1038%2F414883a), PMID 11780055 (https://pu bmed.ncbi.nlm.nih.gov/11780055)
7. Lu, Chao-Yang; Browne, Daniel E.; Yang, Tao & Pan, Jian-Wei (2007), "Demonstration of a Compiled Version of Shor's Quantum Factoring Algorithm Using Photonic Qubits" (http://discov ery.ucl.ac.uk/153160/1/153160.pdf) (PDF), Physical Review Letters, 99 (25): 250504, arXiv:0705.1684 (https://arxiv.org/abs/0705.1684), Bibcode:2007PhRvL..99y0504L (https://ui.ad sabs.harvard.edu/abs/2007PhRvL..99y0504L), doi:10.1103/PhysRevLett.99.250504 (https://do i.org/10.1103%2FPhysRevLett.99.250504), PMID 18233508 (https://pubmed.ncbi.nlm.nih.gov/1 8233508)
8. Lanyon, B. P.; Weinhold, T. J.; Langford, N. K.; Barbieri, M.; James, D. F. V.; Gilchrist, A. & White, A. G. (2007), "Experimental Demonstration of a Compiled Version of Shor's Algorithm with Quantum Entanglement" (http://espace.library.uq.edu.au/view/UQ:134503/UQ134503.pdf) (PDF), Physical Review Letters, 99 (25): 250505, arXiv:0705.1398 (https://arxiv.org/abs/0705.1 398), Bibcode:2007PhRvL..99y0505L (https://ui.adsabs.harvard.edu/abs/2007PhRvL..99y0505 L), doi:10.1103/PhysRevLett.99.250505 (https://doi.org/10.1103%2FPhysRevLett.99.250505), PMID 18233509 (https://pubmed.ncbi.nlm.nih.gov/18233509)
9. Lucero, Erik; Barends, Rami; Chen, Yu; Kelly, Julian; Mariantoni, Matteo; Megrant, Anthony; O'Malley, Peter; Sank, Daniel; Vainsencher, Amit; Wenner, James; White, Ted; Yin, Yi; Cleland, Andrew N.; Martinis, John M. (2012). "Computing prime factors with a Josephson phase qubit quantum processor". Nature Physics. 8 (10): 719. arXiv:1202.5707 (https://arxiv.org/abs/1202.5 707). Bibcode:2012NatPh...8..719L (https://ui.adsabs.harvard.edu/abs/2012NatPh...8..719L). doi:10.1038/nphys2385 (https://doi.org/10.1038%2Fnphys2385).
10. Martín-López, Enrique; Martín-López, Enrique; Laing, Anthony; Lawson, Thomas; Alvarez, Roberto; Zhou, Xiao-Qi; O'Brien, Jeremy L. (12 October 2012). "Experimental realization of Shor's quantum factoring algorithm using qubit recycling". Nature Photonics. 6 (11): 773–776. arXiv:1111.4147 (https://arxiv.org/abs/1111.4147). Bibcode:2012NaPho...6..773M (https://ui.ads abs.harvard.edu/abs/2012NaPho...6..773M). doi:10.1038/nphoton.2012.259 (https://doi.org/10. 1038%2Fnphoton.2012.259).