Factorization using Quantum Computers

June 2020

Contents

Algorithm Used	Method of Implementation	Number factorized
Shor(compiled)	Nuclear Magnetic Resonance(NMR)	15
Shor(compiled)	Photonic qubits using Linear Optics	15
Shor(compiled)	Photonic qubits using photonic chip	15
Shor and Kitaev	Trapped Ion	15
Adiabatic Quantum Computing	Nuclear Magnetic Resonance(NMR)	143
Adiabatic Quantum Computing	Nuclear Magnetic Resonance(NMR)	56,153
Adiabatic Quantum Computing	D-Wave 2000	1028171

1 Introduction

Factorization of numbers is a seemingly abstract mathematical problem which has very important applications. Using classical computation, factorization is a very difficult problem as it takes an exponential amount of time to solve.
RSA is the standard cryptographic algorithm on the internet. It is based on the difficulty of prime factorization as without knowing the factors of the public key, it is impossible to decrypt the message. Currently we use 2048-bit keys, which takes years for a classical computer to crack. However, quantum computers are showing the potential to be able to factorize numbers much faster than their classical counterparts.
In 1994, Peter Shor invented his famous Shor's algorithm, which is a **polynomial-time quantum computer algorithm** for integer factorization. This is an *exponential* speedup!But there is a catch. Although Shor has laid out the theoretical framework to crack RSA encryption, modern quantum computers are still far away from being able to implement the algorithm. This is a very interesting field of research.

In this report, we shall first discuss the problem of prime factorization in detail. We shall look at the classical algorithm and then see how the quantum algorithm speeds up the process.
Experimental implementations of Shor's algorithm will also be discussed. Here we study the NMR, photonic qubit and trapped ion implementation. Finally we also briefly look at another quantum algorithm which has shown great potential to solve the factorization problem - Adiabatic Quantum Computing.

2 Classical Factorization Algorithm

We first look at the classical algorithm which is used to find prime factors of a number. The algorithm is as follows

1. Randomly choose x in the range 1 to N-1. if $gcd(x, N) > 1$ then return gcd(x,N) as a factor.
2. If x and N are coprime, we compute the series, and find the period, r of the series.
$$f(x) = x^a mod N \; where \, a \in \mathbb{N} \tag{1}$$
example: for $x = 2$ and $N = 21$

$$\begin{aligned} 2^0 mod 21 &= 1 \\ 2^1 mod 21 &= 2 \\ 2^2 mod 21 &= 4 \\ 2^3 mod 21 &= 8 \\ 2^4 mod 21 &= 16 \\ 2^5 mod 21 &= 11 \\ 2^6 mod 21 &= 1 \end{aligned}$$

Therefore, the period, $r = 6$. The period is also called the order of x and therefore the procedure of finding r is called **order finding**. The classical step of reducing prime factorization to order finding is necessary before applying a quantum algorithm.

3. If r is even and $x^{r/2} \neq -1(mod\, N)$ then test $gcd(x^{r/2} + 1, N)$, and test to see if one of these is a factor. If not, the algorithm fails.

3 Shor's Algorithm

In the classical algorithm discussed above, the most expensive step is order finding. This is the step which is sped up using Quantum algorithms. In 1994 Peter Shor invented an algorithm called **Shor's algorithm**, which can solve the problem of finding the factors of a given number N in *polynomial time*. Infact, the computational cost of the algorithm is

$$O(n^3 log(n))$$

With number of gates:

$$O(n^2 lognlog(logn))$$

This algorithm requires two other concepts to work. These are quantum fourier transform and phase estimation. Here they will discussed seperately and then used together to find the period of

$$f(x) = x^a mod N$$

3.1 Quantum Fourier Transform

To solve certain problems, it is useful to *transform* the problem into a different problem which is easier to solve. One popular transform which is used in mathematics, physics, computer science, engineering etc. is the discrete Fourier transform.
Given a vector of N complex numbers $x_0, x_1, x_2...X_{N-1}$ the Fourier transform is defined by -

$$y_k \equiv \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi ijk/N} \tag{2}$$

The quantum Fourier Transform performs the transformation on the orthonormal basis $|0\rangle ... |N-1\rangle$ in the following manner

$$|j\rangle \equiv \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle \tag{3}$$

It can be shown that this transformation is unitary, and therefore can be implemented using gates on a quantum computer.

Now when we implement the Fourier transform on a number, it becomes useful to represent the number $|j\rangle$ in base 2(binary), i.e., $j \equiv j_1 j_2 ... j_n . j_l j_{l+1} ... j_m$ which can be expanded as

$$j_1 2^{n-1} + ... + j_n 2^0 + j_l/2 + j_{l+1}/4 + ... + j_m/2^{m-l+1}$$

It can be shown that when the fourier transform is applied to the $|j\rangle$ it can be written as

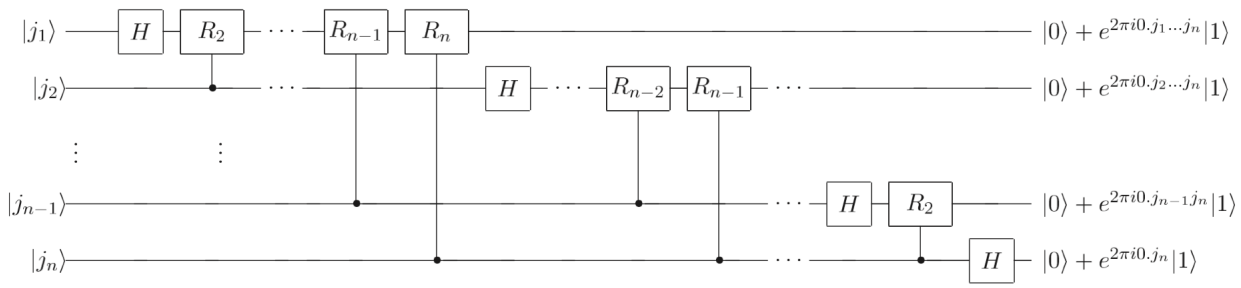
$$|j_1 j_2 ... j_n\rangle \rightarrow \frac{(|0\rangle + e^{2\pi i 0.j_n} |1\rangle)(|0\rangle + e^{2\pi i 0.j_{n-1} j_n} |1\rangle)...(|0\rangle + e^{2\pi i 0.j_1 j_2 ... j_n} |1\rangle)}{2^{n/2}} \tag{4}$$

This method will prove more useful to us for factorising a given number as we will write the number in binary.

To implement the quantum Fourier transform, we use the following gate -

$$R = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix} \tag{5}$$

The final circuit is as follows

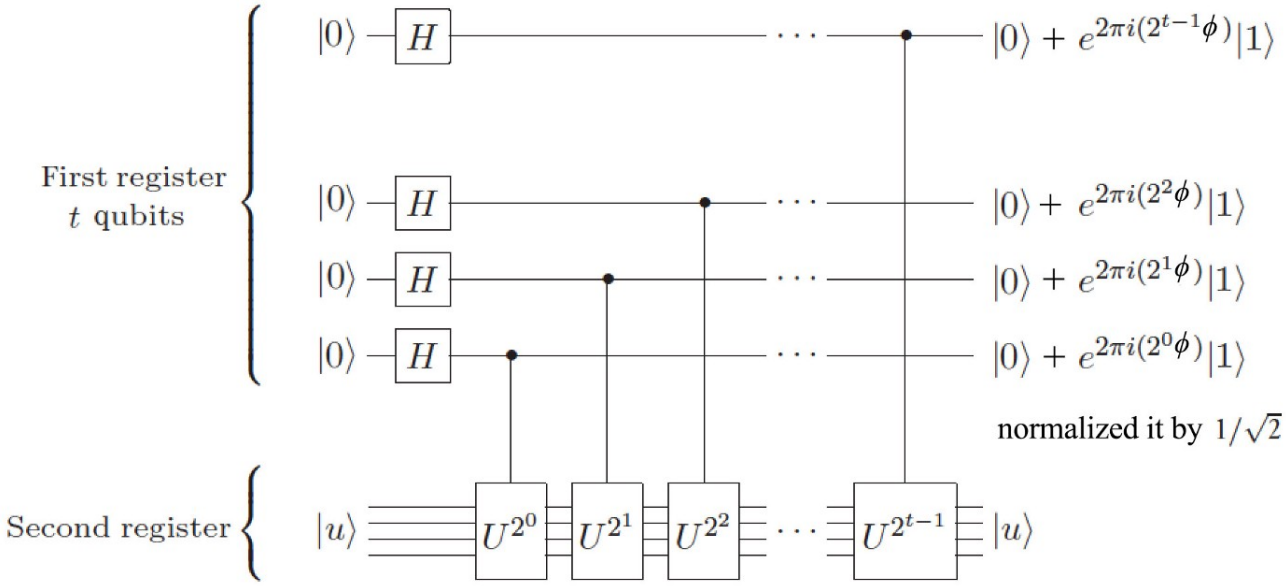


3.2 Phase Estimation

Phase estimation is the process of finding the eigenvalue of a unitary operator. Through phase estimation, we aim to find ψ, given U and $|\psi\rangle$, in the equation

$$U|\psi\rangle = e^{2\pi i\phi}|\psi\rangle \tag{6}$$

(Note that a we have expressed the eigenvalue as $e^{2\pi i\phi}$ because the eigenvalue of a unitary operator is unimodular.)
The circuit which is used for this purpose is



So I can write the output of this circuit as

$$\frac{(|0\rangle + e^{2\pi i(2^{t-1}\phi)}|1\rangle)(|0\rangle + e^{2\pi i(2^{t-2}\phi)}|1\rangle)...(|0\rangle + e^{2\pi i(2^0\phi)}|1\rangle)}{2^{n/2}} \tag{7}$$

But by writing ϕ in binary we can write (7) in a more familiar way.

$$\begin{aligned} \phi &= 0.\phi_1\phi_2...\phi_t \\ 2\phi &= \phi_1.\phi_2\phi_3...\phi_t \\ e^{2\pi i\phi_1.\phi_2...\phi_t} &= e^{2\pi i 0.\phi_2...\phi_t} \end{aligned}$$

In a similar manner, when we multiply ϕ by any power of 2, we shift the binary number $0.\phi_1\phi_2...\phi_t$ to the left, and remove the integer part. When we use this in equation(7), we get the following form -

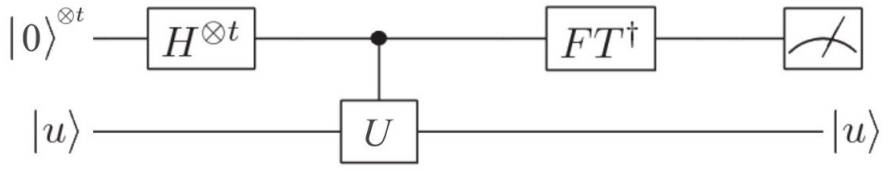
$$\frac{(|0\rangle + e^{2\pi i 0.\phi_t}|1\rangle)(|0\rangle + e^{2\pi i 0.\phi_{t-1}\phi_t}|1\rangle)...(|0\rangle + e^{2\pi i 0.\phi_1...\phi_t}|1\rangle)}{2^{t/2}} \tag{8}$$

Compare this with equation(4), we see that if we apply quantum Fourier transform $|\phi_1...\phi_t\rangle$ we get equation(8). Therefore since we are trying to find ϕ we can apply the inverse Fourier transform to(8)

$$\frac{(|0\rangle + e^{2\pi i 0.\phi_t}|1\rangle)(|0\rangle + e^{2\pi i 0.\phi_{t-1}\phi_t}|1\rangle)...(|0\rangle + e^{2\pi i 0.\phi_1...\phi_t}|1\rangle)}{2^{t/2}} \rightarrow |\phi_1...\phi_t\rangle \tag{9}$$

$$\frac{1}{2^{t/2}}\sum_{j=0}^{2^t-1} e^{2\pi i\phi j}|j\rangle|u\rangle \rightarrow |\tilde{\phi}\rangle|u\rangle \tag{10}$$

The simplified circuit is as follows



3.3 Using Period finding for prime factorization

When we use Shor's algorithm to factorise a number, we first use classical means to convert the problem into an equivalent period finding algorithm i.e., finding the period of

$$f(x) = x^a mod N$$

From here, we use a quantum algorithms to solve this problem. First, we must convert period finding into an equivalent phase estimation problem. For this purpose, we use the unitary operator

$$U|y\rangle \equiv |xy(modN)\rangle \tag{11}$$

The eigenvalue and eigenvector for U are

$$|u_s\rangle \equiv \frac{1}{r}\sum_{k=0}^{r-1} exp[\frac{-2\pi isk}{r}]|x^k modN\rangle \tag{12}$$

$$\begin{aligned} U|u_s\rangle &= \frac{1}{r}\sum_{k=0}^{r-1} exp[\frac{-2\pi isk}{r}]|x^{k+1}modN\rangle \\ &= exp[\frac{2\pi is}{r}]|u_s\rangle \end{aligned} \tag{13}$$

Therefore using phase estimation, we can find the value of s/r which we can use to find r.

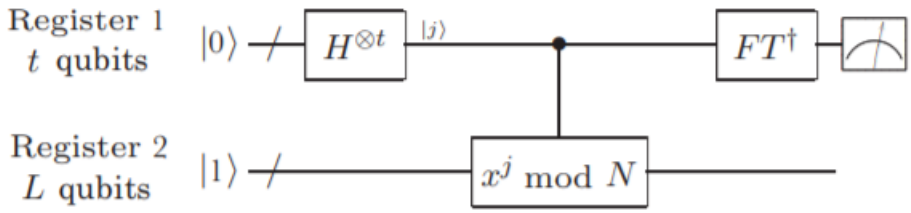
There are still two challenges in applying phase estimation to this problem

1. Remember that to implement phase estimation, we need to know both U and $|u_s\rangle$. While, we know U we still need to find the eigenvectors. One solution is to find a superposition of $|u_s\rangle$

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle = |1\rangle \tag{14}$$

2. We also want to be able to implement a controlled-U^{2^j} operation in an efficient manner. This can be done using a procedure called **modular exponentiation**

So the final circuit for factoring is



The procedure can be summarised in the following manner

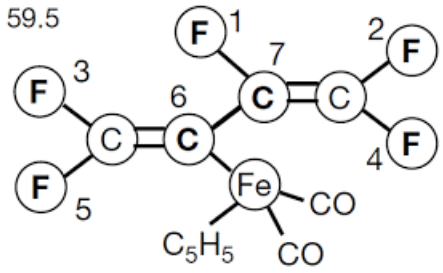
Procedure:

1.	$\|0\rangle\|1\rangle$	initial state
2.	$\rightarrow \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}\|j\rangle\|1\rangle$	create superposition
3.	$\rightarrow \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}\|j\rangle\|x^j \bmod N\rangle$	apply $U_{x,N}$
	$\approx \frac{1}{\sqrt{r2^t}}\sum_{s=0}^{r-1}\sum_{j=0}^{2^t-1} e^{2\pi i s j/r}\|j\rangle\|u_s\rangle$	
4.	$\rightarrow \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\|\widetilde{s/r}\rangle\|u_s\rangle$	apply inverse Fourier transform to first register
5.	$\rightarrow \widetilde{s/r}$	measure first register
6.	$\rightarrow r$	apply continued fractions algorithm

4 Experimental Quantum Factorization

4.1 Nuclear Magnetic Resonance(NMR)

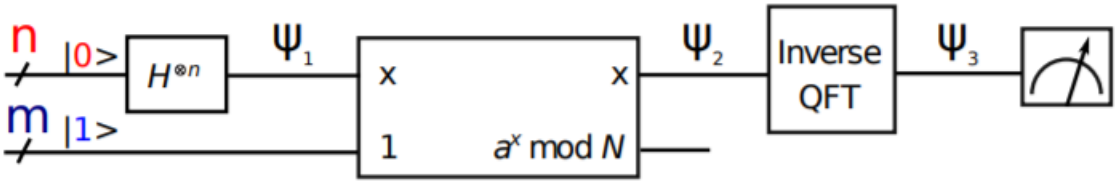
One system which can be used as a quantum computer is a molecule, where qubits are realised in terms of the nuclear spin of the atoms. However, we also need to be able to manipulate these qubits, initialise them and perform operations on them. For this purpose we use NMR(Nuclear Magnetic Resonance).
In the paper [**2**], L. M. K. Vandersypen et al. use the Perfluorobutadienyl molecule shown below. Note that ^{19}F and ^{13}C are spin half nuclei. Therefore this molecule has 7 nuclear spins which means 7 qubits.



The characteristics of NMR based quantum computer are -

1. The coherence time is long(order of seconds)
2. Due to the asymmetry of the molecule, every atom has a different resonant frequency. This property is used to control qubits individually. Using a radio-frequency (RF) pulse at the resonant frequency of the precession frequency, we can rotate the spin of an atom. Therefore it is possible to perform an operation like transforming the state of a qubit form $|0\rangle$ to $|1\rangle$.
3. There is pairwise J(spin) coupling between atoms. Allowing qubits to evolve under this coupling, we can make control operations.

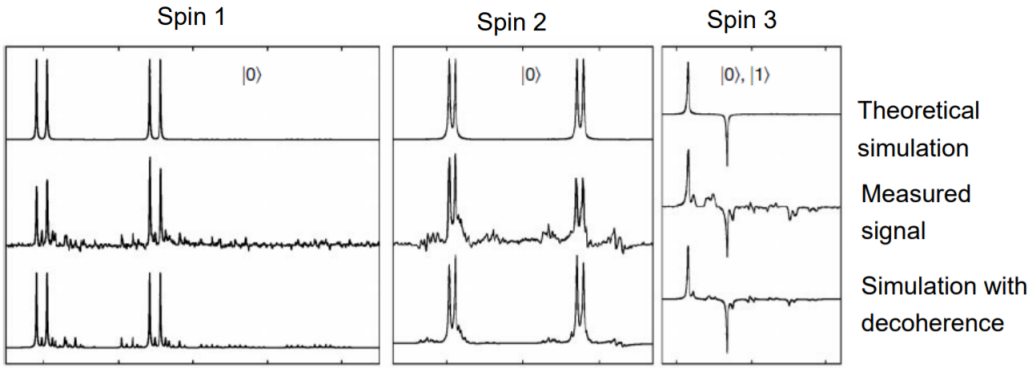
We now use NMR to factor N = 15. Following the algorithm in Section 1, we first make a guess for x where $1 < x < 14$. In [**2**] use x = 11 and x = 7. We use the following circuit to study the circuit



Case: x = 11 In this case,

$$\psi_1 \propto |0\rangle + |1\rangle + |2\rangle + |3\rangle + |4\rangle + |5\rangle + |6\rangle + |7\rangle$$
$$\psi_2 \propto \{|0\rangle + |2\rangle + |4\rangle + |6\rangle\}|1\rangle + \{|1\rangle + |3\rangle + |5\rangle + |7\rangle\}|11\rangle$$
$$\psi_3 \propto \{|0\rangle + |4\rangle\}|1\rangle + \{|0\rangle - |4\rangle\}|11\rangle$$

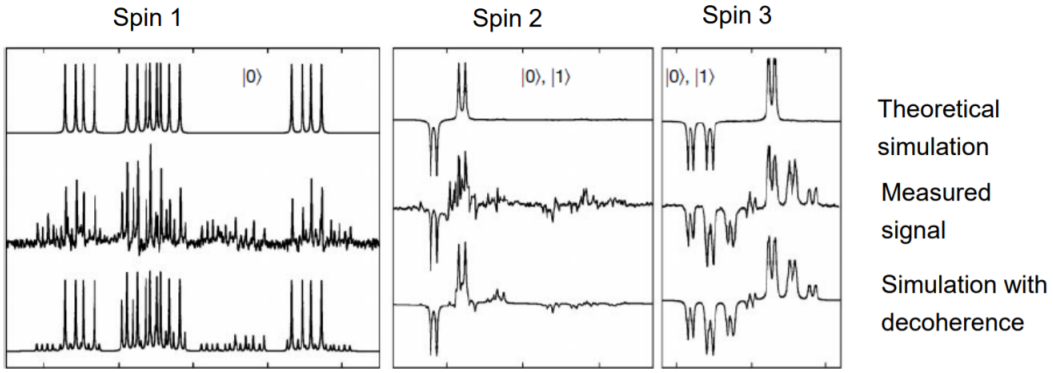
Therefore we desire a superposition of $|000\rangle$ and $|100\rangle$. This is the also what we get as out output. Therefore the period obtained is 4. $r = 2^3/4$ therefore factors are $gcd(11^{2/2} \pm 1, 15) = 3\&5$



Case: x = 7 In this case,

$$\psi_2 \propto \{|0\rangle + |4\rangle\}|1\rangle + \{|1\rangle + |5\rangle\}|7\rangle + \{|2\rangle + |6\rangle\}|4\rangle + \{|3\rangle + |7\rangle\}|13\rangle$$
$$\psi_3 \propto \{|0\rangle + |2\rangle + |4\rangle + |6\rangle\}|1\rangle + \{|0\rangle - i|2\rangle - |4\rangle + i|6\rangle\}|7\rangle$$
$$+ \{|0\rangle - |2\rangle + |4\rangle - |6\rangle\}|4\rangle + \{|0\rangle + i|2\rangle - |4\rangle - i|6\rangle\}|13\rangle$$

Therefore, the desired superposition is $|000\rangle$, $|010\rangle$, $|100\rangle$ and $|110\rangle$. Therefore the factors are $gcd(11^{4/2} \pm 1, 15) = 3\&5$



Therefore the number N=15 is successfully factorised using this scheme. Some things which must be noted about this approach -

1. This is the first instance of Shor's algorithm being implemented experimentally(2001).
2. This is a *compiled algorithm*, which means that the circuits had been optimized based on the input (the number to be factorised). It is not a general circuit which returns factors for any given input.
3. NMR based qubits are difficult to scale to make quantum computers with larger number of qubits.
4. One concern with the NMR implementation is that, there is that entanglement is not observed.

4.2 Photonic Qubits

Using photons to implement quantum algorithms is a popular approach. This approach offers (i)long decoherence times and (ii)precise single qubit operations. A major advantage photons have over the NMR technique is that genuine multiparticle entanglement and multipath interference is observed. Here, we shall discuss two instances of using photons to factorize numbers using Shor's algorithm. A third instance [**5**] is very similar to [**3**] and hence will not be discussed.

4.2.1 Using Linear Optics

In the paper [**3**] Chao-Yang et al., factor N = 15. Following the algorithm discussed in secion 1 and section 2, they take x = 11. Here again, they used a compiled version of Shor's algorithm, which means that the circuit is simplified with prior knowledge of the answer.
The experimental setup involves a photon source and linear optical elements like polarizing beam splitters(PBS) and half wave plates(HWP). The $|0\rangle$ and $|1\rangle$ states are encoded in the photon's horizonatal(H) and vertical(V) polarization. The Hadamard gate can be implemented using a PBS and the CNOT gate is implemented using a combination of PBS and HWP. The simplified circuit for N=15 and x = 11, along with the linear optics network is given below.

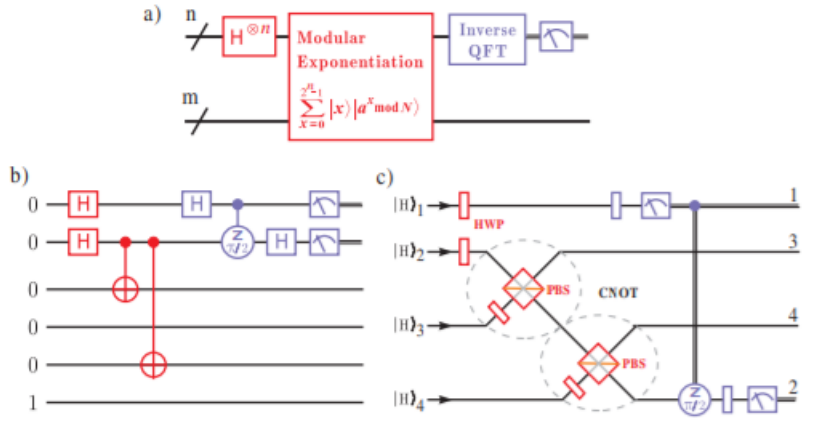
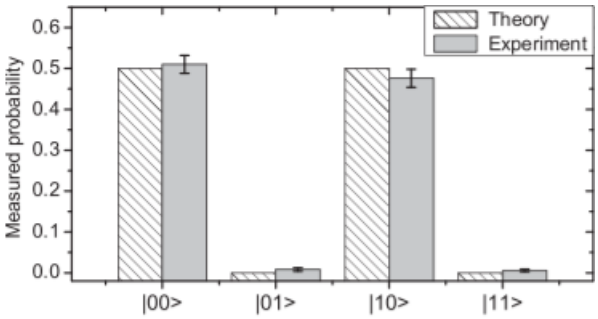


FIG. 1 (color online). Quantum circuit for the order-finding routine of Shor's algorithm. (a) Outline of the quantum circuit. (b) Quantum circuit for $N = 15$ and $a = 11$. The MEF is implemented by two CNOT gates, and the QFT is implemented by Hadamard rotations and two-qubit conditional phase gates. The gate-labeling scheme denotes the axis about which the conditional rotation takes place and the angle of rotation. (c) The simplified linear optics network using HWPs and PBSs to implement the MEF circuit and the semiclassical version of the QFT circuit. The double lines denote classical information.

As shown before, the superposition we expect from this experiment is $|00\rangle$ and $|10\rangle$ (here we are only showing the results for two of the qubits as the third one is easily separable from the other two). This is confirmed by the result of the experiment



4.2.2 Using a Photonic Chip

One of the disadvantages of the above approach using linear optics is that the apparatus can be bulky and is difficult to scale. A solution to this was proposed in the paper [**4**] where Alberto et al., where a photonic chip is used to make the quantum circuit.
On this photonic chip, photons propogate in silica-waveguides(similar to that in an optical fibre) which are micro-fabricated on a silicon chip. On the chip, the silica waveguides are brought together to form a sequence of logical gates.
The logical gates implement a compiled version of Shor's algorithm to factorise N=15 where x = 2.

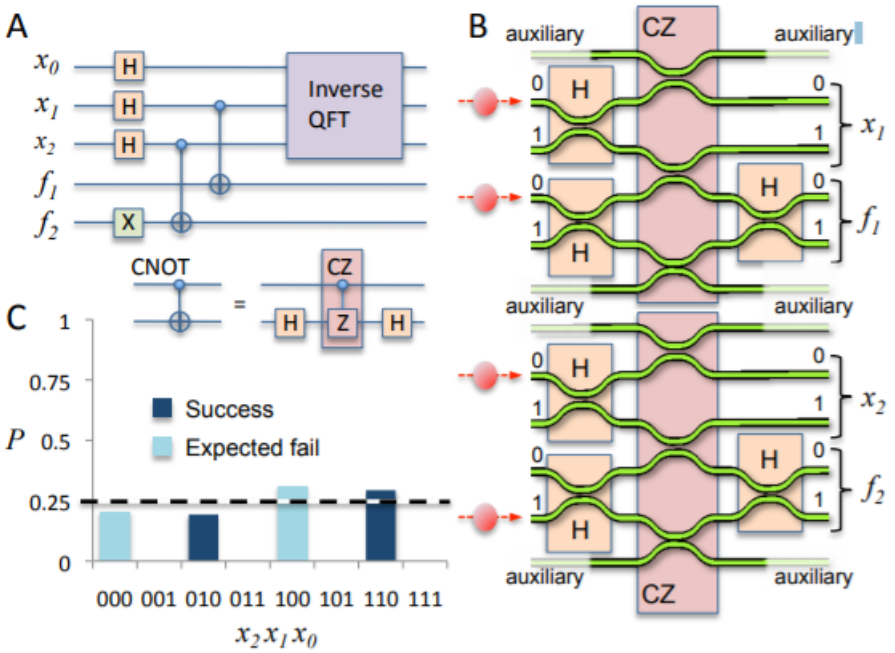
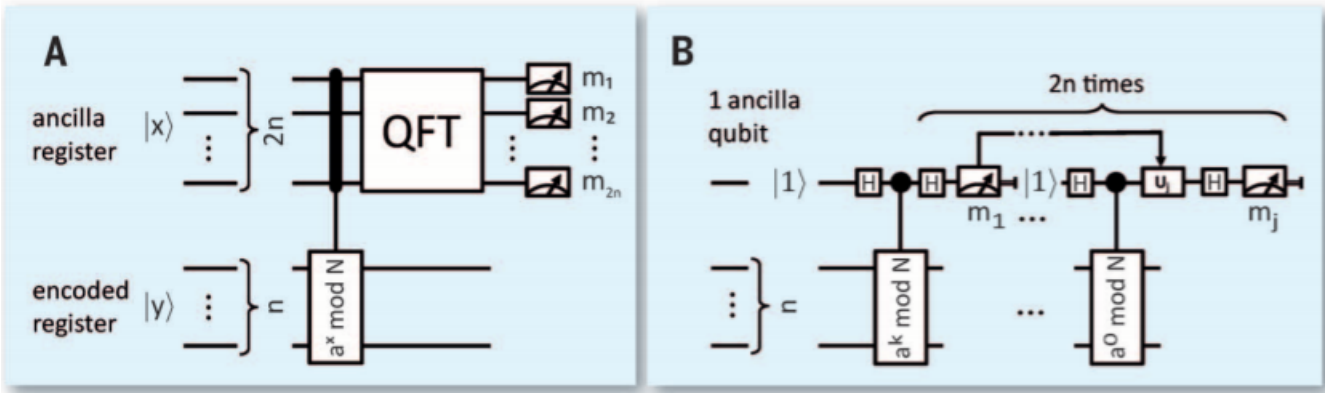


FIG. 1: Integrated optical implementation of Shor's quantum factoring algorithm. (**A**) The quantum circuit. (**B**) Schematic of the waveguide on chip device that implements the quantum computation. The x_n qubits carry the result of the algorithm; f_n are additional qubit required for the computation to work. (**C**) Outcomes of the algorithm.

The 3-bit output from the circuit is $|000\rangle$, $|010\rangle$, $|100\rangle$, $|110\rangle$, corresponding to 0,2,4 and 6 which gives us $r = 4$, giving factors 3 and 5.

4.3 Trapped Ion Quantum Computers

In [**6**], Thomas Monz et al. use a trapped ion quantum computer to implement Shor's algorithm. They used ^{40}Ca ions placed in a linear Paul trap. In this trap, the ions are separated by $5\mu m$ and each can be in either $|0\rangle$ or $|1\rangle$. The qubits are controlled through lasers.
In this experiment, N=15 is factorized. Unlike the previous implementations discussed, the trapped ion computer does not use a compiled version of Shor's algorithm. In traditional Shor's algorithm, 12 qubits will be needed to factorise 15. However, it is difficult to control this many qubits. In 1995, Kitaev has showed that we can factorise numbers with fewer number of qubits(in this case, we only need 5 qubits), if the answer is output 1 qubit at a time. Therefore, we will need 4 qubits to perform the quantum calculations and the fifth qubit transfers information within the computer and is used to output the result. The comparison between the two schemes is shown in the following diagram.



A benefit of this approach is the scalability to larger numbers. As has been discussed the NMR and photonic techniques are not easily scalable. However trapped ions have the potential to build much larger systems. In the paper [**6**], the authors comment that it can be scaled using a segmented trap rather than a Paul trap. Such a trap is already available but it is difficult to control and using it in quantum computers is still in an early stage of development.

4.4 Adiabatic Quantum Computing

So far, we have discussed Shor's algorithm to solve the factorization problem using quantum computers. This follows the circuit model of computation where the computation is performed by a sequence of discrete operations. But there is also another type of quantum algorithms based on **Adiabatic Quantum Computation(AQC)**.
Adiabatic quantum computing is designed to solve *optimization problems* where we need to find the best solution out of many possible solutions. Because, it does not use quantum gates, it is robust against any imperfection in applying the unitary operations.

In AQC, a quantum system is prepared in the ground state of an initial Hamiltonian H_0, and the possible solutions are encoded in in the eigenstates of the problem Hamiltonian H_p. The optimal solution is encoded in the ground state of H_p. The computation consists of the system evolving from H_0 to H_p. If this is performed slowly enough, the system will always stay in the ground state(not in ground state of H_0, but the lowest possible energy level). So in the end, the system will be in the ground state of H_p and hence will denote the optimal solution to the problem. This is summarised in the equation

$$H(t) = [1 - s(t)]H_0 + s(t)H_p \tag{15}$$

where the function $s(t)$ varies from 0 to 1, parametrizes the interpolation. $s(t) = 0$ implies the state H_0 and $s(t) = 1$ implies the state H_p.

In the paper [**7**], Nanyang et al., use AQC(experimentally implemented through NMR) to factorise the number 143(=11x13). They did not use a compiled method, i.e., their method does not require prior knowledge of the answer.

Interestingly in [**8**], Nikesh et al., showed that using the exact same room temperature NMR experiment used in [**7**], to factor an entire class of numbers. They experimentally factorized the number

56153 and said that their method can be used to factorize 291311 as well. They also demonstrated the first quantum factorization of a "triprime" which is the product of three prime numbers. They factorized 175, having factors 5,5 and 7.

Currently the largest number to be factorised is 1028171, which was done by the D-Wave 2000. This machine also uses the AQC framework and implements it through a process called quantum annealing.

References

[1] Nielsen, M., & Chuang, I. (2010). Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511976667

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

[3] Chao-Yang Lu, Daniel E. Browne, Tao Yang, and Jian-Wei Pan, Demonstration of a Compiled Version of Shor's Quantum Factoring Algorithm Using Photonic Qubits, Phys. Rev. Lett. 99, 250504 (2007)

[4] A. Politi, J. C. F. Matthews, J. L. O'Brien, Shors quantum factoring algorithm on a photonic chip, Science 325 (5945) (2009) 1221.

[5] B. P. Lanyon, T. J. Weinhold, N. K. Langford, M. Barbieri, D. F. V. James, A. Gilchrist, and A. G. White, Experimental Demonstration of a Compiled Version of Shor's Algorithm with Quantum Entanglement Phys. Rev. Lett. 99, 250505 (2007)

[6] Thomas Monz,Daniel Nigg, Esteban A. Martinez, Matthias F. Brandl, Philipp Schindler, Richard Rines, Shannon X. Wang, Isaac L. Chuang, Rainer Blatt, Realization of a scalable Shor algorithm, Science 351 (2016)

[7] Nanyang Xu, Jing Zhu, Dawei Lu, Xianyi Zhou, Xinhua Peng, and Jiangfeng Du, Quantum Factorization of 143 on a Dipolar-Coupling Nuclear Magnetic Resonance System, Phys. Rev. Lett. 108, 130501 (2012)

[8] Nikesh S. Dattani, Nathaniel Bryans, Quantum factorization of 56153 with only 4 qubits (2014)