Shor's Algorithm and the Impact & Potential of Quantum Computing

Quantum computing is an emerging technology that is widely discussed topic these days that gets a lot attention. With IBM unveiling its 400 qubit-plus quantum processor system last month, a lot of hype but controversy is stirred up over its opportunities and threats. Put simply, this is due to the quantum computing being better at solving a set of interesting mathematical computations and algorithms. One of the algorithms that results in this quantum advantage is Shor's algorithm. Firstly, Shor's algorithm will be discussed and examples of its relation to the course will be given along the way. Secondly, the applications both opportunities and threats of the specific advantage Shor's algorithm results in. And finally, an overview of the current body of work relating to the topic.

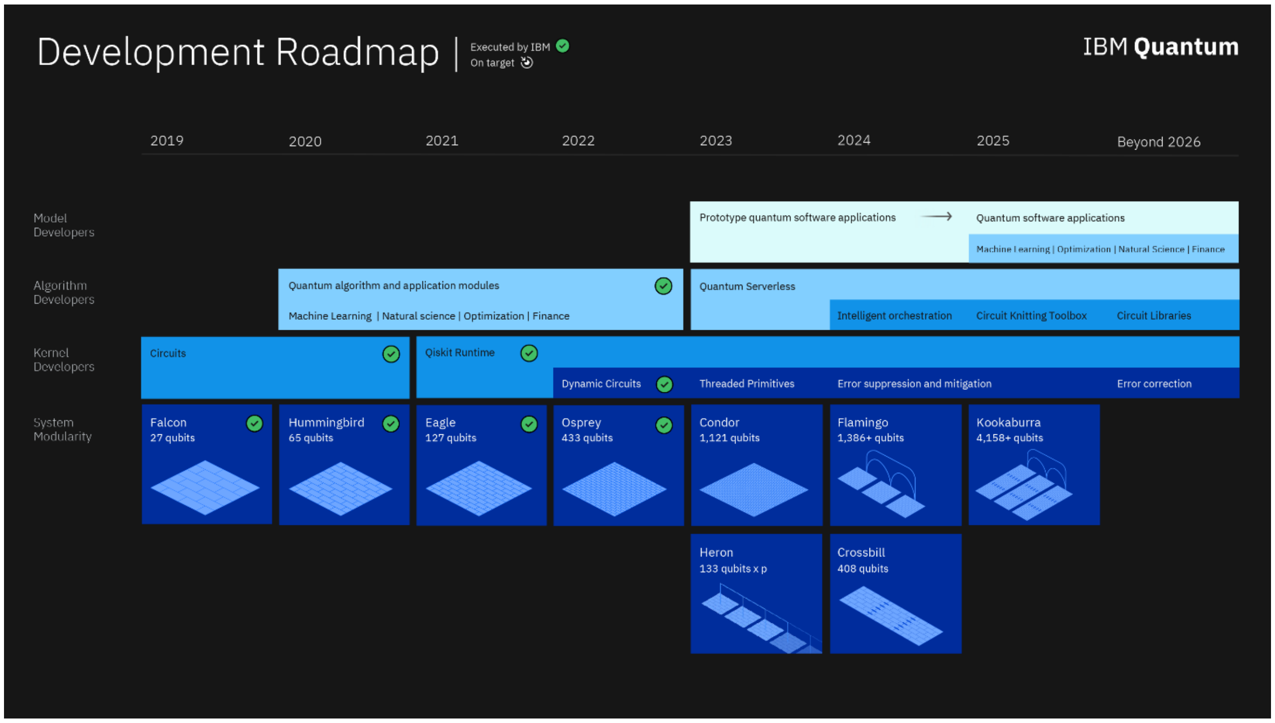

Figure 1 - Qubit Roadmap 2022

Finding the prime factorization of big numbers is known to be computationally expensive to do and is the concept our current digital security and encryption methods are built on. This is because multiplying 2 prime numbers together is way faster and easier than finding the prime factorizations themselves. We went over this while learning about Int mod N and RSA during 2151. This is why Shor's algorithm, an algorithm that allows quantum computers to efficiently (in polynomial time compared to exponential) find the prime factors of a given composite number, got a lot of attention and was seen as a threat to against digital infrastructure and security. Here is how it works:

When determining the prime factors of a number, we want to determine g where

$$N = g * h$$

g does not need to be a direct factor of N, it can also share factors

$$N = a * b, g = a * c$$

For example, both 9 and 18 share a factor of 3 even though 9 is not a factor of 18:

$$9 = 3 * 3, 18 = 3 * 6$$

The largest value that divides both of those numbers without a remainder is called the Greatest Common Factor (GCD). This is where the extremely efficient Euclidean Algorithm that we covered in 2151 comes in. It helps us determine if two numbers share a common factor. For example, the GCD (42, 105) = 21, where 21 is a common factor. Otherwise, If the GCD of 2 numbers is 1, the numbers are "coprime" and share no common factors. As mentioned previously, the g we are looking for does not need to be a direct factor of our target N necessarily but can also share a common factor with it.

This means we can adopt a guess and check approach where we guess a random positive number (g) and check with GCF (g, N):

1. If GCF (g, N) ≠ 1, a factor of N was found, and we are done.
2. If GCF (g, N) = 1, another better guess is needed.

Note that our guess g also has to be less than N because if its larger it wouldn't be a factor.

This approach classically would be considered a slow brute force method but there is an extension we can make that lets us use quantum computers to our advantage:

Again, as mentioned previously, the g we are looking for does not need to be a direct factor of our target N necessarily but can also share a common factor with it.

This means we can extend this guess and check approach by using modular exponentiation to improve our bad guess to a better guess more likely to be the answer, as long as we figure out or are given a p value which accounts for the fact g does not have to be a direct factor necessarily as such:

$$g^p = m*N + 1$$

$$g^p – 1 = m*N$$

$$(g^{p/2} + 1) (g^{p/2} - 1) = m*N$$

Expanding it would look like this: $g^{p/2} g^{p/2} + g^{p/2} - g^{p/2} - 1 = g^p - 1$

p and m are integers, and an example of this would be:

$$\text{GCF}(6, 35) == 1$$

$$6^2 = 36 = 1 * 35 + 1$$

$$(p=2 \ \& \ m=1)$$

Note this involves figuring out the value p, which will be discussed later, so let's assume p is given.

Here is the full example:

Let's say N, our target number is 35, g our guess is 6 and the GCF (6, 35) == 1

Given p = 6,

$$6^{6/2} + 1 = 217$$

$$6^{6/2} - 1 = 215$$

Now without extending the algorithm to use Euclid's Algorithm 215 and 217 would just be non-factors of 35 that we have no use of, but because we do have it, we can do the following:

$$\text{GCF}(217, 35) = 7$$

$$\text{GCF}(215, 35) = 5$$

Here 7 and 5 are the prime number factorizations we are working so hard for.

Now it is possible that in another case after an iteration of our algorithm, our GCF ($g^{p/2} \pm 1$, N) is equal to 1 which would mean we have to try another g and iterate again until we find the right g. The specific time complexity of this algorithm can be noted by Big O notation (that we also went over in 2151) and is found in polynomial time of log(T), where T is the size of our integer N. Now this is assuming we get a value p given to us somehow, but to do so well need the help of quantum computers.

Now on a high level, a quantum computer has 2 key characteristics that allows us to figure this out. First its superposition which allows it to calculate all the possible answers for our p value for our guess. The second is interference, where because of the quantum nature, the correct answer can be mapped to the lowest energy state of the quantum system and the probabilities of the wrong answers will "destructively interfere" giving us the p corresponding to our equation.

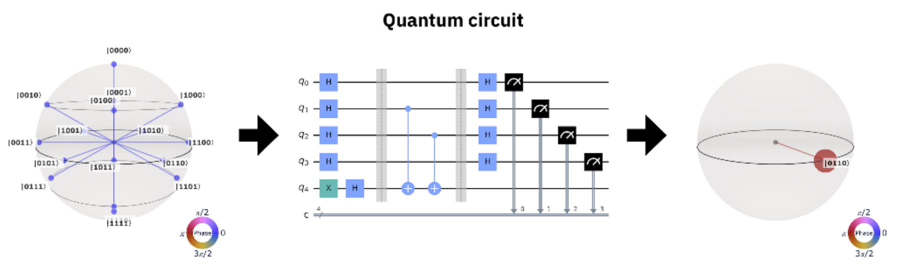

Figure 2 - high level quantum circuit representing destructive interference

Shor’s algorithm is all the classical and quantum sub routines mentioned above that are required to factorize a given number.

Being able to factorize big numbers does not only imply a threat to RSA that was mentioned previously. It also brings a lot of positive applications. It could be used in areas of science and engineering where large prime numbers are important, such as in the optimization of large-scale numerical simulation and design of error-correcting codes (which we went over in hamming codes etc in 2151). Large scale numerical simulation tends to use a lot of multiplication. Here is an example:

```
a*x + b*y = c
#where a, b, and c are large composite numbers and x and y are unknown variables that we
need to solve for
```

In this (over simplified) “simulation”, we need to perform many mathematical operations on a, b, and c, such as multiplication, division, and exponentiation. Instead of computing the above, we can decompose each a, b, and c to into their prime numbers using Shor’s algorithm and rewrite this equation as such:

```
(p1*p2*...*pn)*x + (q1*q2*...*qm)*y = (r1*r2*...*rs)
#where p1, p2, ..., pn, q1, q2, ..., qm, and r1, r2, ..., rs are the prime factors of a,
b, and c, respectively
```

This way the equation is computed way faster. This is because we can now use certain numerical algorithms that are more efficient on expressions with prime numbers than composite numbers, such as the Fast Fourier Transform (in certain cases).

As a result, we can simulate large scale simulations complex physical systems more efficiently and accurately. This could be useful in fields such as computational chemistry or materials science, where researchers need to simulate the behavior of large numbers of atoms or molecules. For example, in material science researchers often use numerical simulations to study the properties of materials. These simulations typically involve solving complex equations that describe the interactions between atoms and molecules in a material. By using Shor’s algorithm, we can for example predict the properties of new materials and identify those with the most potential for energy conversion, storage, and other applications. This could lead to the development of more efficient and cost-effective energy storage and conversion technologies.

Other examples of how this can potentially impact the energy sector are: optimizing production and distribution of solar, wind, and other renewable energies to make them more accessible. Being good at searching vast amounts of data and simulating the behavior of complex systems, means it’s easier to identify the most efficient and cost-effective ways to generate and distribute renewable energy across a complex power grid system.

Quantum computing can also help reduce emissions in the energy sector through more efficient simulation of molecular models for climate models, leading to better identification of efficient and cost-effective energy use in transportation and manufacturing. This could improve the efficiency and sustainability of energy production and use which a lot of investments go towards these days anyways.

Another very interesting interdisciplinary application is in engineering, where repetitive systems and patterns like gear ratio and chip layout require choosing highly factorable numbers that give you more gear ratios, layout options, and integer unrolling factors when designing a part or system. Other interesting current work includes working on using Shor’s algorithm in NP type problem which are problems very hard for a classical computer to solve.

Given the significant impact of computing on society, the potential for quantum computing to revolutionize the healthcare industry by developing new drugs and materials, the potential to improve the energy sector with more efficient storage and conversion technology, and the potential threat to encryption, there is a strong benefit and need to invest in and continue researching quantum computing to fully understand and develop this technology, including new encryption methods.

References:

- https://quantum-computing.ibm.com/composer/docs/iqx/guide/shors-algorithm
  https://qiskit.org/textbook/ch-algorithms/shor.html
- D. Beckman, A. N. Chari, S. Devabhaktuni, and J. Preskill, “Efficient Networks for quantum factoring,” arXiv.org, 21-Feb-1996. [Online]. Available: https://arxiv.org/abs/quant-ph/9602016. [Accessed: 15-Dec-2022].
- M. S. Mahoney, The History of Computing in the History of Technology, Jun. 1988.
- J. H. Park, J. H. Moon, H. J. Kim, M. H. Kong, and Y. H. Oh, “Sedentary lifestyle: Overview of updated evidence of potential health risks,” Korean journal of family medicine, Nov-2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7700832/. [Accessed: 13-Dec-2022].
- M. Haart, “(PDF) Quantum Computing: What it is, how we got here, and who’s working ...,” Mar-2019. [Online]. Available: https://www.researchgate.net/publication/331844245_Quantum_Computing_What_it_is_how_we_got_here_and_who's_working_on_it. [Accessed: 13-Dec-2022].
- R. U. Rasool, H. F. Ahmad, W. Rafique, A. Qayyum, and J. Qadir, “Quantum Computing for Healthcare: A Review,” figshare, 16-Dec-2021. [Online]. Available: https://www.techrxiv.org/articles/preprint/Quantum_Computing_for_Healthcare_A_Review/17198702. [Accessed: 13-Dec-2022].
- Z. Eldredge, “Quantum computing opportunities in renewable energy - researchgate,” Sep-2021. [Online]. Available: https://www.researchgate.net/publication/353528286_Quantum_Computing_Opportunities_in_Renewable_Energy. [Accessed: 14-Dec-2022].
- M. Langione, C. Tillemann-Dick, A. Kumar, and V. Taneja, “Where will quantum computers create value-and when?,” BCG Global, 15-Dec-2021. [Online]. Available: https://www.bcg.com/publications/2019/quantum-computers-create-value-when. [Accessed: 13-Dec-2022].
-

Note: around 1500 ± 50 words not including references.