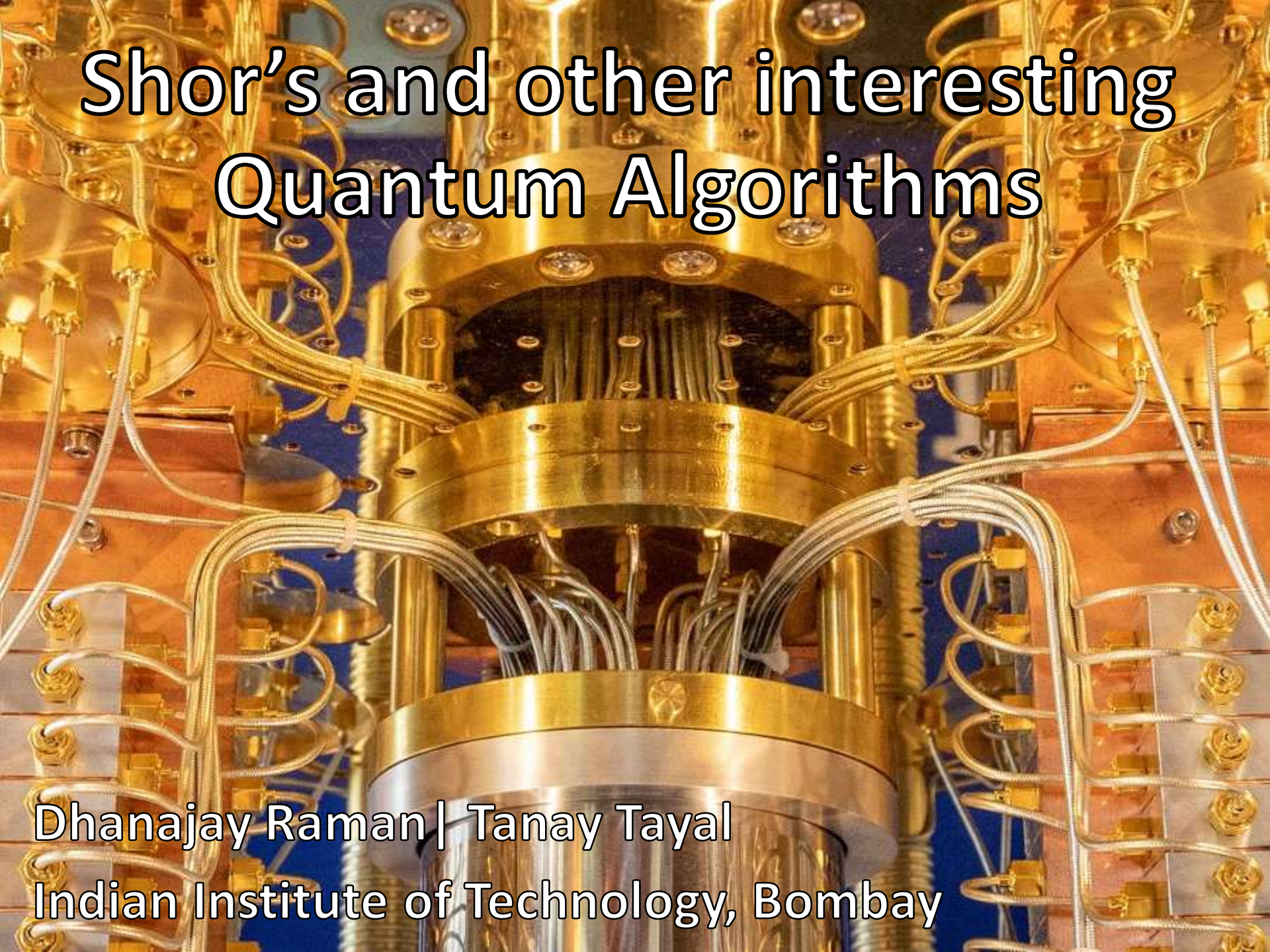
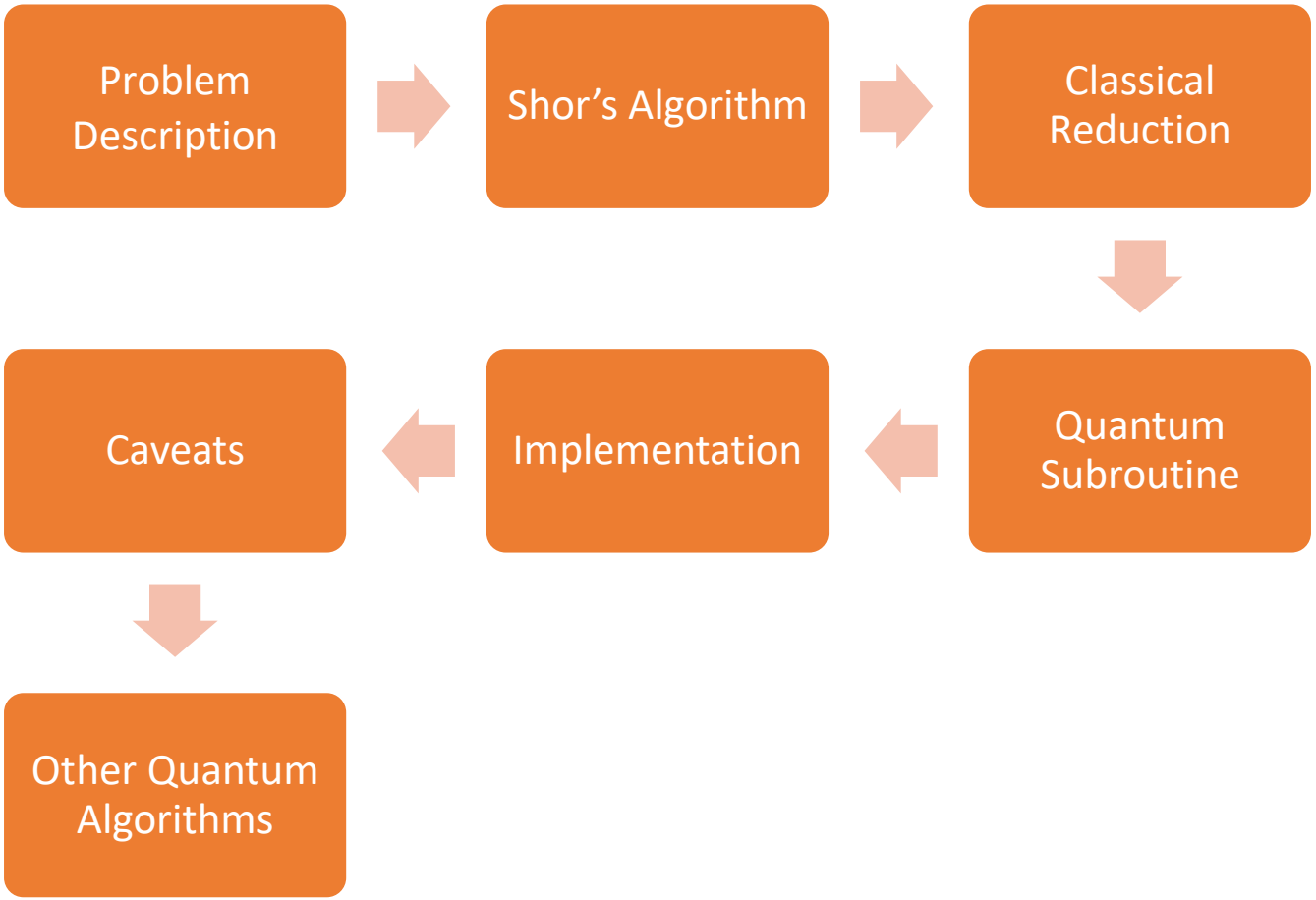
Shor's and other interesting Quantum Algorithms



Dhanajay Raman| Tanay Tayal
Indian Institute of Technology, Bombay

Overview


Problem Description
Shor's Algorithm
Classical Reduction
Quantum Subroutine
Implementation
Caveats
Other Quantum Algorithms

The Problem

- Many encryption systems (like RSA) rely on the difficulty of prime factorization of large numbers into their prime components
- The security of these systems is based on the assumption that no efficient classical algorithm exists for factorization
- The best classical algorithms for factoring large numbers grow exponentially with the size of the numbers, making it computationally impractical for very large numbers
- This computational challenge is crucial in maintaining the security of modern cryptography

The Problem

- Quantum computers are different from classical computers, utilizing quantum mechanical principles like superposition and entanglement
- Quantum computers have the potential to solve certain problems much more efficiently (even exponential speedup is possible)
- Shor's Algorithm poses a theoretical threat to the foundation of current cryptographic practices by promising an efficient method for factoring large numbers

Shor's Algorithm

- At the heart of Shor's Algorithm is the concept of finding the periodicity of a function. The specific function is

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

where a is a randomly chosen integer and N is the number to be factored

- If the period, or the repeat interval, of this function can be determined, it reveals insights that lead directly to the factors of N
- If the factors are not successfully found, the algorithm is repeated with a different random a. The probability of success increases with each iteration

Shor's Algorithm: Classical Reduction

- Pick a random number $1 < a < N$, where N is the number to be factored
- Check if a is coprime to N (i.e., $\gcd(a, N) = 1$). If not, $\gcd(a, N)$ gives a non-trivial factor of N
- Use the chosen a to define the function
$$f(x) = a^x \bmod N$$
- We use the quantum subroutine to find r, the smallest x for which $f(x) = 1$, called the order of a
- If r is even, repeat with a different value of a
- Use r to find factors of N by computing $\mathrm{g} = \gcd\left(a^{r/2} + 1, N\right)$: if g is nontrivial, the other factor is N/g, otherwise repeat with a different value of a

Shor's Algorithm: Quantum Subroutine

- The main steps of the algorithm are:
  - Initial State Preparation
  - Modular Exponentiation
  - Quantum Fourier Transform
  - Measurement of the state
  - Postprocessing to get phase and corresponding period
- For demonstration (and implementation) purposes, we describe the circuit for $N = 15$ and $a = 7$

Shor's Algorithm: Quantum Subroutine

- Registers used:
  - q0 to q7: These are the qubits initially prepared in a superposition of states using Hadamard (H) gates. They serve as the input for the quantum Fourier transform (QFT) that is used at the end of the circuit
  - q8 to q11: These qubits are used to store the outcomes of the modular exponentiation operations. They represent the values of the function $a^x \bmod N$
  - c: This classical register stores the results of the measurements

Shor's Algorithm: Quantum Subroutine

- Initial State Preparation
  - All qubits from q0 to q7 are put through Hadamard gates to create a superposition, allowing for parallel computation across multiple states
- Modular Exponentiation
  - The sequence of blocks labeled with $f(x) = 7^{2^i} \bmod 15$ (where i ranges from 0 to 7) represent the modular exponentiation operation which is crucial for finding the period of the function. This operation is controlled by each qubit from q0 to q7, affecting the state of qubits q8 to q11. This setup essentially prepares the state $|x, a^x \ mod \ N\rangle$ where x is the state represented by qubits q0 to q7

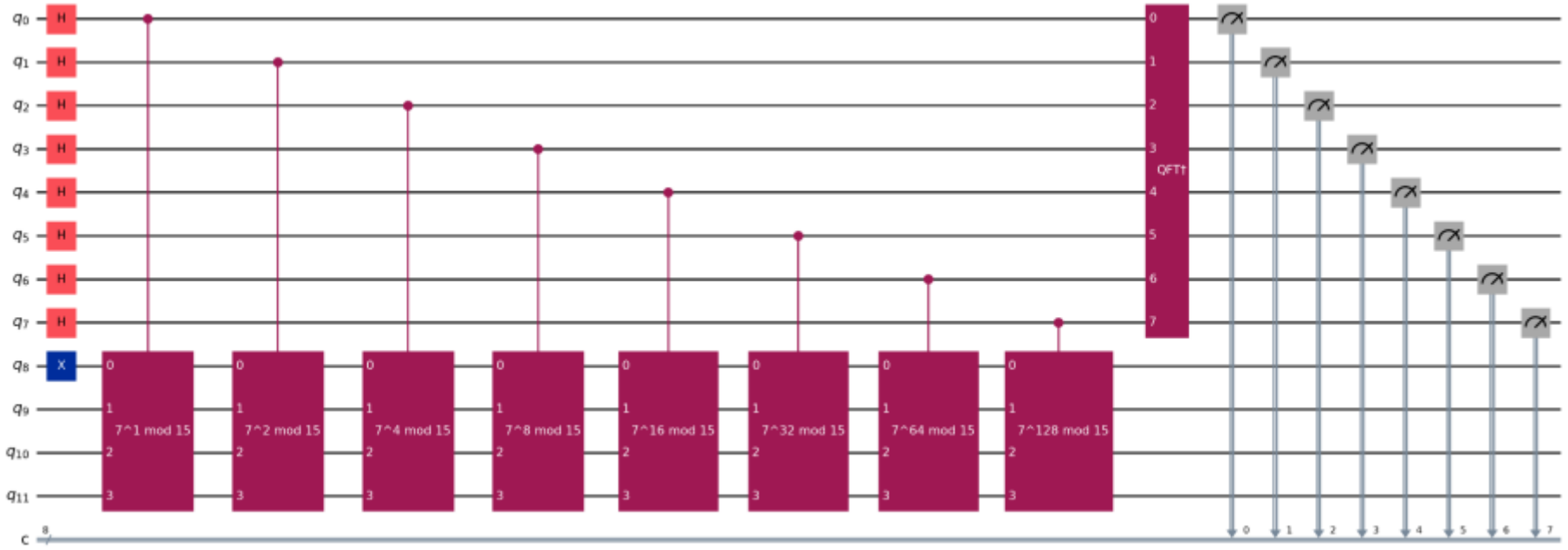
Shor's Algorithm: Quantum Subroutine

- Quantum Fourier Transform (QFT):
  - The series of quantum gates on qubits q0 to q7 after the modular exponentiation constitute the QFT. The QFT is used to transform the quantum state into one where the periodicity of the function $a^x \bmod N$ can be extracted from the probability distribution observed upon measurement
- Measurement:
  - The qubits q0 to q7 are measured, and their state is recorded in the classical register c.

Shor's Algorithm: Quantum Subroutine

- Postprocessing:
  - The final measurement (of register c) gives an 8-bit string
  - Converting it to binary and dividing by $2^8 = 256$ gives us the phase measurement
  - The measured phase is equal to $^s/_r$ where s is some integer between 1 and r
  - Hence rationalizing the phase (and limiting the denominator to be smaller than 15) returns a fraction whose denominator is a good guess of r
  - This r is returned by the function

Qiskit Implementation



Caveats

- Requirement of a Large Quantum Computer
  - Shor's Algorithm needs a significantly large number of qubits to factor numbers of a practical size (such as those used in encryption). Current quantum computers do not yet have the necessary number of reliable qubits.
- Error Rates and Quantum Decoherence
  - Quantum computers are prone to errors and decoherence, which can affect the fidelity of the results.
  - The quantum gates must operate with very high precision, and maintaining coherent quantum states over the time required to perform the calculations is challenging.

Other Quantum Algorithms

- Grover's Search
  - Searches for a solution within an unsorted database with quadratic speedup compared to classical algorithms
  - Can be used to perform brute-force attacks on symmetric cryptographic systems, effectively reducing the security of n-bit keys to n/2 bits in a quantum setting
- Quantum Key Distribution
  - Enables two parties to generate a shared random secret key, which is secure against any eavesdropping, guaranteed by the laws of quantum mechanics
  - Provides a method for secure key distribution, which is essential for maintaining the secrecy of encrypted communications even in the face of quantum computing

References

- Qiskit - https://www.ibm.com/quantum/qiskit
- Shor, P.W. (1994). "Algorithms for quantum computation: Discrete logarithms and factoring". Proceedings 35th Annual Symposium on Foundations of Computer Science. IEEE Comput. Soc. Press. pp. 124–134.
- Grover, Lov K. (1996-07-01). "A fast quantum mechanical algorithm for database search". Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery. pp. 212–219.
- C. H. Bennett and G. Brassard. "Quantum cryptography: Public key distribution and coin tossing". In Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, volume 175, page 8. New York, 1984.