Shor's Algorithm Implementation Using Quantum Rings

Overview

This project implements **Shor's Algorithm** using the **Quantum Rings simulator** to factor semiprime numbers. The algorithm is a **quantum approach to integer factorization**, leveraging **modular exponentiation**, **Quantum Fourier Transform (QFT)**, and **period finding** to determine the prime factors of a given integer **N**.

In cases where quantum computation takes too long, the program also uses the **Greatest Common Divisor (GCD) fallback method** to quickly extract factors when possible.

Implementation Details

- The program is written in **Python** using the **QuantumRingsLib** library.
- The quantum circuit consists of:
    - **Hadamard gates** to create superposition.
    - **Modular exponentiation** to encode periodicity.
    - **Inverse Quantum Fourier Transform (IQFT)** to extract the period.
    - **Measurement operations** to retrieve results.
- The program runs **five attempts** per number to increase the probability of success.
- **Optimization:** Before running quantum computation, the algorithm first checks for common factors using the **GCD method**, which significantly reduces execution time for some inputs.

Scalability & Challenges

- **Why Quantum?**
    - Classical computers take **exponential time** to factorize large numbers (e.g., RSA-2048).
    - Shor's Algorithm theoretically achieves **polynomial-time** factorization, making RSA encryption vulnerable in the future.
- **Challenges Faced:**
    - **Execution Time:** The Quantum Rings simulator takes time for circuit execution, especially for larger numbers.
    - **Resource Limits:** Larger N values require more qubits, increasing computational overhead.
    - **Quantum Noise:** On real quantum hardware, gate errors and decoherence affect accuracy.

- **Solutions & Future Improvements:**
    - Implement **parallelized modular exponentiation** for speed.
    - Use **optimized quantum compilers** to reduce gate depth.
    - Transition to **real quantum hardware** once available.

Results

The table below summarizes the factorization results from the Quantum Rings simulator:

(Note: Execution times may vary based on system load.)

BIT	Semiprime (N)	Prime Factors	Time of execution
8-bit	143	(11,13)	20s
10-bit	899	(29,31)	3m
12-bit	3127	(47,67)	18m

Submission Details

- **Quantum Rings Email:** fatma_h00197@cic-cairo.com
- **GitHub Repository:** [Your GitHub Link Here]
- **Documentation Author:** Fatma H.