Математика. Прикладная математика

УДК 51-73+ 004.932.2

Борисова Влада Владимировна
Дальневосточный федеральный университет
г. Владивосток, Россия
E-mail: borisova.vvl@students.dvfu.ru
Дегтярев Данил Викторович
Дальневосточный федеральный университет
г. Владивосток, Россия
E-mail: degtyarev.dv@students.dvfu.ru
Borisova Vlada Vladimirovna
Far Eastern Federal University
Vladivostok, Russia
E-mail: borisova.vvl@students.dvfu.ru
Degtiarev Danil Viktorovich
Far Eastern Federal University
Vladivostok, Russia
E-mail: degtyarev.dv@students.dvfu.ru

РЕАЛИЗАЦИЯ АЛГОРИТМА КВАНТОВОЙ ФАКТОРИЗАЦИИ ШОРА

THE IMPLEMENTATION OF SHOR'S QUANTUM FACTORIZATION ALGORITHM

Аннотация. Квантовый алгоритм Шора – алгоритм факторизации целых чисел на квантовых компьютерах, который позволяет быстро разложить большие числа на их простые множители. Этот алгоритм является одним из наиболее известных примеров использования квантовых вычислений для решения задач, для которых классические алгоритмы имеют высокую вычислительную сложность. В работе описан принцип работы квантового алгоритма Шора и его применение в различных задачах. Также рассмотрено, как этот алгоритм может быть реализован на квантовых компьютерах и какие вычислительные ресурсы необходимы для его выполнения.

Abstract. Shor's quantum algorithm is a quantum computing-based algorithm for factorizing integers, enabling the rapid decomposition of large numbers into their prime factors. This algorithm stands as one of the most prominent examples of leveraging quantum computation to address problems with high computational complexity in classical algorithms. This paper elucidates the operational principles of Shor's quantum algorithm and its applications across various problem domains. Furthermore, it delves into the implementation of this algorithm on quantum computers and elucidates the computational resources necessary for its execution.

В группе авторов данной статьи – А.Г. Макаров (Дальневосточный федеральный университет, Институт прикладной математики ДВО РАН), А.Е. Боршевников (Дальневосточный федеральный университет), К.С. Солдатов (Дальневосточный федеральный университет, Институт прикладной математики ДВО РАН), К.В. Нефедев (Дальневосточный федеральный университет, Институт прикладной математики ДВО РАН).

Ключевые слова: квантовые вычисления, алгоритм Шора, факторизация чисел, простые множители, вычислительная сложность.

Key words: quantum computing, Shor's algorithm, integer factorization, prime factors, computational complexity.

DOI: 10.22250/20730268_2023_103_3

Введение

С развитием квантовых вычислений стали актуальными вопросы обеспечения информационной безопасности, особенно в контексте разложения чисел. Алгоритм квантовой факторизации Шора, разработанный Питером Шором в 1994 г., является одним из наиболее известных квантовых алгоритмов. Он позволяет эффективно факторизовать большие составные числа на простые множители за полиномиальное время на квантовых компьютерах [1].

Основная идея алгоритма Шора заключается в использовании квантового параллелизма и интерференции для эффективного нахождения периода функции, связанной с факторизуемым числом. Зная период, можно найти простые множители числа. Этот алгоритм имеет большое практическое значение, так как разложение чисел на простые множители – фундаментальная операция в криптографии.

Проблема факторизации

Факторизация больших чисел является вычислительной задачей, сложность которой для классических алгоритмов растет экспоненциально с увеличением числа. Это означает, что по мере увеличения количества цифр или битов в большом числе для классических методов решения задачи факторизации потребуется значительно больше времени и вычислительных ресурсов. Такой рост сложности происходит в геометрической прогрессии, что делает разложение больших чисел на простые множители вычислительно требовательной задачей на классических компьютерах.

Существует несколько эффективных классических алгоритмов факторизации, – например, метод квадратичных форм Шенкса и алгоритм Полларда-Штрассена, имеющие временную сложность порядка $O(N^{1/5+\varepsilon})$ и $O(N^{1/4}\log^4 N)$ соответственно, где N – число, которое необходимо разложить на простые множители [2]. В то же время алгоритм Шора, который является квантовым алгоритмом факторизации, имеет временную сложность порядка $O(\log^3 N)$ на квантовом компьютере [3]. Таким образом, данный алгоритм значительно быстрее классических алгоритмов факторизации. Однако на данный момент квантовые компьютеры находятся в разработке и не являются доступными для широкого использования.

Проблема факторизации имеет большое значение в криптографии, так как многие криптографические протоколы основаны на сложности разложения на простые множители. Например, алгоритм RSA, широко используемый для шифрования данных, основан на трудности факторизации больших составных чисел**.**

Преимущество квантовой факторизации перед классической

Квантовый алгоритм Шора предоставляет значительное преимущество в решении проблемы разложения на простые множители по сравнению с классическими алгоритмами. Алгоритм способен обеспечить экспоненциальное ускорение процесса факторизации. Это означает, что на квантовом компьютере можно факторизовать большие числа значительно быстрее, чем на классическом компьютере [4].

Однако реализация алгоритма Шора на квантовых компьютерах имеет свои сложности. Требуется достаточно большой и стабильный квантовый компьютер с достаточным количеством кубитов и низким уровнем ошибок. Кроме того, необходимо разработать эффективные методы управления квантовыми состояниями и операциями и контроля за ними.

Описание алгоритма квантовой факторизации Шора

Алгоритм Шора для факторизации чисел состоит из двух ключевых частей – классической и квантовой. Классическая часть выполняется на классическом компьютере и предназначена для подготовки данных и анализа результатов. Она играет важную роль в предварительной обработке данных и последующей интерпретации выходных данных квантовой части алгоритма.

Квантовая часть алгоритма Шора использует принципы квантовой механики – такие как суперпозиция (возможность кубитов находиться в нескольких состояниях одновременно) и интерференция (взаимодействие вероятностей различных квантовых состояний) – для эффективного нахождения периода функции, связанной с факторизуемым числом. Эти принципы позволяют квантовому компьютеру параллельно обрабатывать множество данных и усиливают вероятность правильных ответов, что существенно ускоряет процесс факторизации.

Классическая часть алгоритма Шора

1. Выбор случайного числа a**.**

Выбираем случайное целое число a в интервале $1 < a < N$, где N – число, которое нужно факторизовать. Цель выбора a заключается в том, чтобы найти такое число, которое будет взаимно простым с N, т.е. НОД$(a,\ N)$ должен быть равен 1. Это гарантирует, что a не имеет общих делителей с N, за исключением 1, и, следовательно, a не может быть одним из множителей N. Выбор такого a важен для успешного выполнения алгоритма.

2. Нахождение и проверка НОДа.

После выбора a мы вычисляем НОД$(a,\ N)$ с помощью классического алгоритма, – например, алгоритма Эвклида. НОД$(a,\ N)$ позволяет определить, имеют ли a и N общие делители, кроме 1.

Если НОД$(a,\ N)$ не равен 1, то a уже является нетривиальным делителем числа N, и факторизация завершена. Мы успешно нашли один из множителей N и осталось найти второй множитель.

Если НОД$(a,\ N)$ равен 1, это означает, что a и N взаимно просты, и мы переходим к квантовой части алгоритма для нахождения периода функции $f(x) = a^x \ mod\ N$.

Квантовая часть алгоритма Шора

На рис. 1 представлена квантовая схема реализации алгоритма.

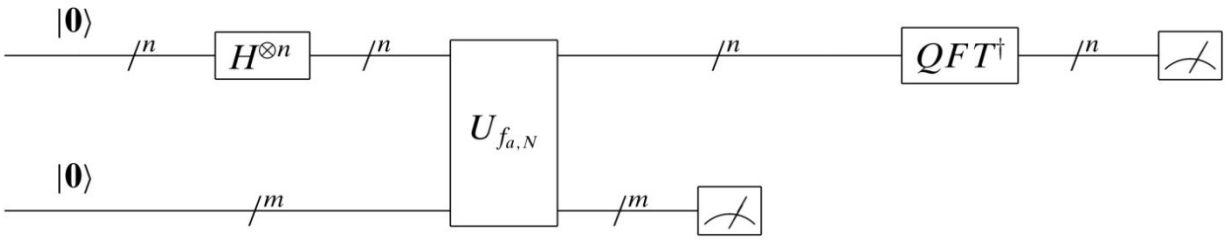

Рис. 1. Квантовая схема реализации алгоритма Шора.

Квантовая часть алгоритма Шора заключается в следующем.

1. Инициализация регистров.

Подготавливаем два квантовых регистра: регистр x (для хранения входных данных) и вспомогательный регистр y. Оба регистра инициализируются в состоянии $|0\rangle$, что можно записать как:

$$|x\rangle = |0\rangle^{\otimes n}, \quad (1)$$

$$|y\rangle = |0\rangle^{\otimes m}, \quad (2)$$

где n – количество кубитов в регистре x, а m – количество кубитов в регистре y.

2. Преобразование Адамара.

Применяем квантовый оператор Адамара (H) ко всем кубитам в регистре x, чтобы создать суперпозицию всех возможных входных значений. Преобразование Адамара применяется следующим образом:

$$H^{\otimes n}\,|x\rangle = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle. \quad (3)$$

Это создает равновероятное распределение всех возможных состояний регистра x.

3. Модулярная функция.

Вводим модулярную функцию:

$$f(x) = a^x \bmod N, \quad (4)$$

где a – случайное целое число; x – значение из регистра x; N – число, которое нужно факторизовать.

Применяем оператор U_f, который реализует операцию $U_f\,|x\rangle\,|y\rangle$, чтобы применить модулярную функцию к состояниям в регистрах x и y:

$$U_f\,|x\rangle\,|y\rangle = |x\rangle\,|y \oplus f(x)\rangle. \quad (5)$$

Это преобразование вносит информацию о модулярной функции $f(x)$ в состояния вспомогательного регистра y.

4. Фиктивное измерение второго регистра.

Выполняем измерение состояний во вспомогательном регистре y, но результат измерения не сохраняется. Это необходимо для подготовки состояний в регистре y к применению обратного квантового преобразования Фурье.

5. Обратное квантовое преобразование Фурье.

Применяем обратное квантовое преобразование Фурье (QFT^{-1}) к состояниям в регистре x, чтобы выделить период r функции $f(x)$. QFT^{-1} определяется следующим образом:

$$QFT^{-1}\,|x\rangle = \frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1} e^{-\frac{2\pi i x y}{2^n}}\,|y\rangle. \quad (6)$$

Обратное квантовое преобразование Фурье переводит состояния в регистре x в состояния, которые содержат информацию о периоде r. Это достигается путем внесения фазовых сдвигов между различными состояниями в регистре y, чтобы амплитуды соответствующих состояний синхронно интерферировали.

6. Измерение первого регистра.

Выполняем измерение состояний в регистре x, получая случайное число x_0, которое представляет одно из возможных значений переменной x. Это число содержит приближенное значение периода r функции $f(x)$.

7. Вычисление периода.

После того как мы измерили состояния регистра x и получили случайное число x_0, можно вычислить приближенное значение периода r с использованием метода непрерывных дробей или других численных методов. Период r можно вычислить следующим образом:

$$r \approx \frac{2^n}{x_0}, \quad (7)$$

где n – количество кубитов в регистре x; x_0 – полученное случайное число после измерения.

8. Нахождение множителей.

После получения приближенного значения периода r можно использовать это значение для определения множителей числа N с помощью классических алгоритмов – таких как алгоритм Евклида. Мы вычисляем НОД(N, $a^{r/2} \pm 1$), где a – случайное целое число, которое выбирали для создания модулярной функции.

Если полученный НОД является нетривиальным делителем числа N, то найдено успешное разложение N на множители.

Если $a^{r/2} \equiv -1 \bmod N$ или если полученный НОД не является нетривиальным делителем N, то мы возвращаемся к началу алгоритма, выбирая другое случайное a и повторяя процесс снова.

Сравнение результатов реализации алгоритма Шора на эмуляторе и реальном квантовом компьютере IBM

В ходе исследования алгоритм Шора был реализован на эмуляторе квантового компьютера с использованием 4 кубитов, написанном на языке программирования Python. Нам удалось разложить число 15 на простые множители 3 и 5, получить схему для квантового компьютера, а также диаграмму результатов измерения состояния кубитов [5]. Дополнительно было проведено испытание алгоритма Шора на реальном квантовом компьютере IBM [6].

Квантовая схема алгоритма Шора (рис. 2) показывает последовательность операций, которые выполняются на квантовом компьютере для разложения числа на простые множители. Сначала кубиты инициализируются, затем используется модулярная функция возведения в степень, обратное квантовое преобразование Фурье и измерение кубитов.

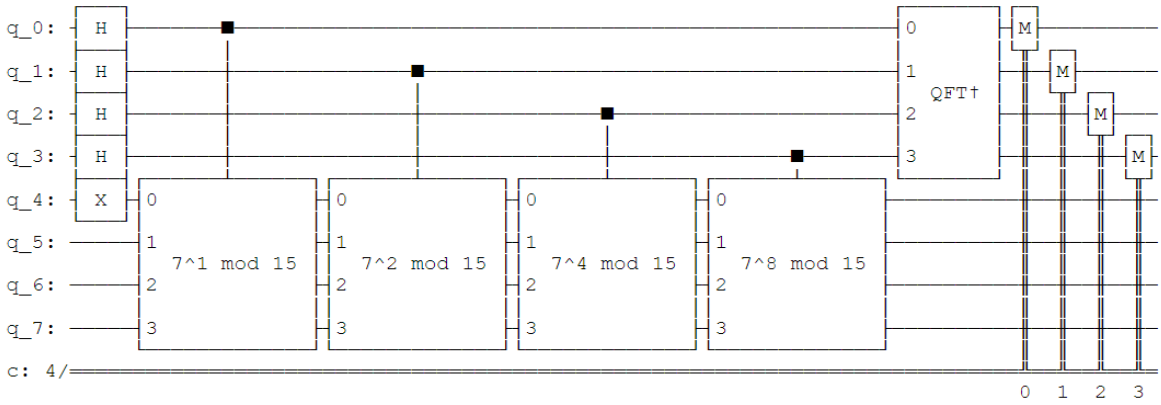

Рис.2. Квантовая схема реализации алгоритма Шора.

Результаты измерений, представленные на рис. 3, показывают распределение вероятностей измеренных состояний кубитов после запуска схемы. Диаграмма позволяет понять, что период функции равен 4. Это видно из наличия четырех выраженных столбцов в диаграмме, представляющей наибольшие вероятности состояний кубитов.

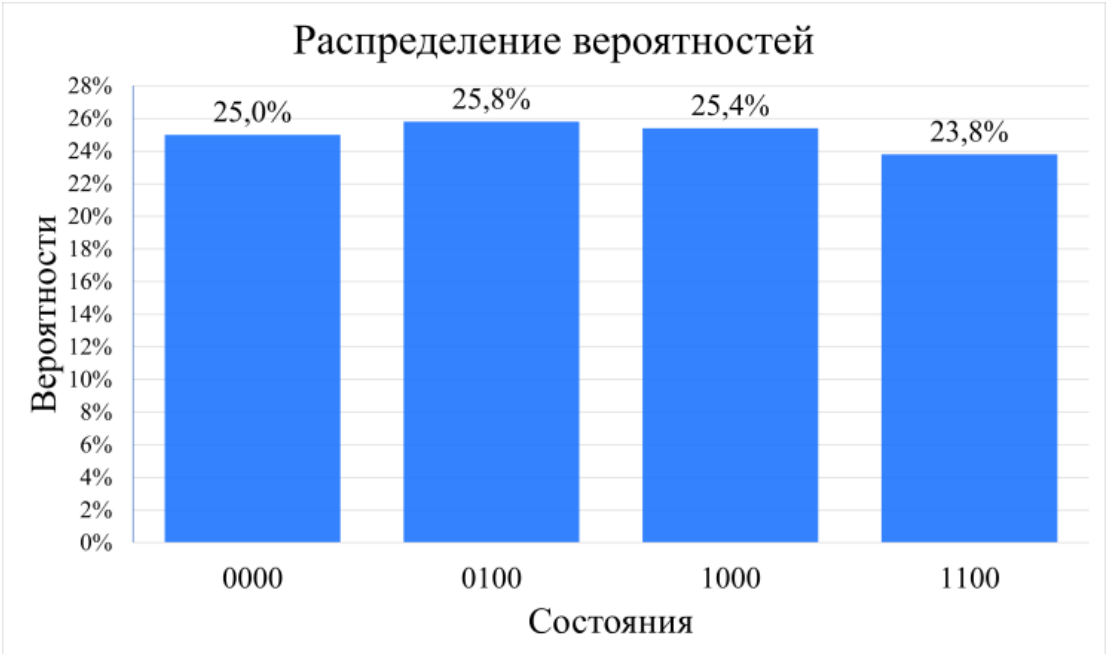

Рис. 3. Результаты измерений на эмуляторе.

В алгоритме Шора, который использует квантовые вычисления, случайные начальные состояния кубитов могут создавать разные вероятности для различных результатов измерений из-за фазовых факторов и воздействия шумов. Однако взаимодействие между квантовыми состояниями дает интерференцию, которая выравнивает вероятности и обеспечивает более равномерное распределение результатов измерений.

После запуска алгоритма на реальном квантовом компьютере IBM была получена диаграмма, отличающаяся от полученной с помощью эмулятора (рис. 4).

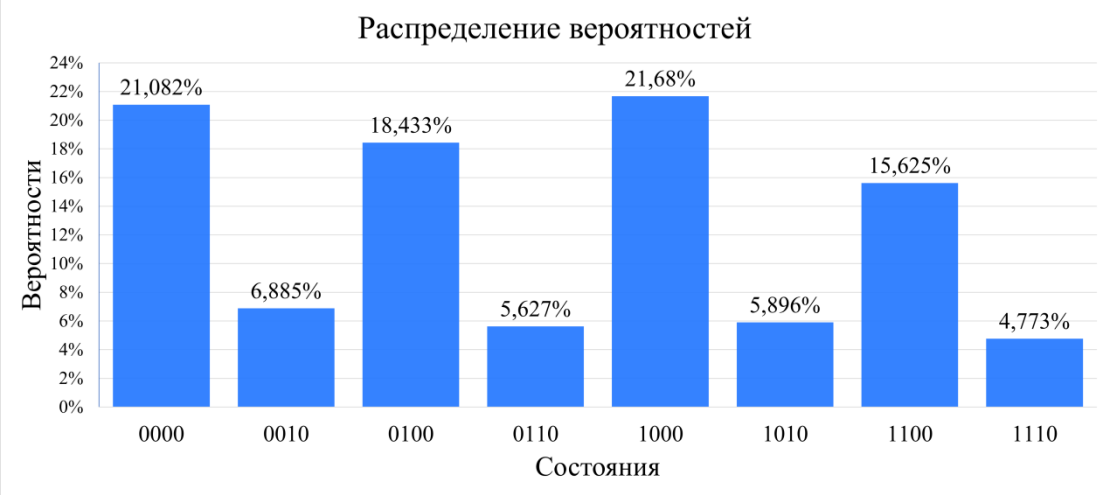

Рис.4. Результаты измерений на реальном квантовом компьютере IBM.

При сравнении с идеальными условиями, полученными при помощи эмулятора на языке Python, мы наблюдаем добавление четырех дополнительных столбцов с низкой вероятностью. Это объясняется высокой степенью погрешности, обусловленной нестабильными состояниями кубитов, подверженными воздействию множества факторов окружающей среды. В случае увеличения числа, подлежащего факторизации, период также увеличится, что приведет к формированию более широкой диаграммы и увеличит количество столбцов вероятностей. Следовательно, процесс вычисления столбцов станет более трудоемким и точность измерений снизится, – возможно, даже до такой степени, что это может привести к недостоверным результатам. Перед нами стоит существенная проблема, характерная для всех текущих квантовых компьютеров, которую предстоит решить в будущем.

Оценка сложности реализации алгоритма

Оценка сложности реализации алгоритма Шора зависит от нескольких факторов – доступности квантового компьютера, уровня опыта и знаний разработчика, а также требуемой точности и надежности результата. При реализации алгоритма необходимо умение программировать на языке, поддерживающем квантовые вычисления (Python или Qiskit). Кроме того, нужен доступ к квантовому компьютеру или эмулятору, способному выполнять квантовые операции. Сложность реализации алгоритма Шора также связана с проблемами, возникающими при работе с квантовыми системами, – такими как декогеренция, ошибки квантовых вентилей и ограниченное количество кубитов. Эти факторы могут привести к ухудшению точности и надежности результатов.

Последствия для криптографии и безопасности

Криптосистемы на основе проблемы факторизации могут быть уязвимы к атакам квантовых компьютеров, так как они основаны на сложности разложения на множители больших чисел, которую квантовые компьютеры могут решать значительно быстрее, чем классические компьютеры. Это

означает, что подобные криптографические системы могут стать уязвимыми для атак, основанных на алгоритме Шора на квантовом компьютере.

Для обеспечения безопасности в условиях наличия квантовых компьютеров необходимо разработать квантово-устойчивые алгоритмы шифрования. Их квантовые свойства используют для обеспечения безопасности передачи информации. Например, квантовые ключи могут быть пригодны для обмена ключами, устойчивыми к атакам квантовых компьютеров.

Развитие квантовых компьютеров требует пересмотра существующих криптографических методов и разработки новых, устойчивых к атакам квантовых вычислителей.

Заключение

В статье была рассмотрена реализация квантового алгоритма Шора. Показаны основные аспекты проблемы разложения на простые множители и преимущества квантовой факторизации перед классической. Описан сам алгоритм Шора и принцип его работы.

Проведена также реализация алгоритма Шора на эмуляторе и реальном квантовом компьютере IBM, с последующим сравнением результатов. Оценка сложности реализации алгоритма Шора показала, что требуется глубокое понимание квантовой физики и математики, а также доступ к квантовым компьютерам или эмуляторам. Однако существуют проблемы (например, декогеренция), которые могут ограничивать развитие квантовых компьютеров.

Реализация алгоритма Шора имеет значительное влияние на криптографию и безопасность. Она вызывает необходимость в разработке квантово-устойчивых алгоритмов и пересмотре существующих методов для обеспечения безопасности информации в условиях наличия квантовых компьютеров. Развитие квантовых компьютеров и реализация алгоритма Шора требуют дальнейших исследований и разработок в области квантовых вычислений и криптографии.

В дальнейшем планируется провести дополнительные исследования и разработки, связанные с алгоритмом квантовой факторизации Шора. Одна из задач – увеличение количества кубитов на квантовом компьютере, что позволит расширить возможности использования алгоритма. Также планируется оптимизировать процессы подготовки состояния кубитов и измерения, чтобы снизить вероятность ошибок в расчетах и повысить эффективность алгоритма. Кроме того, возможны исследования и разработки, связанные с применением алгоритма квантовой факторизации Шора в различных областях. Например, можно исследовать возможность применения алгоритма в криптографии для создания новых методов шифрования и дешифрования, которые будут устойчивы к атакам с использованием квантовых компьютеров.

Исследование проводилось в рамках гранта Президента Российской Федерации для государственной поддержки ведущих научных школ Российской Федерации (НШ-2559.2022.1.2).

---

1. Сысоев, С.С. Введение в квантовые вычисления. Квантовые алгоритмы: учеб. пособие – СПб.: Изд-во Санкт-Петербург. ун-та. – 2019. – 144 с.

2. Василенко, О.Н. Теоретико-числовые алгоритмы в криптографии – М.: МЦНМО. – 2003. – 328 с.

3. Hayward, M. Quantum computing and Shor’s algorithm // Sydney: Macquarie University Mathematics Department. – 2008. – V. 1.

4. Shor, P.W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer // Foundations of Computer Science: Conference Publications, 1997. – P. 1484-1509.

5. Малков, А.В. Моделирование квантовых вычислений с использованием эмулятора Qiskit для языка Python. – М., 2022.

6. Вахний, Т.В., Гуц, А.К., Овчинников, А.В. Выполнение простейших вычислений на квантовом компьютере IBM Q System One // Математическое и компьютерное моделирование: сборник. – М., 2021. – 328 с.