УДК 004.382

ЕМКОСТНАЯ СЛОЖНОСТЬ ОПРЕДЕЛЕНИЯ НОД В АЛГОРИТМЕ ШОРА

В. С. Глухов
Национальный университет «Львовская политехника»

Аннотация. *В статье анализируются результаты нахождения периода r функции $y = a^x \bmod M$, которая используется в алгоритме факторизации Шора для квантовых компьютеров. По условиям данной задачи модуль M является произведением двух простых чисел p и q. В статье анализируются получаемые при различных a решения r, при которых емкостная сложность следующей после нахождения периода задачи, задачи нахождения наибольшего общего делителя НОД($a^{r/2}+1$,M), будет наименьшей.*

Ключевые слова: *цифровой квантовый сопроцессор, факторизация, алгоритм Шора, степень, модуль, НОД.*

Вступление

Квантовые компьютеры (КК) являются аналоговыми и вероятностными вычислительными устройствами [1]. Квантовые компьютеры состоят из кубитов, каждый из которых в любой момент времени может иметь какое-то значение x ($0 \le x \le 1$), которое с точно определенной вероятностью p может измеряться как 0, и с точно определенной вероятностью 1-p – как 1 (квантовая суперпозиция). В ходе вычисления каждый кубит может принимать любое значение в обозначенном диапазоне, но в процессе измерения конечного результата эта суперпозиция с указанной вероятностью обращается в одно из двух классических состояний, 0 или 1.

Критерий ДиВинченцо [2] не запрещает создавать цифровую версию квантового компьютера.

Цифровой квантовый компьютер, а именно, классический процессор и его цифровой квантовый сопроцессор, который может быть реализован на ПЛИС, описан в [3]. Цифровой квантовый сопроцессор работает под управлением классического компьютера, и вместе они образуют цифровой квантовый компьютер (рис. 1). Цифровой квантовый сопроцессор представляет собой набор цифровых блоков, называемых цифровыми кубитами, каждый из которых имеет многобитный вход и однобитный выход. В цифровом кубите имеется генератор псевдослучайных кодов (*PRNG*) для генерации выходного бита, который может принимать значение 0 или 1 с вычисленной вероятностью.

Для некоторых задач число кубитов в квантовом компьютере является полиномиальной функцией сложности задачи. Например, известны алгоритмы факторизации n-битного целого числа с использованием чуть более $2n$ кубитов (алгоритм Шора [4]). Предполагается, что для взлома шифра *RSA* с 2048-битным ключом необходимо не менее 4000 кубитов (но может быть больше в зависимости от используемого алгоритма [5]). Квантовая декогерентность, потребность работы со сверхнизкими температурами, большое потребление, а также ограничения на количество реализуемых межэлементных взаимодействий создают значительные проблемы при построении квантовых компьютеров.

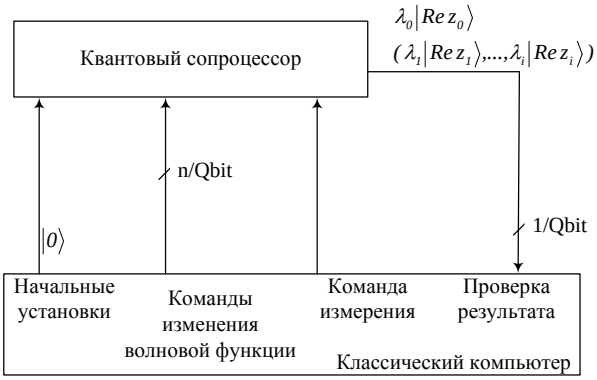

Рис. 1. Классический компьютер с квантовым сопроцессором

Однородные, неоднородные и смешанные квантовый сопроцессоры описаны в [6] и [7].

Цифровой квантовый сопроцесор с сотнями и тысячами цифровых кубит можно реализовать в одной программируемой логической интегральной схеме ПЛИС. Реализация квантового преобразования Фурье [**Ошибка!**

Источник ссылки не найден.] (одной из составляющих алгоритма Шора, QFT^{-1} на рис. 2) в таком квантовом сопроцессоре показана в [9], [10].

В алгоритме Шора задача разложения на множители числа M сводится к задаче определения периода r функции $y = a^x mod M$, которая вычисляется узлами CU (рис. 2), где a – произвольное целое. Показано, что наибольший общий делитель НОД($a^{r/2}$+1, M) может быть делителем числа M. На рис. 2 также изображены узлы H, которые выполняют преобразование Адамара.

1. Цель работы

При реализации алгоритма Шора задачей квантового сопроцессора является нахождения периода r функции $y = a^x mod M$.

Поскольку для произвольных a задача нахождения периода r имеет много решений, то в данной статье делается попытка проанализировать получаемые решения r исходя из сложности дальнейших вычислений, которые должен производить классический компьютер.

2. Емкостная сложность нахождения наибольшего общего делителя в алгоритме Шора

В процессах синтеза, анализа и оптимизации программно-аппаратных моделей (SH-моделей) предлагается использовать пять характеристик сложности: аппаратную, временную, емкостную, программную и структурную [12], [13], которые связаны друг с другом и зависят друг от друга.

После нахождения порядка r функции $y = a^x mod M$ необходимо найти наибольший общий делитель НОД($a^{r/2}$+1, M). Эту задачу выполняет классический компьютер. Поскольку для произвольных a задача нахождения периода r имеет много решений, эти решения можно сравнивать по величине одного из аргументов при нахождении НОД – числа $a^{r/2}$+1. Из-за того, что $a^{r/2} >> 1$, можно сравнивать величины $a^{r/2}$. Для уменьшения анализируемых результатов для анализа берется величина $H = \log_2 a^{r/2} = (r \log_2 a)/2$. Эту величину H можно назвать емкостной сложностью решения, она приближенно представляет разрядность двоичных кодов, которые придется обрабатывать классическому компьютеру при определении НОД($a^{r/2}$+1, M).

3. Исследования емкостной сложности нахождения НОД

Для анализа как a выбраны первые простые числа от 2 до 53 и составное число 6. Как p и q выбраны простые числа от 57 до 97. Величина исследуемой функции $y = a^x mod M$ ограничена ее максимальным при данном исследовании значением $y < 10001$.

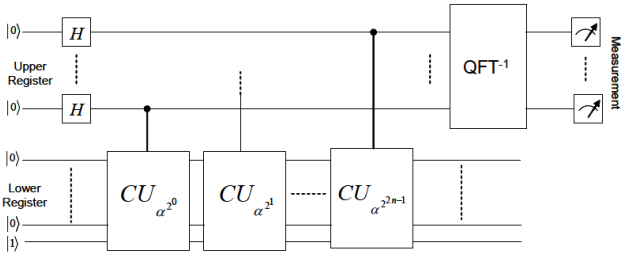

Рис. 2. Квантовый компьютер для квантовой факторизации по алгоритму Шора [11]

Результаты исследования представлены в виде графиков рис. 3 - рис. 12, на горизонтальной оси откладываются значения a, на вертикальной - H.

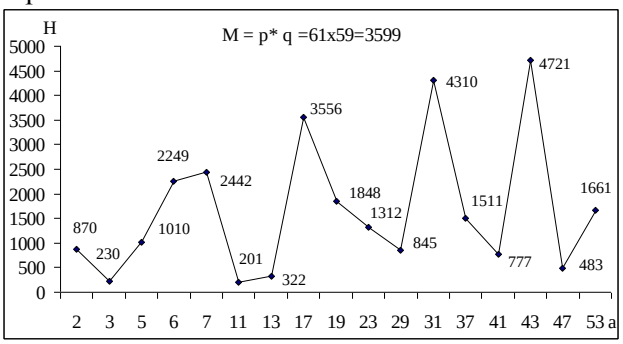

Рис. 3. Емкостная сложность, $M = p * q$ =61x59

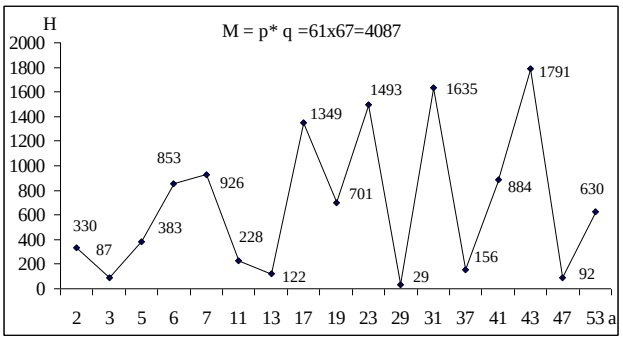

Рис. 4. Емкостная сложность, $M = p * q$ =61x67

Даже для небольших значений M видно, что количество двоичных разрядов H в записи числа $a^{r/2}$ может изменяться в широких пределах от десятков до тысяч бит.

Квантовый сопроцессор может выдать как любой правильный, так и неправильный

результаты. Задачей классического компьютера при роботе с квантовым сопроцессором является проверка правильности полученных результатов.

При выполнении алгоритма Шора дополнительной задачей классического компьютера должен быть выбор правильного решения с наименьшей емкостной сложностью *H*.

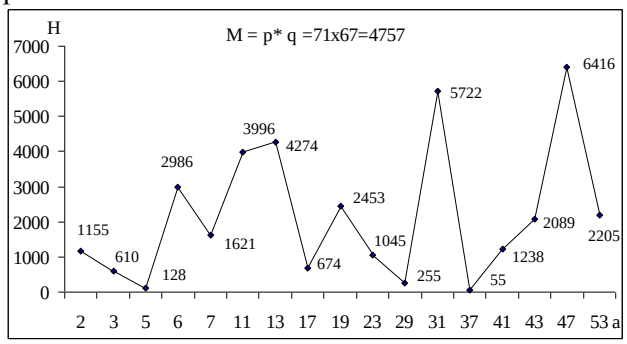

Рис. 5. Емкостная сложность,
$M = p * q$ =71x67

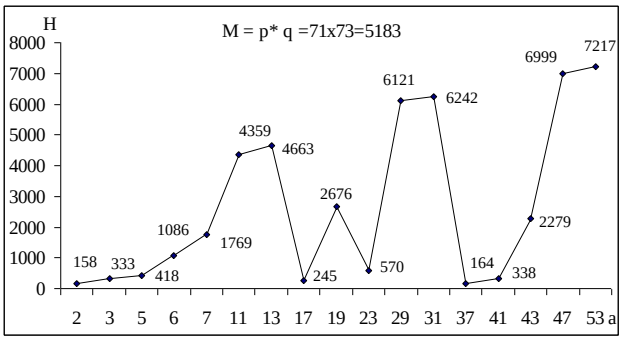

Рис. 6. Емкостная сложность,
$M = p * q$ =71x73

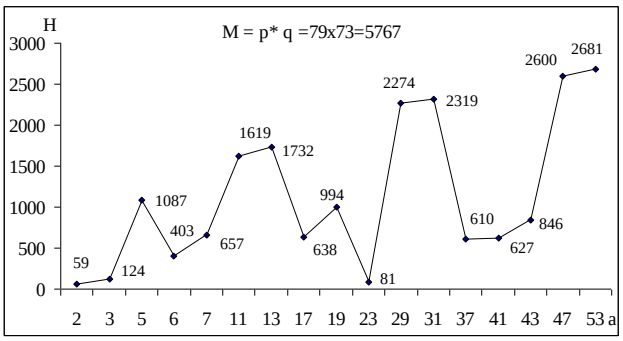

Рис. 7. Емкостная сложность,
$M = p * q$ =79x73

4. Оценка емкостной сложности нахождения НОД($a^{r/2}$+1, M)

Исследование даже такого небольшого количества небольших значений числа M дает возможность установить следующее.

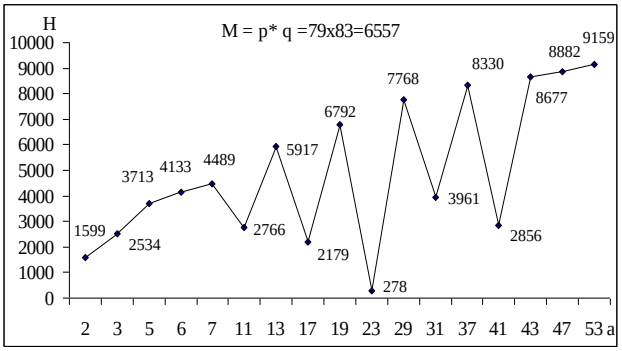

Рис. 8. Емкостная сложность,
$M = p * q$ =79x83

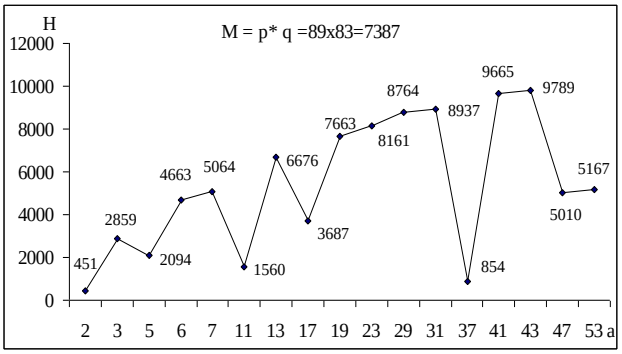

Рис. 9. Емкостная сложность,
$M = p * q$ =89x83

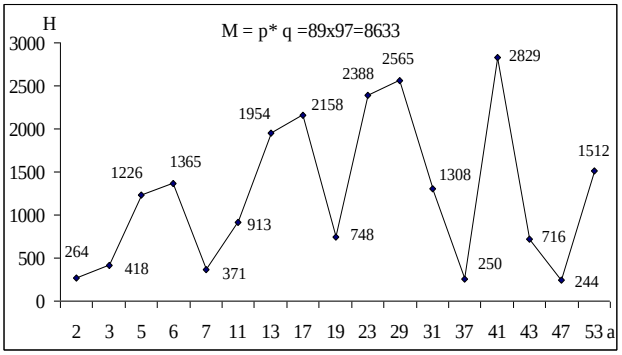

Рис. 10. Емкостная сложность,
$M = p * q$ =89x97

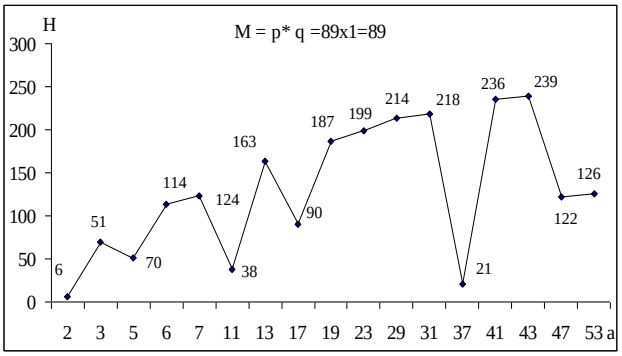

Рис. 11. Емкостная сложность,
$M = p * q$ =89x1

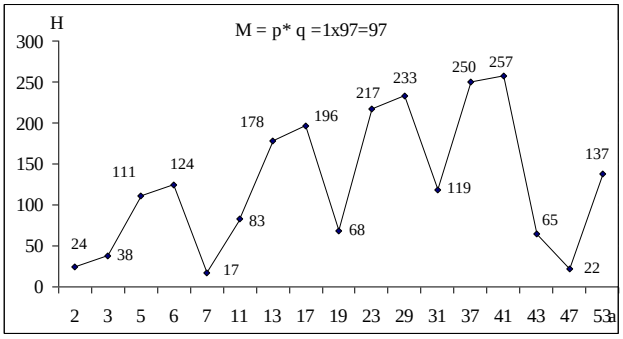

Рис. 12. Емкостная сложность,
$M = p * q$ =1x97

Период r функции $y = a^x modM$, который обеспечивает наименьшую сложность последующей задачи нахождения НОД чаще всего есть решением при a=2 (рис. 13).

Период r функции $y = a^x modM$, который обеспечивает одну из наименьших сложностей последующей задачи нахождения НОД чаще всего появляется при a = 3 и a = 2, но может появляться часто и при других значениях a, например, при a = 37 (рис. 14).

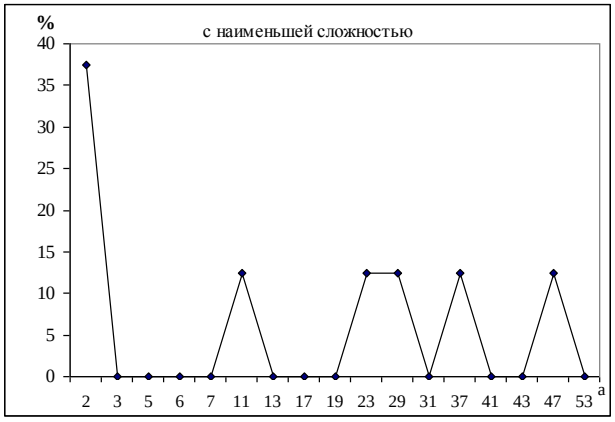

Рис. 13. Частота решения с наименьшей сложностью, %

Рис. 14. Частота решения среди 5 с наименьшей сложностью, %

При некоторых значениях a период r функции $y = a^x modM$, который обеспечивает наименьшую сложность последующей задачи нахождения НОД, может и не существовать, например, при a = 19, a = 31, a = 43, a = 53 (рис. 14).

Для уточнения выявленных закономерностей, особенно при больших M, необходимо проводить дополнительные исследования.

5. Зависимость емкостной сложности нахождения НОД от сомножителей числа M

На рис. 10 - рис. 12 представлены результаты определения сложности H при факторизации составного числа M = 89x97 и при факторизации его сомножителей p = 89 и q = 97 (про факторизацию в данном случае надо говорить в кавычках, поскольку p и q в данном случае – простые числа). Видно, что минимальные значения емкостной сложности H при M составном на рис. 10 определяются минимальными значениями сложности H при замене составного M на его простые сомножители p (рис. 11) и q (рис. 12).

Выводы

В статье сделана попытка проанализировать период r функции $y = a^x modM$ при реализации алгоритма факторизации Шора исходя из емкостной сложности дальнейших вычислений, которые должен производить классический компьютер.

Установлено что:

период r функции $y = a^x modM$, который обеспечивает наименьшую емкостную сложность последующей задачи нахождения НОД чаще всего есть решением при a=2;

период r функции $y = a^x modM$, который обеспечивает одну из наименьших емкостных сложностей последующей задачи нахождения НОД чаще всего появляется при a = 3 и a = 2, но может появляться часто и при других значениях a;

минимальные значения емкостной сложности H при M составном определяются минимальными значениями сложности H при замене составного M на его простые сомножители p и q.

Для уточнения выявленных закономерностей, особенно при больших M, необходимо проводить дополнительные исследования.

Список использованной литературы

1. V. Hlukhov. FPGA Based Digital Quantum Computer Verification [Text] / V. Hlukhov // Proceedings of IEEE 11th International Conference on Dependable Systems, Services and Technologies DESSERT. Ukraine, Kyiv, May 14-18, 2020, pp. 178–182

2. D. P. DiVincenzo. The physical implementation of quantum computation [Text] / D. DiVincenzo // Fortschritte der Physik: Progress of Physics 48 (9-11), 771–783. WILEY-VCH Verlag Berlin GmbH. 2000

3. Valerii Hlukhov, Bohdan Havano. FPGA-based Digital Quantum Coprocessor [Text] / V. Hlukhov, B. Havano // Advances in Cyber-Physical Systems. Volume 3. Number 2. Lviv Polytechnic National University. 2018. pp. 12-31

4. Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer [Text] / P. Shor // Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, Nov. 20–22, 1994, IEEE Computer Society Press, pp. 124–134

5. Applying Moore's Law to Quantum Qubits [Electronic resource] https://quantumcomputingreport.com/our-take/applying-moores-law-to-quantum-qubits/ Copyright © 2019 Quantum Computing Report, All rights reserved/ [Accessed 25 Nov. 2020]

6. Valerii Hlukhov. Comparison of Homogeneous and Heterogeneous Digital Quantum Coprocessors [Text] / V. Hlukhov // Proceedings of International Workshop on Computational Methods and Information Transformation Systems, a satellite of 2020 XV International Scientific and Technical Conference on COMPUTER SCIENCE AND INFORMATION TECHNOLOGIES. Zbarazh Castle, Ukraine, 23-26 September, 2020, pp. 74–77

7. Hlukhov, V. S., Hybrid quantum coprocessors [Text] / V. Hlukhov // IX International scientific and technical conference "Radioengineering Field, Signals, Devices and Systems". Kyiv. Ukraine. November 16 – 22, 2020 (in print)

8. Quantum Fourier transform. [Electronic resource]. *Wikipedia*. Available at: https://en.wikipedia.org/wiki/Quantum_Fourier_transform/ [Accessed 25 Nov. 2020]

9. M. Khalil-Hani, Y. H. Lee, M. N. Marsono. An accurate FPGA-based hardware emulation on quantum Fourier transform [Text] / M. Khalil-Hani, Y. H. Lee, M. N. Marsono // Proceedings of the 13th Australasian Symposium on Parallel and Distributed Computing (AusPDC 2015), Sydney, Australia, 27 – 30 January 2015, pp. 23–30

10. В. Глухов. К-кубитный цифровой квантовый сопроцессор на ПЛИС [Текст]. / В. Глухов // Електротехнічні та комп'ютерні системи. – Одеса : – 2019. Вид-во Наука і техніка. № 31 (107), с. 104–116

11. Archimedes Pavlidis, Dimitris Gizopoulos. Fast Quantum Modular Exponentiation Architecture for Shor's Factoring Algorithm [Text] / A. Pavlidis, D. Gizopoulos // Quantum Information and Computation, Vol. 14, No. 7&8 (2014) 0649–0682. © Rinton Press

12. Черкаський, М. В. SH-модель алгоритму [Текст] / М. Черкаський // Вісник Національного університету "Львівська політехніка" № 433. Видавництво Національного університету «Львівська політехніка». 2001. С.127–134

13. Черкаський, М. В., Хусейн Халід Мурад. Універсальна SH-модель [Текст] / М. Черкаський, Х. М. Хусейн // Вісник Національного університету "Львівська політехніка" № 523 «Комп'ютерні системи та мережі». Львів. Видавництво Національного університету «Львівська політехніка». 2004. С.150–154

References

1. Hlukhov, V. (2020). FPGA Based Digital Quantum Computer Verification. *IEEE 11th International Conference on Dependable Systems, Services and Technologies*, [online] Kyiv, pp. 178-182. Available at: https://ieeexplore.ieee.org/document/9125077/ [Accessed 24 Nov. 2020]

2. DiVincenzo, D. (2000). The physical implementation of quantum computation. Fortschritte der Physik: *Progress of Physics* 48 (9–11), [online] WILEY-VCH Verlag Berlin GmbH, pp. 771–783. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/1521-3978%28200009%2948%3A9/11<771%3A%3AAID-PROP771>3.0.CO%3B2-E/ [Accessed 25 Nov. 2020]

3. Hlukhov, V., Havano, B. (2018). FPGA-based Digital Quantum Coprocessor. *Advances in Cyber-Physical Systems*, (3), pp. 12-31

4. Shor, P. (1994). Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *35th Annual*

Symposium on Foundations of Computer Science, [online] Santa Fe, pp. 124–134. Available at: https://www.jstor.org/stable/2653075?seq=1/ [Accessed 25 Nov. 2020]

5. Applying Moore's Law to Quantum Qubits. (2019). *Quantum Computing Report*. Avaliable at: https://quantumcomputingreport.com/our-take/applying-moores-law-to-quantum-qubits/ [Accessed 25 Nov. 2020]

6. Hlukhov, V. (2020). Comparison of Homogeneous and Heterogeneous Digital Quantum Coprocessors. *International Workshop on Computational Methods and Information Transformation Systems*, a satellite of *2020 XV International Scientific and Technical Conference on Computer Science and Information Technologies*, Zbarazh Castle, pp. 74–77

7. Hlukhov, V. (2020). Hybrid quantum coprocessors. *IX International scientific and technical conference "Radioengineering Field, Signals, Devices and Systems"*. Kyiv (in print)

8. Quantum Fourier transform. (2020). *Wikipedia*. Available at: https://en.wikipedia.org/wiki/Quantum_Fourier_transform/ [Accessed 25 Nov. 2020]

9. Khalil-Hani, M., Lee, Y. and Marsono. M. (2015). An accurate FPGA-based hardware emulation on quantum Fourier transform. *13th Australasian Symposium on Parallel and Distributed Computing*. Sydney, Australia, pp. 23–30

10. Hlukhov, V. (2020). FPGA-based K-Qubit Digital Quantum Coprocessor [K-kubitnyj cifrovoj kvantovyj soprocessor na PLIS]. *Electrotechnic and Computer Systems*, 31(107), pp. 104–116

11. Pavlidis, A. and Gizopoulos, D. (2014) Fast Quantum Modular Exponentiation Architecture for Shor's Factoring Algorithm. *Quantum Information and Computation*, 14, pp. 0649–0682

12. Cherkaskyi, M. (2001). SH-model of algorithm [SH-model algoritmu]. *Bulletin of the Lviv Polytechnic National University "Computer systems and networks"*, 433, pp. 127–134

13. Cherkaskyi, M. and Khusein, Kh. (2004). Universal SH-model [Universalna SH-model]. // *Bulletin of the Lviv Polytechnic National University "Computer systems and networks"*, 523, pp.150–154

CAPACITIVE COMPLEXITY OF DETERMINING GCD IN THE SHOR'S ALGORITHM

V. Hlukhov
Lviv Polytechnic National University

Abstract. *The article analyzes the results of finding the period r of the function $y = a^x \bmod M$ (a is a random number) which is used in the Shor's factorization algorithm for quantum computers. The module M is the product of two primes p and q. The article analyzes the solutions r obtained for various a, for which the capacitive complexity H of finding the greatest common divisor $GCD(a^{r/2} + 1, M)$ is the least.*

A digital quantum computer is a classic processor and its digital quantum coprocessor. A digital quantum coprocessor with hundreds and thousands of digital qubits can be implemented in one programmable logic integrated circuit FPGA. In the Shor's algorithm, the factorization problem of the number M reduces to the problem of determining the period r of the function y. It is known that $GCD(a^{r/2} + 1, M)$ can be a divisor of the number M

The task of the quantum coprocessor in implementing the Shor's algorithm is to find the period r. After that it is necessary to find the GCD. Since for random a the problem of finding the period r has many solutions, these solutions can be compared by the value of one of the arguments when finding the GCD - the number $a^{r/2}$. In this case, $H = (r\log_2 a)/2$ is taken for analysis. It approximately represents the bit depth of binary codes that a classic computer will have to process when determining the GCD.

H can vary over a wide range from tens to thousands of bits even for small values of M. In this research the period r, which ensures the least complexity of the subsequent task of finding the GCD, is most often a solution for a = 3 and a = 2, but it can also occur often with other values of a. To clarify the revealed patterns, especially for large M, it is necessary to conduct additional research.

Keywords*: digital quantum coprocessor, factorization, Shor's algorithm, power, module, GCD.*

ЄМНІСНА СКЛАДНІСТЬ ВИЗНАЧЕННЯ НСД В АЛГОРИТМІ ШОРА

В. С. Глухов
Національний університет «Львівська політехніка»

Анотація. *У статті проаналізовано результати пошуку періоду r функції $y = a^x \bmod M$ (a - випадкове число), яка використовується в алгоритмі факторизації Шора для квантових комп'ютерів. Модуль M є добутком двох простих чисел p і q. У статті проаналізовано розв'язки r, отримані для різних a, для яких ємнісна складність H знаходження найбільшого спільного дільника $НСД(a^{r/2} + 1, M)$ є найменшою.*

Цифровий квантовий комп'ютер - це класичний процесор та його цифровий квантовий копроцесор. Цифровий квантовий копроцесор з сотнями і тисячами цифрових кубітів може бути реалізований в одній програмованій логічній інтегральній схемі ПЛІС. В алгоритмі Шора задача факторизації числа M зводиться до задачі визначення періоду r функції y. Відомо, що $НСД(a^{r/2} + 1, M)$ може бути дільником числа M

Завданням квантового копроцесора при реалізації алгоритму Шора є пошук періоду r. Після цього необхідно знайти НСД. Оскільки для випадкового a проблема пошуку періоду r має багато рішень, ці рішення можна порівняти за значенням одного з аргументів при знаходженні НСД – числа $a^{r}/^{2}$. При цьому аналізується $H = (r\log_2 a)/2$. Він приблизно представляє розрядність двійкових кодів, які класичний комп'ютер повинен буде обробляти при визначенні НСД.

H може змінюватися в широкому діапазоні від десятків до тисяч біт навіть при малих значеннях M. У цьому дослідженні період r, який забезпечує найменшу складність подальшого завдання пошуку НСД, найчастіше є рішенням для a = 3 і a = 2, але він може часто траплятися і для інших значеннь a. Для уточнення виявлених закономірностей, особливо для великих M, необхідно провести додаткові дослідження.

Ключові слова: *цифровий квантовий копроцесор, факторизація, алгоритм Шора, степінь, модуль, НСД.*

Получено 03.04.2020

Глухов Валерий Сергеевич, доктор технических наук, профессор, профессор кафедры электронных вычислительных машин Национального университета «Львовская политехника», ул. С. Бандеры, 12, Львов, Украина, E-mail: glukhov@polynet.lviv.ua , м/т.: +38-063-75-72-330.

Valeriy Hlukhov, Dr. of Science, Professor, Professor of the Department of Computer Engineering, Lviv Polytechnic National University, S. Bandera Str., 12, Lviv, Ukraine, E-mail: glukhov@polynet.lviv.ua, м/т.: +38-063-75-72-330.

ORCID ID:0000-0002-0542-7447