

Моделирование квантовых алгоритмов принятия решений на классическом симуляторе: визуализация квантовой программной инженерии в образовательном процессе

В. С. Ульянов¹, С. В. Ульянов^{2,3*}

¹ ФГБОУ ВО «Московский государственный университет геодезии и картографии», г. Москва, Российская Федерация

Адрес: 105064, Российская Федерация, г. Москва, Гороховский пер., д. 4

² ГБОУ ВО Московской области «Университет «Дубна», г. Дубна, Российская Федерация

Адрес: 141982, Российская Федерация, Московская область, г. Дубна, ул. Университетская, д. 19

³ Международная межправительственная организация Объединенный институт ядерных исследований, г. Дубна, Российская Федерация

Адрес: 141980, Российская Федерация, Московская область, г. Дубна, ул. Жолио-Кюри, д. 6

*ulyanovsv46_46@mail.ru

Аннотация

Отличие квантового от классического алгоритма заключается в следующем: задача, решаемая квантовым алгоритмом (КА), закодирована в структуре квантовых операторов, применяемых к входному сигналу. Вид входного сигнала в структуру КА в этом случае всегда один и тот же. Выходной сигнал КА включает в себя информацию о решении закодированной проблемы. В результате КА задается функция для анализа, и КА определяет ее свойство в виде ответа без количественных вычислений, т. е. КА изучает качественные свойства функций. Ядром любого КА является набор унитарных квантовых операторов или квантовых вентилях. На практике квантовый вентиль представляет собой унитарную матрицу с определенной структурой. Размер этой матрицы растет экспоненциально с увеличением количества входных данных, что существенно ограничивает моделирование КА на классическом компьютере с фон-неймановской архитектурой. Модели квантовых поисковых алгоритмов применяются для решения задач информатики, таких как поиск в неструктурированной базе данных, квантовая криптография, инженерные задачи, проектирование систем управления, робототехника, интеллектуальные контроллеры и т. д. Алгоритм Deutsch относится к классу алгоритмов принятия решений и подробно объясняется вместе с реализациями на локальном компьютерном симуляторе. В представленной статье описывается практический подход к моделированию двух из самых известных КА принятия решения на классических компьютерах: Deutsch и Deutsch – Jozsa алгоритмов.

Ключевые слова: алгоритм принятия решения, квантовые схемы, квантовые алгоритмические вентилях, синтез квантовых операторов

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Ульянов В. С., Ульянов С. В. Моделирование квантовых алгоритмов принятия решений на классическом симуляторе: визуализация квантовой программной инженерии в образовательном процессе // Современные информационные технологии и ИТ-образование. 2023. Т. 19, № 2. С. 298-323. doi: <https://doi.org/10.25559/SITITO.019.202302.298-323>

© Ульянов В. С., Ульянов С. В., 2023



Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.



Quantum Decision-Making Algorithms Modeling in the Classical Simulator: the Quantum Software Engineering Visualization for the Educational Process

V. S. Ulyanov^a, S. V. Ulyanov^{b,c,*}

^a Moscow State University of Geodesy and Cartography, Moscow, Russian Federation

Address: 4, Gorokhovskiy per., Moscow 105064, Russian Federation

^b Dubna State University, Dubna, Russian Federation

Address: 19 Universitetskaya St., Dubna 141980, Moscow Region, Russian Federation

^c Joint Institute for Nuclear Research, Dubna, Russian Federation

Address: 6 Joliot-Curie St., Dubna 141980, Moscow region, Russian Federation

* ulyanovsv46_46@mail.ru

Abstract

The difference between classical and quantum algorithms (QA) is as follows: the problem solved by QA is coded in the structure of the quantum operators. Input to QA in this case is always the same. Output of QA says which problem coded. In some sense, give a function to QA to analyze and QA returns its property as an answer without quantitative computing. QA studies qualitative properties of the functions. The core of any QA is a set of unitary quantum operators or quantum gates. In practical representation, quantum gate is a unitary matrix with particular structure. The size of this matrix grows exponentially with an increase in the number of inputs, which significantly limits the QA simulation on a classical computer with von Neumann architecture. Quantum search algorithm (QSA) – models apply for the solution of computer science problems as searching in unstructured database, quantum cryptography, engineering tasks, control system design, robotics, smart controllers, etc. Grover's algorithm is explained in details, along with implementations on a local computer simulator. The presented article describes a practical approach to modeling one of the most famous QA on classical computers, the Grover algorithm.

Keywords: quantum search algorithms, quantum circuits, quantum algorithmic gate, quantum operators design

Conflict of interests: The authors declare no conflict of interests.

For citation: Ulyanov V. S., Ulyanov S. V. Quantum Decision-Making Algorithms Modeling in the Classical Simulator: the Quantum Software Engineering Visualization for the Educational Process. *Modern Information Technologies and IT-Education*. 2023;19(2):298-323. doi: <https://doi.org/10.25559/SITITO.019.202302.298-323>



Введение

Переход на исследование, разработку и внедрение квантовых ИТ на современном этапе оказался крайне трудным для системной и программной инженерии, особенно для ИТ-поколений, начиная с середины 90-х. Прежде всего данная проблема связана с отсутствием необходимых знаний и опыта работа с подобными наукоемкими ИТ. Подробный анализ [1] сложившейся к 2023 г. ситуации показал еще больший разрыв и отставание отечественных разработок от современных разработок. Одним из направлений успешного развития является квантовая программная инженерия на основе квантовых вычислений, квантовых алгоритмов и квантового программирования, имеющих уже инженерные и коммерческие приложения¹ [2-5].

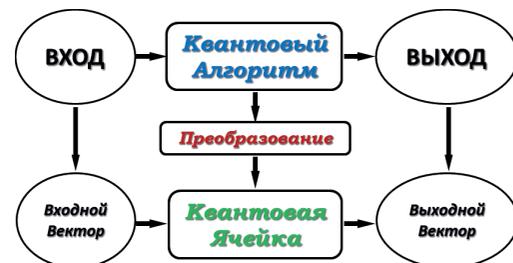
Примечание. Первая квантовая революция породила понятие квантового бита и квантовых вычислений. Вторая квантовая революция привела к необходимости разработки квантового компьютера, способного ускорить решения исследуемой задачи и решать алгоритмически неразрешимые (классическими алгоритмами) задачи. Квантовая программная инженерия представляет начало третьей квантовой революции [6], опирается на методы квантового программирования и квантовые симуляторы квантовых алгоритмов (КА) принятия решения или квантового поиска² [2-4].

Цель работы. Целью данной работы является демонстрация эффективного моделирования КА принятия решений, представленного в форме квантовой схемы, через разработанную ИТ проектирования и преобразования автоматически в соответствующую программируемую квантовую алгоритмическую ячейку (КАЯ) (quantum algorithmic gate (QAG)). Такой подход позволяет объяснить наглядно роль квантовых логических «парадоксов» [7, 8] при моделировании КА в учебных и образовательных процессах, а также сформировать когнитивное восприятие истинности суждения о непротиворечивости семантики нетрадиционного результата принятия решения через наглядную визуализацию [1], [8-14] результатов работы КА. В данной работе на доступном уровне рассматривается метод проектирования квантовых алгоритмических ячеек (КАЯ) и структура симулятора КА на примере алгоритмов принятия решений Deutsch и Deutsch — Jozsa. Моделированием на классическом компьютере показан квантовый эффект принятия правильного решения с уверенностью 50 %. Данный эффект имеет квантовую природу и в классических алгоритмах принятия решения аналога не имеет. Приведена математически корректная и физически строгая интерпретация квантовых операторов суперпозиции, запутанных состояний и интерференции на языке программной инженерии и логической непротиворечивости результатов измерения (toss coin) КА принятия решений в условиях эффективного случайного поиска решений.

Особенности подхода математического моделирования КА.

Математической моделью, данной КАЯ, является матричный оператор. При применении векторного представления квантового регистра входного состояния КАЯ полученный выходной результат будет являться векторным представлением нужного состояния выходного регистра (см. рис. 1).

Основой квантовых вычислений являются три оператора, действующих на квантовые когерентные состояния: Суперпозиция, Квантовая корреляция (Запутанные состояния) и Интерференция. Когерентными являются такие состояния, эволюция которых является решениями соответствующих уравнений Шредингера, с минимальной неопределенностью (в смысле Гейзенберга это состояния с «максимально классическими свойствами»). Преобразование Адамара создает суперпозицию классических состояний. Квантовые операторы, такие как контролируемое отрицание (controlled NOT – CNOT), создают робастные связанные состояния. Квантовое преобразование Фурье осуществляет интерференцию. Эффективное применение набора операций, необходимых для квантовых вычислений, включает также фазу настройки амплитуд суперпозиций, перестановок, аппроксимацию матриц преобразований и обобщение коррекции фаз блока матричных преобразований [9-15]. Эти операции обобщают все необходимые действия, применяемые в квантовых поисковых алгоритмах и алгоритмах принятия решений, реализуемых на классических компьютерах³ [12], [15-17].



Р и с. 1. Блочный подход к моделированию квантовых алгоритмов на классических компьютерах

Fig. 1. The gate approach for simulation of quantum algorithms using classical computers

Источник: здесь и далее в статье все рисунки составлены авторами.

Source: Hereinafter in this article all figures were made by the authors.

В данной работе демонстрируется применение данного подхода к моделированию КА на классических компьютерах. КА определяют свойство исследуемой функции, минуя количественные вычисления. В качестве примера вычислений рассмотрено моделирование КА Deutsch и Deutsch — Jozsa. В проблеме Deutsch классический алгоритм определения качественного свойства исследуемой функции требует четырех действий, и по количественному результату определяется искомое свойство. КА определяет искомое свойство за одну итерацию.

¹ Ivancova O. V., Korenkov V. V., Ulyanov S. V. Quantum Software Engineering. Quantum supremacy modelling. Part I: Design IT and information analysis of quantum algorithms: Educational and methodical textbook. Textbook. Moscow : KURS, 2020. 328 p.; Ivancova O. V., Korenkov V. V., Ulyanov S. V. Quantum Software Engineering. Quantum supremacy modelling. Part II: Quantum search algorithms simulator – computational intelligence toolkit: Educational and methodical textbook. Moscow : KURS, 2020. 344 p.

² Там же.

³ Ulyanov S. V. System and Method for Control Using Quantum Soft Computing. United States Patent US-6578018B1. USA, 2003. URL: <https://patents.google.com/patent/US6578018B1/en> (дата обращения: 01.04.2023).



рацию, и для определения достоверного ответа квантового Deutsch-алгоритма требуется вероятность измерения результата 0.5. Данный эффект носит чисто квантовый характер и классического аналога не имеет. *Примечание.* Рассмотренный подход моделирования КА на симуляторах с классической архитектурой существенно отличается от методов моделирования, обсуждаемых в [16], [18-20].

Рассмотрим конкретный пример проектирования КАЯ для алгоритма Deutsch и его моделирование на компьютере с классической архитектурой.

1. Моделирование на классическом компьютере квантового алгоритма принятия решения Дойча

Сформулируем задачу Дойча [9-12].

1.1. Проблема Дойча

Входом КА всегда является бинарная функция f . Эта функция представляется в качестве отображения, определяющего изображение каждой входной бинарной строки. Функция $f: \{0,1\} \rightarrow \{0,1\}$ является постоянной, тогда и только тогда, когда $\exists y \in \{0,1\}: \forall x \in \{0,1\}: f(x) = y$;

- Функция $f: \{0,1\} \rightarrow \{0,1\}$ является балансирующей, если $|\{x \in \{0,1\}: f(x) = 0\}| = |\{x \in \{0,1\}: f(x) = 1\}|$.

Проблема Дойча относится к классу задач принятия решения и формулируется следующим образом:

Вход	Постоянная или балансирующая функция f
Проблема	Решить, к какому классу функция f относится

Мы можем различить 4 возможных варианта функций $f_i: \{0,1\} \rightarrow \{0,1\}$, которые могут быть заданы с помощью следующих отображений:

Постоянные функции			
x	$f_1(x)$	x	$f_2(x)$
0	0	0	1
1	0	1	1

Балансирующие функции			
x	$f_3(x)$	x	$f_4(x)$
0	0	0	1
1	1	1	0



Р и с. 2. Схематическая диаграмма квантовых алгоритмов
F i g. 2. Schematic diagram of Quantum Algorithms

Набор $\{f_i\}_{i \in \{1,2,3,4\}}$ является набором входов алгоритма. Любая функция f_i представляется в виде соответствующего отображения. На структурном уровне квантовый алгоритм представлен на рис. 2.

Отметим, что схема на рис. 2 является общей для проектирования КА [3-5], [12], [15].

1.2. Кодирование

Блок кодирования преобразует функцию f в унитарную матрицу U_F , которая содержит искомые свойства исследуемой функции. Для примера рассмотрим кодирование в случае $f = f_3$. В данном случае мы имеем следующее отображение:

x	$f(x)$
0	0
1	1

Шаг 1: Функция f преобразуется в обратимую функцию $F: \{0,1\}^2 \rightarrow \{0,1\}^2$, такую, что $F(x_0, y_0) = (x_0, f(x_0) \oplus y_0)$, или в логическом представлении:

y_0	$F(x_0, y_0)$
0	$(x_0, f(x_0))$
1	$(x_0, \neg f(x_0))$

Замечание. Оператор отрицания « \neg », действующий на бинарную строку, меняет значения каждого бита исходной строки на противоположные:

$$p = (p_0, \dots, p_{n-1})$$

$$\neg p = ((p_0 + 1) \bmod 2, \dots, (p_{n-1} + 1) \bmod 2)$$

Следовательно, если $f = f_3$ функция F определяется как:

(x_0, y_0)	$F(x_0, y_0)$
(0,0)	(0,0)
(0,1)	(0,1)
(1,0)	(1,1)
(1,1)	(1,0)

Шаг 2: Таблица отображений F преобразуется в таблицу отображений U_F :

$$\forall s \in \{0,1\}^2: U_F[\tau(s)] = \tau[F(s)]$$

Таким образом, отображением U_F является:

$ x_0 y_0\rangle$	$U_F x_0 y_0\rangle$
$ 00\rangle$	$ 00\rangle$
$ 01\rangle$	$ 01\rangle$
$ 10\rangle$	$ 11\rangle$
$ 11\rangle$	$ 10\rangle$

В представлении базисных векторов как вектор столбцов:

v	$U_F v$
$(1,0,0,0)^T$	$(1,0,0,0)^T$
$(0,1,0,0)^T$	$(0,1,0,0)^T$
$(0,0,1,0)^T$	$(0,0,0,1)^T$
$(0,0,0,1)^T$	$(0,0,1,0)^T$



Замечание. Оператор транспонирования, действующий на вектор-столбец (строку), преобразует вектор в соответствующую строку (столбец):

$$\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}^T = (x_1, \dots, x_n); \quad (x_1, \dots, x_n)^T = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}.$$

Шаг 3: Унитарная матрица, ассоциированная с полученным отображением U_F , строится путем перестановки рядов матрицы истинности размерности 4×4 : Первый и второй ряды отображаются сами в себя, третий и четвертый меняются местами:

$$U_F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Замечание. Общий способ построения матрицы U_F заключается в выражении каждого вектора $U_F(|s\rangle)$ как линейной комбинации всех базисных векторов. Координатами подобных базисных векторов являются все нули, за исключением одного базисного вектора, соответствующего изображению $|s\rangle$ под действием U_F :

$$U_F|00\rangle = 1|00\rangle + 0|01\rangle + 0|10\rangle + 0|11\rangle$$

$$U_F|01\rangle = 0|00\rangle + 1|01\rangle + 0|10\rangle + 0|11\rangle$$

$$U_F|10\rangle = 0|00\rangle + 0|01\rangle + 0|10\rangle + 1|11\rangle$$

$$U_F|11\rangle = 0|00\rangle + 0|01\rangle + 1|10\rangle + 0|11\rangle$$

Мы вычисляем элементы $[U_F]_{ij}$ как координаты вектора $U_F|j\rangle$ относительно вектора $|i\rangle$, где $|i\rangle, |j\rangle$ являются бинарными последовательностями. Это означает:

$$[U_F]_{ij} = 1 \Leftrightarrow U_F|j\rangle = |i\rangle.$$

Значения $[U_F]_{ij}$ называются *амплитудами вероятности* вектора $|j\rangle$, который может быть преобразован в вектор $|i\rangle$

под действием оператора U_F .

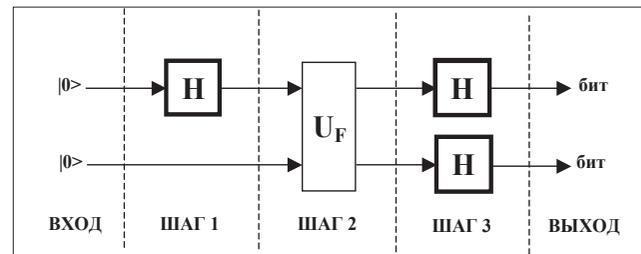
Например, амплитуда вероятности отображения вектора $|00\rangle$ в вектор $|00\rangle$ под действием оператора U_F равна 1, т.к. $U_F|00\rangle = 1|00\rangle$. В тоже время амплитуда вероятности отображения $|00\rangle$ в $|01\rangle$ равна 0, т.к. $U_F|00\rangle = 0|01\rangle$.

Следуя данному подходу, можно построить таблицу:

U_F	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	1	0	0	0
$ 01\rangle$	0	1	0	0
$ 10\rangle$	0	0	1	
$ 11\rangle$	0	0	1	0

1.3. Квантовый блок

Полученная после кодирования матрица U_F внедряется в квантовую схему, действующую на входное состояние $|00\rangle$. Данная схема выражается в виде следующей диаграммы (см. рис. 3):



Р и с. 3. Схематическое представление Deutsch-алгоритма
Fig 3. The structure of the Deutsch algorithm

Каждый тонкий квадрат означает классический матричный оператор размерности $n \times n$, где n - число линий, входящих и выходящих из прямоугольника. Матричный оператор называется классическим, если он отображает один базисный вектор в другой базисный вектор. Например, оператор U_F является классическим.

Примечание. Отметим некоторые характерные свойства матрицы Адамара H , которые используются при построении КАЯ квантового Deutsch-алгоритма. Обозначим кэт-векторы:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{и} \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Действие матрицы Адамара на кэт-векторы приводит к следующему результату:

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle);$$

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$



Таким образом, матричный оператор Адамара H обладает неклассическими свойствами:

- 1) из классических состояний $(|0\rangle, |1\rangle)$ (первый шаг, блок H на рис. 3) порождает неклассическое состояние «квантовая суперпозиция» и извлекает новые состояния $(|1\rangle, |0\rangle)$, скрытые в исходных состояниях (противоречия аксиоматике Клини — Колмогорова: действие классического объекта на классический порождает классический объект);
- 2) порождает неклассическое (по отношению к интерпретации в аксиоматике вероятности по Колмогорову) понятие «амплитуда» вероятности с положительным и отрицательным значениями;
- 3) в результате действия матричного оператора Адамара H из состояния в один бит получен кубит (кот Шредингера);
- 4) матричный оператор Адамара H порождает новое явление — «конструктивная» и «деструктивная» интерференция:

$$H \cdot \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \frac{1}{\sqrt{2}} \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) = \frac{1}{2} \left(\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right) = |0\rangle'$$

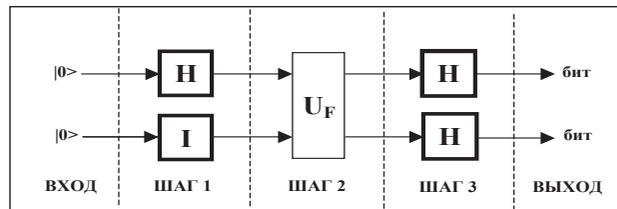
усиливая амплитуду вероятности одного состояния $|0\rangle$ и подавляя амплитуду вероятности другого состояния $|1\rangle$ (см. Шаг 3 на рис. 3).

Толстые прямоугольники на рис. 3 означают неклассические операторы. Неклассические операторы отображают (по крайней мере) один из базисных векторов в суперпозицию базисных векторов.

Диаграмма, представленная на рис. 3, может быть преобразована в соответствующую КАЯ. Исходная схема должна быть дополнена недостающими операторами, например шаг 1 на рис. 3, вторая линия означает, что на данном шаге второй базисный вектор остался неизменным. Мы говорим, что на данный вектор подействовал матричный оператор идентичности I . Таким образом, мы завершаем схему. Оператор идентичности является классическим и определяется следующим образом:

I	$ 0\rangle$	$ 1\rangle$
$ 0\rangle$	1	0
$ 1\rangle$	0	1

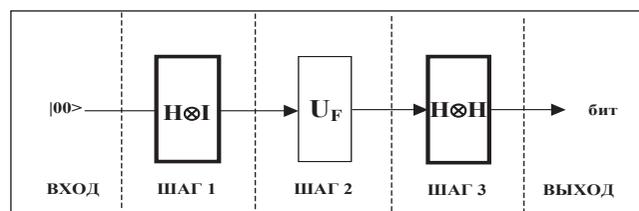
Далее мы должны построить матричный оператор, соответствующий каждому шагу, действие которого соответствует текущему действию матричных операторов на параллельные линии. В результате получаем схему на рис. 4:



Р и с. 4. Квантовый Deutsch-алгоритм (Второе схематическое представление)

F i g. 4. The Quantum Deutsch algorithm (Second schematic variant)

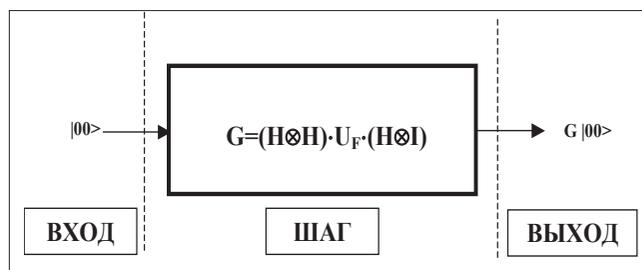
Наконец, можно построить уникальный матричный оператор, действие которого эквивалентно последовательному применению операторов, задействованных в алгоритме (см. рис. 5).



Р и с. 5. Квантовый Deutsch-алгоритм (Третье схематическое представление)

F i g. 5. The Quantum Deutsch algorithm (Third schematic variant)

Подобный оператор является композицией составляющих операторов, полученной путем матричного произведения матриц операторов, выполненного в обратном порядке применения операторов. В итоге мы получаем схему, изображенную на рис. 6.



Р и с. 6. Квантовый Deutsch-алгоритм (Окончательное представление)

F i g. 6. The Quantum Deutsch algorithm (Final schematic variant)

Схема на рис. 6 представляет собой программируемую ячейку для реализации Deutsch-алгоритма.



1.4. Формирование КАЯ (QAG)

Рассмотрим кратко алгоритм проектирования КАЯ для Deutsch-алгоритма:

Вычисляем $(H \otimes I)$. Получаем матрицу, размерности 4×4 . Присваиваем строкам и столбцам полученной матрицы соответствующие базисные векторы. Амплитуды вероятностей отображения одного базисного вектора в другой под действием оператора $(H \otimes I)$ могут быть получены на пересечении соответствующих строк и столбцов. Для примера, вектор $|00\rangle$; его амплитуда вероятности отображения в $|01\rangle$ есть произведение между амплитудой вероятности $|0\rangle$ отобразиться в $|0\rangle$ под действием оператора H на амплитуду вероятности $|0\rangle$ отобразиться в $|1\rangle$ под действием оператора I

Следовательно,

I	$ 0\rangle$	$ 1\rangle$	H	$ 0\rangle$	$ 1\rangle$
$ 0\rangle$	1	0	$ 0\rangle$	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$
$ 1\rangle$	0	1	$ 1\rangle$	$\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$

Мы можем автоматически вычислить операторы $H \otimes I$ и $H \otimes H$:

$H \otimes I$	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	$\frac{1}{\sqrt{2}}$	0	$\frac{1}{\sqrt{2}}$	0
$ 01\rangle$	0	$\frac{1}{\sqrt{2}}$	0	$\frac{1}{\sqrt{2}}$
$ 10\rangle$	$\frac{1}{\sqrt{2}}$	0	$-\frac{1}{\sqrt{2}}$	0
$ 11\rangle$	0	$\frac{1}{\sqrt{2}}$	0	$-\frac{1}{\sqrt{2}}$
$H \otimes H$	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
$ 01\rangle$	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$
$ 10\rangle$	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$
$ 11\rangle$	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$

Оператор U_F для случая $f = f_3$:

U_{F_3}	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	1	0	0	0
$ 01\rangle$	0	1	0	0
$ 10\rangle$	0	0	0	1
$ 11\rangle$	0	0	1	0

Финальная программируемая ячейка

$$G_3 = (H \otimes H) \cdot (U_{F_3} \cdot (H \otimes I))$$

получается следующим образом:

$U_{F_3} \cdot (H \otimes I)$	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	$\frac{1}{\sqrt{2}}$	0	$\frac{1}{\sqrt{2}}$	0
$ 01\rangle$	0	$\frac{1}{\sqrt{2}}$	0	$\frac{1}{\sqrt{2}}$
$ 10\rangle$	0	$\frac{1}{\sqrt{2}}$	0	$-\frac{1}{\sqrt{2}}$
$ 11\rangle$	$\frac{1}{\sqrt{2}}$	0	$-\frac{1}{\sqrt{2}}$	0

G_3	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	$\frac{1}{2}$	$\frac{1}{2}$	0	0
$ 01\rangle$	0	0	$\frac{1}{2}$	$-\frac{1}{2}$
$ 10\rangle$	0	0	$\frac{1}{2}$	$\frac{1}{2}$
$ 11\rangle$	$\frac{1}{2}$	$-\frac{1}{2}$	0	0

Для остальных вариантов исходных функций, квантовые ячейки вычисляются соответственно:



x	$f_1(x)$
0	0
1	0

x	$f_4(x)$
0	1
1	0

x	$f_2(x)$
0	1
1	1

(x_0, y_0)	$F_1(x_0, y_0)$
(0,0)	(0,0)
(0,1)	(0,1)
(1,0)	(1,0)
(1,1)	(1,1)

(x_0, y_0)	$F_4(x_0, y_0)$
(0,0)	(0,1)
(0,1)	(0,0)
(1,0)	(1,0)
(1,1)	(1,1)

(x_0, y_0)	$F_2(x_0, y_0)$
(0,0)	(0,1)
(0,1)	(0,0)
(1,0)	(1,1)
(1,1)	(1,0)

Для каждого логического отображения легко строятся соответствующие матричные операторы:

$ x_0 y_0\rangle$	$U_{F_1} x_0 y_0\rangle$
$ 00\rangle$	$ 00\rangle$
$ 01\rangle$	$ 01\rangle$
$ 10\rangle$	$ 10\rangle$
$ 11\rangle$	$ 11\rangle$

U_{F_1}	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	1	0	0	0
$ 01\rangle$	0	1	0	0
$ 10\rangle$	0	0	1	0
$ 11\rangle$	0	0	0	1

$ x_0 y_0\rangle$	$U_{F_2} x_0 y_0\rangle$
$ 00\rangle$	$ 01\rangle$
$ 01\rangle$	$ 00\rangle$
$ 10\rangle$	$ 11\rangle$
$ 11\rangle$	$ 10\rangle$

U_{F_2}	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	0	1	0	0
$ 01\rangle$	1	0	0	0
$ 10\rangle$	0	0	0	1
$ 11\rangle$	0	0	1	0

$ x_0 y_0\rangle$	$U_{F_4} x_0 y_0\rangle$
$ 00\rangle$	$ 01\rangle$
$ 01\rangle$	$ 00\rangle$
$ 10\rangle$	$ 10\rangle$
$ 11\rangle$	$ 11\rangle$

U_{F_4}	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	0	1	0	0
$ 01\rangle$	1	0	0	0
$ 10\rangle$	0	0	1	0
$ 11\rangle$	0	0	0	1



Различные U_{F_i} ($i = 1,2,4$) генерируют различные программируемые ячейки

$$G_i = (H \otimes H) \cdot (U_{F_i} \cdot (H \otimes I)):$$

G_1	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	$1/2^{1/2}$	$1/2^{1/2}$	0	0
$ 01\rangle$	$1/2^{1/2}$	$-1/2^{1/2}$	0	0
$ 10\rangle$	0	0	$1/2^{1/2}$	$1/2^{1/2}$
$ 11\rangle$	0	0	$1/2^{1/2}$	$-1/2^{1/2}$

G_2	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	$1/2^{1/2}$	$1/2^{1/2}$	0	0
$ 01\rangle$	$-1/2^{1/2}$	$1/2^{1/2}$	0	0
$ 10\rangle$	0	0	$1/2^{1/2}$	$1/2^{1/2}$
$ 11\rangle$	0	0	$-1/2^{1/2}$	$1/2^{1/2}$

G_4	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	$1/2^{1/2}$	$1/2^{1/2}$	0	0
$ 01\rangle$	0	0	$-1/2^{1/2}$	$1/2^{1/2}$
$ 10\rangle$	0	0	$1/2^{1/2}$	$1/2^{1/2}$
$ 11\rangle$	$-1/2^{1/2}$	$1/2^{1/2}$	0	0

Наконец, различные программируемые ячейки генерируют различные суперпозиции состояний:

$G_1 00\rangle$	=	$1/2^{1/2}$	$ 00\rangle$	+	$1/2^{1/2}$	$ 01\rangle$
$G_2 00\rangle$	=	$1/2^{1/2}$	$ 00\rangle$	-	$1/2^{1/2}$	$ 01\rangle$
$G_3 00\rangle$	=	$1/2^{1/2}$	$ 00\rangle$	+	$1/2^{1/2}$	$ 11\rangle$
$G_4 00\rangle$	=	$1/2^{1/2}$	$ 00\rangle$	-	$1/2^{1/2}$	$ 11\rangle$

Заметим, что векторы $G_1|00\rangle$ и $G_2|00\rangle$ могут быть выражены в виде тензорного произведения более простых векторов:

$G_1 00\rangle$	=	$1/2^{1/2}$	$ 0\rangle$	\otimes	($ 0\rangle$ + $ 1\rangle$)
$G_2 00\rangle$	=	$1/2^{1/2}$	$ 0\rangle$	\otimes	($ 0\rangle$ - $ 1\rangle$)

Это невозможно для $G_3|00\rangle$ и $G_4|00\rangle$.

В этом случае эти два вектора образуют смешанное (запутанное — entangled) состояние.

Определение 1: Вектор v размерности 2^n образует запутанное (entangled) состояние тогда и только тогда, когда он не может быть представлен в виде тензорного произведения n векторов размерности 2. Математически данное определение выражается как:

$$\neg \exists v_1, \dots, v_n : v = v_1 \otimes \dots \otimes v_n.$$

Когда КАЯ сгенерировала выходной вектор, являющийся комплексной линейной комбинацией (суперпозицией) базисных векторов, выполняется операция измерения.

Предполагается, что измерение является недетерминированной операцией, на вход которой подается линейная комбинация базисных векторов. Вероятность каждого из векторов быть измеренным определяется квадратом его комплексной координаты в исходной суперпозиции.

Определение 2: Квантовым измерением является недетерминированная операция, входом которой является комплексная линейная комбинация базисных векторов

$v = \sum_{i=1}^n \alpha_i v_i$, а выходом является один из базисных векторов v_i . При этом вероятность наблюдения вектора v_i

равна квадрату его комплексной координаты α_i :

$v = \alpha_1 v_1 + \dots + \alpha_n v_n$	$\xrightarrow{\text{измерение } t}$	<table border="1"> <tr> <th>Вектор</th> <th>Вероятность</th> </tr> <tr> <td>v_1</td> <td>$\ \alpha_1\ ^2$</td> </tr> <tr> <td>\vdots</td> <td>\vdots</td> </tr> <tr> <td>v_n</td> <td>$\ \alpha_n\ ^2$</td> </tr> </table>	Вектор	Вероятность	v_1	$\ \alpha_1\ ^2$	\vdots	\vdots	v_n	$\ \alpha_n\ ^2$
Вектор	Вероятность									
v_1	$\ \alpha_1\ ^2$									
\vdots	\vdots									
v_n	$\ \alpha_n\ ^2$									

Применив операцию измерения к суперпозициям, рассматриваемым в данном примере, получаем:

Суперпозиция базисных векторов перед измерением	Результат измерения	
	Вектор	Вероятность
$G_1 00\rangle = 1/\sqrt{2} 00\rangle + 1/\sqrt{2} 01\rangle$	$ 00\rangle$ $ 01\rangle$	$\ 1/\sqrt{2}\ ^2 = 0.5$ $\ 1/\sqrt{2}\ ^2 = 0.5$
$G_2 00\rangle = 1/\sqrt{2} 00\rangle - 1/\sqrt{2} 01\rangle$	$ 00\rangle$ $ 01\rangle$	$\ 1/\sqrt{2}\ ^2 = 0.5$ $\ 1/\sqrt{2}\ ^2 = 0.5$
$G_3 00\rangle = 1/\sqrt{2} 00\rangle + 1/\sqrt{2} 11\rangle$	$ 00\rangle$ $ 11\rangle$	$\ 1/\sqrt{2}\ ^2 = 0.5$ $\ 1/\sqrt{2}\ ^2 = 0.5$
$G_4 00\rangle = 1/\sqrt{2} 00\rangle - 1/\sqrt{2} 11\rangle$	$ 00\rangle$ $ 11\rangle$	$\ 1/\sqrt{2}\ ^2 = 0.5$ $\ 1/\sqrt{2}\ ^2 = 0.5$

После измерения квантовый блок заканчивает свою работу. Таким образом, только один базисный вектор сохраняется. В отличие от классического алгоритма определения свойства функции за четыре итерации, в алгоритме Дойча квантовый блок определяет искомое качественное свойство функции только за одну итерацию (квантовое превосходство — quantum supremacy).

1.5. Декодирование

В блоке декодирования осуществляется интерпретация полученного в результате работы квантового блока базисного вектора. В случае измерения вектора $|00\rangle$ о типе анализируемой функции сказать ничего нельзя. В тоже время, если полученный вектор $|01\rangle$ или $|11\rangle$, мы знаем, что в первом случае функциями могут быть только f_1 или f_2 , а во втором случае только f_3 или f_4 . Очевидно, что только применяя G_1 и G_2 можно получить финальную суперпозицию, содержащую $|01\rangle$ с ненулевой амплитудой вероятности.

Аналогично только G_3 и G_4 могут произвести суперпозицию, содержащую $|11\rangle$.

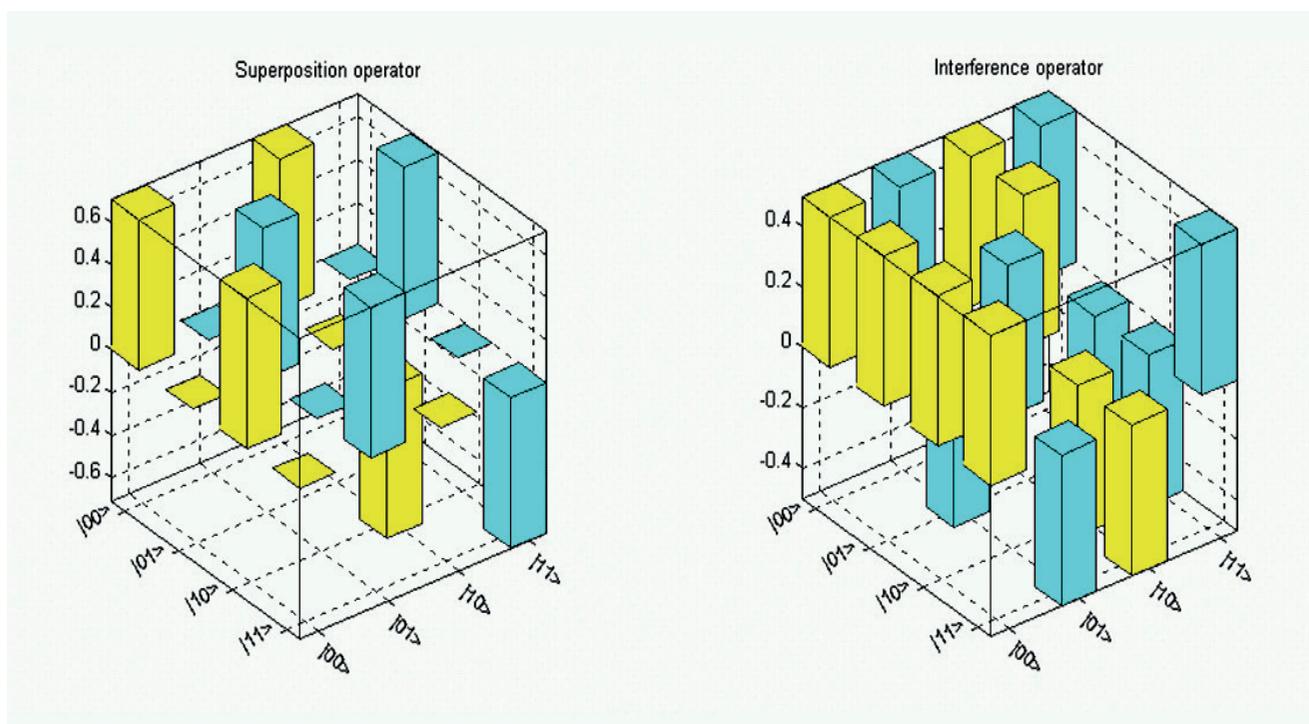
Так как функции f_1 и f_2 являются постоянными, а f_3 и f_4 являются балансирующими, декодирование происходит согласно следующей схеме:

Измеренный вектор	Ответ
$ 00\rangle$	Ничего сказать нельзя
$ 01\rangle$	Функция f является постоянной
$ 11\rangle$	Функция f является балансирующей

Таким образом, КА с вероятностью 50 % дает однозначный правильный результат о типе входной функции после выполнения всего одного обращения к квантовому блоку. Любой классический алгоритм для однозначно правильного результата требует вычисления всех возможных значений исследуемой функции.

2. Моделирование алгоритма Дойча

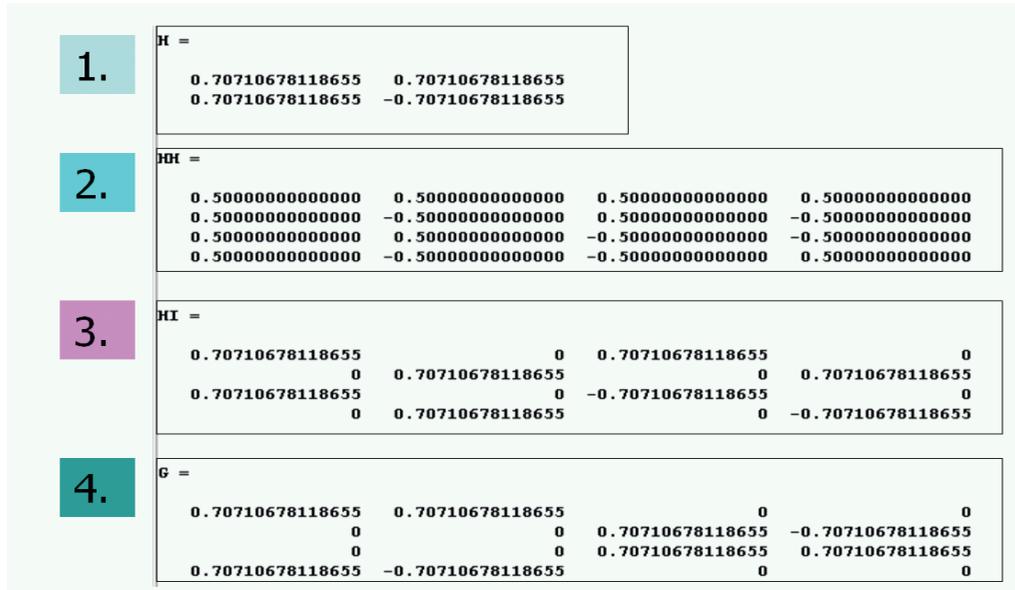
Для моделирования работы данного алгоритма использована система MATLAB®. На вход программы подается одна из возможных функций для анализа, выходом является измеренный базисный вектор. Операции кодирования, построения оператора U_F и сборки конечного квантового блока выполняются автоматически согласно алгоритму, описанному выше. Промежуточные результаты выводятся на экран монитора в процессе выполнения алгоритма. На рис. 7 показаны операторы суперпозиции интерференции.



Р и с.7. Операторы суперпозиции и интерференции
Fig. 7. Superposition & Interference operators



Результаты построения квантовой ячейки G и составляющих ее операторов $H \otimes H$, $H \otimes I$ и $(H \otimes H) \cdot U_F$ продемонстрированы на рис. 8.



Р и с. 8. Вычисление квантовых операторов КАЯ

F i g. 8. The Quantum operator's calculation of QAG

На рис. 9 показан результат применения ячейки G к начальному вектору $|00\rangle$, а также результаты измерения и декодирования.

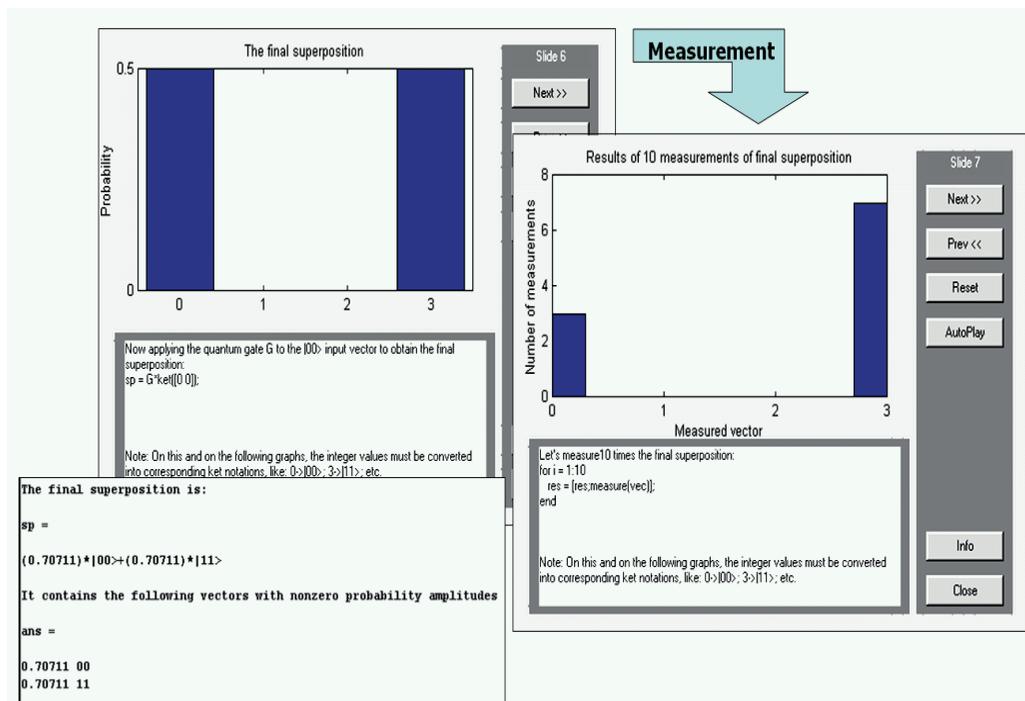
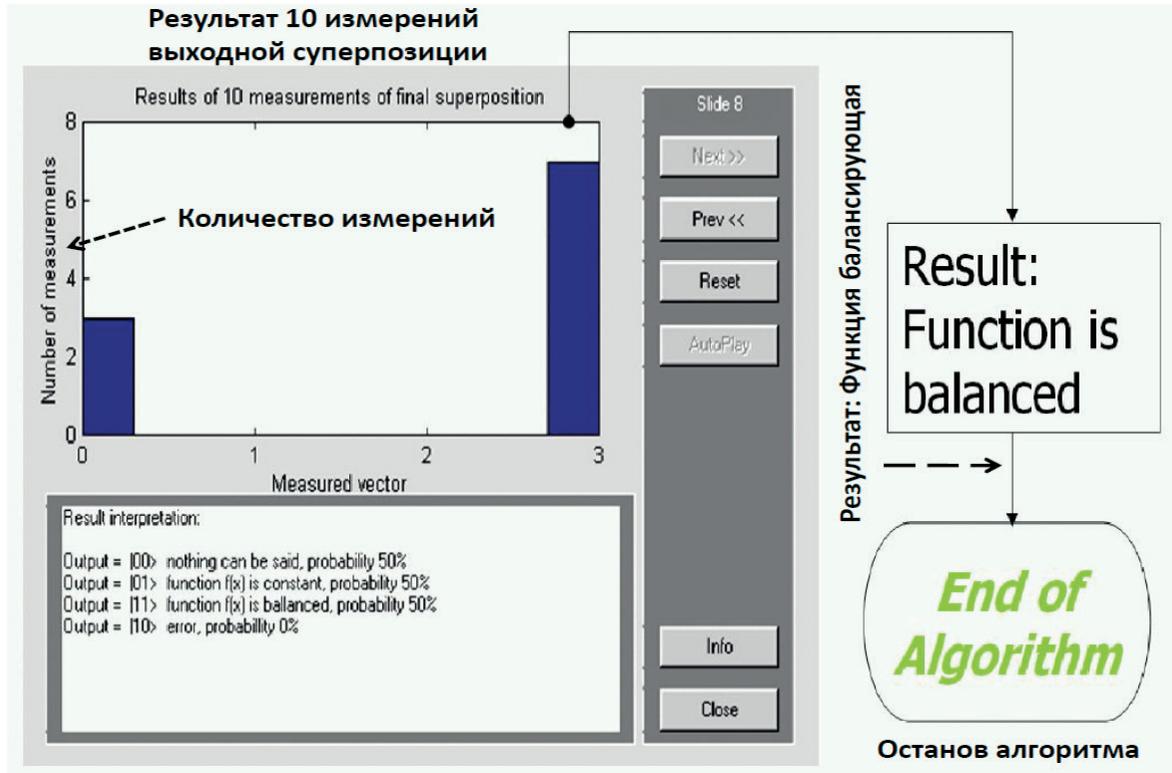
Р и с. 9. Применение оператора G в КАЯ для входного вектора $|00\rangle$ и его измерениеF i g. 9. The G operator application in QAG for the $|00\rangle$ input vector and its measurement

Рисунок 10 показывает содержательную интерпретацию выходного результата (декодирование).



Р и с. 10. Интерпретация результата (декодирование)
F i g. 10. The result interpretation (decoding)

Результат работы квантового алгоритма: тип исследуемой функции — «балансирующая функция». Структура программы позволяет дальнейший переход к более сложным квантовым алгоритмам, таким как алгоритм Deutsch — Jozsa, поисковый алгоритм Гровера и факторизации Шора с небольшими изменениями в исходном коде [12], [15]. Ввиду ограничения объема данного материала полный текст программы приводится в последующих работах.

3. Описание КА принятия решений Deutsch — Jozsa

Проблема квантового алгоритма Deutsch — Jozsa формулируется следующим образом:

Вход	Постоянная или балансирующая функция $f : \{0,1\}^n \rightarrow \{0,1\}$
Проблема	Решить, к какому классу относится функция f

Проблема алгоритма Deutsch — Jozsa такая же, как и у алгоритма Deutsch, но сформирована и обобщена для $n > 1$. Обобщение приведенной постановки задачи дано в [21] и моделируется по изложенной методике.

Шаги реализации алгоритма Deutsch — Jozsa

Шаг 0: Кодирование Шаг 0.1: Формирование инъективной функции F

Шаг 0.2: Приготовление таблицы отображения оператора запутывания

Шаг 1: Приготовление квантовых операторов

Шаг 1.1: Приготовление оператора суперпозиции

Шаг 1.2: Приготовление оператора запутывания, используя шаг 0.2

Шаг 1.3: Приготовление оператора интерференции

Шаг 1.4: Сборка квантовой ячейки

Шаг 2: Выполнение алгоритма

Шаг 2.1: Применение оператора суперпозиции

Шаг 2.2: Применение оператора запутывания

Шаг 2.3: Применение оператора интерференции

Шаг 2.4: Измерение и интерпретация результатов

3.1. Кодирование

С целью иллюстрации алгоритма и выявления важных аспектов рассмотрим случай для $n = 2$. В дальнейшем рассмотрим кодирование сбалансированной или постоянной функции f для $n > 0$.

Пример: Кодирование постоянной единичной функции

Рассмотрим случай, когда функция f — постоянная единичная функция:

$$n = 2, \quad \forall x \in \{0,1\}^n : f(x) = 1$$

Функция f отображается следующим образом:



x	$f(x)$
00	1
01	1
10	1
11	1

Смысл блока кодирования состоит в преобразовании таблицы отображения функции f в унитарный оператор U_f , действующий в Гильбертовом пространстве⁴.

Шаг 1. Таблица отображения функции f преобразуется в таблицу отображения обратимой функции $F: \{0,1\}^{n+1} \rightarrow \{0,1\}^{n+1}$, такую, что $F(x_0, x_1, y_0) = (x_0, x_1, f(x_0, x_1) \oplus y_0)$, то есть:

x_0, x_1, y_0	$F(x_0, x_1, y_0)$
000	001
010	011
100	101
110	111
001	000
011	010
101	100
111	110

Шаг 2. Таблица отображений обратимой функции F преобразуется в таблицу отображений оператора U_F согласно следующему правилу:

$$\forall s \in \{0,1\}^2 : U_F [\tau(s)] = \tau[F(s)]$$

Таким образом, получаем:

$ x_0, x_1, y_0\rangle$	$U_F x_0, x_1, y_0\rangle$
$ 000\rangle$	$ 001\rangle$
$ 010\rangle$	$ 011\rangle$
$ 100\rangle$	$ 101\rangle$
$ 110\rangle$	$ 111\rangle$
$ 001\rangle$	$ 000\rangle$
$ 011\rangle$	$ 010\rangle$
$ 101\rangle$	$ 100\rangle$
$ 111\rangle$	$ 110\rangle$

Шаг 3. Таблицу отображений U_F преобразуем в унитарный матричный оператор $[U_F]_{ij}$

$$[U_F]_{ij} = 1 \Leftrightarrow U_F |j\rangle = |i\rangle$$

U_f	$ 000\rangle$	$ 001\rangle$	$ 010\rangle$	$ 011\rangle$	$ 100\rangle$	$ 101\rangle$	$ 110\rangle$	$ 111\rangle$
$ 000\rangle$	0	1	0	0	0	0	0	0
$ 001\rangle$	1	0	0	0	0	0	0	0
$ 010\rangle$	0	0	0	1	0	0	0	0
$ 011\rangle$	0	0	1	0	0	0	0	0
$ 100\rangle$	0	0	0	0	0	1	0	0
$ 101\rangle$	0	0	0	0	1	0	0	0
$ 110\rangle$	0	0	0	0	0	0	0	1
$ 111\rangle$	0	0	0	0	0	0	1	0

Используя тензорное произведение, матричный оператор U_f может быть записан в следующем виде:

$$U_f = I \otimes I \otimes C = {}^2I \otimes C,$$

где \otimes — тензорное произведение, I — оператор идентичности и C — оператор отрицания:

$$C = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Оператор C преобразует $|0\rangle$ в $|1\rangle$ и $|1\rangle$ в $|0\rangle$. Если U_f применить к тензорному произведению трех векторов размерности 2, результирующим вектором будет тензорное произведение трех векторов, к первым двум из которых будет применен оператор идентичности, а к третьему — оператор отрицания.

В случае, когда матричный оператор может быть записан в виде тензорного произведения матричных оператор меньшей размерности, эволюция входных векторов происходит независимо друг от друга и никакой корреляции среди векторов нет.

3.1.1. Кодирование постоянной нулевой функции

Рассмотрим случай кодирования постоянной нулевой функции:

$$n = 2, \quad \forall x \in \{0,1\}^n : f(x) = 0$$

Отображение функции f

x	$f(x)$
00	1
01	1
10	1
11	1

Шаг 1. Таблица отображения обратимой функции F выглядит следующим образом:

x_0, x_1, y_0	$F(x_0, x_1, y_0)$
000	000
010	010
100	100
110	110
001	001
011	011
101	101
111	111

⁴ n -мерное Гильбертово пространство является n -мерным пространством комплексных чисел с определенным на нем скалярным произведением векторов.



Шаг 2. Таблица отображения F преобразуется в унитарный матричный оператор U_f

$ x_0, x_1, y_0\rangle$	$U_f x_0, x_1, y_0\rangle$
000⟩	000⟩
010⟩	010⟩
100⟩	100⟩
110⟩	110⟩
001⟩	001⟩
011⟩	011⟩
101⟩	101⟩
111⟩	111⟩

Шаг 3. Преобразуем табличный вид оператора U_f в матричный. Получаем матрицу размерностью $2^3 \times 2^3$:

U_f	000⟩	001⟩	010⟩	011⟩	100⟩	101⟩	110⟩	111⟩
000⟩	1	0	0	0	0	0	0	0
001⟩	0	1	0	0	0	0	0	0
010⟩	0	0	1	0	0	0	0	0
011⟩	0	0	0	1	0	0	0	0
100⟩	0	0	0	0	1	0	0	0
101⟩	0	0	0	0	0	1	0	0
110⟩	0	0	0	0	0	0	1	0
111⟩	0	0	0	0	0	0	0	1

Используя тензорное произведение, унитарный матричный оператор U_f может быть представлен следующим образом:

$$U_f = I \otimes I \otimes I = {}^3I,$$

где \otimes — тензорное произведение, I — тождественная матрица.

Структура U_f такова, что все векторы входного тензорного произведения преобразуются независимо друг от друга.

3.1.2. Кодирование сбалансированной функции

x	$f(x)$
00	0
01	1
10	1
11	0

Шаг 1. Таблица отображений функции F принимает вид:

x_0, x_1, y_0	$F(x_0, x_1, y_0)$
000	000
010	011
100	101
110	110
001	001
011	010
101	100
111	111

Шаг 2. F преобразуется в U_f :

$ x_0, x_1, y_0\rangle$	$U_f x_0, x_1, y_0\rangle$
000⟩	000⟩
010⟩	010⟩
100⟩	100⟩
110⟩	110⟩
001⟩	001⟩
011⟩	011⟩
101⟩	101⟩
111⟩	111⟩

Шаг 3. Переводим табличный вид оператора U_f в матричный. Получаем матрицу размерностью $2^3 \times 2^3$:

U_f	000⟩	001⟩	010⟩	011⟩	100⟩	101⟩	110⟩	111⟩
000⟩	1	0	0	0	0	0	0	0
001⟩	0	1	0	0	0	0	0	0
010⟩	0	0	1	0	0	0	0	0
011⟩	0	0	0	1	0	0	0	0
100⟩	0	0	0	0	1	0	0	0
101⟩	0	0	0	0	0	1	0	0
110⟩	0	0	0	0	0	0	1	0
111⟩	0	0	0	0	0	0	0	1

Матрицу нельзя записать в виде тензорного произведения матриц меньшей размерности. Мы можем представить матрицу в виде вложенных матриц:

U_F	00⟩	01⟩	10⟩	11⟩
00⟩	C	0	0	0
01⟩	0	I	0	0
10⟩	0	0	I	0
11⟩	0	0	0	C

Это означает, что матричный оператор, действующий на третий вектор из входного тензорного произведения, зависит от значений остальных двух векторов.

Например, у нас есть входные векторы $|0\rangle$ и $|0\rangle$, тогда матричным оператором, действующим на третий вектор, является оператор идентичности I . Если же у нас входные векторы $|0\rangle$ и $|1\rangle$, то третий вектор преобразуется под действием оператора отрицания C . Этот оператор создает запутанность состояний, а именно корреляцию среди векторов тензорного произведения. В данном случае невозможно представить матричный оператор запутывания через тензорное произведение более простых операторов, таких как I и C ; это было возможно в случае операторов запутанности только для постоянных функций, представленных выше.



3.1.3. Общий вид кодирования f при $n = 2$

В общем случае функция f отображается следующим образом:

x	$f(x)$
00	f_{00}
01	f_{01}
10	f_{10}
11	f_{11}

где $f_i \in \{0,1\}$, $i = 00,01,10,11$.

Функция f является постоянной, если $\exists y \in \{0,1\} : \forall x \in \{0,1\}^n : f(x) = y$. Функция f - сбалансированная, если $|\{f_i : f_i = 0\}| = |\{f_i : f_i = 1\}|$.

Шаг 1. Обратимую функцию F представляем в табличном виде ($f \rightarrow F$):

$x_0, x_1 y_0$	$F(x_0, x_1 y_0)$
000	00 f_{00}
010	01 f_{01}
100	10 f_{10}
110	11 f_{11}
001	00 $\neg f_{00}$
011	01 $\neg f_{01}$
101	10 $\neg f_{10}$
111	11 $\neg f_{11}$

Шаг 2. F преобразуется в U_f :

$ x_0, x_1 y_0\rangle$	$U_f x_0, x_1 y_0\rangle$
$ 000\rangle$	$ 00 f_{00}\rangle$
$ 010\rangle$	$ 01 f_{01}\rangle$
$ 100\rangle$	$ 10 f_{10}\rangle$
$ 110\rangle$	$ 11 f_{11}\rangle$
$ 001\rangle$	$ 00 \neg f_{00}\rangle$
$ 011\rangle$	$ 01 \neg f_{01}\rangle$
$ 101\rangle$	$ 10 \neg f_{10}\rangle$
$ 111\rangle$	$ 11 \neg f_{11}\rangle$

Шаг 3. Матрица, соответствующая унитарному оператору U_f , может быть записана как:

U_F	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	M_{00}	0	0	0
$ 01\rangle$	0	M_{01}	0	0
$ 10\rangle$	0	0	M_{10}	0
$ 11\rangle$	0	0	0	M_{11}

где $M_i = I$, если $f_i = 0$ и $M_i = C$, если $f_i = 1$ при $i = 00,01,10,11$. Структура этой матрицы такова, что первые два вектора всегда остаются неизменными, при этом операторы $M_i \in \{I, C\}$ применяются только к третьему вектору. Если все M_i одинаковые, то рассматриваемая функция постоянная. Если $|\{M_i : M_i = I\}| = |\{M_i : M_i = C\}|$, это значит, что функция f — сбалансированная.

3.1.4. Общий случай

Рассмотрим общий случай при $n > 0$. Функция f имеет следующий вид:

x	$f(x)$
0..0	$f_{0..0}$
0..1	$f_{0..1}$
..	$f_{..}$
1..1	$f_{1..1}$

где $f_i \in \{0,1\}$, $i \in \{0,1\}^n$. Функция f является постоянной, если $\exists y \in \{0,1\} : \forall x \in \{0,1\}^n : f(x) = y$. Функция f — сбалансированная, если $|\{f_i : f_i = 0\}| = |\{f_i : f_i = 1\}|$.

Шаг 1. Обратимую функцию F представляем в табличном виде:

$x \in \{0,1\}^n$	$F(x_0, x_1 y_0)$
0..00	0..0 $f_{0..0}$
...	...
1..10	1..1 $f_{1..1}$
0..10	0..1 $\neg f_{1..1}$
...	...
1..11	1..1 $\neg f_{1..1}$

Шаг 2. F преобразуется в U_f :

$ x\rangle$	$F x\rangle$
$..00\rangle$	$..0f_{0..0}\rangle$
...	...
$..10\rangle$	$..1f_{1..1}\rangle$
$..10\rangle$	$..1\neg f_{1..1}\rangle$
...	...
$..11\rangle$	$..1\neg f_{1..1}\rangle$

Шаг 3. Матрица, соответствующая унитарному оператору U_f , может быть записана следующим образом:

U_f	$ 0..0\rangle$	$ 0..1\rangle$	$ 1..0\rangle$	$ 1..1\rangle$
$ 0..0\rangle$	$M_{0..0}$	0	0	0
$ 0..1\rangle$	0	$M_{0..1}$	0	0
..
$ 1..1\rangle$	0	0	0	$M_{1..1}$

где $M_i = I$ при $M_i = C$ при $f = 0$, $i \in \{0,1\}^n$.

Эта матрица оставляет неизменными первые n векторов, а к последнему применяет матричный оператор $M_i \in \{I, C\}$.

Если все M_i совпадают с C или I , то функцию U_f можно записать как ${}^n I \otimes I$ или ${}^n I \otimes C$ соответственно.

В данном случае никакой запутанности (entanglement) состояний не будет. Однако, если выполняется условие $|\{M_i : M_i = I\}| = |\{M_i : M_i = C\}|$, то функция f является сбалансированной, и оператор создает квантовую корреляцию среди векторов.

3.2. Квантовый блок

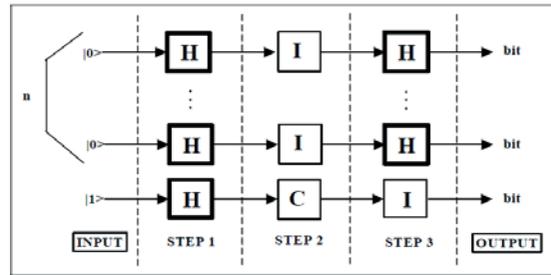
Полученная матрица U_f из блока кодирования погружается в квантовый блок. Для наилучшего анализа структуры квантового блока рассмотрим два случая, когда функция f является сбалансированной и постоянной. Так же будет рассмотрен общий случай при $n = 2$ и при $n > 0$.

3.2.1. f - постоянная единичная функция

Если f — постоянная единичная функция, то U_f представляется следующим образом:

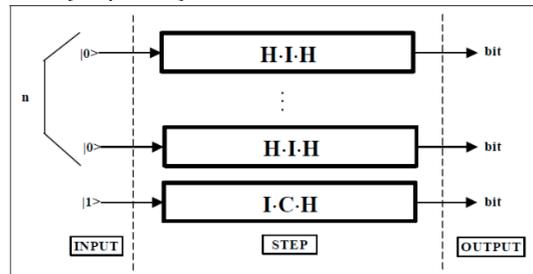
$$U_f = {}^n I \otimes C$$

Это означает, что U_f может быть разложена на $n+1$ оператора, преобразующих одновременно $n+1$ вектор размерности 2 входного тензорного произведения (см. рис. 10).



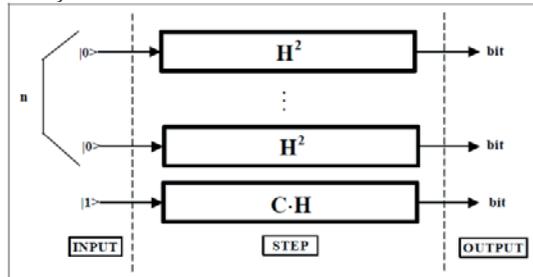
Р и с.11. Постоянная единичная функция. Первая схема
F i g. 11. Constant identity function. First scheme

За счет того, что U_f не создает корреляции, мы можем рассматривать преобразования входных параметров отдельно (см. рис. 12).



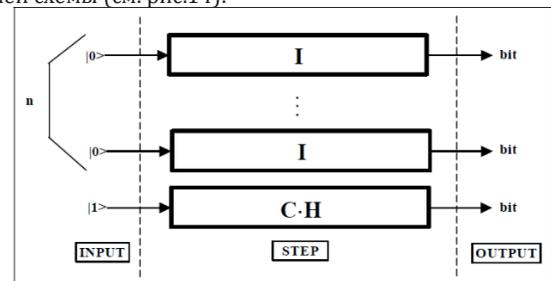
Р и с. 12. Постоянная единичная функция. Вторая схема
F i g. 12. Constant identity function. Second scheme

Зная, что $M \cdot I = M$, где M — матрица такой же размерности, что и I , мы можем упростить нашу схему (см.рис.13):



Р и с.13. Постоянная единичная функция. Третья схема
F i g. 13. Constant identity function. Third scheme

Полагая, что $H^2 = I$, можем провести еще одно упрощение нашей схемы (см. рис.14):



Р и с.14. Постоянная единичная функция. Четвертая схема
F i g. 14. Constant identity function. Fourth scheme

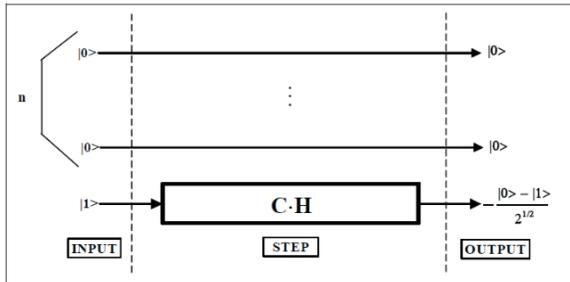


Рассмотрим преобразования, происходящие с входным вектором:

$$I|0\rangle = |0\rangle,$$

$$C \cdot H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

Легко заметить, что, если f — постоянная единичная функция, то первые n векторов остаются неизменными. Таким образом, нашу схему можно привести к следующему виду (см. рис.15):



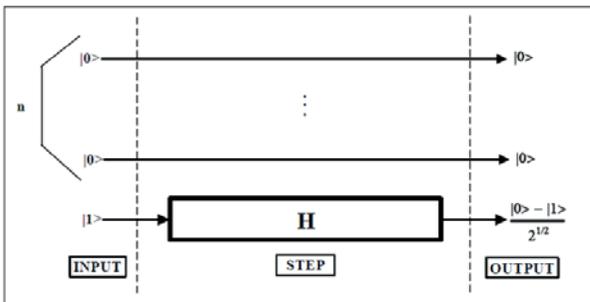
Р и с. 15. Постоянная единичная функция. Пятая схема
F i g. 15. Constant identity function. Fifth scheme

3.2.2. f — постоянная нулевая функция

В данном случае U_f представляется следующим образом:

$$U_f = {}^n I \otimes I.$$

Проведя подобный анализ, который был описан выше, мы получим следующую схему (см. рис. 16):



Р и с. 16. Постоянная единичная функция. Конечная схема
F i g. 16. Constant identity function. Final scheme

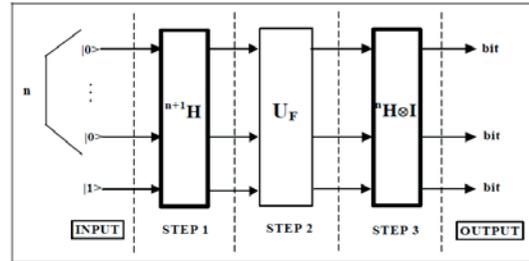
3.2.3. Общий случай для $n=2$

При $n = 2$ унитарный матричный U_f принимает следующий

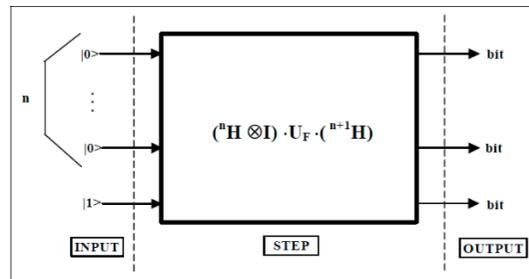
вид:

U_f	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	M_{00}	0	0	0
$ 01\rangle$	0	M_{01}	0	0
$ 10\rangle$	0	0	M_{10}	0
$ 11\rangle$	0	0	0	M_{11}

где $M_i \in \{I, C\}, i = 00,01,10,11$. Схема преобразований и квантовая схема алгоритма Deutsch — Jozsa показаны на рис. 17 и 18 соответственно:



Р и с. 17. Схема преобразований
F i g. 17. Scheme of transformations



Р и с. 18. Квантовая схема алгоритма Deutsch
F i g. 18. Quantum scheme of the Deutsch algorithm

Вычислим квантовую ячейку $G = ({}^n H) \cdot U_f \cdot ({}^{n+1} H)$ для $n=2$.

${}^3 H$	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	$H/2$	$H/2$	$H/2$	$H/2$
$ 01\rangle$	$H/2$	$-H/2$	$H/2$	$-H/2$
$ 10\rangle$	$H/2$	$H/2$	$-H/2$	$-H/2$
$ 11\rangle$	$H/2$	$-H/2$	$-H/2$	$H/2$

${}^2 H \otimes I$	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	$I/2$	$I/2$	$I/2$	$I/2$
$ 01\rangle$	$I/2$	$-I/2$	$I/2$	$-I/2$
$ 10\rangle$	$I/2$	$I/2$	$-I/2$	$-I/2$
$ 11\rangle$	$I/2$	$-I/2$	$-I/2$	$I/2$

$U_f \cdot {}^3 H$	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	$M_{00}H/2$	$M_{00}H/2$	$M_{00}H/2$	$M_{00}H/2$
$ 01\rangle$	$M_{01}H/2$	$-M_{01}H/2$	$M_{01}H/2$	$-M_{01}H/2$
$ 10\rangle$	$M_{10}H/2$	$M_{10}H/2$	$-M_{10}H/2$	$-M_{10}H/2$
$ 11\rangle$	$M_{11}H/2$	$M_{11}H/2$	$-M_{11}H/2$	$-M_{11}H/2$

G	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	$\begin{pmatrix} M_{00} + M_{01} + M_{10} + \\ M_{11} \\ H/4 \end{pmatrix}$	$\begin{pmatrix} M_{00} - M_{01} + M_{10} - \\ M_{11} \\ H/4 \end{pmatrix}$	$\begin{pmatrix} M_{00} + M_{01} - M_{10} - \\ M_{11} \\ H/4 \end{pmatrix}$	$\begin{pmatrix} M_{00} - M_{01} - M_{10} + \\ M_{11} \\ H/4 \end{pmatrix}$
$ 01\rangle$	$\begin{pmatrix} M_{00} - M_{01} + M_{10} - \\ M_{11} \\ H/4 \end{pmatrix}$	$\begin{pmatrix} M_{00} + M_{01} + M_{10} + \\ M_{11} \\ H/4 \end{pmatrix}$	$\begin{pmatrix} M_{00} - M_{01} - M_{10} + \\ M_{11} \\ H/4 \end{pmatrix}$	$\begin{pmatrix} M_{00} + M_{01} - M_{10} - \\ M_{11} \\ H/4 \end{pmatrix}$
$ 10\rangle$	$\begin{pmatrix} M_{00} + M_{01} - M_{10} - \\ M_{11} \\ H/4 \end{pmatrix}$	$\begin{pmatrix} M_{00} - M_{01} - M_{10} + \\ M_{11} \\ H/4 \end{pmatrix}$	$\begin{pmatrix} M_{00} + M_{01} + M_{10} + \\ M_{11} \\ H/4 \end{pmatrix}$	$\begin{pmatrix} M_{00} - M_{01} + M_{10} - \\ M_{11} \\ H/4 \end{pmatrix}$
$ 11\rangle$	$\begin{pmatrix} M_{00} - M_{01} - M_{10} + \\ M_{11} \\ H/4 \end{pmatrix}$	$\begin{pmatrix} M_{00} + M_{01} - M_{10} - \\ M_{11} \\ H/4 \end{pmatrix}$	$\begin{pmatrix} M_{00} - M_{01} + M_{10} - \\ M_{11} \\ H/4 \end{pmatrix}$	$\begin{pmatrix} M_{00} + M_{01} + M_{10} + \\ M_{11} \\ H/4 \end{pmatrix}$

При входном векторе $|001\rangle$ получаем:

$$G|001\rangle = |00\rangle \otimes \frac{H|1\rangle}{4}(M_{00} + M_{01} + M_{10} + M_{11}) + |01\rangle \otimes \frac{H|1\rangle}{4}(M_{00} - M_{01} + M_{10} - M_{11}) +$$

$$+ |10\rangle \otimes \frac{H|1\rangle}{4}(M_{00} - M_{01} - M_{10} + M_{11}) + |11\rangle \otimes \frac{H|1\rangle}{4}(M_{00} - M_{01} - M_{10} + M_{11}).$$

Рассмотрим оператор $\frac{H}{4}(M_{00} + M_{01} + M_{10} + M_{11})$,

предположив, что f — сбалансированная функция

$$|\{M_i : M_i = I\}| = |\{M_i : M_i = C\}|.$$

Получаем следующее:

$M_{00} + M_{01} + M_{10} + M_{11}$	$ 0\rangle$	$ 1\rangle$
$ 0\rangle$	2	2
$ 1\rangle$	2	2
$\frac{H}{4}(M_{00} + M_{01} + M_{10} + M_{11})$	$ 0\rangle$	$ 1\rangle$
$ 0\rangle$	$\left(\frac{1}{2}\right)^{\frac{1}{2}}$	0
$ 1\rangle$	$\left(\frac{1}{2}\right)^{\frac{1}{2}}$	0

Получаем, что

$$\frac{H|1\rangle}{4}(M_{00} + M_{01} + M_{10} + M_{11}) = 0.$$

Это говорит о том, что амплитуда вероятности преобразования состояния входного регистра $|001\rangle$ в $|000\rangle$ или $|001\rangle$ равна нулю. Рассмотрим теперь матричные операторы:

$$\frac{H}{4}(M_{00} + M_{01} + M_{10} + M_{11}),$$

$$\frac{H}{4}(M_{00} - M_{01} + M_{10} - M_{11}),$$

$$\frac{H}{4}(M_{00} - M_{01} - M_{10} + M_{11}),$$

$$\frac{H}{4}(M_{00} - M_{01} - M_{10} + M_{11}),$$

предполагая, что f — постоянная нулевая функция, то есть $\forall i : M_i = I$.

$M_{00} + M_{01} + M_{10} + M_{11}$	$ 0\rangle$	$ 1\rangle$
$ 0\rangle$	4	0
$ 1\rangle$	0	4

$\frac{H}{4}(M_{00} + M_{01} + M_{10} + M_{11})$	$ 0\rangle$	$ 1\rangle$
$ 0\rangle$	$\left(\frac{1}{2}\right)^{\frac{1}{2}}$	$\left(\frac{1}{2}\right)^{\frac{1}{2}}$
$ 1\rangle$	$\left(\frac{1}{2}\right)^{\frac{1}{2}}$	$\left(-\frac{1}{2}\right)^{\frac{1}{2}}$

$M_{00} - M_{01} + M_{10} - M_{11}$	$ 0\rangle$	$ 1\rangle$
$ 0\rangle$	0	0
$ 1\rangle$	0	0
$M_{00} + M_{01} - M_{10} - M_{11}$	$ 0\rangle$	$ 1\rangle$
$ 0\rangle$	0	0
$ 1\rangle$	0	0
$M_{00} - M_{01} - M_{10} + M_{11}$	$ 0\rangle$	$ 1\rangle$
$ 0\rangle$	0	0
$ 1\rangle$	0	0



где $M_i \in \{I, C\}$, $i \in \{0, 1\}^n$.

Вычислим квантовую ячейку $G = ({}^n H) \cdot U_f \cdot ({}^{n+1} H)$:

${}^{n+1} H$	$ 0..0\rangle$...	$ j\rangle$...	$ 1..1\rangle$
$ 0..0\rangle$	$H/2^{n/2}$...	$H/2^{n/2}$...	$H/2^{n/2}$
...
$ i\rangle$	$H/2^{n/2}$...	$(-1)^{i \cdot j} H/2^{n/2}$...	$(-1)^{i \cdot (1..1)} H/2^{n/2}$
...
$ 1..1\rangle$	$H/2^{n/2}$...	$(-1)^{(1..1) \cdot j} H/2^{n/2}$...	$(-1)^{(1..1) \cdot (1..1)} H/2^{n/2}$

${}^n H \otimes I$	$ 0..0\rangle$...	$ j\rangle$...	$ 1..1\rangle$
$ 0..0\rangle$	$I/2^{n/2}$...	$I/2^{n/2}$...	$I/2^{n/2}$
...
$ i\rangle$	$I/2^{n/2}$...	$(-1)^{i \cdot j} I/2^{n/2}$...	$(-1)^{i \cdot (1..1)} I/2^{n/2}$
...
$ 1..1\rangle$	$I/2^{n/2}$...	$(-1)^{(1..1) \cdot j} I/2^{n/2}$...	$(-1)^{(1..1) \cdot (1..1)} I/2^{n/2}$

$U_f \cdot {}^{n+1} H$	$ 0..0\rangle$...	$ j\rangle$...	$ 1..1\rangle$
$ 0..0\rangle$	$M_{0..0} H/2^n$...	$M_{0..0} H/2^{n/2}$...	$M_{0..0} H/2^{n/2}$
...
$ i\rangle$	$M_i H/2^{n/2}$...	$(-1)^{i \cdot j} M_i H/2^{n/2}$...	$(-1)^{i \cdot (1..1)} M_i H/2^{n/2}$
...
$ 1..1\rangle$	$M_{1..1} H/2^{n/2}$...	$(-1)^{(1..1) \cdot j} M_{1..1} H/2^{n/2}$...	$(-1)^{(1..1) \cdot (1..1)} M_{1..1} H/2^{n/2}$

Рассматриваем только первую колонку оператора G , потому что данный оператор применяется для входного регистра $|0..01\rangle$.

G	$ 0..0\rangle$...
$ 0..0\rangle$	$(M_{0..0} + \dots + M_i + \dots + M_{1..1}) H/2^n$...
...
$ i\rangle$	$(\sum_{j \in \{0,1\}^n} (-1)^{i \cdot j} M_j) H/2^n$...
...
$ 1..1\rangle$	$(\sum_{j \in \{0,1\}^n} (-1)^{(1..1) \cdot j} M_j) H/2^n$...

Рассмотрим случай, когда функция f – постоянная. Это означает, что $\forall i: M_i = I$:

$$\frac{1}{2^n} (\sum_{j \in \{0,1\}^n} (-1)^{i \cdot j} M_j) H = 0.$$

Следовательно, вектор $|0..01\rangle$ преобразуется в суперпозицию, содержащую векторы $|0..00\rangle$ или $|0..01\rangle$. Работа квантового блока заканчивается операцией измерения.

Вышеупомянутые результаты показывают возможность измерения и их вероятности:

Суперпозиция базисных векторов (до измерения)	Результат измерения	
	Вектор	Вероятность
Постоянная функция: $G 0..01\rangle = 0..0\rangle \otimes (a_0 0\rangle + a_1 1\rangle)$	$ 0..00\rangle$ $ 0..01\rangle$	$\ a_0\ ^2$ $\ a_1\ ^2$
Сбалансированная функция: $G 0..01\rangle = \sum_{i \in \{0,1\}^n - \{0..00, 0..01\}} a_i i\rangle$	$\forall i \in \{0,1\}^n - \{0..00, 0..01\} : i\rangle$	$\ a_i\ ^2$

В алгоритме Deutsch — Jozsa квантовый блок повторяется всего один раз.

3.3. Декодирование

Как и в Deutsch-алгоритме, после измерения заключительного вектора нужно интерпретировать результат, то есть выявить свойство функции f (f — постоянная или сбалансированная). Фактически полученная структура квантовой ячейки G такова, что после измерения только состояния $|0..00\rangle$ и $|0..01\rangle$ имеют ненулевую амплитуду вероятности, исключительно в случае сбалансированной функции f . В случае, когда функция

случае сбалансированной функции f . В случае, когда функция f — сбалансированная, эти векторы имеют нулевые коэффициенты в линейной комбинации базисных векторов, получаемой после применения квантовой ячейки G к входному состоянию.

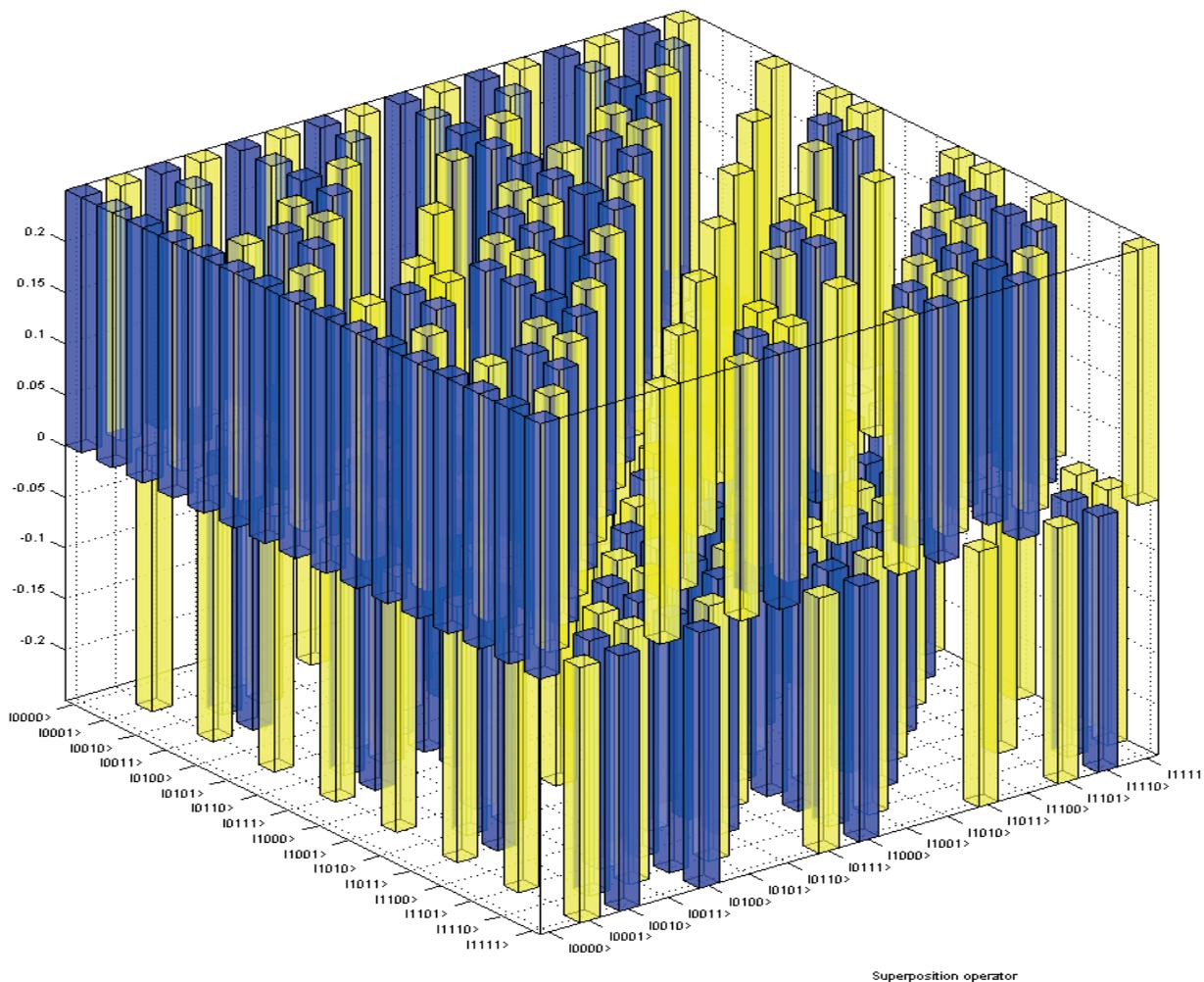
Таким образом, имеем следующее:

Результирующий вектор (после измерения)	Ответ
$ 0..00\rangle$	f – постоянная
$ 0..01\rangle$	f – постоянная
Остальные	f – сбалансированная

4. Моделирование квантового алгоритма Deutsch — Jozsa

Для моделирования работы данного алгоритма использована система MATLAB® [12]. На вход программы подается одна из возможных функций для анализа, выходом является измеренный базисный вектор. Операции кодирования, построения оператора U_F и сборки конечного квантового блока выполняются автоматически согласно алгоритму, описанному выше.

Промежуточные результаты выводятся на экран в процессе выполнения алгоритма. Рассмотрим работу квантового алгоритма Deutsch — Jozsa согласно шагам, описанным в алгоритме для постоянной и сбалансированной функций при $n = 3$. На шаге 0 (*encoding*) формируется отображение оператора запутывания. Для шага 1 на рис. 19 изображен оператор суперпозиции (${}^n H \otimes H$).



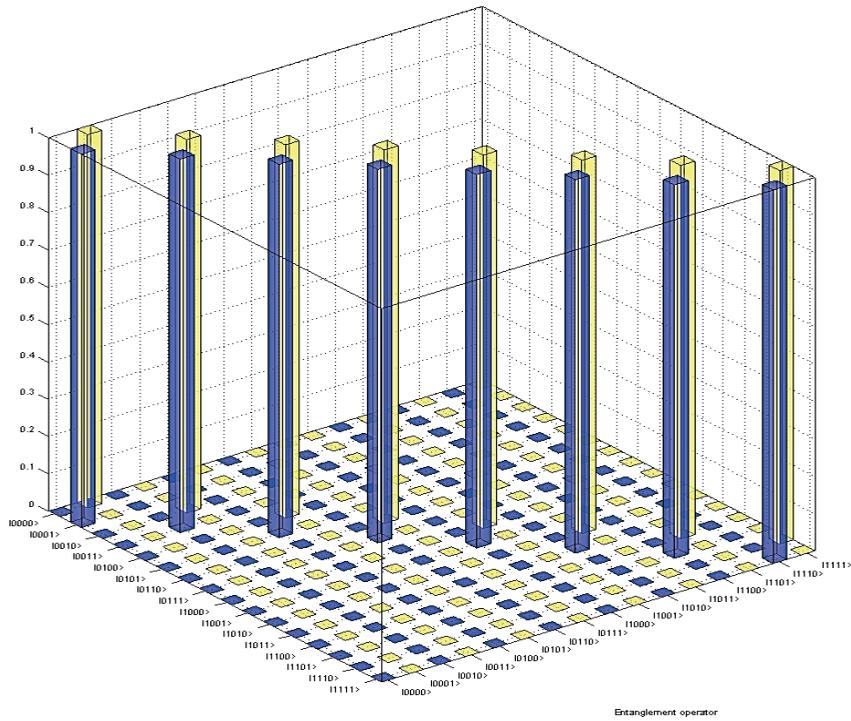
Р и с. 19. Оператор суперпозиции
F i g. 19. Superposition operator

На рис. 20 и 21 изображены операторы запутывания для постоянной функции ($\forall x \in \{0,1\}^3 : f(x)=1$ и $\forall x \in \{0,1\}^3 : f(x)=0$). Для сбалансированной имеем

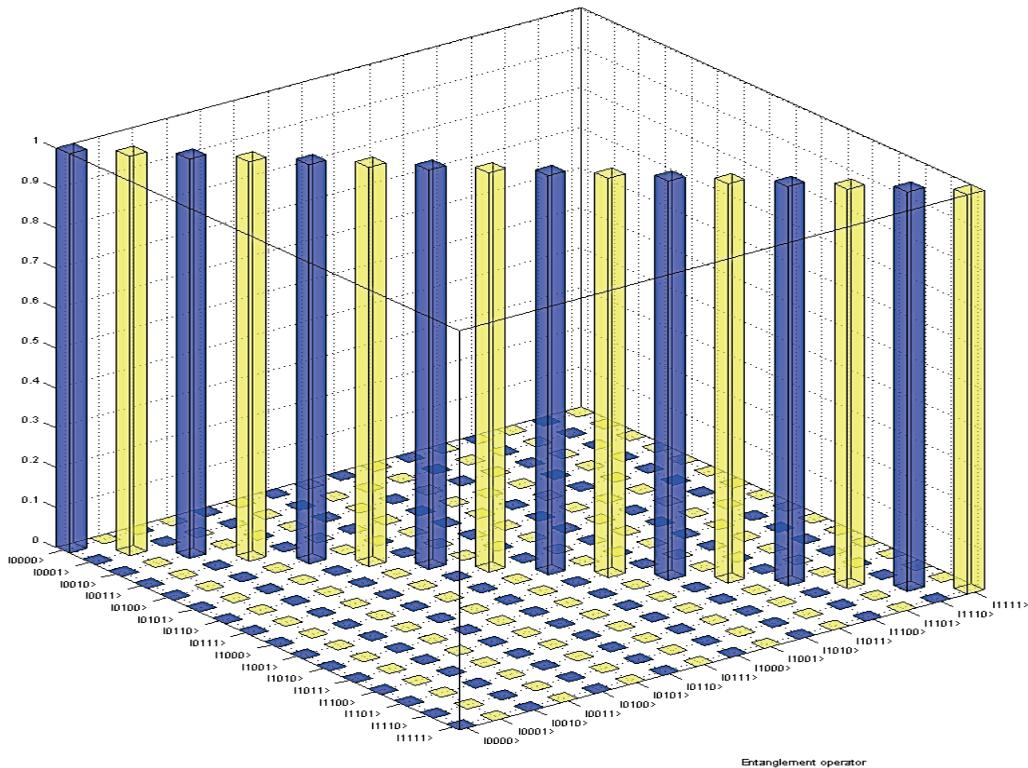
$$f(x) = \begin{cases} 1, & x = \{000, 010, 100, 111\} \\ 0, & x = \{001, 011, 101, 110\} \end{cases}$$

рис. 22 соответственно.



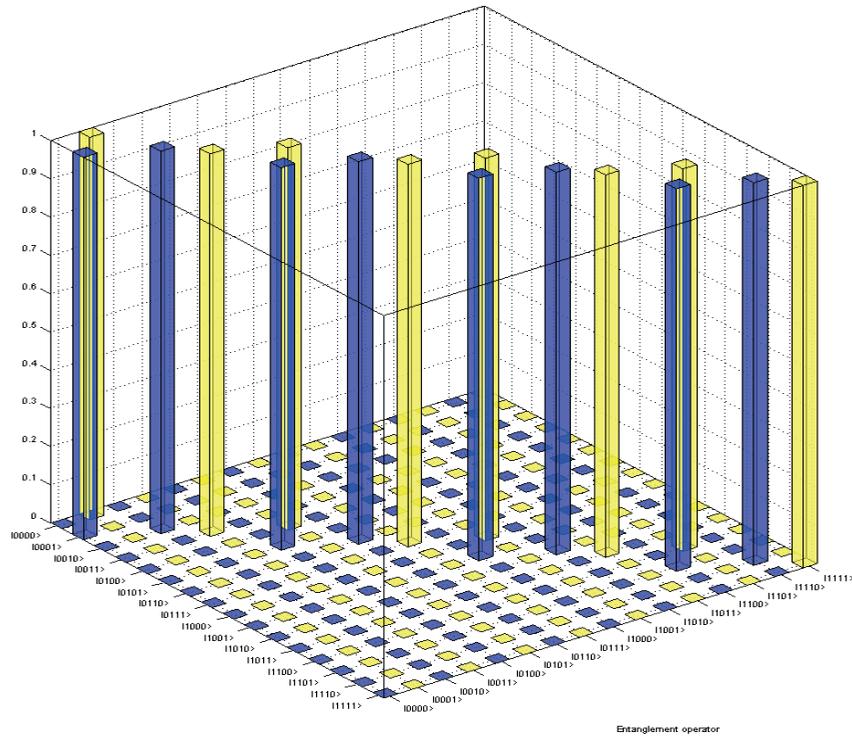


Р и с. 20. Оператор запутывания для единичной постоянной функции
F i g. 20. Entanglement operator for identity constant function



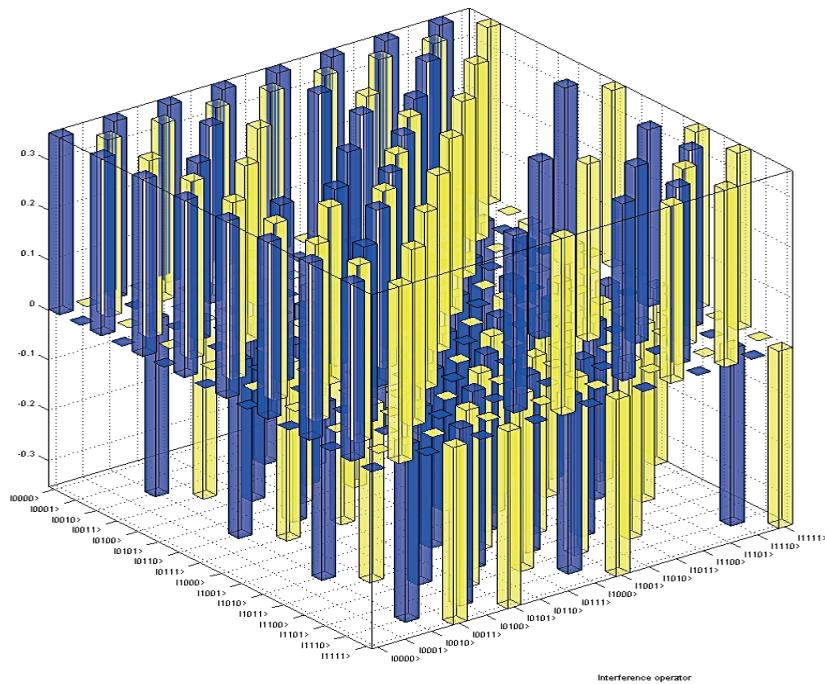
Р и с. 21. Оператор запутывания для нулевой постоянной функции
F i g. 21. Entanglement operator for zero constant function





Р и с. 22. Оператор запутывания для сбалансированной функции
F i g. 22. Entanglement operator for balanced function

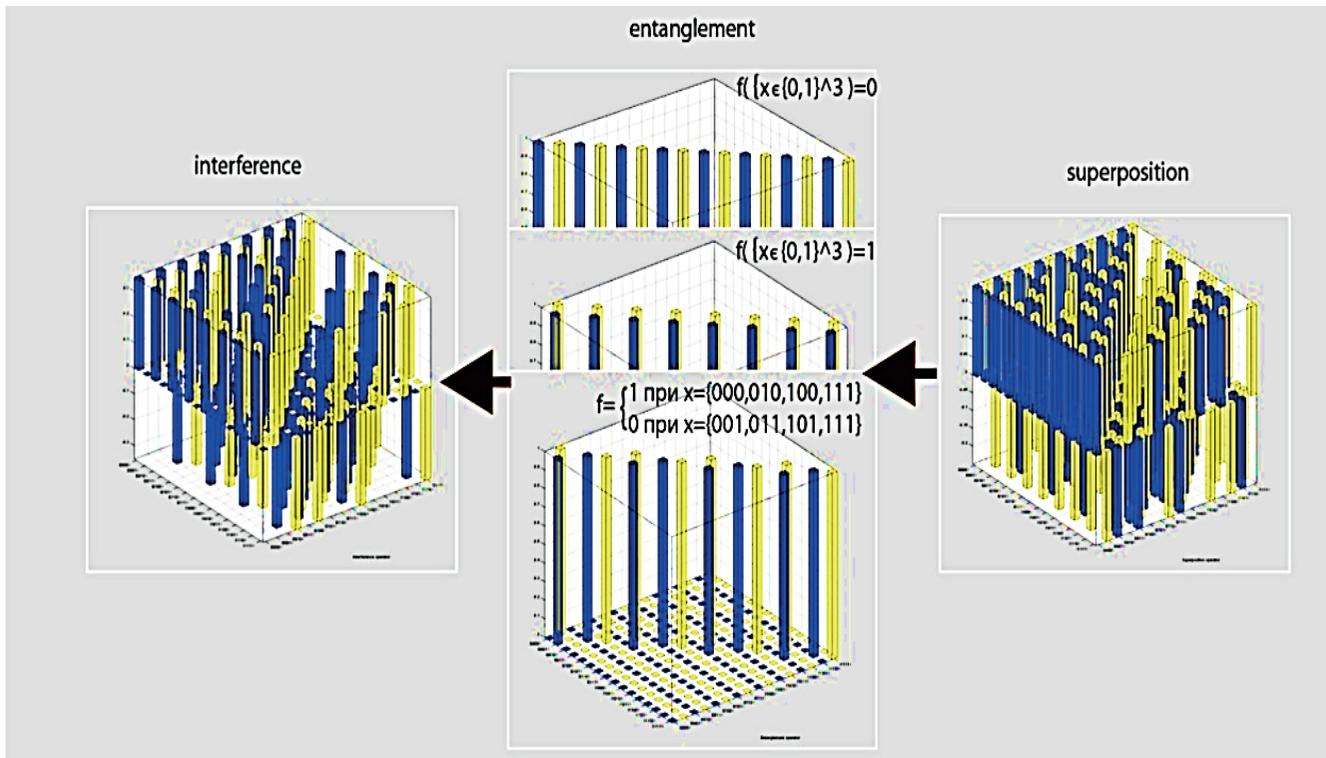
Оператору интерференции (${}^n H \otimes I$) соответствует рис. 23.



Р и с. 23. Оператор интерференции
F i g. 23. Interference operator

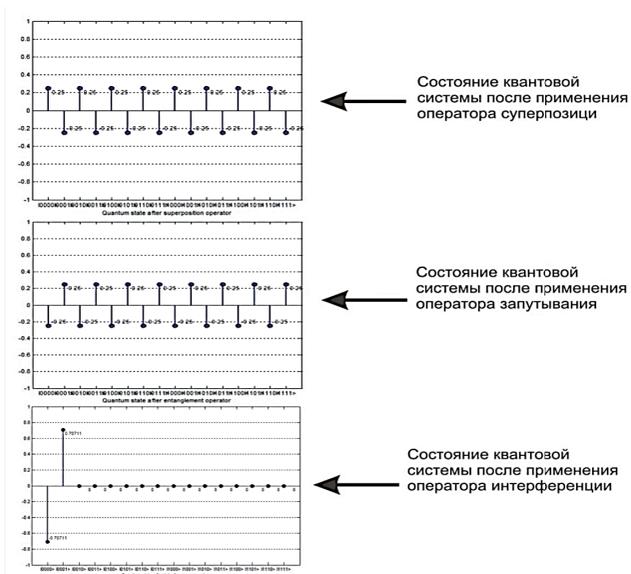


После приготовления квантовых операторов осуществляется сборка квантовой ячейки (рис. 24).

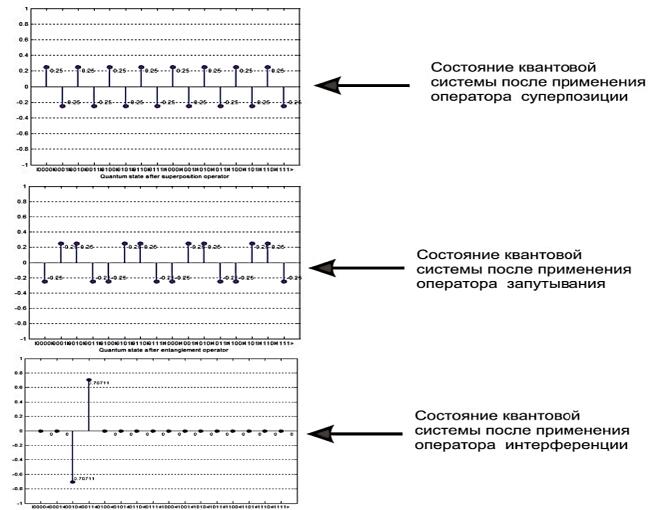


Р и с. 24. Сборка квантовой ячейки
F i g. 24. The quantum gates assembly

Рисунки 25 и 26 показывают результат выполнения алгоритма для постоянной и сбалансированной функций (шаг 2).

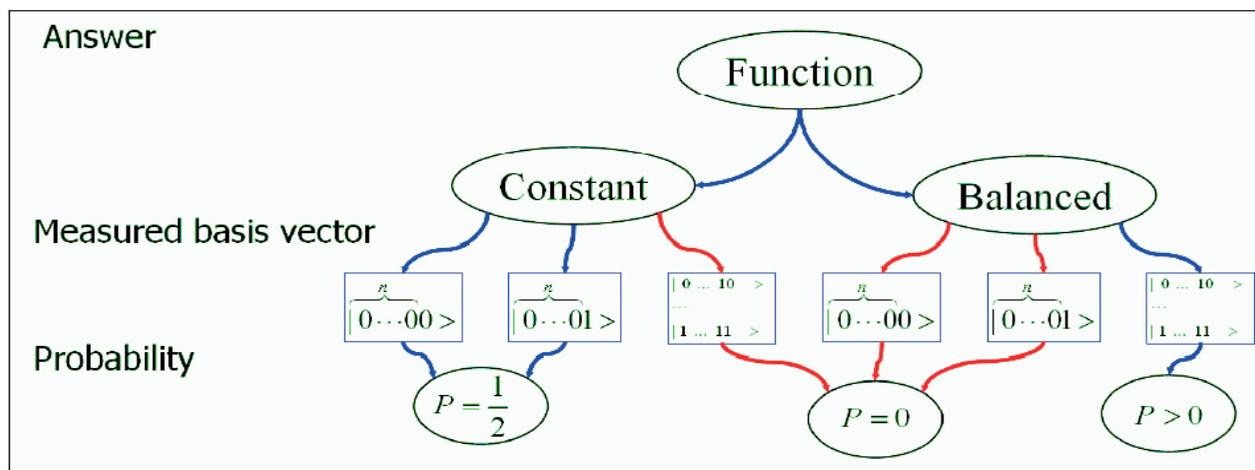


Р и с. 25. Результат выполнения алгоритма при постоянной входной функции
F i g. 25. The algorithm execution result with a constant input function



Р и с. 26. Результат выполнения алгоритма при сбалансированной входной функции
F i g. 26. The algorithm execution result with a balanced input function

Интерпретация результата показана на рис. 27.



Р и с. 27. Интерпретация результатов
F i g. 27. The results interpretation

В квантовом алгоритме принятия решений Deutsch — Jozsa операторы интерференции (${}^n H \cdot I$) и суперпозиции (${}^{n+1} H$) имеют совершенно разные математическую и физическую структуры:

- оператор интерференции извлекает качественную информацию о свойстве входной функции при применении оператора ${}^n H$ и выделяет это качество с помощью оператора идентичности I ;
- с помощью оператора суперпозиции происходят параллельные вычисления, робастное извлечение качества функции происходит за счет применения оператора запутанности. Для алгоритма Deutsch — Jozsa достаточно только одной итерации без количественной оценки качеств функции f для выявления искомого качества функции с вероятностью до $2/3$.

Выводы

В настоящее время идет бурное изучение возможностей и необходимости применения квантовых вычислений для разработки базиса квантовой программной инженерии⁵ [1], [2-5], [22-25]. Необходимость построения квантового компьютера и разработка схемы квантовых вычислений возникает по двум причинам. Первая причина — технологическая. Фактическое развитие полупроводниковых технологий и технологий изготовления больших интегральных схем неизбежно приводит к тому, что для записи бита классической информации требуются все более и более микроскопические объекты, приходя, по

существу, к отдельным атомам и молекулам. Поведение этих объектов уже не укладывается в рамки классического описания, и потому приходится либо считать, что достигнут технологический предел миниатюризации, либо решать проблему организации вычислений на квантовых объектах.

Вторая причина, которая стимулирует исследование в области квантовых компьютеров — это проявляющиеся принципиальные ограничения, которые возникают при использовании классических компьютеров, когда речь идет о возрастающем числе данных и экспоненциальном росте времени вычисления для многих практически интересных и важных классических алгоритмов.

В данной работе был продемонстрирован один из подходов к реализации квантовых алгоритмов принятия решений Deutsch и Deutsch — Jozsa. Данный алгоритм позволяет определить качественные характеристики функции без количественной оценки всего за одну итерацию. К достоинствам выбранного подхода можно отнести: наглядность представления входных данных, операторов и результатов их применения, простота реализации. Однако есть существенный недостаток: это экспоненциально большое время вычислений и экспоненциально большой объем занимаемой памяти, связанные с прямым матричным представлением квантовых операторов. В связи с этим при моделировании данным подходом квантовый алгоритм может оперировать только с небольшим входным регистром, следствием чего является узкий диапазон входных данных, обрабатываемых алгоритмом.

⁵ Satanassi S. Quantum computers for high school: design of activities for an I SEE teaching module. Scuola di Scienze Dipartimento di Fisica e Astronomia Corso di Laurea Magistrale in Fisica: Dissertation. Bologna : Alma Mater Studiorum Universit' a di Bologna, 2018. URL: https://iseeproject.eu/wp-content/uploads/2017/04/Quantum-computers-for-high-school_-_SS.pdf (дата обращения: 01.04.2023).



References

- [1] Bagdasaryn N., Korenkov V., Reshetnikov P., Tyatyushkina O., Ulyanov S. Nonstandard logic of education background in end-to-end information technologies and cognitive computing. Pt. 1: Bcon's Problem, Big Data Analytic and Intelligent Model Theory of Cognition Physical Processes. *Sistemnyj analiz v nauke i obrazovanii* = System Analysis in Science and Education. 2019;(1):1-38. (In Russ., abstract in Eng.) EDN: SOJBID
- [2] Serrano M.A., Pérez-Castillo R., Piattini M. Quantum Software Engineering. Cham: Springer; 2022. 302 p. <https://doi.org/10.1007/978-3-031-05324-5>
- [3] Ivancova O., Korenkov V., Ryabov N., Ulyanov S. Quantum Software Engineering: Quantum Gate-Based Computational Intelligence Supremacy. In: Voevodin V., Sobolev S. (eds.) Supercomputing. RuSCDays 2020. *Communications in Computer and Information Science*. Vol. 1331. Cham: Springer; 2020. p. 110-121. https://doi.org/10.1007/978-3-030-64616-5_10
- [4] Ulyanov S.V., Tyatyushkina O.Yu., Korenkov V.V. Quantum software engineering Pt II: Quantum computing supremacy on quantum gate-based algorithm models. *Sistemnyj analiz v nauke i obrazovanii* = System Analysis in Science and Education. 2021;(1):81-129. EDN: EIMGEI
- [5] Ulyanov S.V., Reshetnikov A.G. Cognitive intelligent robust control system based on quantum fuzzy inference for robotics and mechatronics. In : 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY). Subotica, Serbia: IEEE Computer Society; 2017. p. 000255-000260. <https://doi.org/10.1109/SISY.2017.8080563>
- [6] Fox M.F.J., Zwickl B.M., Lewandowski H.J. Preparing for the quantum revolution: What is the role of higher education? *Physical Review Physics Education Research*. 2020;16(2):020131. <https://doi.org/10.1103/PhysRevPhysEducRes.16.020131>
- [7] Uljanov S., Andreev E., Afanas'eva O., Barbashin M., Reznikova N., Samigulina J. Logical and quantum paradoxes of quantum and soft computational intelligence. *Sistemnyj analiz v nauke i obrazovanii* = System Analysis in Science and Education. 2010;(2):112-129. (In Russ., abstract in Eng.) EDN: MSYRIF
- [8] Ulyanov S.V., Litvintseva L.V., Mishin A.A., Sorokin S.V., Fukuda T., Tyatyushkina O.Yu., Kolbenko E.V., Nefedov N.Yu., Petrov S.P., Polunin A.S. "Paradox" of quantum self-organization of knowledge bases and robust intelligent control systems. *Nechetkie Sistemy i Myagkie Vychisleniya* = Fuzzy Systems and Soft Computing. 2011;6(1):67-106. (In Russ., abstract in Eng.) EDN: OTWEKP
- [9] Nielsen M.A., Chuang I.L. Quantum Computation and Quantum Information: 10th Anniversary Ed. Cambridge: Cambridge University Press; 2010. 702 p. <https://doi.org/10.1017/CBO9780511976667>
- [10] Benenti G., Casati G., Strini G. Principles of Quantum Computation and Information. Vol. I: Basic Concepts. Singapore: World Scientific; 2004. 272 p. <https://doi.org/10.1142/5528>
- [11] Benenti G., Casati G., Strini G. Principles of Quantum Computation and Information. Vol. II: Basic Tools and Special Topics. Singapore: World Scientific; 2007. 444 p. <https://doi.org/10.1142/5838>
- [12] Ulyanov S.V., Ulyanov V.S. Quantum Algorithmic Gate-Based Computing: Grover Quantum Search Algorithm Design in Quantum Software Engineering. arXiv:2304.13703. 2023. Available at: <https://arxiv.org/abs/2304.13703> (accessed 01.04.2023).
- [13] Abhijith J. et al. Quantum Algorithm Implementations for Beginners. *ACM Transactions on Quantum Computing* 2022;3(4):18. <https://doi.org/10.1145/3517340>
- [14] Childs A.M., Coudron M., Gilani A.S. Quantum Algorithms and the Power of Forgetting. In: 14th Innovations in Theoretical Computer Science Conference (ITCS 2023). Leibniz International Proceedings in Informatics (LIPIcs). 2023. Vol. 251. p. 37:1-37:22. <https://doi.org/10.4230/LIPIcs.ITCS.2023.37>
- [15] Aaronson S. Open problems related to quantum query complexity. *ACM Transactions on Quantum Computing*, 2021;2(4):14. <https://doi.org/10.1145/3488559>
- [16] Cumming R., Thomas T. Using a quantum computer to solve a real-world problem – what can be achieved today? arXiv:2211.13080. 2022. Available at: <https://arxiv.org/abs/2211.13080> (accessed 01.04.2023).
- [17] Lin L. Lecture Notes on Quantum Algorithms for Scientific Computation. arXiv:2201.08309. 2022. Available at: <https://arxiv.org/abs/2201.08309> (accessed 01.04.2023).
- [18] Nyman P. Simulation of Quantum Algorithms with a Symbolic Programming Language. arXiv:0705.3333. 2007. Available at: <https://arxiv.org/abs/0705.3333> (accessed 01.04.2023).
- [19] Juliá-Díaz B., Burdis J.M., Tabakin F. QDENSITY – A Mathematica quantum computer simulation. *Computer Physics Communications*. 2009;180(3):474. <https://doi.org/10.1016/j.cpc.2008.10.006>
- [20] Johansson N., Larsson J.-Å. Efficient classical simulation of the Deutsch – Jozsa and Simon's algorithms. *Quantum Information Processing*. 2017;16:233. <https://doi.org/10.1007/s11128-017-1679-7>
- [21] Li H., Qiu D., Leo L. Distributed exact quantum algorithms for Deutsch-Jozsa problem. arXiv:2303.10663. 2023. Available at: <https://arxiv.org/abs/2303.10663> (accessed 01.04.2023).
- [22] Satanassi S., Fantini P., Spada R., Levrini O. Quantum Computing for high school: an approach to interdisciplinary in STEM for teaching. *Journal of Physics: Conference Series*. 2021;1929:012053. <https://doi.org/10.1088/1742-6596/1929/1/012053>
- [23] Seegerer S., Michaeli T., Romeike R. Quantum Computing As a Topic in Computer Science Education. In: Proceedings of the 16th Workshop in Primary and Secondary Computing Education (WiPSCE '21). New York, NY, USA: Association for Computing Machinery; 2021. Article number: 13. <https://doi.org/10.1145/3481312.3481348>



- [24] Kushimo T., Thacker B. Investigating students' strengths and difficulties in Quantum Computing. arXiv:2212.03726. 2022. Available at: <https://arxiv.org/abs/2212.03726> (accessed 01.04.2023).
- [25] Angara P.P., Stege U., MacLean A., Müller H.A., Markham T. Teaching Quantum Computing to High-School-Aged Youth: A Hands-On Approach. *IEEE Transactions on Quantum Engineering*. 2022;3:3100115. <https://doi.org/10.1109/TQE.2021.3127503>

*Поступила 01.04.2023; одобрена после рецензирования 21.05.2023; принята к публикации 03.06.2023.
Submitted 01.04.2023; approved after reviewing 21.05.2023; accepted for publication 03.06.2023.*

Об авторах:

Ульянов Виктор Сергеевич, доцент кафедры информационных технологий, ФГБОУ ВО «Московский государственный университет геодезии и картографии» (105064, Российская Федерация, г. Москва, Гороховский пер., д. 4), кандидат технических наук, доцент, **ORCID: <https://orcid.org/0000-0002-6600-2602>**, ulyanovik@mail.ru

Ульянов Сергей Викторович, профессор кафедры системного анализа и управления Института системного анализа и управления, ГБОУ ВО Московской области «Университет «Дубна» (141982, Российская Федерация, Московская область, г. Дубна, ул. Университетская, д. 19); главный научный сотрудник Лаборатории информационных технологий имени М.Г. Мещерякова, Международная межправительственная организация Объединенный институт ядерных исследований (141980, Российская Федерация, Московская область, г. Дубна, ул. Жолио-Кюри, д. 6), доктор физико-математических наук, профессор, **ORCID: <https://orcid.org/0000-0001-7409-9531>**, ulyanovsv46_46@mail.ru

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the authors:

Viktor S. Ulyanov, Associate Professor of the Department of Computer Science, Moscow State University of Geodesy and Cartography (4, Gorokhovskiy per., Moscow 105064, Russian Federation), Cand. Sci. (Eng.), Associate Professor, **ORCID: <https://orcid.org/0000-0002-6600-2602>**, ulyanovik@mail.ru

Sergey V. Ulyanov, Professor of the Department of System Analysis and Management, Institute of System Analysis and Management, Dubna State University (19 Universitetskaya St., Dubna 141980, Moscow Region, Russian Federation); Chief researcher of the Meshcheryakov Laboratory of Information Technologies, Joint Institute for Nuclear Research (6 Joliot-Curie St., Dubna 141980, Moscow region, Russian Federation), Dr. Sci. (Phys.-Math.), Professor, **ORCID: <https://orcid.org/0000-0001-7409-9531>**, ulyanovsv46_46@mail.ru

All authors have read and approved the final manuscript.

