

УДК 004.415.2, 004.588

МОДЕЛИРОВАНИЕ НА КЛАССИЧЕСКОМ КОМПЬЮТЕРЕ КВАНТОВОГО АЛГОРИТМА ПРИНЯТИЯ РЕШЕНИЯ ДЕЙЧА

Полунин Алексей Сергеевич¹, Ульянов Сергей Викторович²

¹ Студент;

ГОУ ВПО «Международный Университет природы, общества и человека «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: Aleksey_Polunin@mail.ru.

² Доктор физико-математических наук, профессор;

ГОУ ВПО «Международный Университет природы, общества и человека «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: ulyanovsv@mail.ru.

Демонстрируется эффективность моделирования квантового алгоритма принятия решения Дейча на классическом компьютере.

Ключевые слова: квантовый алгоритм, Дейча, классический компьютер, моделирование.

DECISION-MAKING QUANTUM ALGORITHM MODELLING OF DEUTCH ON THE CLASSICAL COMPUTER

Polunin Aleksey¹, Ulyanov Sergey²

¹ Student;

Dubna International University of Nature, Society, and Man,
Institute of system analysis and management;
141980, Dubna, Moscow reg., Universitetskaya str., 19;
e-mail: Aleksey_Polunin@mail.ru.

² Doctor of Science in Physics and Mathematics, professor;

Dubna International University of Nature, Society, and Man,
Institute of system analysis and management;
141980, Dubna, Moscow reg., Universitetskaya str., 19;
e-mail: ulyanovsv@mail.ru.

Simulation effectiveness of Deutsch quantum decision making algorithm on classical computer is demonstrated.

Keywords: Deutch quantum algorithm, classical computer, simulation.

Введение

Целью данной статьи является эффективная демонстрация моделирования квантового алгоритма принятия решения, представленного в форме квантовой схемы, через преобразования автоматически в соответствующую программируемую квантовую ячейку (КЯ). Математической моделью данной КЯ является матричный оператор. При применении векторного представления квантового регистра входного состояния КЯ, полученный выходной результат будет являться векторным представлением нужного состояния выходного регистра (см. рис. 1).



Рис. 1. Блочный подход к моделированию квантовых алгоритмов на классических компьютерах

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 (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. Эти операции обобщают все необходимые действия, применяемые в квантовых поисковых алгоритмах, реализуемых на классических компьютерах. Мы продемонстрируем применение данного подхода к моделированию квантовых алгоритмов на классических компьютерах. В качестве примера вычислений будет рассмотрено моделирование квантового алгоритма Дейча.

Моделирование квантового алгоритма Дейча

Сформулируем задачу Дейча [1, 2].

Проблема Дейча

Входом квантового алгоритма 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 Дейча формулируется следующим образом:

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

Мы можем различить 4 возможных варианта функций $f_i : \{0,1\} \rightarrow \{0,1\}$. Они 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

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:

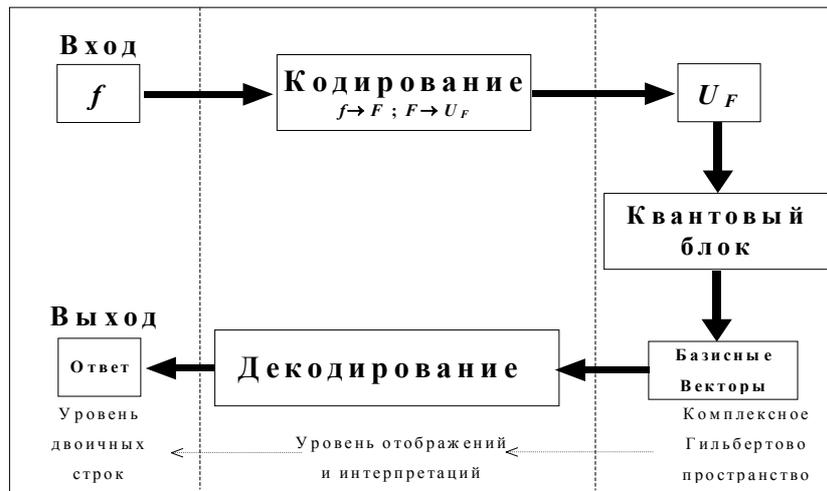


Рис. 2. Схематическая диаграмма квантовых алгоритмов

Кодирование

Блок кодирования преобразует функцию f в унитарную матрицу U_F . 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 $f = f_3$, то 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 :

$$\begin{aligned} U_F|00\rangle &= 1|00\rangle + 0|01\rangle + 0|10\rangle + 0|11\rangle \\ U_F|01\rangle &= 0|00\rangle + 1|01\rangle + 0|10\rangle + 0|11\rangle \\ U_F|10\rangle &= 0|00\rangle + 0|01\rangle + 0|10\rangle + 1|11\rangle \\ U_F|11\rangle &= 0|00\rangle + 0|01\rangle + 1|10\rangle + 0|11\rangle. \end{aligned}$$

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 $|j\rangle$ быть преобразованным 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	0	1
$ 11\rangle$	0	0	1	0

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):

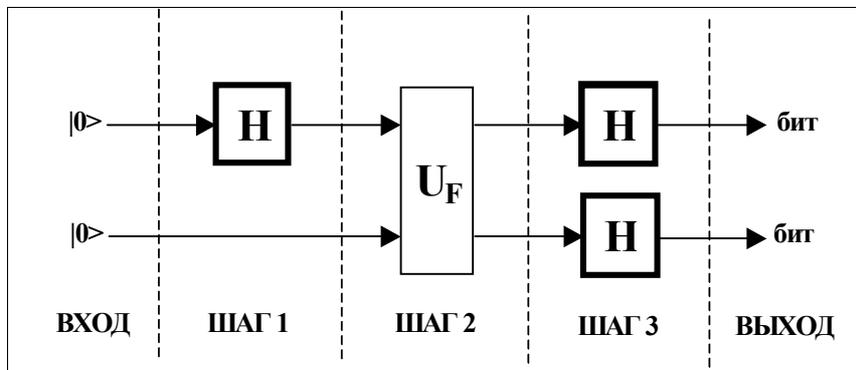


Рис. 3. Схематическое представление алгоритма Дейча

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. Толстые прямоугольники 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 КЯ. 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. 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:

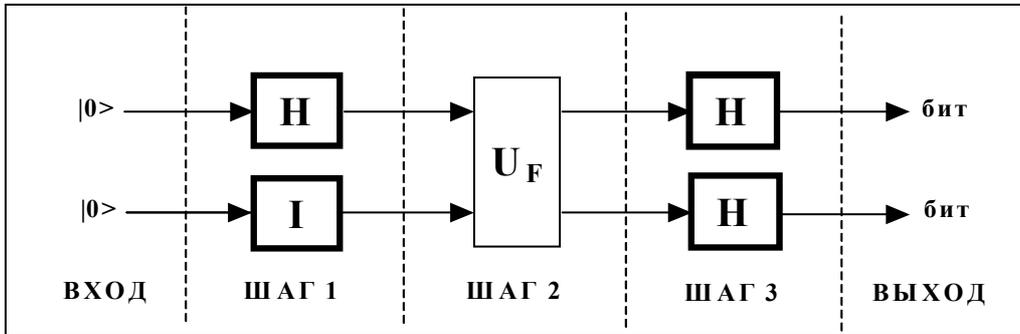


Рис. 4. Квантовый алгоритм Дейча (Второе схематическое представление)

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

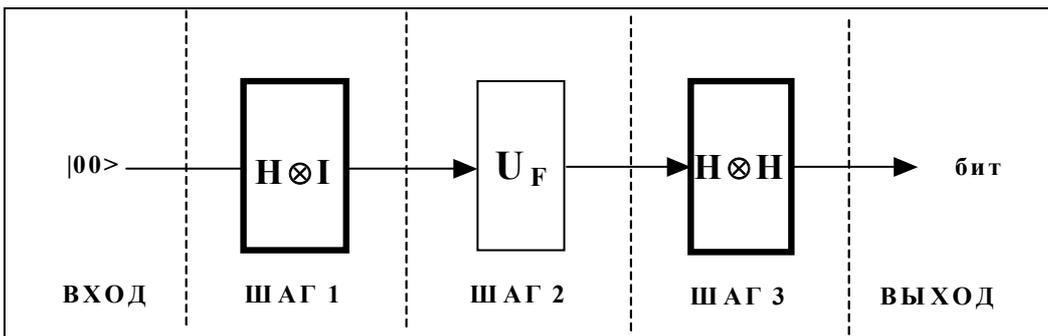


Рис. 5. Квантовый алгоритм Дейча (Третье схематическое представление)

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 (точнее программируемую ячейку для реализации алгоритма Дейча):

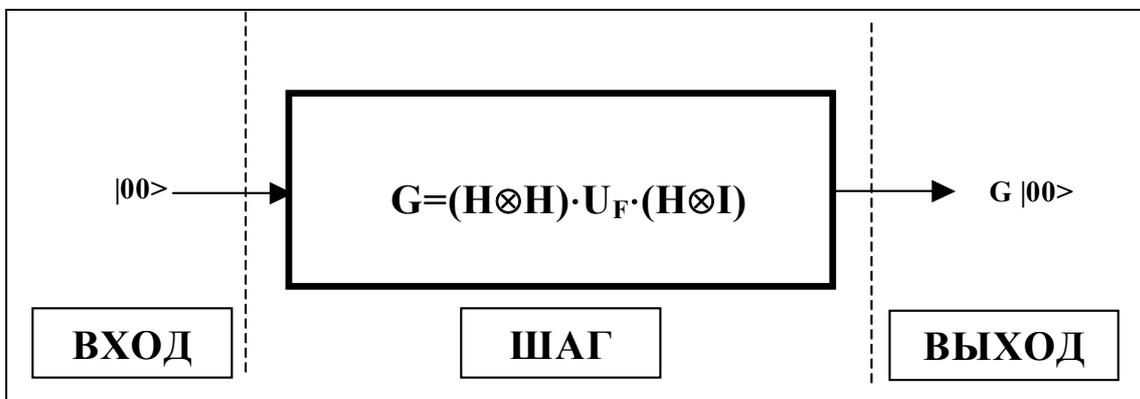


Рис. 6. Квантовый алгоритм Дейча (Окончательное представление)

Формирование КЯ

1. Вычисляем $(H \otimes I)$. Получаем матрицу, размерности 4×4 . Присваиваем строкам и столбцам полученной матрицы соответствующие базисные вектора. Амплитуды вероятностей отображения одного базисного вектора в другой под действием оператора $(H \otimes I)$ могут быть получены на пересечении соответствующих строк и столбцов.

Рассмотрим для примера вектор $|0\rangle$. Его амплитуда вероятности отображения в вектор $|01\rangle$ 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$
$ 0\rangle$	1	0
$ 1\rangle$	0	1

H	$ 0\rangle$	$ 1\rangle$
$ 0\rangle$	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$
$ 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 $f = f_3$:

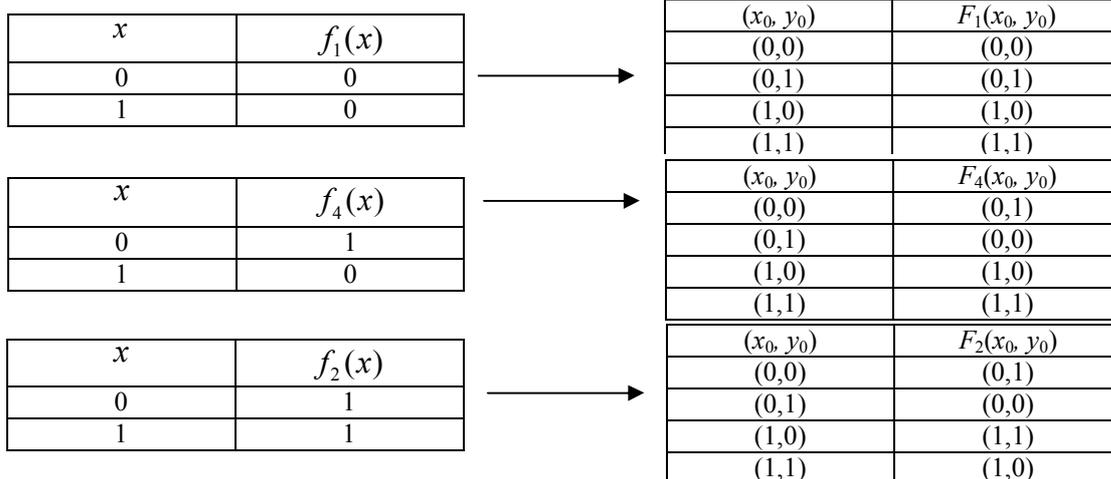
U_{F_3}	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$ 00\rangle$	1	0	0	0
$ 01\rangle$	0	1	0	0
$ 10\rangle$	0	0	0	1
$ 11\rangle$	0	0	1	0

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

G_3	$ 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

Для остальных вариантов исходных функций, квантовые ячейки вычисляются соответственно:



Для каждого отображения легко строятся соответствующие матричные операторы:

$ x_0y_0\rangle$	$U_{F_1} x_0y_0\rangle$
$ 00\rangle$	$ 00\rangle$
$ 01\rangle$	$ 01\rangle$
$ 10\rangle$	$ 10\rangle$
$ 11\rangle$	$ 11\rangle$

 \rightarrow

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_0y_0\rangle$	$U_{F_2} x_0y_0\rangle$
$ 00\rangle$	$ 01\rangle$
$ 01\rangle$	$ 00\rangle$
$ 10\rangle$	$ 11\rangle$
$ 11\rangle$	$ 10\rangle$

 \rightarrow

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_0y_0\rangle$	$U_{F_4} x_0y_0\rangle$
$ 00\rangle$	$ 01\rangle$
$ 01\rangle$	$ 00\rangle$
$ 10\rangle$	$ 10\rangle$
$ 11\rangle$	$ 11\rangle$

 \rightarrow

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:

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

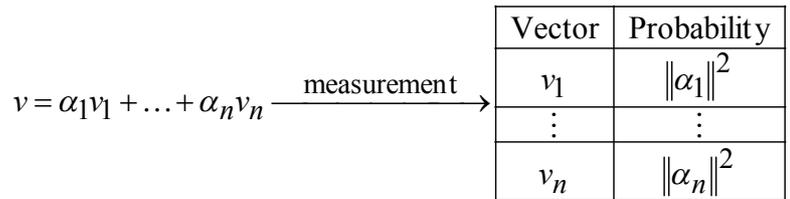
Это не возможно для $G_3|00\rangle$ и $G_4|00\rangle$. Говорят, что эти два вектора образуют смешанное (entangled) состояние.

Определение 1: Вектор v размерности 2^n образует смешанное (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 КЯ 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 :



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

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

$G_3 00\rangle = \frac{1}{\sqrt{2}} 00\rangle + \frac{1}{\sqrt{2}} 11\rangle$	$ 00\rangle$ $ 11\rangle$	$\ \frac{1}{\sqrt{2}}\ ^2=0.5$ $\ \frac{1}{\sqrt{2}}\ ^2=0.5$
$G_4 00\rangle = \frac{1}{\sqrt{2}} 00\rangle - \frac{1}{\sqrt{2}} 11\rangle$	$ 00\rangle$ $ 11\rangle$	$\ \frac{1}{\sqrt{2}}\ ^2=0.5$ $\ \frac{1}{\sqrt{2}}\ ^2=0.5$

После измерения, квантовый блок заканчивает свою работу.

Таким образом, только один базисный вектор сохраняется.

В отличие от классического алгоритма определения свойства функции за четыре шага, в алгоритме Дейча квантовый блок определяет искомое качественное свойство функции только за одну итерацию.

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, с вероятностью 50% квантовый алгоритм 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.

Моделирование алгоритма Дейча

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 в процессе выполнения алгоритма.

Структура программы позволяет дальнейший переход к более сложным квантовым алгоритмам, таким как алгоритм Дейча-Йосса, поисковый алгоритм Гровера и факторизации Шора с небольшими

изменениями в исходном коде. В виду ограничения объема данного материала, полный текст программы не приводится.

На рис. 7 показаны операторы суперпозиции интерференции:

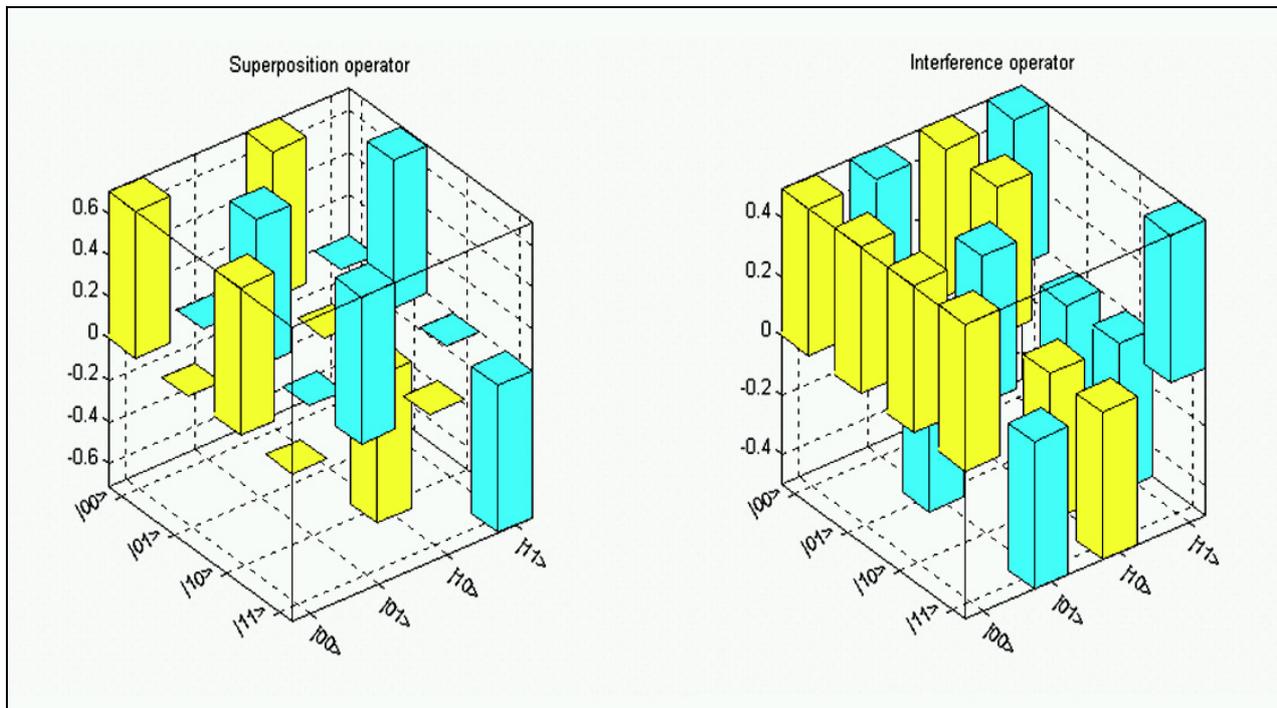


Рис. 7. Операторы суперпозиции и интерференции

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.

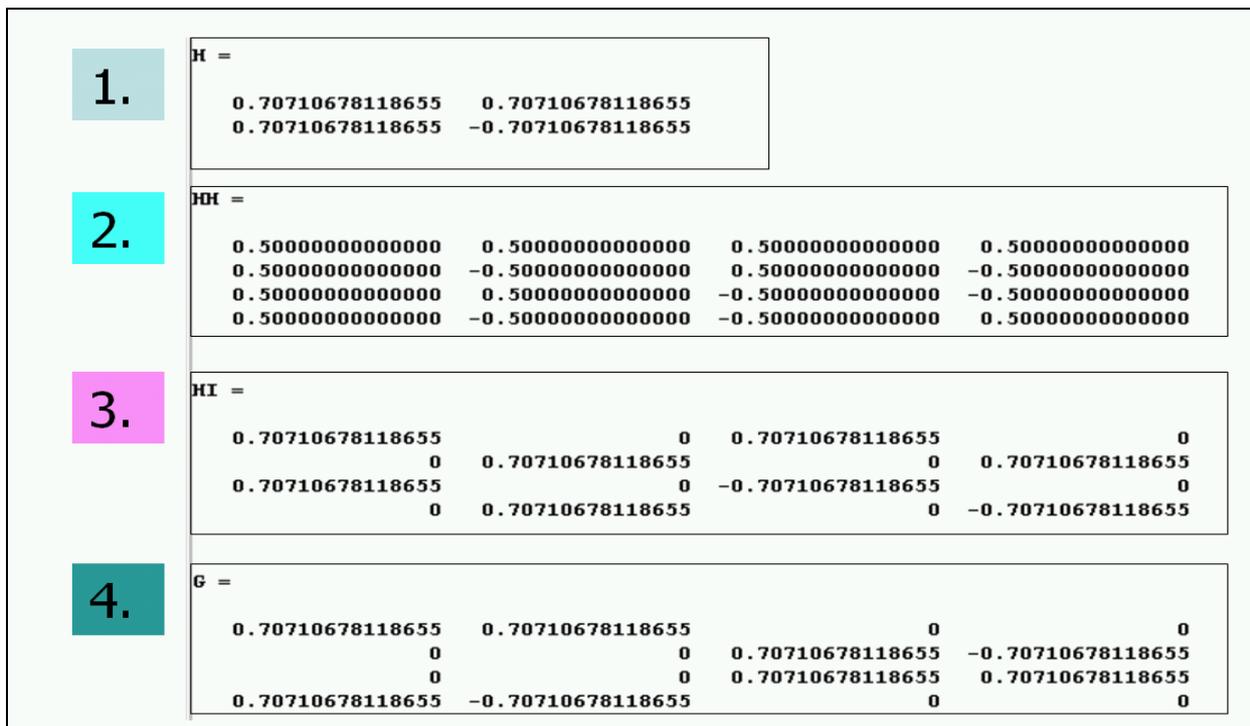


Рис. 8. Вычисление квантовых операторов КЯ

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

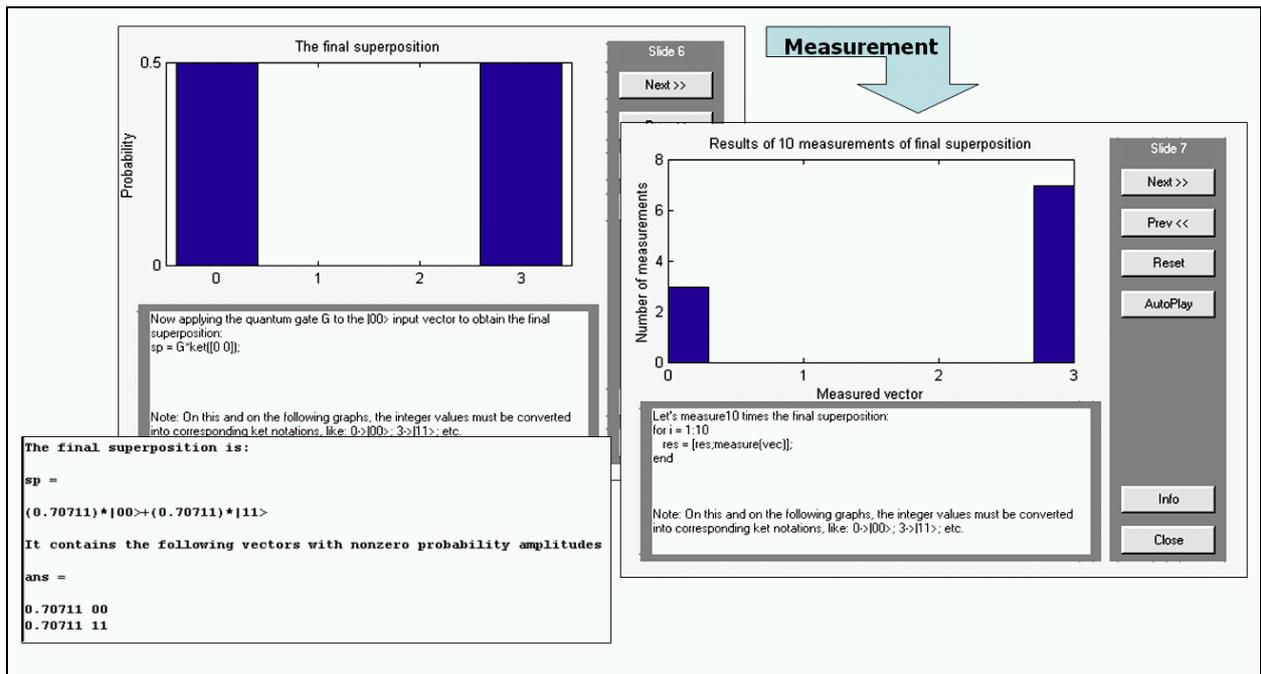


Рис. 9. Применение оператора G в КЯ для входного вектора $|00\rangle$ и его измерение

Рисунок 10 показывает содержательную интерпретацию выходного результата (декодирование).

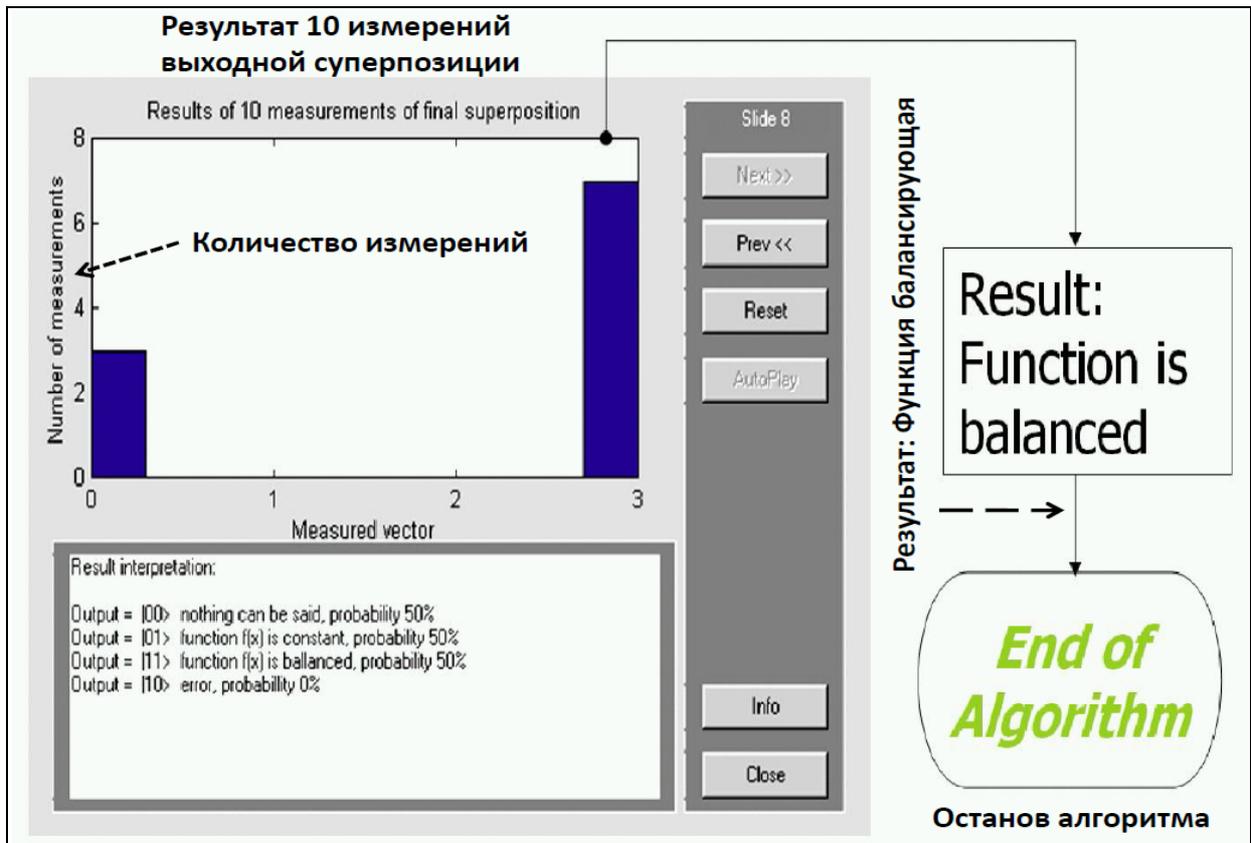


Рис. 10. Интерпретация результата (декодирование)

Выводы

Показано, что нетривиальные эффекты квантовых алгоритмов возможно моделировать эффективно на классическом компьютере. Так, например, в рассмотренном примере, с вероятностью 50% квантовый алгоритм 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.

Список литературы

1. Benenti G., Casati G., Strini G. Principles of quantum computation and information // Singapore: World Scientific. – 2004. – Vol. I. – 2007. – Vol. II.
2. Ulyanov S.V., Litvintseva L.V. Ulyanov S.S. Quantum information and quantum computational intelligence: Design & classical simulation of quantum algorithm gates. – Universita degli Studi di Milano: Polo Didattico e di Ricerca di Crema Publ. – 2005. – Vol. 80.