# Квантовые вычисления
# Глава 3

Сысоев Сергей Сергеевич

Санкт-Петербургский государственный университет
Математико-механический факультет
http://se.math.spbu.ru/

Цель этой главы – познакомить вас с некоторыми простейшими квантовыми алгоритмами, демонстрирующими преимущества квантовых вычислений над классическими. Кроме того, мы разберем простейший демонстрационный прототип квантового компьютера на двух кубитах, реализующий алгоритм Дэвида Дойча – первый алгоритм, разработанный для модели квантовых вычислений.

## 1 Задача Дойча

Итак, задача Дойча. Предположим, что у нас есть черный ящик, реализующий функцию $f$, отображающую один бит в один бит ($f : \{0,1\} \to \{0,1\}$).

Черный ящик – это общепринятая метафора, означающая, что механизм устройства внутри ящика недоступен для анализа, и про функцию $f$ мы ничего не можем узнать. Мы можем только вызывать ее от разных параметров, обращаясь к черному ящику, как к оракулу.

Функций, отображающих один бит в один бит всего 4:

$$\begin{aligned} f(x) &= 0 \\ f(x) &= 1 \\ f(x) &= x \\ f(x) &= \bar{x} \end{aligned} \tag{1}$$

Две из них константы – 0 и 1, и две сбалансированных – тождественная и NOT. Постановка задачи Дойча такова: является ли функция $f$, реализованная в черном ящике константой?

При этом, по коварному замыслу производителя черный ящик с интересующей нас функцией работает очень медленно – один вызов $f$ он осуществляет за 24 часа. Сколько нам нужно времени, чтобы ответить на вопрос задачи Дойча?

Ясно, что одного вызова функции недостаточно. Нужно вызвать ее два раза – по одному для каждого из возможных параметров. Посмотрим, помогут ли нам квантовые вычисления справиться с этой задачей быстрее.

Мы помним, что эволюция квантовой системы унитарна. Следовательно, вместо функции $f$ в черном ящике должен быть реализован унитарный

оператор $U_f$, который, в отличии от функции $f$ должен быть обратим (иначе он не будет унитарным).

Квантовый оракул, реализующий функцию $f$, можно определить, например, вот так:

$$U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle \tag{2}$$

В отличие от функции $f$, отображающей один бит в один бит, оператор $U_f$ работает в пространстве двух кубитов. При этом кубит $|x\rangle$, несущий значение аргумента, выполняет роль сохранения прообраза $f(x)$ для обеспечения обратимости и не затрагивается действием оператора. Оператор $U_f$ выполняет перестановку базисных векторов и не «склеивает» разные базисные вектора. Следовательно, он унитарен.

Кроме того, он несет в себе всю информацию о функции. Если в регистр $y$ мы положим 0, то для всех разных значений $x$ мы можем получить значение $f(x)$ в этом регистре.

$$\begin{aligned} U_f |0\rangle |0\rangle &\to |0\rangle |f(0)\rangle \\ U_f |1\rangle |0\rangle &\to |1\rangle |f(1)\rangle \end{aligned} \tag{3}$$

Поскольку оператор должен быть определен на всех базисных векторах, нам придется определить $U_f$ и для векторов, содержащих 1 в регистре $y$. При этом нельзя допустить, чтобы образы векторов $|x\rangle |0\rangle$ и $|x\rangle |1\rangle$ оказались одним вектором вида $|x\rangle |y\rangle$, и определение (2) предотвращает подобное "склеивание" векторов:

$$\begin{aligned} U_f |0\rangle |1\rangle \to |0\rangle |1 \oplus f(0)\rangle \neq |0\rangle |f(0)\rangle = U_f |0\rangle |0\rangle \\ U_f |1\rangle |1\rangle \to |1\rangle |1 \oplus f(1)\rangle \neq |1\rangle |f(1)\rangle = U_f |1\rangle |0\rangle \end{aligned} \tag{4}$$

Определив оракул $U_f$ таким образом, мы получаем еще одно полезное свойство. Посмотрим, как $U_f$ действует на состояния вида $|x\rangle (|0\rangle - |1\rangle)$:

$$\begin{aligned} U_f \frac{1}{\sqrt{2}} |x\rangle (|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}} |x\rangle (|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle) = \\ = \frac{1}{\sqrt{2}} (-1)^{f(x)} |x\rangle (|0\rangle - |1\rangle) \end{aligned} \tag{5}$$

Получается, что для такого состояния $U_f$ не изменяет регистр $y$. Вместо этого регистр $x$ домножается на -1, если $f(x) = 1$.

В отличие от классического случая, где мы насчитали всего 4 функции вида $\{0,1\} \to \{0,1\}$, в квантовых вычислениях существует бесконечно много двухкубитных операторов. Рассмотрим матрицы и схемы оракулов для различных функций $f$.

**Константа (f(x) = 0).**

$$U_f |x\rangle |y\rangle = |x\rangle |y \oplus 0\rangle = |x\rangle |y\rangle$$

$$U_f = I$$


**Fig. 1.** Схема оператора $U_f$. f(x) = 0.

Схема устройства проста – пустой набор квантовых гейтов (fig. 1).
**Константа (f(x) = 1).**

$$\begin{aligned} |00\rangle &\to |01\rangle \\ |01\rangle &\to |00\rangle \\ |10\rangle &\to |11\rangle \\ |11\rangle &\to |10\rangle \end{aligned} \tag{6}$$

$$U_f = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = I \otimes X$$

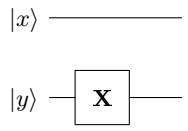


**Fig. 2.** Схема $U_f$. f(x) = 1.

**Сбалансирована. (f(x) = x.)**

$$\begin{aligned} |00\rangle &\to |00\rangle \\ |01\rangle &\to |01\rangle \\ |10\rangle &\to |11\rangle \\ |11\rangle &\to |10\rangle \end{aligned} \tag{7}$$

$$U_f = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = CNOT$$

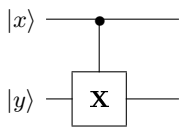


**Fig. 3.** Схема $U_f$. f(x) = x.

**Сбалансирована. ($f(x) = \bar{x}$.)**

$$\begin{aligned} |00\rangle &\to |01\rangle \\ |01\rangle &\to |00\rangle \\ |10\rangle &\to |10\rangle \\ |11\rangle &\to |11\rangle \end{aligned} \tag{8}$$

$$U_f = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

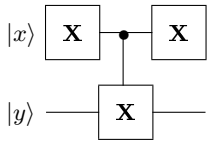


**Fig. 4.** Схема $U_f$. $f(x) = \bar{x}$.

**Схема устройства.**
Алгоритм, решающий задачу Дойча, выглядит следующим образом (5).

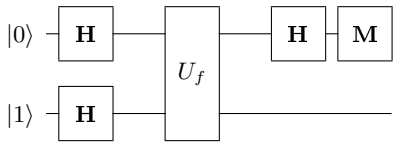


**Fig. 5.** Алгоритм Дойча.

На вход устройства подается 2 кубита, $|0\rangle$ в регистре $x$ и $|1\rangle$ в регистре $y$. К обоим кубитам применяется преобразование Адамара и неизвестный квантовый оракул (один из рассмотренных нами ранее). После этого к кубиту в регистре $x$ снова применяется преобразование Адамара. Буква $M$ на схеме означает измерение (Measurement) – измеряется только кубит в регистре $x$. Как видите, вызов оракула в этом алгоритме происходит всего один раз (вместо классических двух).

Давайте посмотрим, к чему это приведет.

$$|01\rangle \xrightarrow{H_2} \frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) =$$

$$= \frac{1}{2}|0\rangle(|0\rangle - |1\rangle) + \frac{1}{2}|1\rangle(|0\rangle - |1\rangle) \xrightarrow{U_f}$$

(см. (5))

$$\xrightarrow{U_f} \frac{1}{2}(-1)^{f(0)}|0\rangle(|0\rangle - |1\rangle) + \frac{1}{2}(-1)^{f(1)}|1\rangle(|0\rangle - |1\rangle) =$$

$$= \frac{1}{2}((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle)(|0\rangle - |1\rangle) \tag{9}$$

Нам остается только применить преобразование Адамара к первому кубиту и, после этого, его измерить.

Рассмотрим состояние первого кубита в (9). Если функция f – константа, то оба показателя степеней (-1) будут равны, и это состояние представляет собой:

$$\pm\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \pm H(|0\rangle)$$

Если же функция $f$ сбалансирована ($f(0) \neq f(1)$), то степени (-1) будут разными, и состояние будет выглядеть так:

$$\pm\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \pm H(|1\rangle)$$

Таким образом, после применения оператора Адамара к первому кубиту (мы помним, что $H^{-1} = H$), в регистре $x$ будет находится $|0\rangle$, если $f$ константа и $|1\rangle$, если $f$ сбалансирована. Измерение первого кубита в стандартном базисе позволит нам определить, с какой из этих ситуаций мы имеем дело.

На (fig. 6) представлена реализация алгоритма Дойча на симуляторе квантового компьютера ("http://qc-sim.appspot.com") для функции $f(x) = x$.

Результат измерения первого кубита – $|1\rangle$ соответствует ожидаемому для сбалансированной функции.

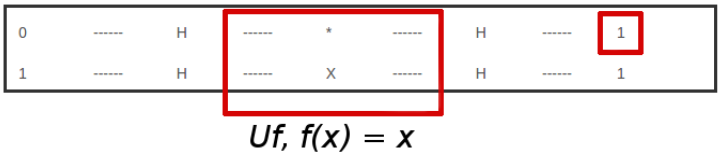


**Fig. 6.** Алгоритм Дойча на симуляторе квантового компьютера.

**Fig. 7.** Внешний вид квантового компьютера.

## 2 Квантовый компьютер на фотонах

Для демонстрации алгоритма Дойча рассмотрим простейший квантовый компьютер на двух кубитах, который вы при желании можете собрать у себя дома или в лаборатории (fig. 7).

Предлагаемый вашему вниманию прототип представляет собой набор устройств, размещенных в определенном порядке на тяжелой ровной плите из ДСП (fig. 8).

Красный лазер (1) излучает фотоны с длиной волны 650 нм. Каждый фотон будет носителем сразу двух кубитов: один кубит будет кодироваться поляризацией фотона, второй – его путем в интерферометре. Для одного «сеанса» вычислений нам, вообще говоря, достаточно одного фотона.

Линейный поляризатор (2), знакомый вам по второй главе, пропускает в интерферометр только фотоны с нужной поляризацией. Пройдя его, фотон входит в интерферометр Маха-Цендера, состоящий из двух полупрозрачных зеркал (светоделительных кубиков) (3) и двух обычных металлических зеркал (4).

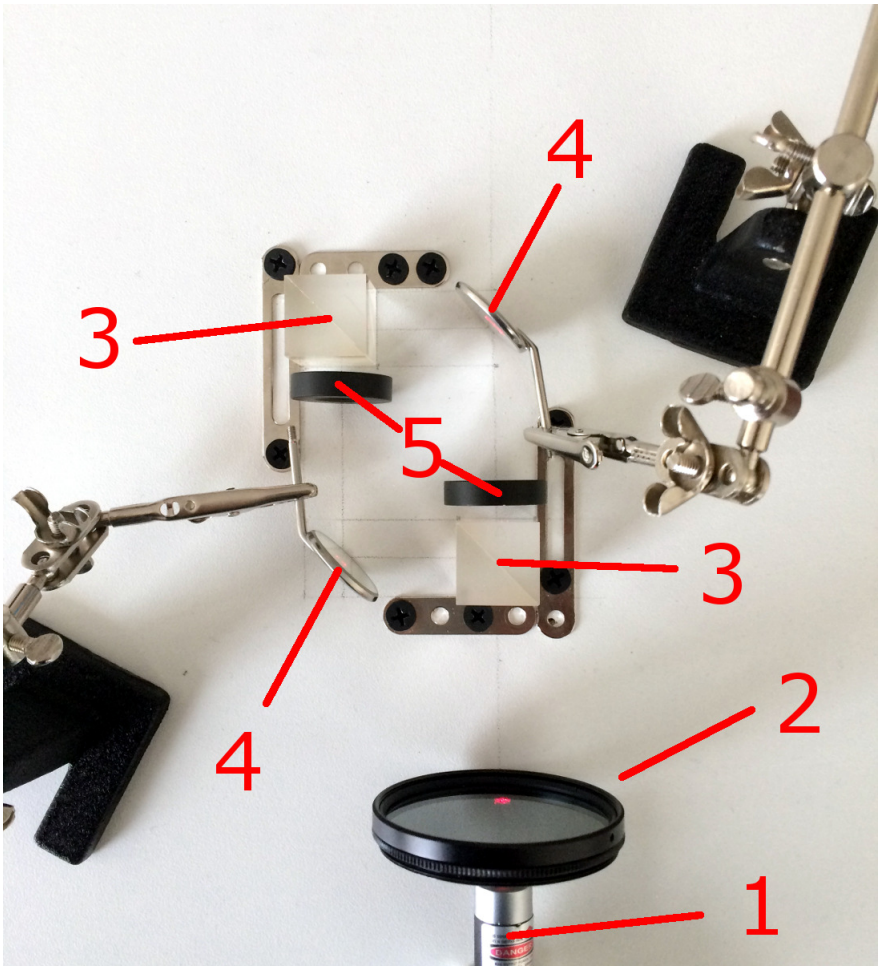


**Fig. 8.** Квантовый компьютер. Компоненты.

Обратите внимание, что светоделительные кубики неполяризационные. Они не должны никак влиять на поляризацию фотонов.

Зеркала должны быть полностью металлические, без стеклянного покрытия. Такими пользуются, например, стоматологи. Стеклянное покрытие является дополнительной отражающей поверхностью, и добавляет ненужное

нам расщепление фотона, т.к. он может отразиться и от стекла, и от металла зеркала.

Функцию оракула в компьютере выполняют две полуволновые пластины для красного света (5). О них чуть позже.

Результатом измерения является вид интерференционной картины, полученной на экране при выходе лучей из интерферометра.

Интерферометр Маха-Цендера действует следующим образом (fig. 9):

1. При прохождении полупрозрачного зеркала внутри первого кубика фотон имеет две равновероятные возможности – отразиться от зеркала и пойти по левому плечу интерферометра, или пройти его насквозь и пойти по правому плечу. Таким образом, для одного фотона до момента его измерения мы имеет суперпозицию двух состояний – фотон в левом и в правом плечах интерферометра.
2. Далее, на обоих путях стоят зеркала, отражающие «раздвоившийся» фотон на второй кубик, где для каждого из путей опять возникает две возможности – пройти насквозь или отразиться.

Рассмотрим верхний выход. В него попадает левый фотон, прошедший второй кубик насквозь и правый фотон, отразившийся от него. При этом мы помним, что это один и тот же фотон. Значит, он может проинтерферировать сам с собой, и на экране сверху (да и слева тоже) мы должны увидеть интерференционную картину.

**Полуволновая пластина. Принцип действия.**

Разберем принцип действия полуволновой пластины.

Мы уже говорили (в главе 2) о кристаллах с оптической осью. Линейные поляризаторы, например, пропускают свет, поляризованный вдоль одной оси, и не пропускают фотоны, поляризованные вдоль второй, перпендикулярной оси.

Похожим образом устроены кристаллы, используемые в волновых пластинах. У этих кристаллов тоже есть ось (fig. 10), и фотоны, поляризованные вдоль этой оси так же беспрепятственно проходят через кристалл. Луч, прошедший через кристалл по этой оси (ось "Fast"), называется "обыкновенным".

Фотоны, поляризованные вдоль ортогональной оси, тоже проходят, но замедляются кристаллом (формируют "необыкновенный" луч). При этом можно подобрать такую толщину кристалла, чтобы вышедший из него фотон необыкновенного луча запоздал по сравнению с обыкновенным на половину длины волны ($\lambda/2$). Это равносильно умножению волны на $-1$.

Давайте посмотрим, что будет с лучом, наклоненным по отношению к оси кристалла на угол $\theta$ (fig. 10). Вертикальная его составляющая проходит без изменений, а горизонтальная домножается на $-1$. Происходит отражение направления поляризации относительно оптической оси полуволновой пластины.

Носителем кубитов в рассматриваемом квантовом компьютере является один

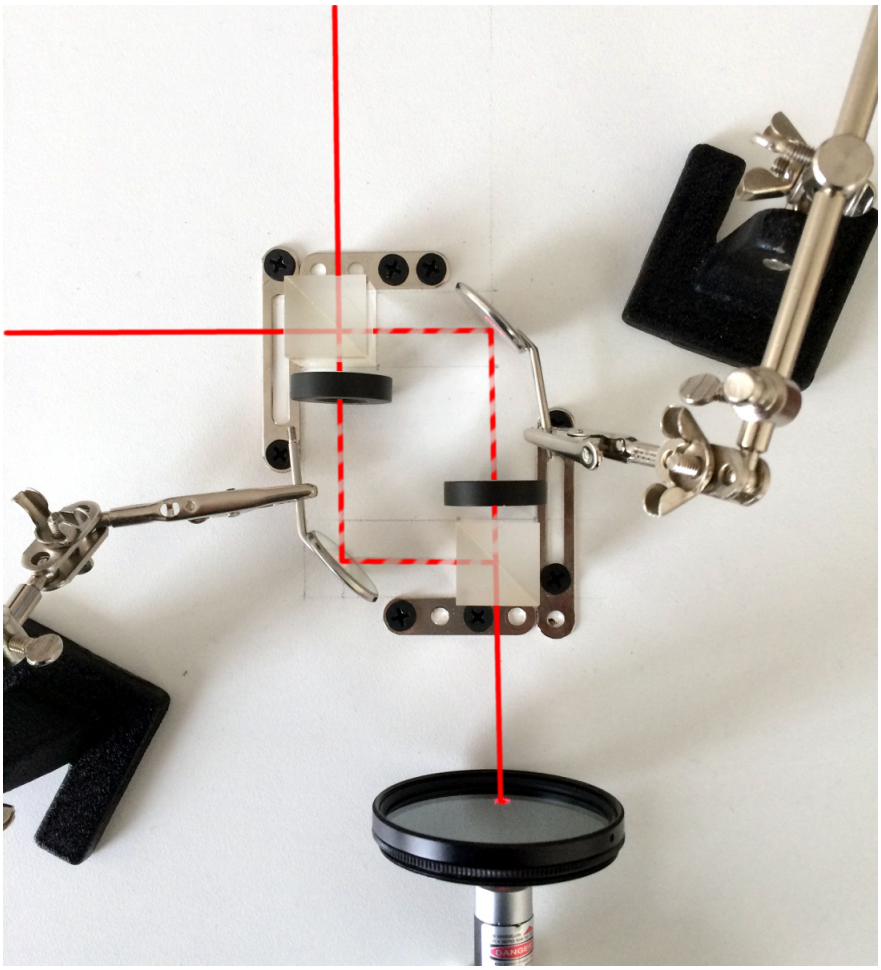

**Fig. 9.** Квантовый компьютер. Интерферометр Маха-Цендера.

фотон. Первый кубит (регистр $|x\rangle$ в задаче Дойча) кодируется путем фотона в интерферометре Маха-Цендера.
$|0\rangle$ – правое плечо.
$|1\rangle$ – левое плечо.
Второй кубит (регистр $|y\rangle$) кодируется поляризацией фотона относительно плоскости основания. $|0\rangle$ – горизонтальная поляризация.
$|1\rangle$ – вертикальная поляризация.

**Работа компьютера. Поляризатор.** Внутрь интерферометра попадают только фотоны, прошедшие поляризатор. Абсолютная ориентация поляризатора для вычислений не важна, но мы договорились о кодировании

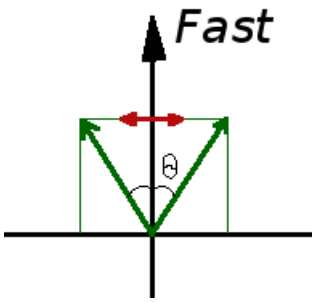


**Fig. 10.** Волновая пластина $\lambda/2$.

второго кубита его поляризацией относительно плоскости основания, поэтому мы ориентируем поляризатор под углом $-\pi/4$ к плоскости основания (fig. 11).

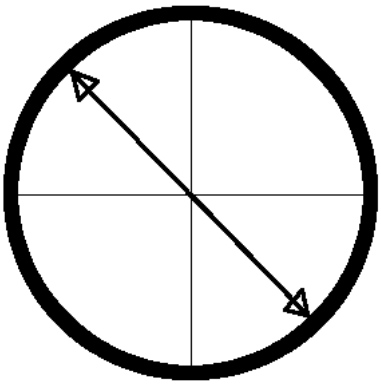

**Fig. 11.** Ориентация линейного поляризатора.

Таким образом, после прохождения поляризатора мы имеем состояние:

$$\frac{1}{\sqrt{2}}|0\rangle\left(|0\rangle - |1\rangle\right)$$

Поляризатор в этом устройстве выполняет роль преобразования Адамара на втором кубите в алгоритме Дойча.

**Работа компьютера. Первый кубик.**

Первый кубик играет роль преобразования Адамара на первом кубите. После него мы имеем состояние:

$$\frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$$

**Работа компьютера. Оракул.**

Мы прошли первый этап схемы Дойча – операторы Адамара. Настало время оракула. В задаче Дойча возможно только 4 вида квантового оракула, каждый из которых нам необходимо уметь реализовывать с помощью полуволновых пластин.

Полуволновые пластины мы ориентируем под прямым углом к поляризатору (fig. 12).

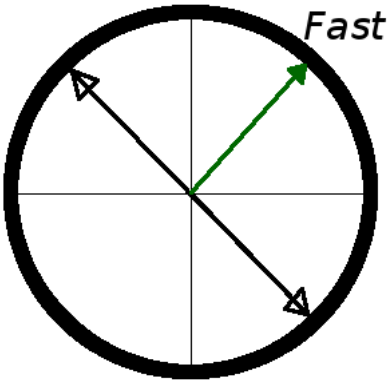


**Fig. 12.** Ориентация полуволновой пластины.

При угле $\pi/2$ с направлением поляризации фотонов, полуволновые пластины действуют, как умножение состояния на (-1) – замедление на половину длины волны:

$$|x\rangle\,(|0\rangle - |1\rangle) \rightarrow -\,|x\rangle\,(|0\rangle - |1\rangle)$$

Для $f(x) = 0$ мы имеем схему тождественного оператора (fig. 13). Полуволновых пластин не требуется ни на левом, ни на правом пути.

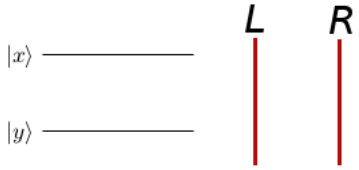


**Fig. 13.** Схема $U_f$ и ее реализация на полуволновых пластинах. $f(x) = 0$

Для $f(x) = 1$ второй кубит подвергается воздействию оператора NOT независимо от значения первого кубита (fig. 14). Полуволновые пластины поставлены на обоих путях.

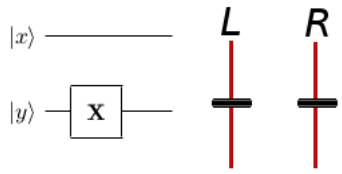


**Fig. 14.** Схема $U_f$ и ее реализация на полуволновых пластинах. $f(x) = 1$

Для $f(x) = x$ второй кубит подвергается воздействию оператора NOT только если первый кубит равен 1 – это соответствует левому плечу интерферометра (fig. 15). Полуволновая пластина ставится на левом пути.

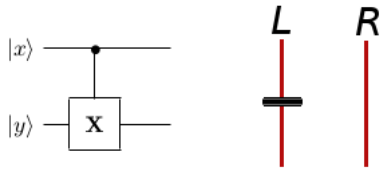


**Fig. 15.** Схема $U_f$ и ее реализация на полуволновых пластинах. $f(x) = x$

Для $f(x) = \bar{x}$ второй кубит подвергается воздействию оператора NOT только если первый кубит равен 0 – это соответствует правому плечу интерферометра (fig. 16). Полуволновая пластина ставится на правом пути.

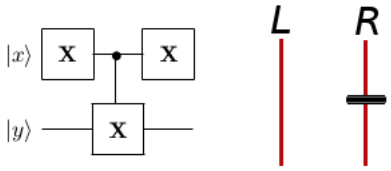


**Fig. 16.** Схема $U_f$ и ее реализация на полуволновых пластинах. $f(x) = \bar{x}$

**Работа компьютера. Второй кубик.**

Второй кубик сводит два плеча интерферометра вместе, выполняя роль преобразования Адамара на первом кубите.

**Работа компьютера. Измерение.**

Настало время измерения первого кубита. На (fig. 17) вы видите два луча, демонстрирующие ситуацию, в которой интерферометр не настроен. Настройка интерферометра осуществляется поворотом зеркал таким образом, чтобы свести эти два луча в одну точку.

После настройки мы увидим интерференционную картину, представленную на (fig. 18).

Такая интерференционная картина была получена при двух конфигурациях оракула – без полуволновых пластин, и с пластинами на обоих путях, потому что, когда пластины стоят на обоих путях, оба луча задерживаются одинаково, и это не меняет характера интерференции.

Интерференционная картина на (fig. 18) соответствует ситуации, когда f – константа.

На картинке некоторые светлые полосы отмечены гелевой ручкой.

Если полуволновая пластина стоит только на одном из плечей интерферометра ($f$ сбалансирована), то один из лучей задерживается по фазе на $\pi$, и в тех местах, где раньше были максимумы интерференционной картины, возникнут тени, так как в эти точки лучи станут приходить в противофазе.

Это мы и видим на картинке (fig. 19)!

Таким образом, настроив интерферометр и пропустив через него всего один фотон, мы можем понять, какого вида оракул сконфигурирован внутри интерферометра.

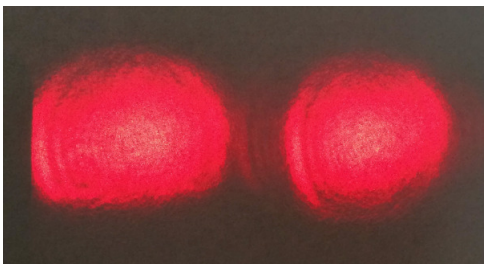

**Fig. 17.** Интерферометр не настроен.

## 3 Задача Дойча-Джозы

Рассмотрим еще несколько простых задач, которые помогут нам сформировать представление о том, что такое типичный квантовый алгоритм.

**Постановка задачи Дойча-Джозы**

Функция $f : \{0,1\}^n \to \{0,1\}$ преобразовывает n-битное число в один бит. При этом известно, что $f$ либо константа, либо сбалансирована (возвращает

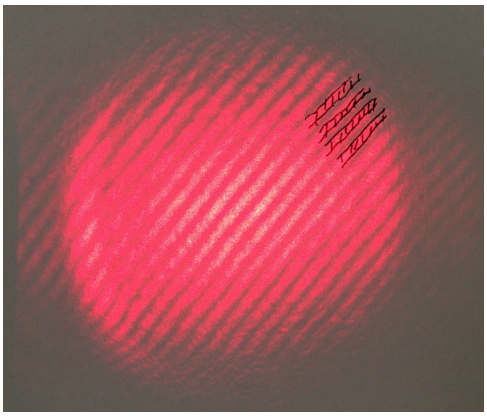

**Fig. 18.** Интерференционная картина. $f$ – константа.

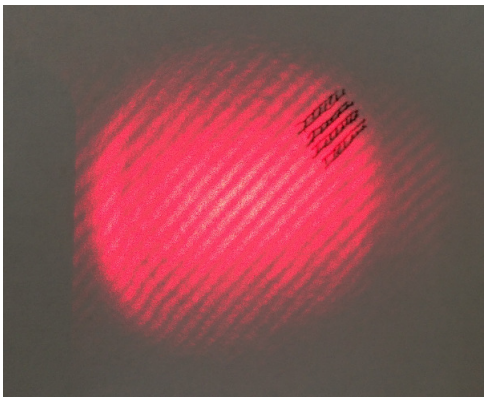

**Fig. 19.** Интерференционная картина. $f$ – сбалансирована.

1 ровно на половине своей области определения). Необходимо выяснить, какая ситуация имеет место на самом деле. $f$ реализована в виде черного ящика.

В классическом случае для того, чтобы достоверно убедиться, что $f = const$, потребуется $2^{n-1}$ обращений к оракулу. Квантовый алгоритм (fig. 20) позволяет определить тип функции за одно обращение к квантовому оракулу.

Рассмотрим действие алгоритма:

$$|0\rangle^n |1\rangle \xrightarrow{H_{n+1}} \frac{1}{2^{\frac{n+1}{2}}} \sum_{x=0}^{2^n-1} |x\rangle (|0\rangle - |1\rangle) \xrightarrow{U_f}$$

$$\xrightarrow{U_f} \frac{1}{2^{\frac{n+1}{2}}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle (|0\rangle - |1\rangle) \tag{10}$$

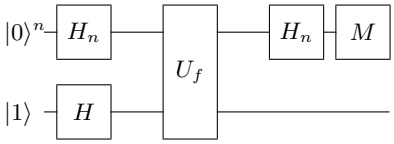


**Fig. 20.** Алгоритм Дойча-Джозы.

Если $f = const$, то $(-1)^{f(x)}$ можно вынести из под знака суммы, и первые $n$ кубитов находятся в состоянии:

$$\frac{1}{2^{n/2}}(-1)^f \sum_{x=0}^{2^n-1} |x\rangle = (-1)^f H_n |0\rangle^n$$

и, после применения оператора $H_n$ измерение входного регистра даст вектор $|0\rangle^n$ с вероятностью 1.

Если же функция сбалансирована, то состояние во входном регистре можно представить следующим образом:

$$\frac{1}{2^{n/2}}\left( \sum_{x:f(x)=0} |x\rangle - \sum_{x:f(x)=1} |x\rangle \right) \tag{11}$$

Обратим внимание, что оператор Адамара, примененный к любому вектору $|x\rangle$, возвращает сумму векторов, в которой при векторе $|0\rangle^n$ всегда стоит коэффициент $+1$:

$$H_n |x\rangle = \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} (-1)^{x \bullet y} |y\rangle = \frac{1}{2^{n/2}} (|0\rangle^n + \sum_{y=1}^{2^n-1} (-1)^{x \bullet y} |y\rangle)$$

Количество слагаемых в обеих суммах в (11) одинаково, так как функция $f$ сбалансирована. Следовательно, после применения к этим суммам оператора $H_n$, все вектора $|0\rangle^n$, получившиеся из первой суммы, взаимоуничтожатся с векторами $|0\rangle^n$, получившимися из второй суммы (поскольку перед второй суммой стоит минус). Таким образом, в результирующем состоянии не будет вектора $|0\rangle^n$, и он не может быть получен в результате измерения входного регистра.

Если в результате измерения мы получили вектор $|0\rangle^n$, то функция $f$ – константа. Если же мы получили любой другой вектор, функция $f$ сбалансирована.

На (fig. 21) приведен пример реализации алгоритма Дойча-Джозы на квантовом симуляторе для функции "четность" ($f(x) = 1 \iff x \mod 2 = 0$).

Мы видим, что результат измерения входного регистра равен $|100\rangle$, что соответствует результату "функция сбалансирована".

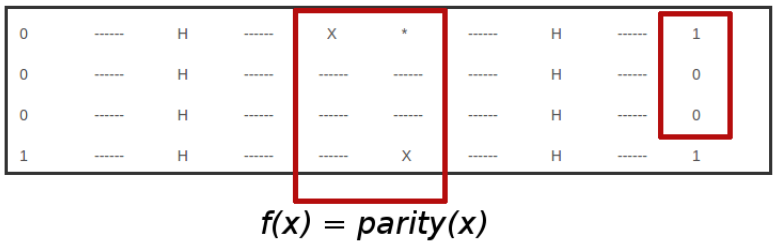


**Fig. 21.** Алгоритм Дойча-Джозы для функции "четность".

## 4 Задача Бернштейна-Вазирани

**Постановка задачи.**

В черном ящике реализована функция $f$:

$$f : \{0,1\}^n \to \{0,1\}$$

$$f(x) = a \bullet x$$

Необходимо найти число $a$.

В классическом случае вычисление каждого бита числа $a$ потребует отдельного обращения к оракулу. Тот же самый квантовый алгоритм (fig. 20), что мы использовали в задаче Дойча-Джозы, позволяет определить $a$ за одно обращение к квантовому оракулу.

Поскольку мы используем тот же алгоритм, что и в предыдущей задаче, мы не будем повторять разбор первых шагов и начнем с состояния (10), получаемого во входном регистре после вызова оракула $U_f$:

$$\frac{1}{2^{\frac{n+1}{2}}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle (|0\rangle - |1\rangle) =$$

$$= \frac{1}{2^{\frac{n+1}{2}}} \sum_{x=0}^{2^n-1} (-1)^{a \bullet x} |x\rangle (|0\rangle - |1\rangle) = H_n |a\rangle \xrightarrow{H_n}$$

$$\xrightarrow{H_n} |a\rangle$$

Таким образом, после применения схемы измерение входного регистра даст вектор $|a\rangle$.

На (fig. 22) приведен пример реализации алгоритма Бернштейна-Вазирани на квантовом симуляторе для функции $f(x) = 163 \bullet x$.

Мы видим, что результат измерения входного регистра равен $|10100011\rangle$, что является двоичным представлением числа 163.

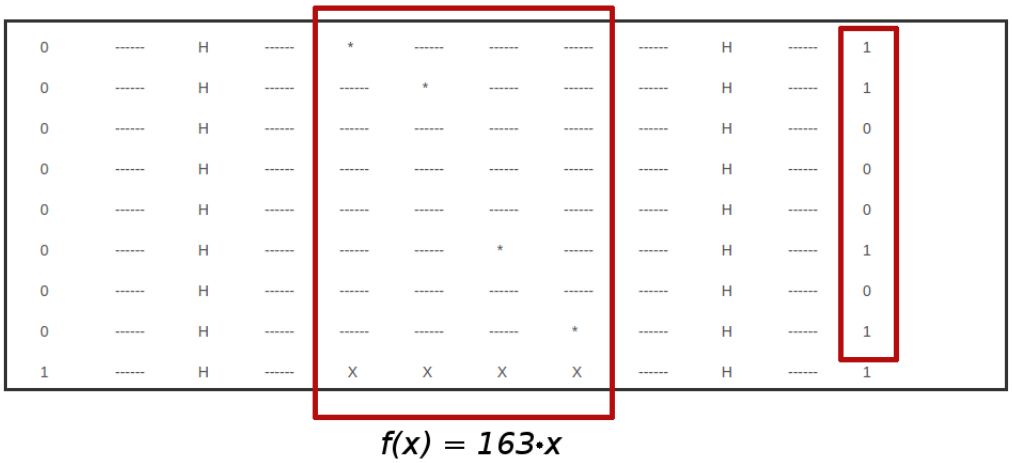


**Fig. 22.** Алгоритм Бернштейна-Вазирани. a = 163 = 0b10100011.

## 5 Задача Саймона

**Постановка задачи.**

Функция $f : \{0,1\}^n \to \{0,1\}^n$ реализована в виде черного ящика. Известно, что $f$ имеет своеобразный период $a$:

$$\exists! a \neq 0 : \ \forall x \ f(x) = f(y) \iff y = x \oplus a$$

Необходимо определить число $a$.

Для классических вычислений эта задача сложная. В худшем случае может потребоваться порядка $2^{n-1} + 1$ обращений к оракулу и столько же памяти для хранения результатов вызова функции. В среднем же потребуется $O(2^{n-2})$ вызовов $f$, если предположить, что задачу придется решать многократно для разных функций.

Квантовый алгоритм для решения задачи представлен на (fig. 23).

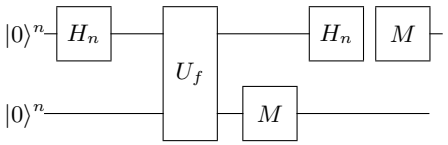


**Fig. 23.** Алгоритм Саймона.

Рассмотрим действие алгоритма:

$$|0\rangle^n |0\rangle^n \xrightarrow{H_n} \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle |0\rangle^n \xrightarrow{U_f} \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle |f(x)\rangle =$$

Для каждого значения $f(x)$ существует ровно два прообраза этого значения – $x$ и $(x \oplus a)$, поэтому мы можем вынести $f(x)$ для всех таких пар за скобки:

$$= \frac{1}{2^{n/2}} \sum_{x \neq x \oplus a} (|x\rangle + |x \oplus a\rangle) |f(x)\rangle$$

После измерения второго регистра в нем останется какое-то конкретное значение $f(x)$, а в первом регистре окажется сумма прообразов этого значения:

$$\frac{1}{\sqrt{2}}(|x\rangle + |x \oplus a\rangle) |f(x)\rangle$$

Далее по схеме следует применение преобразования Адамара к первому регистру:

$$\frac{1}{\sqrt{2}}(|x\rangle + |x \oplus a\rangle) \xrightarrow{H_n} \frac{1}{2^{\frac{n+1}{2}}} \sum_{y=0}^{2^n-1} (-1)^{x \bullet y} |y\rangle + \frac{1}{2^{\frac{n+1}{2}}} \sum_{y=0}^{2^n-1} (-1)^{(x \oplus a) \bullet y} |y\rangle =$$

$$= \frac{1}{2^{\frac{n+1}{2}}} \sum_{y=0}^{2^n-1} ((-1)^{x \bullet y} + (-1)^{x \bullet y \oplus a \bullet y}) |y\rangle =$$

$$= \frac{1}{\dots} \sum_{y: a \bullet y = 0} |y\rangle \tag{12}$$

В сумме (12) останутся только векторы $|y\rangle$, номера которых удовлетворяют условию:

$$a \bullet y = 0 \tag{13}$$

Для определения числа $a$ нам необходимо $n$ линейно-независимых уравнений вида (13), и для их получения придется запустить алгоритм Саймона $O(n)$ раз.

И так, в этой главе мы разобрали 4 квантовых алгоритма и познакомились с простейшим прототипом квантового компьютера.

Во всех квантовых алгоритмах присутствовали следующие этапы:

1. Подготовка суперпозиции всех возможных входных данных для интересующей нас функции.
2. Применение самой функции (квантового оракула).
3. Преобразование результата таким образом, чтобы вероятность интересующего нас исхода была близка к единице.

Для базового знакомства с квантовыми вычислениями пройденного на данный момент материала может быть достаточно. Тех же, кому интересны алгоритмы решения реальных практических задач, я приглашаю перейти к следующей клаве.

## 6 Дополнительные материалы

1. **Deutsch, David. "Quantum theory, the Church-Turing principle and the universal quantum computer".** Proceedings of the Royal Society A. 400 (1818) (July 1985): 97–117.
2. **John Preskill. Lecture notes for "Physics 219/Computer Science 219. Quantum Computation" (Formerly Physics 229)** http://www.theory.caltech.edu/people/preskill/ph229/index.html
3. **Симулятор квантового компьютера.** http://qc-sim.appspot.com