

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПУТЕЙ СООБЩЕНИЯ ИМПЕРАТОРА НИКОЛАЯ II (МИИТ)

ЮРИДИЧЕСКИЙ ИНСТИТУТ

МОРГУНОВ Р. Б.
КОПЛАК О. В.

КЛАССИЧЕСКИЕ
И КВАНТОВЫЕ АЛГОРИТМЫ
РЕШЕНИЯ МНОГОФАКТОРНЫХ
ПРОБЛЕМ ГУМАНИТАРНОЙ СФЕРЫ

*Методические указания
к выполнению практических работ
по информатике*

МОСКВА—2017

УДК 621.37

ББК 32.86

М—79

Моргунов, Р. Б. Классические и квантовые алгоритмы решения многофакторных проблем гуманитарной сферы : метод. указания к выполнению практических работ по информатике / Р. Б. Моргунов, О. В. Коплак. — М. : Юридический институт МИИТ, 2017. — 123 с.

Издание включает описание физических принципов и инженерных решений для хранения и обработки информации, примеры решения задач и сами задачи, а также методические указания по выполнению практических работ для студентов Юридического института МИИТа по дисциплинам цикла «Информатика». Также может быть полезно для студентов и аспирантов других специальностей.

© Юридический институт МИИТ, 2017

© Моргунов Р. Б., Коплак О. В. 2017

Изд. заказ 4	Формат 60×90 ¹ / ₁₆	Тираж 20 экз.
Подписано в печать 31.02.2017	Усл.-печ. л. 7,8	Уч.-изд. л. 3,4

127994, Москва, ул. Образцова, д. 9, стр. 9.

Содержание

Введение	4
Раздел 1. Количество информации.....	6
1.1. Меры информации.....	6
1.2. Беспорядок и информация	10
1.3. Лабораторная работа. Расчет энтропии и количества информации.....	13
Раздел 2. Современные устройства хранения информации	18
2.1. Элементарная ячейка памяти.....	18
2.2. Лабораторная работа. Симулятор ячейки памяти.....	21
2.3. Создание элементов и схем классической логики и памяти.....	26
2.4. Лабораторная работа. Симулятор логических элементов	36
Раздел 3. Проблемы миниатюризации и экономии энергии	46
3.1. Энергетический предел миниатюризации элементов памяти	46
3.2. Неклассическая логика, наноманиты и их взаимодействие	52
3.3. Лабораторная работа. Симулятор однобитной ячейки памяти.....	57
Раздел 4. Спинтроника	60
Раздел 5. Принципы квантовых вычислений	71
5.1. Физические квантовые вычислительные устройства	71
5.2. Квантовые логические операции.....	75
Раздел 6. Лабораторные работы по квантовым вычислениям.....	85
K1. Визуализация квантовых преобразований.....	85
K2. Квантовый алгоритм Гровера	90
K3. Квантовый «композитор».....	95
K4. Элементарные квантовые алгоритмы.....	104
K5. Квантовый оператор Адамара (Hadamard gate)	112
K6. Двухкубитные квантовые схемы	117
Рекомендуемая литература	123

Введение

Почти каждый студент сегодня использует флэш-память или аналогичное устройство для хранения данных. Часто нам кажется, что этим мы и современны. Однако настоящее ощущение включенности в современные технологии дарит лишь знание того, как работают эти устройства. Данные методические указания являются попыткой адаптации современных знаний об информационных технологиях для использования неспециалистами в области электроники и нанотехнологий. Здесь будут представлены лишь наиболее важные принципы и примеры технологий, как повсеместно используемых для обработки информации, так и тех, что могут стать актуальными уже в недалеком будущем. Освоение представленного материала потребует лишь знания небольшой доли школьной математики и совсем немного почти бытовых знаний о простых электрических и магнитных явлениях.

Интригующей темой, затронутой в пособии, является *квантовый компьютеринг*. Хотя это словосочетание порой вызывает опасение за уровень собственного образования, смешанное с недоверием к самой теме, нужно помнить, что для работы на классическом компьютере большинство людей совершенно не вникают в процессы, которые в нем происходят. Часто пользователи в равной степени не знают устройство как классического, так и квантового компьютера, что, однако, не мешает им справляться с поставленными задачами. Вместе с тем квантовые компьютеры быстро и уверенно входят в нашу жизнь: Газпромбанк прокладывает линию защищенной квантовой криптографической связи длиной 100 км, реальные квантовые вычисления может совершить любой желающей по системам удаленной связи с квантовым компьютером, располагающимся в фирмах D-wave, IBM, Google и др. Знакомство с простейшими квантовыми вычислениями на примере лабораторных работ является одной из целей нашей книги.

Другая цель — знакомство студентов гуманитарных специальностей с информационными технологиями, разработанными именно для решения проблем социальной, юридической, экономической направленностей. Известно, что проблемы гуманитарной сферы характеризуются многофакторностью, низкой степенью формализации, гигантской разветвленностью, а также вероятностным характером решений.

В методическом руководстве будет показано, что перечисленные характеристики почти слово в слово повторяют основные особенности квантового компьютера, который появился не потому, что технические возможности, наконец, позволили реализовать его, а главным образом потому, что задачи, стоящие перед обществом, все острее напоминают о необходимости прогресса именно в гуманитарной сфере. К таким задачам относится поиск неструктурированной информации в базах данных, распознавание образов, семантический анализ текстов, оценка качества информации и многое другое.

В настоящем издании в основном затронуты те аспекты, где в цепочке информатика-математика-физика трудно выбрать, что важнее. От физических процессов зависит возможность реализации устройств и способы их математического описания, которые, в свою очередь, формируют разделы информатики как технической дисциплины. Поэтому глубокое комплексное понимание информационных технологий возможно лишь при упомянутом комплексном подходе к их изучению.

Методические указания содержат простые контрольные вопросы к каждой главе и практические задания для лабораторных работ. Они предназначены для студентов нефизических специальностей вузов, которые желали бы знать о принципах работы простых повседневных устройств и уметь рассказать об этом своим детям. Предполагается использование методического руководства в цикле дисциплин информатики для экономических, юридических и гуманитарных специальностей вузов.

Раздел 1. Количество информации

1.1. Меры информации

Оценка различных устройств и информационных технологий основана на количественной мере информации, которая должна быть стандартизована и должна допускать сравнение между физически различными приборами, хранящими, записывающими, обрабатывающими или передающими информацию. Сегодня не нужно быть специалистом, чтобы ориентироваться в единицах информации. Все знают, что ее количество оценивается битами, байтами и соответствующими кило-, мега-, гигабайтами. Однако проблема в том, что эти единицы измерения хорошо подходят лишь для классических устройств и с трудом определяются для произвольной системы. Когда говорят, «нам сообщили много информации», подразумевается, скорее, ее качество и субъективно оцениваемая полезность, а не количество. Как правильно посчитать количество переданной или хранимой информации? Для этого были предложены различные подходы, позволяющие унифицировать понятие количества информации. Сделать это непросто потому, что подход к оценке объема информации зависит от конкретной задачи. Можно дать следующее определение количеству информации

КОЛИЧЕСТВО ИНФОРМАЦИИ — положительное число Q , характеризующее информацию по разнообразию, упорядоченности, или выбору состояний отображаемой системы. Эта мера обладает свойством аддитивности, т.е. при сложении информации Q находят как сумму значений этого параметра для двух систем.

Имеется три подхода к количественной оценке:

- структурный;
- статистический;
- семантический.

В рамках *структурного подхода* полезность информации не учитывается. Формальному пересчету подлежат элементарные события, с

помощью которых передается информация, их комбинации, а также их вероятности.

Статистический подход берет за основу полностью разупорядоченную систему, отсчитывая от нее любые упорядоченные конструкции из элементов системы. Он более глубок по сравнению со структурным подходом, так как соединяет фундаментальные представления об информации в разных науках.

Наконец, *семантический подход* оценивает именно ценность и востребованность информации. Он является наименее формальным и, как следствие, наиболее субъективным. Речь о нем зашла потому, что именно этот подход наиболее адекватен для решения проблем гуманитарной сферы, которые значительно сложнее любых технических и математических задач в плане их адаптации для компьютера.

Приведем в качестве примеров наиболее широко используемые методы оценки. Одним из первых появился **метод Хартли** (1928), который рассматривает систему, способную пребывать в n состояниях (либо, производят n опытов с равновероятными исходами — бросают кубик, и вероятность выпадения шестерки равна $1/6$). В такой системе каждое состояние можно закодировать двоичными кодами длины L . Какова должна быть эта длина кода, чтобы описать все возможные состояния системы?

Для этого число возможных комбинаций должно быть больше или равно n . А наименьшее возможное число комбинаций:

$$Q = \zeta \log_a n.$$

Это число характеризует вариативность системы, или возможность принимать различные состояния. Эту величину и принимают за меру информационной емкости системы. Коэффициент ζ можно определить по-разному потому, что логарифмы по разным основаниям пересчитываются друг в друга. Поэтому для НАТурального логарифма $Q = \zeta \ln n$ по определению принимают $\zeta = 1$, а единица информации НАТ (natural). Если логарифм по основанию 2, то речь идет о двоичной системе. Соответственно, $\zeta = 1/\ln 2$, $Q = \zeta \ln_2 n$, а единицу измерения называют **бит** (англ. *bit* — **binary digit**). Наконец, в десятичной системе $\zeta = 1/\ln 10$, $Q = \zeta \lg n$, а единицу измерения называют ДИТ (от французского dix — десять).

На практике прижились биты. Бит — количество информации, необходимое для различения двух равновероятных сообщений

(например «орел—решка», «намагничено—размагничено» и др. — см. рис. 1.1.1).

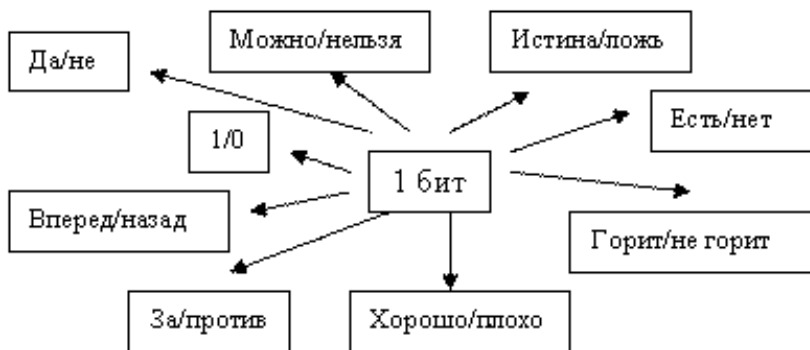


Рис. 1.1.1. Примеры информации, которая соответствует одному биту

Бит — это и наименьшая «порция» памяти компьютера, необходимая для хранения одной из двух логических значений двумерных машинных кодов «0» и «1».

Пример. Необходимо поднять груз на определенный этаж 8-этажного здания (нумерация этажей 0—7, $n = 8$). Сколько бит информации полностью определяют задание?

$$N = \log_2 n = \log_2 8 = 3.$$

Таким образом, 3 бита информации нужно сообщить грузчикам для выполнения задания. Как передают эти биты? Например, простыми «однобитными» вопросами и ответами. Для того чтобы установить,

1. Выше 7-го этажа? Нет = 0.
2. Выше 4-го этажа? Да = 1.
3. Выше 5-го этажа? Да = 1.
4. Выше 8-го этажа? Да = 1.

Итог: этаж номер 6.

Допустим, что была поставлена задача — доставить груз в определенную квартиру, коих на каждом этаже 4. Тогда будет нужно еще 2 бита информации $\log_2 4 = 2$. Такой же результат

получим, если вместо независимой нумерации этажей и квартир на этажах (два источника неопределенности) мы будем иметь только сквозную нумерацию квартир (один обобщенный источник):

$$H = \log_2 n + \log_2 m = \log_2 8 + \log_2 4 = \log_2 (n \times m) = \log_2 32 = 5,$$

Для n равновероятных состояний системы величина $Q = \ln n$ характеризует глубину нашего незнания о системе без дополнительной информации.

Метод Хартли применим для равновероятных событий. Его обобщение на случай событий разной вероятности было предложено К. Шенноном. **Формула Шеннона** позволяет рассчитать количество информации, зная число состояний системы n и учитывая вероятность попадания в это состояние p_i :

$$Q = \sum p_i \log_2 p_i.$$

Здесь суммирование производится по различным состояниям системы, которым приписываются соответствующие вероятности p_i . Для равновероятных состояний $p_i = 1/n$, и мера информации $Q = \log_2 n$.

Пример. Рассмотрим систему, состоящую из нескольких ячеек (камеру хранения). Событием будем считать помещение предмета в одну из ячеек. Если, например, известно, где находится предмет, например в k -й ячейке, то все $p_i = 0$, кроме $p_k = 1$, то тогда $Q = \lg 1 = 0$, и здесь мы новой информации не получаем (как и следовало ожидать).

Пример. Выясним, сколько информации содержится в двузначном числе. Будем считать, что оба разряда числа могут принимать произвольные значения: старший разряд пробегает значения от 1 до 9 (9 значений), а младший — от 0 до 9 (10 значений). Всего двузначных чисел 90 (10—99). Поэтому количество информации $Q = \log_2 90 \approx 6,5$. Этот же результат можно получить аддитивным сложением информации для каждого из разрядов $Q = \log_2 90 = \log_2 9 + \log_2 10$. Приблизительное значение $\log_2 10$ равно 3,32. Получается, что с помощью десятичной единицы можно передать в 3,32 больше информации, чем при пе-

редаче двоичной единицы. А младший разряд в двузначном числе несет больше информации, чем старший разряд.

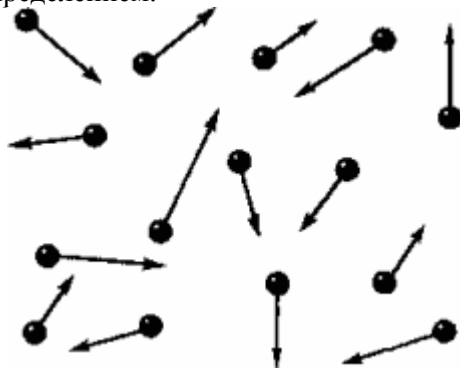
1.2. Беспорядок и информация

Удивительным фактом кажется то, что представления о мерах информации, предложенных в информатике и математике, оказываются очень близки к понятиям, развитым в естественных науках. Рассматривая статистические закономерности случайных событий (например, движения молекул в газе), физики пришли к выводу, что даже в системах, содержащих гигантское количество частиц и, соответственно, их состояний, можно эти состояния пересчитать N , а затем выяснить, какие из них реализуются большим числом способов n . Отношение $P = n/N$ будет характеризовать вероятность найти систему в данном состоянии с числом реализаций n . Интуитивно ясно, что замкнутая система (на которую не влияют внешние воздействия) должна самопроизвольно приходить именно в наиболее вероятное состояние с самым большим значением P и самым большим количеством ее реализаций n . Этот очень простой принцип (здесь доведенный до наивности) называется вторым законом термодинамики. (Первый закон термодинамики — это просто закон сохранения энергии). К информатике и технологиям хранения и обработки информации второй закон термодинамики имеет прямое отношение.

Состояние, которое система займет с наибольшей вероятностью, как уже было отмечено, должно реализовываться максимальным количеством неотличимых способов. Например, если в газе поменять местами молекулы, от этого его состояние (давление, температура) не изменится. Чем больше таких замен допускает данное состояние, тем оно более вероятно. Одновременно с этим про самое вероятное (равновесное) состояние можно также сказать и то, что в нем присутствует максимальный беспорядок. В самом деле, раз именно для этого состояния наблюдается наибольшая возможность менять местами все объекты и их параметры, значит именно оно самое разупорядоченное.

Например, молекулы газа в комнате могут быть равномерно распределены по ней (равновесное состояние) (рис. 1.2.1). Кому не нравятся молекулы (у гуманитариев это часто случается), можно думать, например, о хаотичном распределении букв (рис. 1.2.1).

Если представить себе, что комната разбита на клетки, то при равномерном распределении можно менять клетки местами и молекулы переставлять из клетки в клетку — ничего при этом не изменится. Теперь представим себе значительно менее вероятное состояние, когда все молекулы в комнате собрались в небольшой ее части. Теперь мы не можем без изменения состояния менять местами пустые и заполненные клетки потому, что тогда изменится объем, давление и др. При этом это состояние можно назвать более упорядоченным по сравнению с равномерным распределением.



СХАІСХЕІІХНАІСНХІХІКСНАІСЕХІ
 ІНХІІСНАІСАІСНАЕКЕАХІКЕСІСНА
 НХІСХІЕКІХІХЕІСНЕІНАІЕНКХКХ
 ХАКХНСКАІСІЕКІХНАІСНХЕКХІСНА
 ІСНАІКАЕХКІСНАІКХЕХЕІСНАХКЕК
 СНАІСІНКХІАІСНАХЕКЕХСНАКСІЕЕ

Рис. 1.2.1. Хаотичное распределение объектов

Поэтому физики говорят, что беспорядок в замкнутых системах стремится возрастать, и в равновесии максимален. Это другая формулировка второго закона термодинамики. Что же этот закон представляет собой с точки зрения информатики? Очевидно, что наиболее разупорядоченное состояние несет наименьшую информацию о си-



Надгробие Л. Больцмана

стеме, а с ростом порядка количество информации растёт (см. параграф 1.1). Получается, что для того чтобы получить (передать, обработать) информацию, нужно затратить энергию на упорядочение системы. Другими словами — **обработка информации всегда связана с затратами энергии**, нужной для «наведения порядка».

Все эти рассуждения применительно к физическим системам (которыми, безусловно, являются и ячейки памяти компьютера) приводят к поистине великой формуле Людвига Больцмана:

$$S = k \ln W.$$

В этой простейшей формуле $W = n/N$ — уже упомянутая вероятность данного состояния системы, k — постоянный коэффициент, называемый константой Больцмана, S — количественная мера беспорядка в системе, называемая энтропией. На кладбище Вены покоится прах австрийского физика Больцмана (1844—1906). На его надгробии начертана упомянутая формула, которая носит его имя.

Формула Больцмана связывает информатику (подсчет состояний) и физику (энергию, отвечающую данной энтропии). В этой формуле, удивительно похожей на формулу Шеннона, полученную из других соображений, пересекается микромир (наблюдение за отдельными частицами) и макромир (результатирующие свойства системы). С помощью формулы Больцмана решена проблема формирования общего из огромного количества частных. Поэтому, как и художественное полотно или музыкальное произведение, формула Больцмана носит философский гуманитарный характер, хотя одновременно и является

простейшим почти арифметическим примером из математики. Ее изящество и математическая простота сочетаются с глубокими разнообразными смыслами и применениями. Иногда ее воспринимают слишком примитивно. Тогда это приводит к парадоксам. Суть одного известнейшего парадокса заключается в том, что рассматривалось два сосуда с разными температурами, соединённых узкой трубкой с затворками, которыми управлял маленький человечек — «демон». «Демон» измерял скорость отдельных летящих молекул и выбирал самые быстрые, пропуская их в сосуд с высокой температурой, а более медленные оставались в сосуде с низкой температурой. Постепенно теплый сосуд должен был разогреться еще больше, а холодный еще больше охладиться без затрат энергии. Из этого мысленного эксперимента следует нарушение второго закона термодинамики. Парадокс может быть разрешён при помощи теории информации. Для измерения скорости молекулы «демон» должен был бы получить **информацию** о её скорости. Но всякое получение информации — материальный процесс, сопровождающийся возрастанием энтропии. Количественный анализ показал, что приращение энтропии при измерении (хаотичное колебание дверки и самого «демона» молекулами) превосходит по абсолютной величине уменьшение энтропии, вызванное перераспределением молекул «демоном». В дальнейшем формула Больцмана потребует нам для определения энергетического порога работоспособности логических устройств.

1.3. Лабораторная работа. Расчет энтропии и количества информации

Цель работы: освоение методов расчета количества информации.

Порядок выполнения работы

1. Изучить теоретическую часть.
2. Выполнить задания.
3. Представить полученные данные для проверки преподавателю.

Для работы в двоичной системе исчисления, а также перевода чисел из одной системы в другую и выполнения арифметических

действий существуют онлайн калькуляторы. Например, предоставляемый на страничке <https://numsys.ru/> (рис. 1.3.1).

The screenshot shows the website numsys.ru with a browser window. The main heading is "Системы счисления" (Number Systems). Below it is a form for converting numbers between bases. The first row has a "Перевести" (Convert) button, followed by input fields for "2014", "из 10" (from 10), "-ной в 2" (-base to 2), "-ную" (-base), and another "Перевести" button. The second row has input fields for "1110", a dropdown menu set to "прибавить к" (add to), input fields for "1001", a dropdown menu set to "в 2" (in 2), "-ной системе счисления" (-base number system), and a "Вычислить" (Calculate) button. Below the form is a paragraph of text: "Мы работаем с неотрицательными действительными числами не длинее 50-ти символов, в системах счисления с двоичной по тридцатишестичисленной, без обода и выходных." (We work with non-negative real numbers not longer than 50 symbols, in number systems from binary to base 36, without leading zeros and trailing digits). Below this text are two social media-like buttons: "В Мне нравится" (I like it) with a count of 7296, and "G+1" with a count of 556. At the bottom, there is a footer with "© 2011–2014 | Обратная связь | Мобильная версия" (© 2011–2014 | Feedback | Mobile version).

Рис. 1.3.1. Онлайн калькулятор для переходов между системами исчисления

Логарифмы тоже можно рассчитать онлайн, но проще это сделать калькулятором, который имеется как приложение в любом современном компьютере. Для наших целей потребуется его инженерный вид, который можно выбрать в Меню калькулятора (рис. 1.3.2).

Задания

1. Перевести числа из десятичной системы в двоичную с помощью онлайн калькулятора.
2. Выполнить действия в двоичной системе $324 \times (345 - 213)$.
3. Вернуться в десятичную систему и убедиться, что ответ совпадает с обычными вычислениями.
4. Рассчитать количество энергии, которое соответствует упорядочению букв из хаоса РТЭЯПНО в слово ЭНТРОПИЯ.

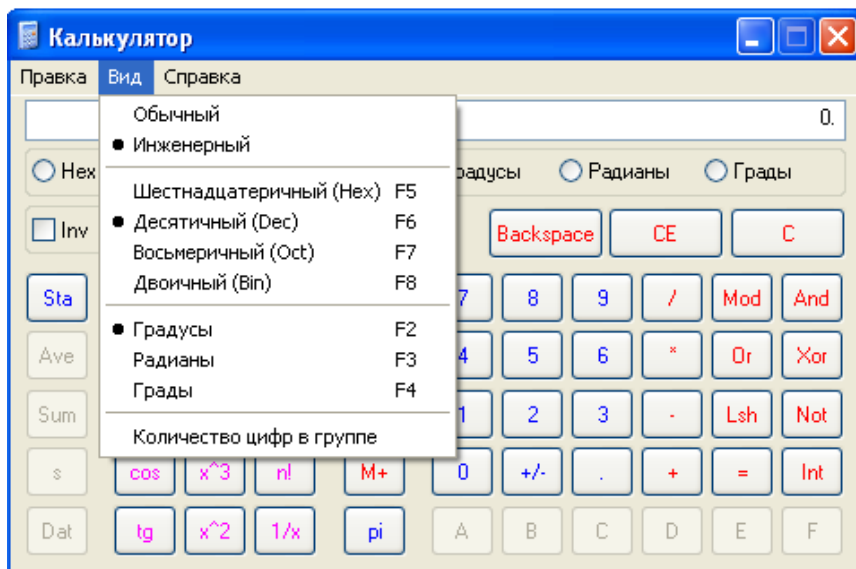


Рис. 1.3.2. Инженерный калькулятор с возможностью выбора системы исчисления

5. Откройте онлайн калькулятор Шеннона на странице <http://www.shannonentropy.netmark.pl/>. Перед вами появится панель для введения туда строки из нулей и единиц или других символов, частоту повторения которых будет рассчитывать калькулятор (рис. 1.3.3).

6. Введите туда объект. Например, 1100101. Рассчитайте энтропию, связанную с разупорядоченностью символов в вашей строке.

7. Предложите другие объекты (например, 1111111 и др.), рассчитайте их энтропии, сравните между собой, сделайте выводы.

8. Проверьте значения, которые выдает онлайн калькулятор, совпадают ли они с предсказаниями формулы Шеннона по образцу, который предоставляет калькулятор на результирующей странице (рис. 1.3.4).

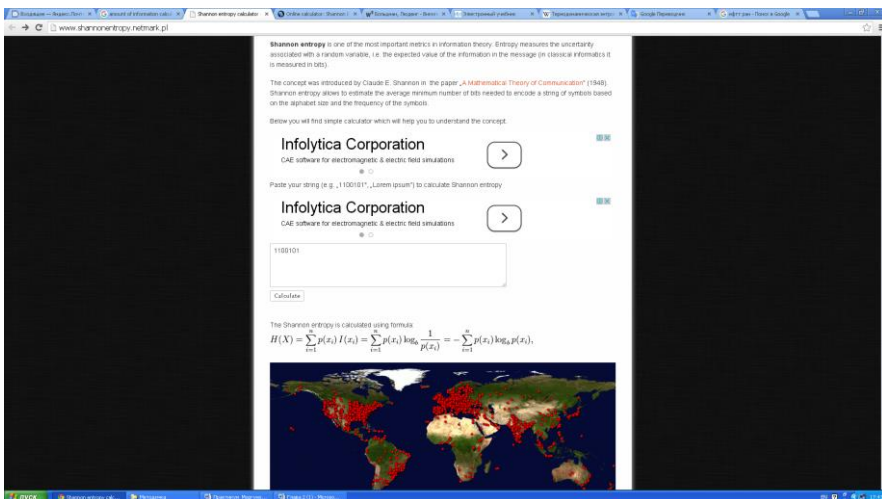


Рис. 1.3.3. Онлайн калькулятор Шеннона

Real example how to calculate and interpret information entropy

Your string is: **1100111**

Alphabet of symbols in the string: **0 1**

Frequencies of alphabet symbols:

- **0.286 -> 0**
- **0.714 -> 1**

Shannon entropy can be calculated as follow:

Infolytica Corporation

CAE software for electromagnetic & electric field simulations



$$H(X) = -[(0.286 \log_2 0.286) + (0.714 \log_2 0.714)]$$

$$H(X) = -[(-0.516) + (-0.347)]$$

$$H(X) = -[-0.86312]$$

$$H(X) = 0.86312$$

Рис. 1.3.4. Результаты расчета калькулятором Шеннона

Контрольные вопросы

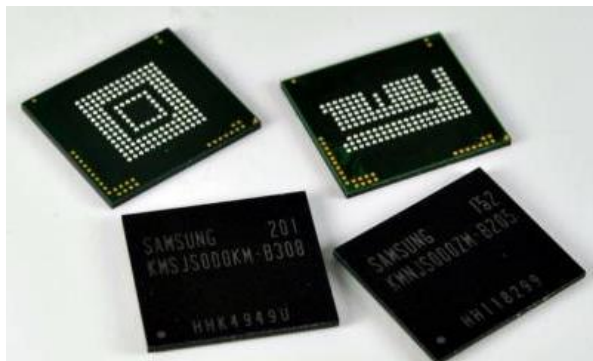
1. Чему равно количество информации, если стало известно, что одного из десяти студентов отчислили за неуспеваемость?
2. Если бы алфавит состоял из трех букв a , b , c с вероятностями их использования 0,25; 0,55; 0,2, то какое бы количество информации приходилось бы на символ сообщения, составленного с помощью такого алфавита.
3. Как называется количественная мера информации?
4. Каким образом связана информация и беспорядок?
5. Как называется количественная мера беспорядка?
6. Как выглядит формула Шеннона.
7. Какие величины связывает формула Больцмана?
8. Что такое «аддитивность информации»?
9. В чем разница между единицами измерения информации НАТ, БИТ, ДИТ?
10. Какую минимальную энергию нужно иметь, чтобы принять (обработать, передать) один бит информации?
11. В чем заключается парадокс «Демона Максвелла»?
12. Какие меры информации вы знаете?

Раздел 2. Современные устройства хранения информации

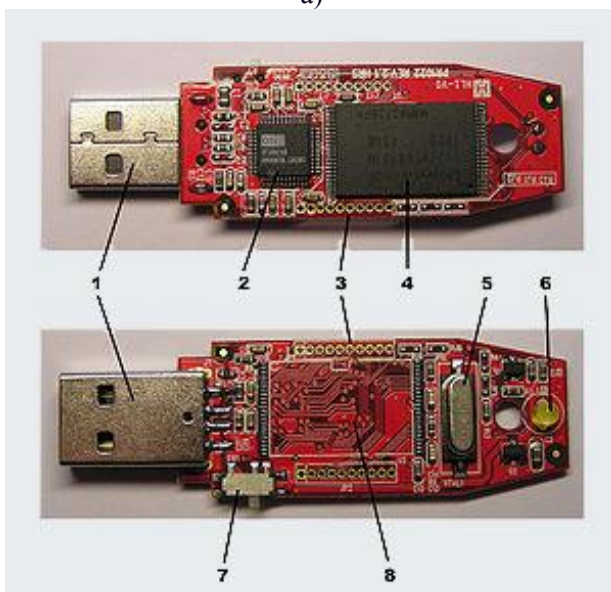
2.1. Элементарная ячейка памяти

Существует необозримое количество моделей компьютерной памяти, основанной на различных принципах. В большинстве окружающих нас компьютеров используются гальванические транзисторные технологии. Однако инженерия классических компьютеров уже представляет собой мостик к устройствам новых поколений. Так, память случайного доступа RAM развивается в магнитную память случайного доступа MRAM, спиновую память SRAM и другие устройства, которые вышли на рынок компьютерной техники. Это строго наномасштабные технологии, т.е. такие, в которых используются устройства размером 30—60 нм. Уменьшение масштабов устройств меняет и принципы их работы. На смену классическим вычислениям приходят квантовые вероятностные расчеты. Примеры физических реализаций таких машин содержатся в руководстве. Проблема уменьшения энергозатрат в расчете на одну логическую операцию заставляет стремиться к так называемому пределу Ландауэра, который ограничивает минимально возможные затраты, устанавливая эквивалентность энергии и информации. Поэтому помимо квантовых устройств появляются экзотические классические устройства с энергозатратами, близкими к 17 мэВ на одну операцию (это и есть предел Ландауэра). Они совсем не похожи на современные элементы компьютера и также будут коротко рассмотрены в пособии. Какие из них окажутся в скором времени востребованными, неизвестно, однако главные направления разработки этих устройств будут представлены в этом руководстве.

Рассмотрим сначала вопрос о физических принципах работы *Flash*-памяти. Она содержит миллиарды элементов, собранные в едином корпусе и, более того, располагающиеся на поверхности единого монокристалла кремния или германия (рис 2.1.1). Этот вид памяти был разработан японцем Fujio Masuoka в 1984 г. Компания *Intel* выпустила на рынок первый флеш-чип в 1988 г.



а)



б)

Рис. 2.1.1. Микросхемы флэш-памяти (а) и разобранный USB-Flash накопитель (б) и его основные элементы: 1— USB-контакт, 2 — контроллер, 3 — многослойная печатная плата, 4 — блок NAND памяти, 5 — кварцевый генератор опорной частоты, 6 — светодиод, 7— переключатель защиты от записи, 8 — место для дополнительной памяти

Прежде чем рассматривать архитектуру этой микросхемы и принципы ее работы, разберем сначала, как работает ее элементарный «кирпичик», хранящий информацию. Единичный элемент этой микросхемы — полевой транзистор с плавающим затвором, изобретенный в 1970-х гг. (от англ. *field effect transistor (FET)*). Имеется великое множество различных типов таких транзисторов, однако мы не будем углубляться в подробности, ставя целью лишь знакомство с самым простым прибором.

Этот прибор устроен несложно (рис 2.1.2). У него есть четыре электрода: управляющий затвор, плавающий затвор, сток и исток. Все это выполнено на кремниевом монокристалле, однако различные области транзистора легированы разными концентрациями фосфора, что и обеспечивает возможность порогового переключения транзистора. В конечном итоге, именно эта особенность (пороговое переключение) и нужна для хранения логических нулей и единиц. Между затвором и каналом, прямо в диэлектрике, расположен еще один проводник, который и называют плавающим затвором.

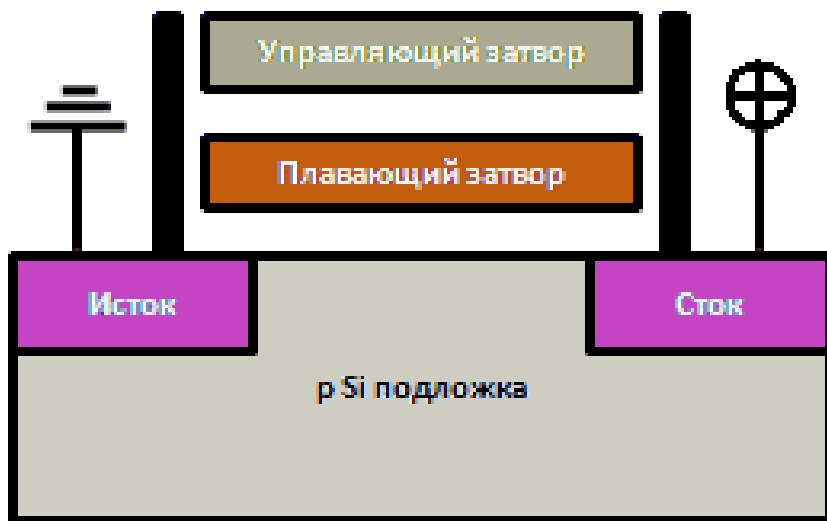


Рис. 2.1.2. Схема полевого транзистора с плавающим затвором

Если приложить между стоком и истоком разность потенциалов с положительным потенциалом на затворе, то через полевой транзистор от стока к истоку потечет ток. Величина этого тока оказывается достаточной, чтобы «пробить» диэлектрик и доставить часть электронов на затвор. Тогда плавающий затвор, оказавшись отрицательно заряженным, создаст электрическое поле, которое начинает препятствовать протеканию тока в канале, в результате чего транзистор закрывается. На этом этапе *логическая единица сохранена* в приборе — если теперь отключить питание, электроны с плавающего затвора не смогут уйти из-за диэлектрического барьера, а запасенный там заряд останется неизменным на долгие годы. Плавающий затвор можно и разрядить, приложив к основному затвору напряжение противоположного знака. Это вызовет уход электронов и приведет систему в состояние логического нуля. Можно сказать, что эта процедура стирает логическую единицу. Так и происходит запись и хранение информации — если на затворе накоплен отрицательный заряд, это состояние принимают за логическую единицу, а если заряда нет, считается, что это логический ноль.

Существует большое количество симуляторов работы транзистора *FET*. С помощью одного из них мы осуществим все вышеописанные процедуры, чтобы почувствовать, как он работает.

2.2. Лабораторная работа. Симулятор ячейки памяти

1. Сначала разберемся в работе обычного полевого транзистора без плавающего затвора. Такие транзисторы могут применяться в качестве усилителей сигнала, либо в качестве переключателя. Для этого перейдем по ссылке: URL: <http://www-g.eng.cam.ac.uk/mmg/teaching/linearcircuits/loader.swf?device=mosfet.swf>, и перед вами появится панель (рис. 2.1.3).

Слева цветом отмечены элементы транзистора и материал, из которого они изготовлены. Металлические электроды (черным цветом), диэлектрический слой оксида кремния (темно-серым цветом), объем кристалла кремния, легированного бором, физики говорят про него, как про *p-doped* или р-легированный слой, в котором избыточны положительные носители заряда — дырки (светло-серым цветом), белым цветом в процессе работы будут показаны изменения слоя, обед-

ненного зарядом (*depletion region*) и, наконец, голубым цветом изображены части кристалла, легированного фосфором. В этих слоях главными носителями заряда являются отрицательно заряженные электроны.

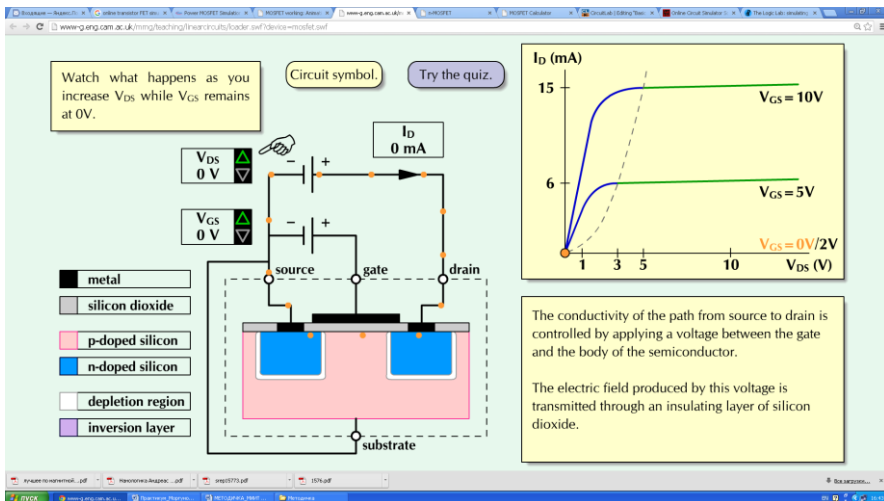


Рис. 2.1.3. Симулятор полевого транзистора с плавающим затвором без подключенных напряжений

2. Попытаемся сначала увеличивать напряжение V_{GS} на обычном затворе. При этом будем наблюдать рост запирающего слоя, в котором будут накапливаться электроны (*inversion layer*). Канал проводимости от истока (*source*) к стоку (*drain*) контролируется напряжением, приложенным между затвором и объемом полупроводника V_{GS} . Увеличение этого напряжения вызывает рост числа подвижных электронов в канале, увеличивая глубину инверсионного слоя (*inversion layer*). При самом высоком напряжении 10 В ток все еще не начинает течь, однако ситуация теперь выглядит так, как показано на рис. 2.1.4. Носители заряда в канале есть, но нет причины, которая вызвала бы возникновение тока. Для того чтобы она появилась, нужно приложить напряжение между истоком и стоком (рис. 2.1.4).

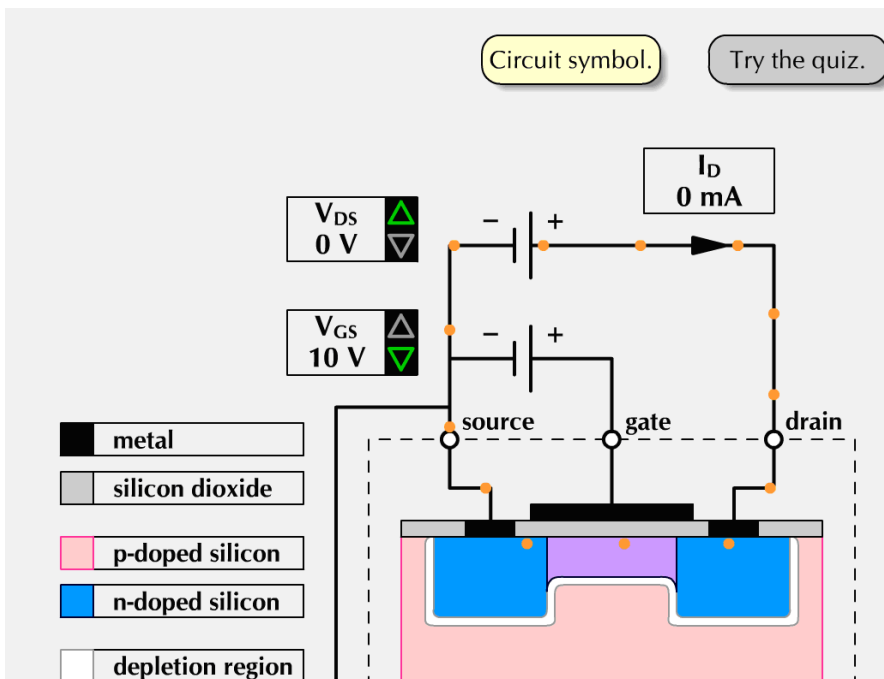


Рис. 2.1.4. Симулятор полевого транзистора с максимальным напряжением 10 В на затворе

3. Устанавливаем промежуточное значение $V_{GS} = 5$ В, а затем увеличиваем напряжение между истоком и стоком V_{DS} , которое вызывает движение зарядов в канале (рис. 2.1.5).

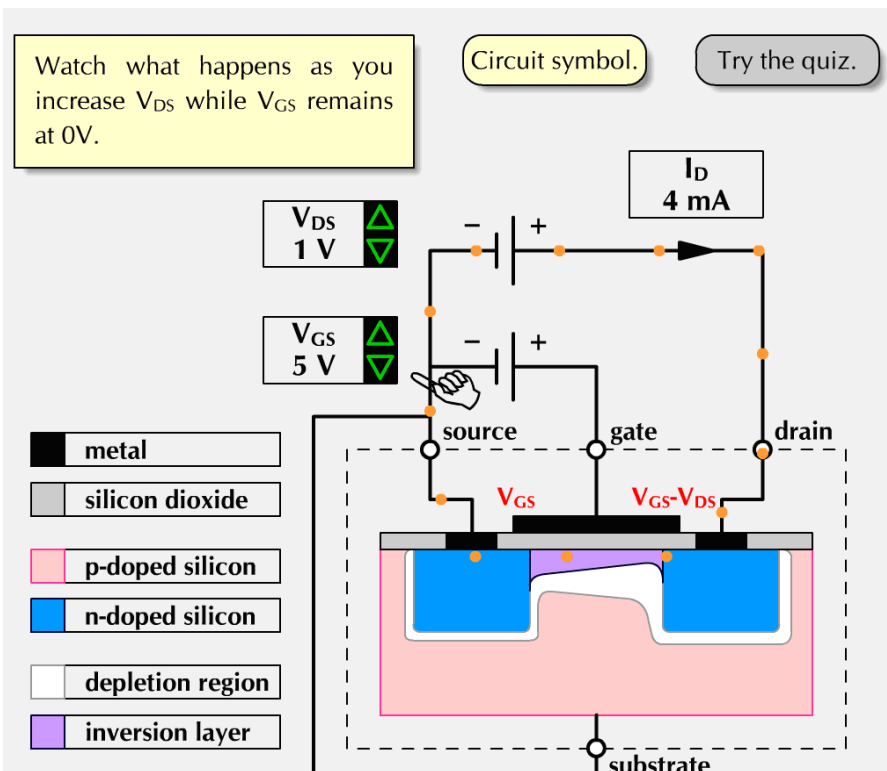


Рис. 2.1.5. Симулятор полевого транзистора с напряжением $V_{GS} = 5$ В на затворе и приложенным напряжением между истоком и стоком $V_{DS} = 1$ В

4. По данным переключения V_{GS} и V_{DS} построить таблицу токов I_D через сток.

5. Открыв страничку <http://www.falstad.com/circuit/e-nmosfet.html>, менять значения V_{GS} и V_{DS} , наблюдая за изменением силы тока в цепи. Построить зависимости силы тока в цепи исток-сток от этих напряжений, руководствуясь данными, выводимыми на симуляторе (рис. 2.1.6).

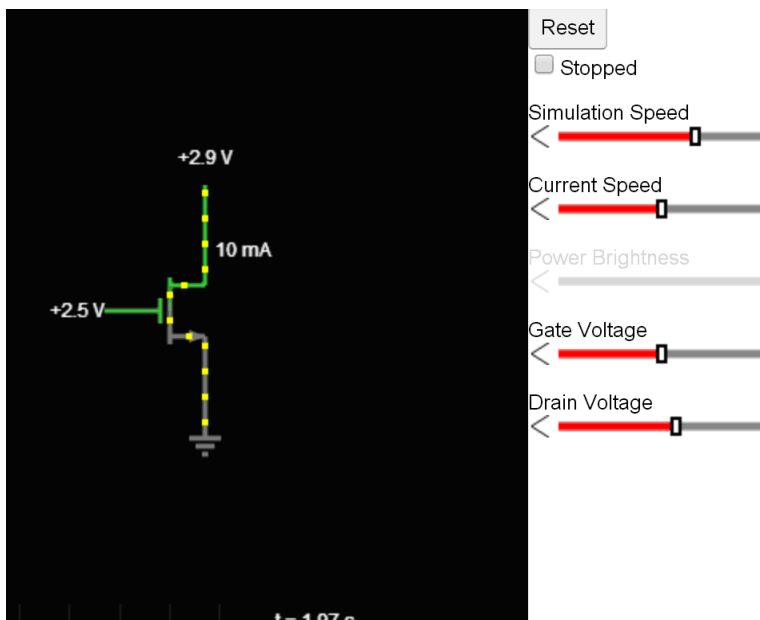


Рис. 2.1.6. Симулятор полевого транзистора

В качестве отчета по лабораторной работе представить преподавателю таблицу токов I_D в зависимости от напряжений V_{GS} и V_{DS} . Показать зависимости $I_D(V_{GS})$ и $I_D(V_{DS})$ в MS Excel. Написать выводы о работе полевого транзистора.

Контрольные вопросы

1. Из каких материалов делают транзисторы?
2. Какие электроды есть у полевого транзистора?
3. Каково назначение разных электродов?
4. Какие функции выполняет напряжение, приложенное к затвору?
5. Какие функции выполняет напряжение между стоком и истоком?
6. Для чего применяется транзистор без плавающего затвора?
7. Как изменяется зона обеднения по мере увеличения напряжения на затворе?

8. Что происходит с электронами проводимости при включении напряжения между истоком и стоком?
9. Чем отличается полевой транзистор с плавающим затвором от обычного полевого транзистора?
10. Каким образом происходит запоминание логического нуля или логической единицы в транзисторе с плавающим затвором?
11. Что такое плавающий затвор и зачем он нужен?
12. Сколько времени может храниться информация на флэш-устройстве? Чем ограничивается время хранения?
13. Из каких компонент состоит «флэшка»?

2.3. Создание элементов и схем классической логики и памяти

Слово *FLASH* часто применительно к типу памяти переводится как «вспышка». Однако на самом деле это слово было использовано впервые в 1989—1990 г. корпорацией *Toshiba* для описания работы нового типа памяти, и его контекст означал «быстрый, стремительный». Вообще, считается что компания *Intel* впервые создала в 1988 г. флэш-память с двумерной архитектурой (ее называют NOR). А в 1989 г. корпорация *Toshiba* предложила трехмерную память *NAND*. Обе памяти сегодня находят применение в микросхемах флэш-памяти. Теперь нужно понять, как осуществляются логические операции и записываются слова и цифры с помощью транзисторов с плавающим затвором. Заряженный плавающий затвор своим электрическим полем препятствует протеканию тока в цепи исток—сток. Например, если при отсутствии заряда приложить на «основной» затвор напряжение +1В, в цепи стока начинает протекать ток. Если же плавающий затвор заряжен, приложение этого же напряжения не вызовет электрический ток, потому что электрическое поле плавающего затвора будет ему мешать. Можно себе представлять, что внутреннее напряжение в транзисторе будет включено встречно внешнему напряжению. В результате электрический ток потечет только при напряжении, например, +5 В. Таким образом, получается пороговое значение напряжения. Поэтому, подав, к примеру +4,5В мы сможем по наличию или отсутствию тока стока сделать вывод о наличии или отсутствии заряда на плавающем затворе. А это эквивалентно считыванию одного бита информации!

Каким образом такое устройство (полевой транзистор) позволяет создать *Flash*-память? Микросхема памяти состоит из миллионов и даже миллиардов транзисторов, расположенных строго определенным образом. В зависимости от их взаимного расположения различают два вида флэш-памяти: *NOR*- и *NAND*-память. Их принципиальное отличие состоит в том, каким образом соединены эти транзисторы.

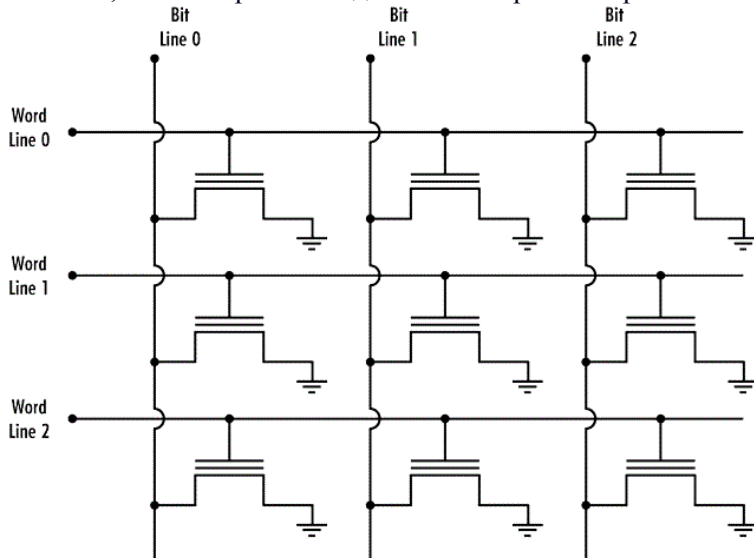


Рис. 2.2.1. Схема расположения транзисторов в двумерной флэш-памяти *NOR*

Память *NOR* основана на двумерном расположении транзисторов и проводников между ними. То есть имеется множество строк и столбцов, как в матрице. На пересечениях этих строк и столбцов установлено по одной ячейке памяти (по одному транзистору). Проводящие линии, расположенные в строчку, называются линиями бит, а те, что соединяют столбцы, называются линиями слов. Стоки всех транзисторов подключаются к горизонтальным линиям бит, а все их затворы к вертикальным линиям слов.

Для считывания информации из какой-то конкретной ячейки соответствующий транзистор подключен затвором на одну из линий слов, а его сток — на одну из линий бит. Подавая пороговое напря-

жение на линию слов, соответствующее затвору транзистора, мы вызываем или не вызываем в нем ток, что соответственно означает считывание единицы или нуля.

В отличие от двумерной памяти *NOR*, ячейки *NAND*-памяти представляют собой трехмерный массив. К каждой линии бит подключено сразу несколько вертикально расположенных транзисторов. Это уменьшает количество линий и увеличивает масштабируемость устройства, что и является главным преимуществом памяти *NAND*. Как считать состояние определенного транзистора в этой структуре? Для понимания процесса рассмотрим схему на рис. 2.2.2.

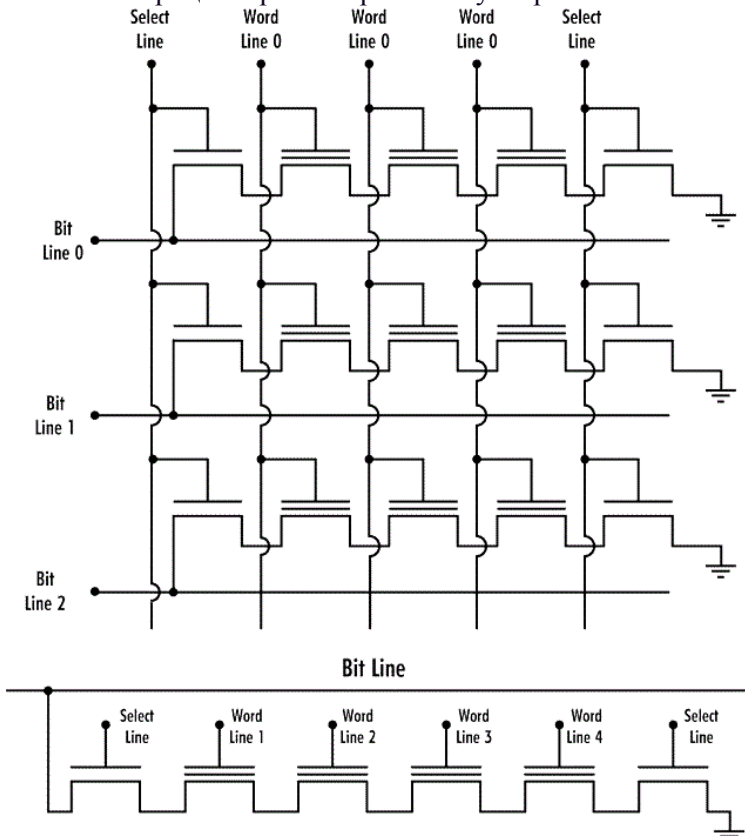


Рис. 2.2.2. Схема расположения транзисторов в трехмерной флэш-памяти *NAND*

В этих схемах нет ничего сложного — просто повторяющиеся элементы, отдельно исследованные нами в первой лабораторной работе. Нам не потребуется ни распутывать эти провода, ни создавать их самим. Приведенные картинки служат лишь для иллюстрации самой технологии создания миниатюрной памяти.

В этой схеме каждая линия бит соответствует нескольким ячейкам. Если на линии имеется хотя бы один закрытый транзистор, то на линии бит есть высокое напряжение (рис. 2.2.3).

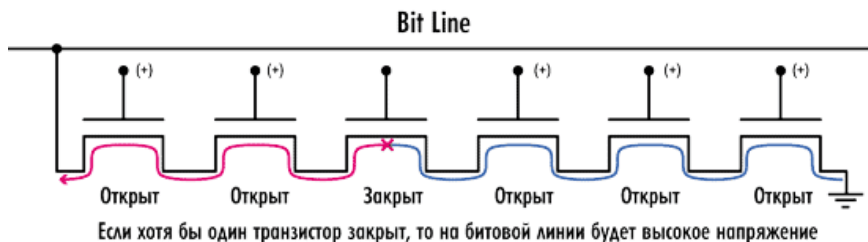


Рис. 2.2.3. Отдельная строка транзисторов из трехмерной флэш-памяти NAND

Низкое напряжение на линии бит будет только в том случае, если нет закрытых транзисторов. Если хотя бы один транзистор закрыт, это приведет к отсутствию тока через него и через всю остальную цепочку, независимо от состояния других транзисторов в ней. Чтобы считать состояние конкретного транзистора в этой паутине устройств, недостаточно подать на линию слов (на затвор транзистора) пороговое напряжение и следить за сигналом на линии бит. Нужно перед этим еще открыть все остальные транзисторы. Для этого на затвор считываемого транзистора подается пороговое напряжение, а на затворы всех остальных транзисторов — подается повышенное напряжение, которое открывает их все независимо от состояния плавающего затвора. В результате, определив, нулевой или ненулевой ток будет в линии бит, мы определим, в каком состоянии был именно тот транзистор, с которого нужно было считать информацию, поскольку остальные транзисторы в этот момент будут точно открыты и не дадут вклада в сигнал.

Рассмотрим теперь классические элементы логики, создаваемые из полевых транзисторов для проведения вычислений. В качестве первого элемента рассмотрим *NOR*, именем которого называется один из видов флэш-памяти. Нужно отметить, что все действия, которые совершаются этими элементами, были описаны математиком Булем задолго до появления компьютеров, а соответствующая, развитая им алгебра была названа Булевой алгеброй.

На рис. 2.2.4 показана схема с двумя входами *A* и *B* (мы теперь всегда будем использовать для них английское обозначение *In*), и одним выходом *C* (с соответствующим обозначением *Out*).

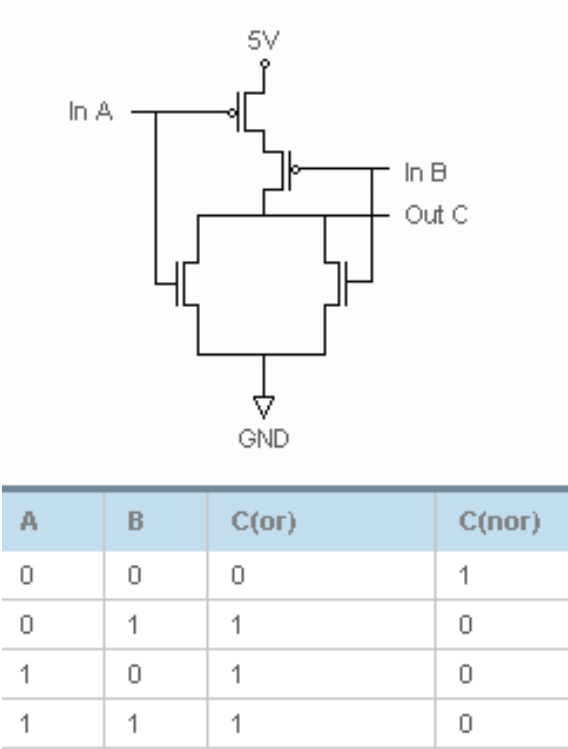


Рис. 2.2.4. Схема логического элемента *NOR* (*Not OR*), который в булевой алгебре обозначает отрицание «ИЛИ». В таблице для сравнения показано, какие выходные сигналы будут в случае операций *OR* и *NOR*

Эта схема дает возможность реализации таблицы, которая описывает, что будет на выходе в зависимости от сигналов на входе. Такие таблички используются везде для описания работы логических устройств, и, как мы увидим в дальнейшем, вполне пригодны для случая квантовых компьютеров.

С помощью этой схемы осуществляется преобразование входных напряжений в выходные, соответствующие «0» и «1». Они необходимы, потому что для чтения и записи информации используются различные напряжения.

В рассматриваемой структуре флэш-памяти на хранение одного бит информации используется один транзистор. На самом деле в энергозависимых типах памяти для этого требуется несколько транзисторов и конденсатор. Однако при этом значительно на несколько порядков величины уменьшаются размеры микросхем и упрощается технология их изготовления вместе с ценой. Один бит в ячейки памяти размещается не всегда: *Intel* предложил память *StrataFlash*, в которой каждая ячейка хранит 2 бита информации и более (до 9). Это так называемая технология многоуровневых ячеек, в которых заряд делится между несколькими уровнями, каждому из которых соответствует определенная комбинация бит. В принципе количество уровней не ограничено, однако в действительности при нарастании их числа появляются значительные шумы и происходит утечка электронов при продолжительном хранении. Современные микросхемы памяти хранят информацию годами, а число циклов стирания-записи в них варьируется в диапазоне 10^5 — 10^6 . Флэш-память с архитектурой *NOR* плоха тем, что в ней нельзя уменьшать размер микросхемы путем уменьшения размеров транзисторов, поскольку к каждому транзистору надо обеспечить отдельный контакт. В этом свойстве (масштабируемости) значительно выигрывает архитектура *NAND*.

Схема устройства *NAND* показана на рис. 2.2.5 *NAND* — *Not AND* — в булевой математике используется для отрицания «И». Сама операция «И» означает, что для появления выходного сигнала на входе А И на входе В должны быть ненулевые сигналы (логические единицы). Элемент ИЛИ допускает наличие сигнала ИЛИ на входе А ИЛИ на входе В, т.е. одного сигнала достаточно, чтобы на выходе тоже появилась логическая единица. Отличается память *NAND* от *NOR* только логической схемой. В случае с памятью *NAND* площадь устройства

можно значительно уменьшить за счет размеров ячеек. Главный недостаток заключается в низкой (по сравнению с *NOR*) скорости работы. Лучшие черты представленных устройств (масштабируемость и скорость) перенимаются комбинированными системами *DiNOR*. Системы *NOR* и *NAND* не лучше и не хуже друг друга, просто они находят применение в разных областях и каждая из них по-своему эффективна.

Область применения разных типов флэш-памяти зависит от требуемых для конкретной задачи скорости записи/считывания и надежности хранения информации. В памяти *NAND* ячейки собраны в блоки (кластеры). Поэтому каждое прочтение извлекает сразу несколько бит, а следовательно, и скорость такого устройства должна быть выше. Однако это только в том случае, если обращение к ячейке адресное, что само по себе замедляет процесс, поскольку требуется время на ее поиск. Такие системы значительно проигрывают по времени системам со случайным доступом *RAM (Random Access Memory)*. Об этих системах подробнее речь пойдет далее, однако их общее определение, согласно Википедии, таково: произвольным доступом (также называемым случайным доступом или прямым доступом, англ. *random access*) понимают возможность обратиться к любому элементу последовательности за равные промежутки времени, не зависящие от размеров последовательности (в отличие от последовательного доступа, когда чем дальше расположен элемент, тем больше требуется времени для доступа). Говорят, что структура данных поддерживает произвольный доступ, если возможен доступ к любому элементу за одинаковое время независимо от позиции элемента. Таким образом, *NAND* значительно проигрывает в операциях с произвольным доступом и не позволяет напрямую работать с байтами информации.

Технология флэш-памяти очень перспективна. При этом, разумеется, устройства хранения данных этого типа значительно дороже жесткими дисков. Применение флэш-памяти все еще ограничивается мобильными устройствами, хотя эта область информационных технологий тоже отнюдь не мала. Предполагается, что применение флэш-памяти будет расширяться. Заметны следующие тенденции рынка. Будут развиваться интегрированные решения, которые включают лучшие качества различных систем. Это, например, *on-chip (single-chip)* устройства, интегрирующие в одной микросхеме флэш-

память, процессор, память случайного доступа и специальное программное обеспечение. Такие системы позволяют хранить данные и одновременно выполнять программный код. Это один из путей уменьшения стоимости и размеров систем, одновременно с увеличением их производительности.

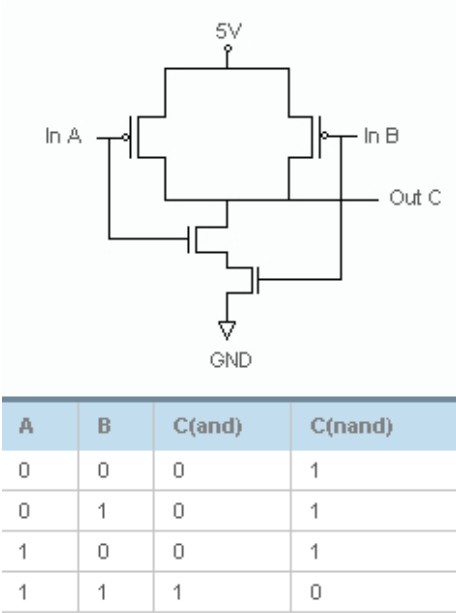


Рис. 2.2.5. Схема логического элемента *NAND*, который в булевой алгебре обозначает отрицание «И». В таблице для сравнения показано, какие выходные сигналы будут в случае операций *AND* и *NAND*

Рассмотрим также флэш-память *superAND* с функциями управления (рис. 2.2.6), которые раньше были отдельно, а теперь встроены прямо в ту же микросхему. К ним относятся, например, функции контроля испорченных секторов, коррекции ошибок текущего состояния ячеек памяти. Плохие испорченные участки всегда встречаются (не забывайте, что их сотни миллионов). В новых устройствах имеется около 3% нерабочих ячеек, что является нормой и не нарушает пригодность устройства в целом. Со временем количество таких элементов, разумеется, увеличивается (нагрев, механические вибрации,

намагничивание внешними полями). Чтобы это не привело к выходу из строя всего прибора, во флэш-памяти предусмотрен резервный объем. Когда появляется испорченный сектор, автоматически меняется его адрес в таблице размещения файлов, и он выбывает из игры, будучи замененным на исправный сектор из резервной области.

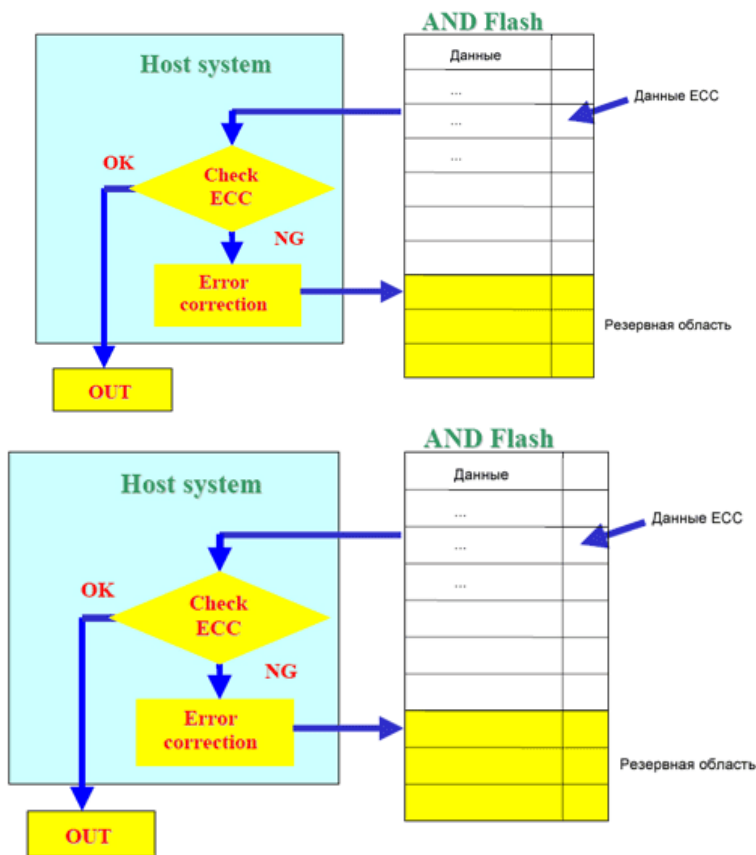


Рис. 2.2.6. Флэш-память superAND с функциями управления

Конечно, на приведенных выше схемах памяти все выглядит ровно и аккуратно. На самом деле, если разрезать «флэшку» и изучать ее в электронном микроскопе, рисунки получатся не такими уж идеальными (<https://geektimes.ru/post/253940/>) (рис. 2.2.7 и 2.2.8).

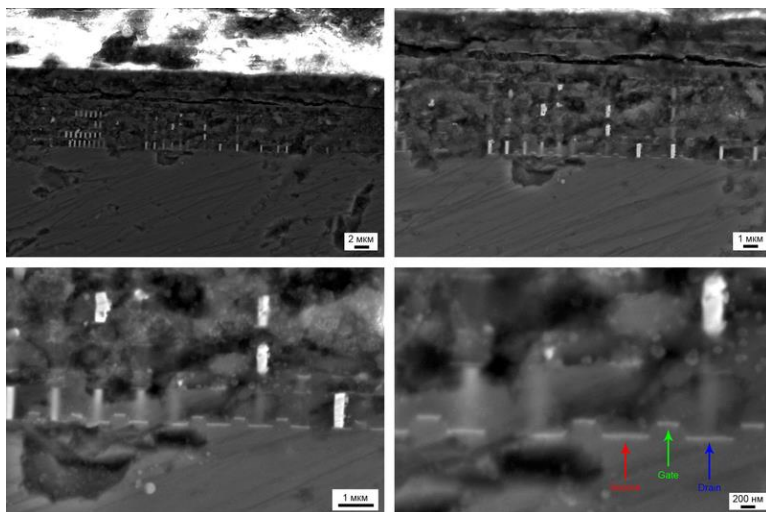


Рис. 2.2.7. Стрелками показаны сток, исток, затвор в элементах флэш-памяти

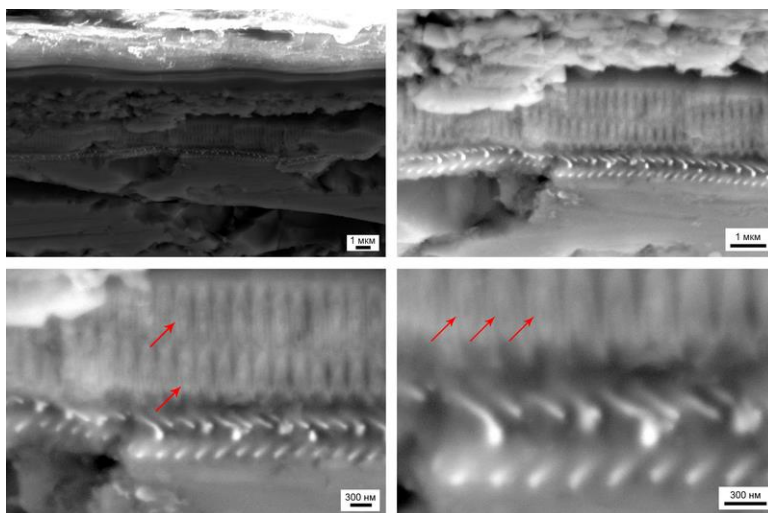


Рис. 2.2.8. Набор элементов памяти (показаны стрелками).
Наличие двух строк элементов говорит о том,
что в одной ячейке хранится 2 бита информации

2.4. Лабораторная работа. Симулятор логических элементов

Для изучения логических элементов следует перейти на страничку <http://www.neuroproductions.be/logic-lab/>. Перед вами откроется панель, показанная на рис. 2.2.9.

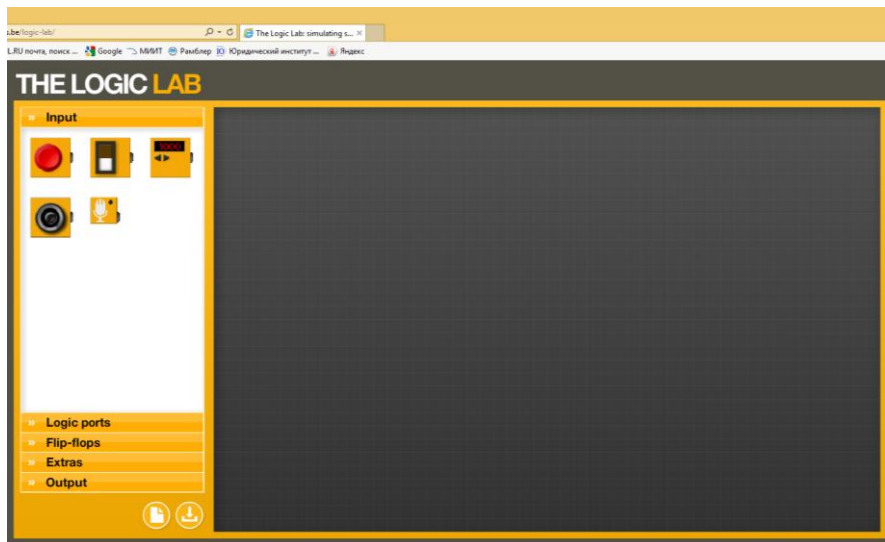


Рис. 2.2.9. Исходная панель для моделирования

На этой панели собраны простейшие схемы, осуществляющие логические операции. Слева имеется панель, позволяющая выбирать элементы, необходимые для создания схемы. Чтобы работа с этими устройствами стала доступной, нужно опуститься вниз странички и найти там примеры *Examples/showcase* (рис. 2.2.10). Начнем с исследования работы отдельных элементов. Для этого нужно нажать на панель *Logic Ports*. Тогда все элементы «оживут». Мышкой вы можете теперь передвигать эти элементы, выбрасывать их, добавлять новые, изменять подключение проводов и т.д. и т.п.

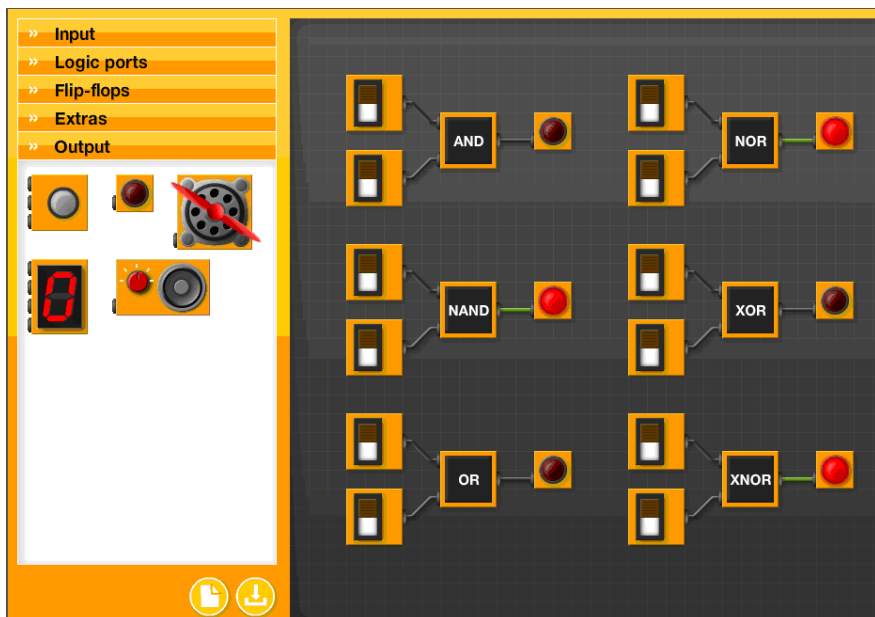


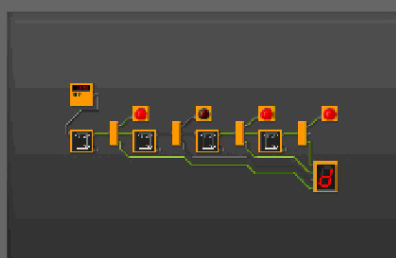
Рис. 2.2.10. Панель классического симулятора логических операций

Перечислим элементы этой панели, которые нужно выбирать. На рис. 2.2.11 показана панель логических операций (*Logic ports*) (рис. 2.2.12). Вы видите на ней набор уже знакомых (*AND*, *NAND*, *OR*, *NOR*) и некоторых новых (*XOR*, *XNOR*, *NOT*) операций. Все эти элементы расположены на панели. Для операций *AND*, *NAND*, *OR*, *NOR* таблички их работы были приведены выше. В этой работе первым заданием будет построение аналогичных таблиц для новых элементов *XOR*, *XNOR*, *NOT*.

EXAMPLES / SHOWCASE



LOGIC PORTS
by The Logic Lab



BINARY COUNTER
by The Logic Lab

Рис. 2.2.11. Панели действующих онлайн образцов классической логики

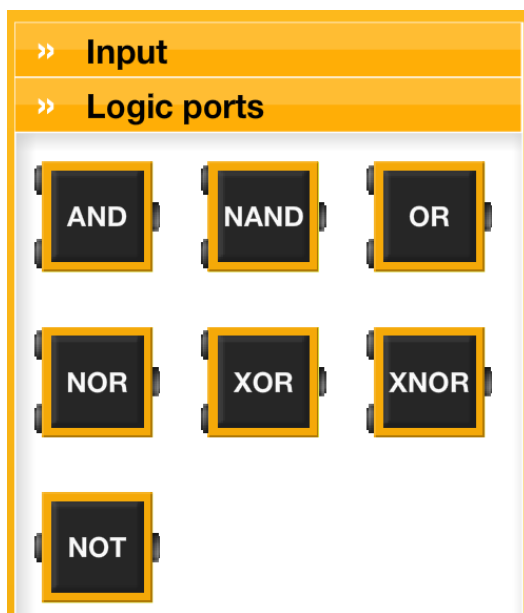


Рис. 2.2.12. Панель логических операций

С этой целью рассмотрим сначала панель *Input*, отвечающую за источники сигналов, подаваемых на логические устройства (рис. 2.2.13). Они нужны для того, чтобы заставить работать логические устройства, подавать на их вход определенные сигналы. Из них нам понадобится переключатель «белое красное» — он при нажатии дает 0 или 1 на выходе. Это самое простое и самое нужное устройство, позволяющее медленно и спокойно разобраться, во что преобразуется сигнал на выходе. Можно заметить, что к каждому логическому элементу таких однопроводных источников может быть подключено несколько. В этом случае реализуются логические операции сравнения. Имеется также генератор импульсов регулируемой длительности, кнопка, микрофон. Эти элементы вам предстоит изучить самостоятельно.

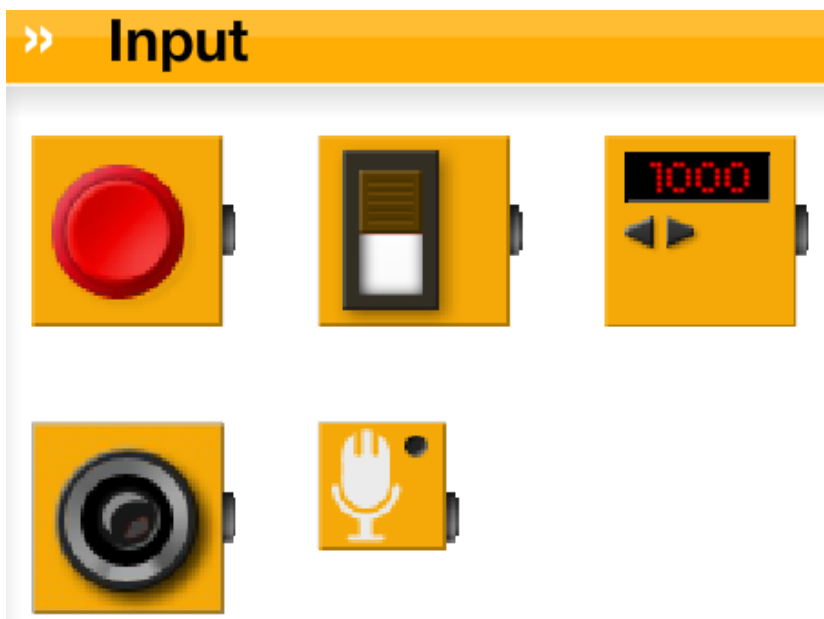


Рис. 2.2.13. Панель источников входного сигнала

Итак, у нас имеются в распоряжении источники сигналов, а также логические преобразователи разных типов. Но для того чтобы понять, как работают эти устройства, нужно уметь регистрировать вы-

ходные сигналы. Для этого имеется также целый арсенал устройств (рис. 2.2.14). Самое простое устройство — лампочка с одним входом. (Разумеется, в реальности у лампочки два электрода, но этим обстоятельством пренебрегают.) Но кроме лампочки есть и моторчик. Он нужен не только для того, чтобы улучшать настроение, но и для тех, кто плохо различает цвета. Есть и динамик с сигналом и регулятором для управления громкостью, счетчик и другие элементы.

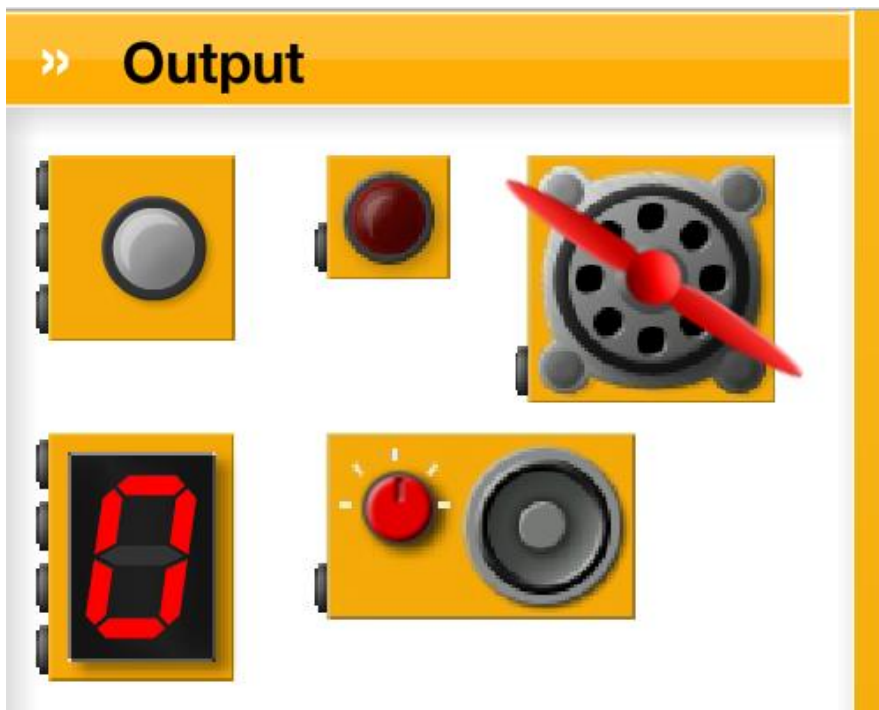


Рис. 2.2.14. Панель регистраторов сигналов

» Flip-flops

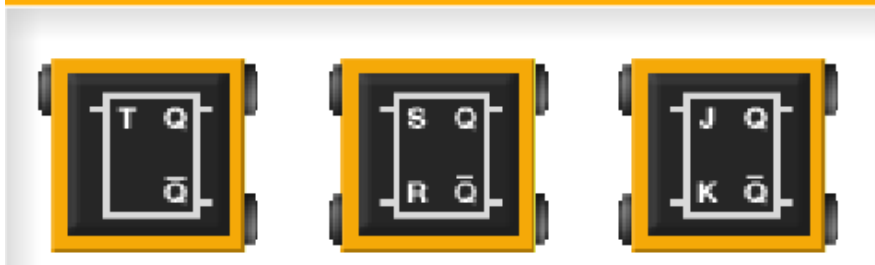


Рис. 2.2.15. Панель триггеров

Наконец, имеются также элементы *Flip-flops*, что означает «опрокидывание» (рис. 2.2.15). Это название появилось потому, что эти элементы как бы опрокидывают входящий сигнал (превращают 0 в 1, и тогда это называется *flop*), или меняют сигналы местами (*flip-flop*). Они называются триггерами. У этих элементов есть особенность — они запоминают записанное состояние и сохраняют его после отключения сигнала, т.е. как раз и являются элементами памяти. Попробуйте собрать схему триггера, изображенную на рис. 2.2.16, чтобы убедиться в этом.

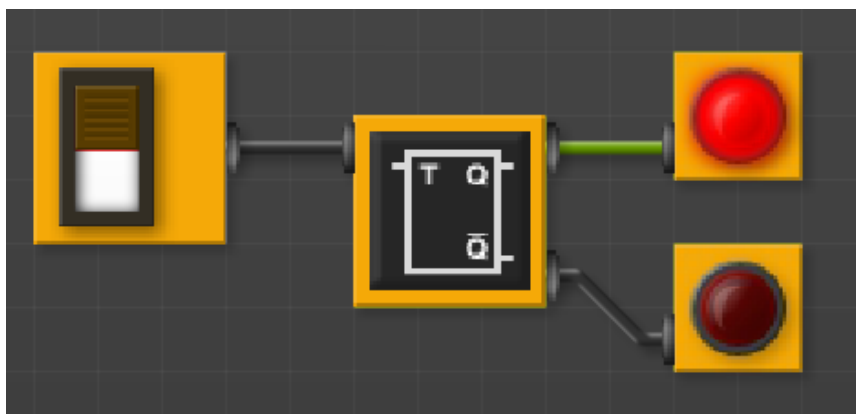


Рис. 2.2.16. Схема триггера

Обратимся к Википедии за определением триггера.

Триггер — электронное устройство, которое долго находится в одном из двух устойчивых состояний и позволяет менять их поочередно под действием внешних импульсов.

О состоянии, которое «запомнил» триггер, можно судить по выходному сигналу. Смена состояний триггеров случается за очень короткое время. Поэтому их относят к импульсным устройствам. Таким образом, триггер запоминает двоичную информацию. Память триггера — это его способность оставаться в одном из двух стабильных состояний и после прекращения действия переключающего сигнала. Приняв одно из состояний за «1», а другое за «0», можно считать, что триггер хранит (помнит) один разряд числа, записанного в двоичном коде.

Триггеры подразделяются на две большие группы — *динамические* и *статические*. Названы они так по способу представления выходной информации. *Динамический* триггер представляет собой управляемый генератор, одно из состояний которого (единичное) характеризуется наличием на выходе непрерывной последовательности импульсов определённой частоты, а другое (нулевое) — отсутствием выходных импульсов. Смена состояний производится внешними импульсами. К *статическим* триггерам относят устройства, каждое состояние которых характеризуется неизменными уровнями выходного напряжения (выходными потенциалами): высоким — близким к напряжению питания и низким — около нуля. Статические триггеры по способу представления выходной информации часто называют потенциальными. Статические (потенциальные) триггеры, в свою очередь, подразделяются на две неравные по практическому значению группы — симметричные и несимметричные триггеры. Симметричные триггеры отличает симметрия схемы и по структуре, и по параметрам элементов обоих плеч. Для несимметричных триггеров характерна неидентичность параметров элементов отдельных каскадов, а также и связей между ними. *Двухступенчатые* триггеры (*flip-flop*, *шлёпающие*) делятся на триггеры со статическим управлением и триггеры с динамическим управлением. При одном уровне сигнала на входе *C* информация, в соответствии с логикой работы триггера, записывается в первую ступень (вторая ступень заблокирована для записи). При другом уровне этого сигнала происходит копирование со-

стояния первой ступени во вторую (первая ступень заблокирована для записи), выходной сигнал появляется в этот момент времени с задержкой, равной задержке срабатывания ступени.

Наконец, есть дополнительные устройства *Extras* (рис. 2.2.17), которые представляют собой разветвитель, нужный для того, чтобы один и тот же сигнал направить сразу на несколько устройств, и прерыватель, который с заданной частотой прерывает подаваемый на него сигнал.

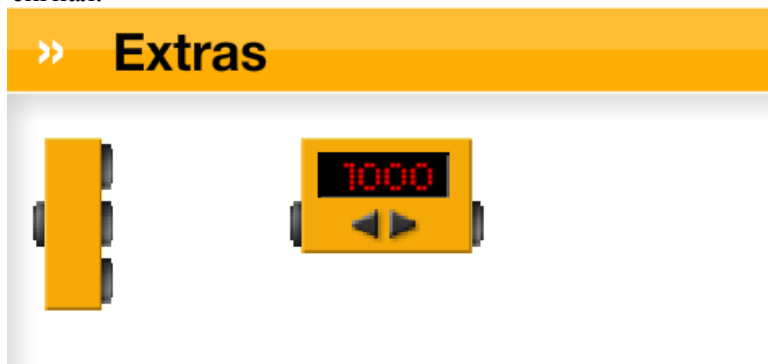


Рис. 2.2.17. Разветвитель и прерыватель с регулируемой частотой

Действия каждого из вышеописанных элементов вы сможете изучить самостоятельно, если в схему 2.2.16 будете вставлять их вместо имеющихся там элементов

Посмотрим, что можно собрать из этих устройств. Начать нужно с простого. На схеме для датчика *NAND* (сначала вдоволь наигравшись с ней и убедившись, что она выполняет ранее приведенную таблицу) замените, например, один из источников постоянного напряжения генератором с регулируемой частотой (рис. 2.2.18). Вместо лампочки на выходе поставьте звуковой сигнал с регулируемым тоном (или пропеллер, или счетчик). Попробуйте все элементы. Вы убедитесь, что оператор *NAND* будет отрицать «и», т.е. он будет пропускать сигнал от генератора тогда, когда источник постоянного тока выключен, либо пропускать сигнал включенного источника в тот момент, когда на генераторе нет импульса. В любом случае вы услышите последовательность гудков. Однако важно проследить именно за тем, как и в какие моменты она формируется.

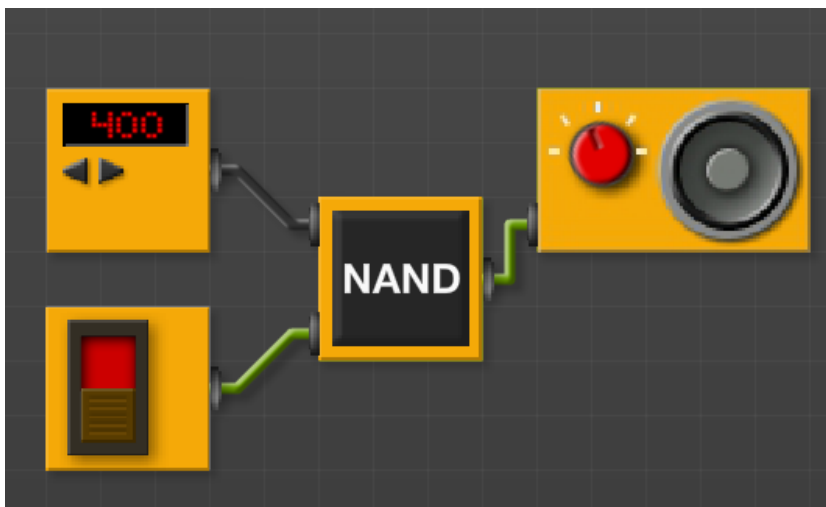


Рис. 2.2.18. Схема триггера

Задания

1. Соберите поочередно схемы, подобные рис. 2.2.9 для исследования свойств триггеров. Создайте таблицы работы всех типов триггеров, имеющихся в меню по образцу.

S	R	Q(t)	Q(t)
0	1	0	1
1	0	1	0
0	0	сохраняет состояние	сохраняет состояние
1	1	не определено	не определено

2. Запустите двухразрядный (бинарный) счетчик (*binary counts*) и объясните принцип его работы (рис. 2.2.19). Преобразуйте бинарный счетчик, заменив в нем как можно больше элементов, не нарушая принцип и результат его работы.

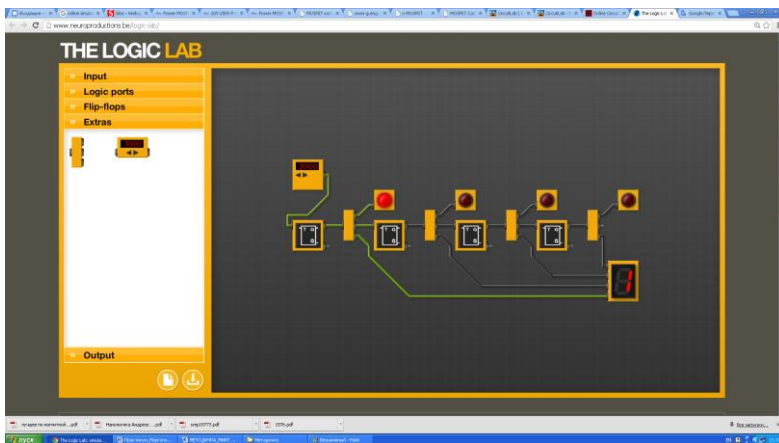


Рис. 2.2.19. Бинарный счетчик

3. Выполните операции сложения $3+4$ и $10+12$ по следующему алгоритму:

- переведите числа в двоичный код;
- с помощью симулятора соберите схему, складывающую числа;
- получите результат в двоичном коде;
- переведите в десятичную форму и убедитесь в правильности сложения.

Контрольные вопросы

1. Что такое триггер?
2. Какие типы триггеров вам известны?
3. Чем отличаются флэш-памяти типа *NOR* и *NAND*?
4. Что такое технология *RAM*?
5. Как работает функция коррекции ошибок во флэш-памяти?
6. Перечислите логические операции, которые были использованы в лабораторной работе.
7. Рассчитайте количество информации и соответствующее минимальное количество энергии, необходимое для выполнения команды *NAND*.
8. Опишите устройство элементарной ячейки памяти, основанное на транзисторе *FET*.
9. Из каких элементов состоит «флэшка»?

Раздел 3. Проблемы миниатюризации и экономии энергии

3.1. Энергетический предел миниатюризации элементов памяти

Основное препятствие к уменьшению размеров компьютеров — рост тепловых потерь, которые возникают при переключении состояний логических элементов, по мере уплотнения элементов. В вышеописанных ситуациях классических компьютеров сегодняшнего дня эти потери очень велики и возникают в результате протекания электрического тока по проводникам. Хотя в расчете на один транзистор не требуются высокие электрические мощности, но при учете, что размеры транзистора составляют 10—30 нм, получается, что на 1 кв. см, на котором расположены миллионы таких устройств, приходится довольно высокая рассеиваемая тепловая мощность. Эта мощность разогревает транзисторы, увеличивает шум и в конечном итоге выводит из строя всю систему. В современных бытовых компьютерах мощность, рассеиваемая процессором, составляет 40—130 Вт, что равно теплу, выделяемому лампой накаливания, и имеется множество способов борьбы с этим теплом. Самый простой — установка радиатора на микропроцессор (рис. 3.1.1).



Рис. 3.1.1. Устройство охлаждения микропроцессора

Однако предлагаются и реализуются другие способы, например, охлаждение с помощью термоэлектрических систем, которые охлаждают компьютер при протекании тока по ним. Это такой микрохолодильничек, который располагают между радиатором и процессором (рис. 3.1.2).

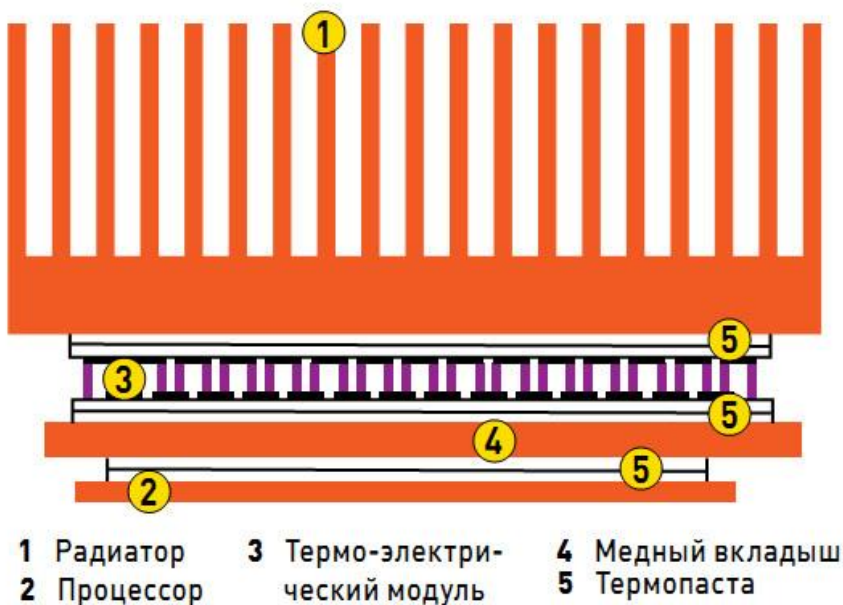


Рис. 3.1.2. Термоэлектрическое охлаждение компьютера

Применяются также системы жидкостного охлаждения компьютеров (рис. 3.1.3).



Рис. 3.1.3. Система жидкостного охлаждения компьютера

Наиболее прогрессивные системы снабжены термотрубками (рис. 3.1.4).

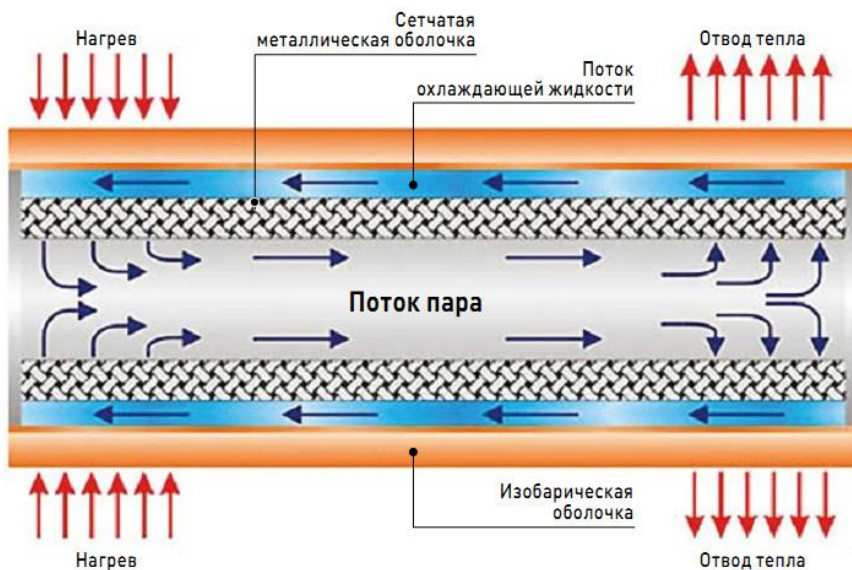


Рис. 3.1.4. Термотрубка для охлаждения процессора

Внутри термотрубки имеется циркуляция легко испаряемого вещества, которое эффективно забирает лишнюю тепловую энергию при испарении.

Все эти меры способны изменить ситуацию на десятки процентов, не улучшая ее радикально потому, что не устраняется главная причина тепловыделения — электрический ток. К тому же все перечисленные меры очень дороги и не позволяют радикально изменить свойства компьютера. Далее будет показано, что существуют современные технологии, позволяющие минимизировать присутствие в схемах именно электрических токов. Расплатой за это является необходимость разработки компьютеров, основанных на иных физических принципах, по сравнению с теми, что были описаны в предыдущих разделах.

Здесь пойдет речь о пределе, который можно достигнуть на пути минимизации тепловыделения логических устройств. Оказывается, что независимо от физического устройства и принципов работы логического элемента, при потере 1 бит информации выделяется минимально возможная теплота, количество которой W описывается формулой:

$$W = kT \ln 2$$

Здесь $k = 1.38 \cdot 10^{-23}$ Дж/К, T — температура устройства.

Это утверждение называется принципом Ландауэра, потому что было выдвинуто в 1961 г. Р. Ландауэром (IBM). Поскольку причиной выделения тепла является уже известный нам факт, что количество информации соответствует определенному количеству энергии, а на каждый акт обработки информации необратимо тратится энергия, то единственным способом обойти ограничение, накладываемое принципом Ландауэра, является использование вычислений, совершаемых при использовании обратимых физических процессов. Выражением Шеннона — Фон-Неймана — Ландауэра (*Shannon-von Neumann-Landauer, SNL*) называют минимальную энергию, необходимую для переключения бита E_{bit} (это минимальная высота барьера, необходимая для разделения двух состояний ячейки памяти)¹:

$$E_{\text{bit}} > k_B T \ln 2 = 0.017 \text{ эВ при } T = 300 \text{ К.}$$

¹ См.: *Landauer R. Irreversibility and heat generation in the computing process // IBM Journal of Research and Development. 1961. Vol. 5. P. 183—191.*

Если барьер между двумя состояниями (например, транзистора) сделать меньше, то случайные флуктуации будут приводить к очень быстрой потере информации и невозможности совершения логических операций. Если барьер увеличивать, то придется на переключение тратить очень много энергии и напрасно разогревать систему. В сущности, в современных компьютерах так и происходит. Каждый логический элемент рассеивает примерно 10^4 эВ на одну логическую операцию, т.е. примерно в миллион раз больше, чем предел Ландауэра. Таким образом, энерго-совершенство компьютеров можно улучшить в миллионы раз, если изменить принципы их работы.

Требования, предъявляемые к минимальным размерам, потребляемой энергии, скорости работы приборов современной полупроводниковой электроники стремительно растут. Это ставит задачу поиска и внедрения в технику альтернативных материалов и структур, работающих на неклассических принципах. Современные электронные логические элементы работают в двоичном коде и используют в работе заряд электрона. Так, например, в классическом полевом транзисторе открытому состоянию отвечает логический «0», закрытому — «1». На каждое изменение логического состояния транзистора затрачивается электрическая энергия E_{bit} , которая, в конечном счете, трансформируется в тепловую энергию, разогревающую транзистор. Чем выше тактовая частота микропроцессора и чем выше количество транзисторов в нем, тем выше рассеиваемая мощность. Учитывая, что система может находиться в двух состояниях, легко получить приведенную формулу. Теоретическому минимуму рассеивающей мощности на 1 бит E_{bit} соответствует тактовая частота процессора примерно 10 ГГц и количество транзисторов в нем примерно 10^8 . По некоторым прогнозам это значение будет достигнуто в 2020 г., а размер отдельного транзистора составит примерно 20 нм (рис. 3.1.5). Принцип Ландауэра является естественным ограничением производительности микропроцессоров до критического значения тепловой мощности примерно 10^3 Вт, «отводить» которую становится невозможно.

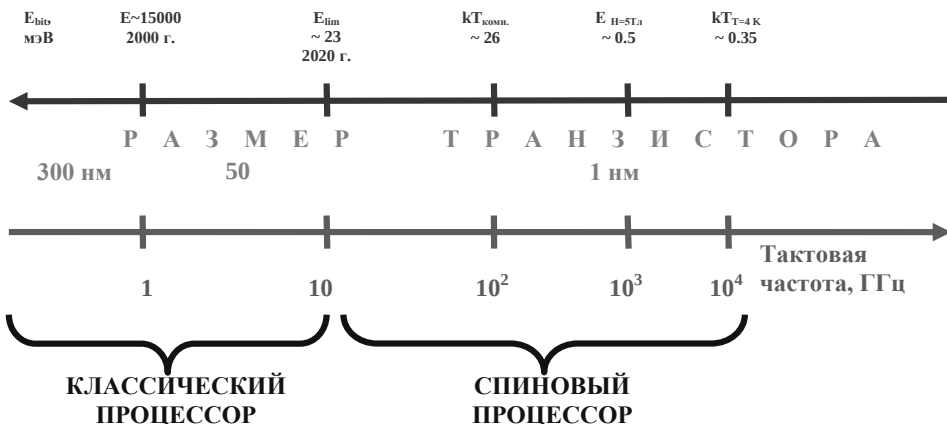


Рис. 3.1.5. Рост тактовой частоты классических и спиновых процессоров

В спиновом процессоре логическим элементам «0» и «1» отвечают спины (магнитные моменты) атомов, ориентированные «вверх» или «вниз». Скорость работы такого устройства будет определяться частотой переходов спинов из одного состояния в другое. На такие системы, которые не пребывают в равновесии, принцип Ландауэра, справедливый только для равновесных систем, не распространяется. Для энергии $E = 0,5$ мэВ, которая соответствует разности энергий для атомов со спинами «вниз» и «вверх» в магнитном поле 5 Тл, тактовая частота спинового транзистора составит примерно 1 ТГц (см. рис. 3.1.5). Поэтому создание спиновых логических устройств с пониженной рассеиваемой мощностью E_{bit} позволит на порядок снизить энергопотребление электроники.

На рис. 3.1.6 показано увеличение плотности записи по мере смены физических принципов, на которых основаны устройства памяти. Видно, что достижение ситуации, когда отдельный атом будет использован для хранения бита информации, приведет не только к преодолению предела Ландауэра, но и к многократному увеличению плотности записи и компактированию компьютеров.

Рост поверхностной плотности записи на магнитных дисках

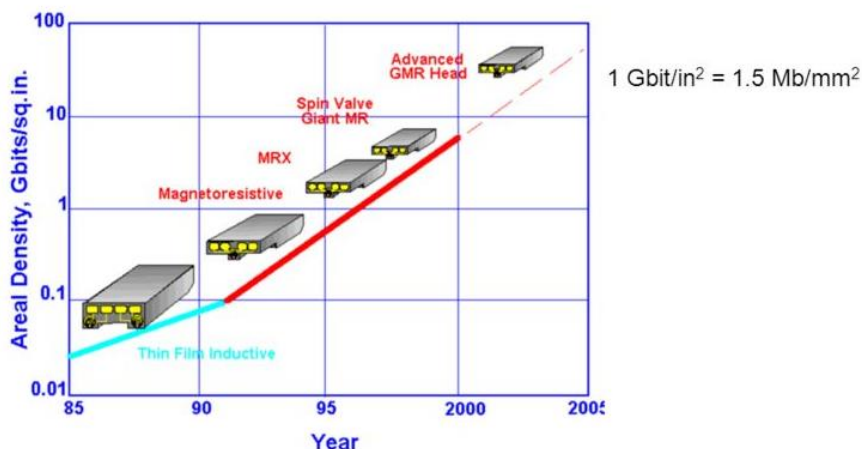


Рис. 3.1.6. Эволюция физических принципов записи информации и ее плотности

3.2. Неклассическая логика, наноманиты и их взаимодействие

Одним из примеров нестандартной технологии логических элементов является использование наноманитов, которые приближают логические операции к пределу Ландауэра, т.е. уменьшают выделение энергии. В Еврокомиссии создана специальная группа, которая в течение двух лет должна разработать классификацию нанопродукции. Среди подходов к определению понятия «нанотехнологии» имеются следующие:

- 1) в техническом комитете ISO/TK 229 под нанотехнологиями дают следующее определение: знание и управление процессами, как правило, в масштабе 1 нм, но не исключающее масштаб менее 100 нм в одном или более измерениях, когда ввод в действие размерного эффекта (явления) приводит к возможности новых применений; использование свойств объектов и материалов в масштабе нанометров, ко-

торые отличаются от свойств свободных атомов или молекул, а также от объемных свойств вещества, состоящего из этих атомов или молекул, для создания более совершенных материалов, приборов, систем, реализующих эти свойства;

2) нанотехнология определяется как совокупность методов и приемов, обеспечивающих возможность контролируемым образом создавать и модифицировать объекты, включающие компоненты с размерами менее 100 нм, хотя бы в одном измерении, и в результате этого получившие принципиально новые качества, позволяющие осуществлять их интеграцию в полноценно функционирующие системы большего масштаба.

Простейшее устройство, созданное для выполнения логических операций при минимально возможных затратах энергии, представлено на рис. 3.2.1.

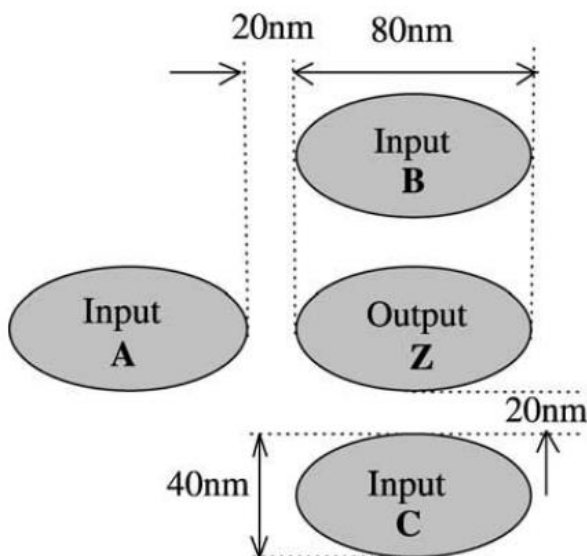


Рис. 3.2.1. Схема устройства, состоящее из 4 наночастиц, способное выполнять логические операции с энергозатратами, близкими к пределу Ландауэра¹

¹ Anisul Haque S. et al. Magnetic logic gate for binary computing // Science and Technology of Advanced Materials. 2004. № 5. S. 79—82.

Это устройство содержит всего четыре небольшие эллипсоидальные наночастички, расположенные на поверхности полупроводника и способные изменять свою ориентацию во внешнем магнитном поле, подобно стрелке компаса. Если теперь представить, что намагниченность этих частиц можно изменять и контролировать, то одно из состояний намагниченности можно принять логическим нулем, а другое — единицей. Современное оборудование позволяет не только видеть сами частицы, но и определять, в какую сторону намагничен эллипсоид (рис.3.2.2). Белое пятнышко на этих фотографиях соответствует «северному полюсу» наномagnита, а черное пятнышко — «южному полюсу».

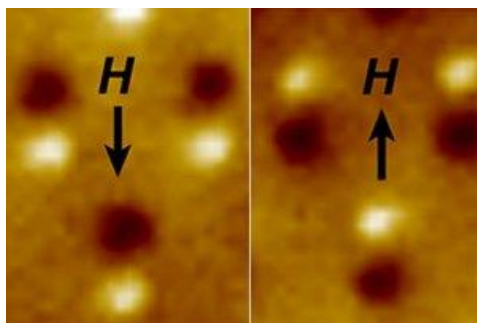


Рис. 3.2.2. Изображение наночастиц, намагничиваемых внешним полем H в разных логических состояниях. Изображение получено магнитным микроскопом.

Различные возможные состояния четверки наночастиц в зависимости от приложенного магнитного поля представлены на рис. 3.2.3. Интуитивно ясно, например, что два магнетика, расположенные рядом, будут стремиться занять противоположные направления по намагниченности. Именно поэтому три «входные частицы» (слева) комбинацией своих магнитных полей могут управлять состоянием выходной частицы. Всего имеется восемь равновесных конфигураций намагниченности у такой системы, потому что $2^3 = 8$ отвечает количеству комбинаций из трех частиц по два состояния у каждой. На рис. 3.2.3 указано также время переключения между состояниями в наносекундах, что вполне отвечает современным частотам работы процессора.

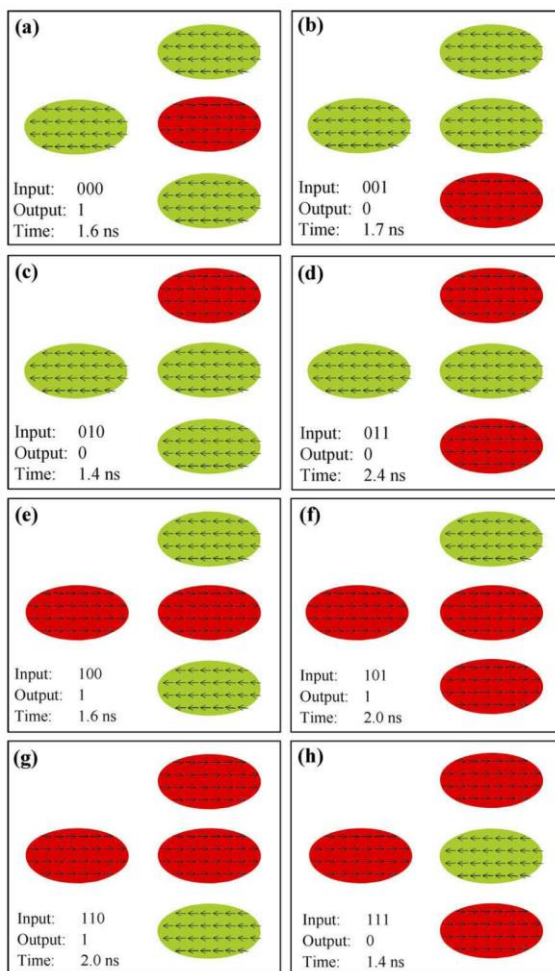


Рис. 3.2.3. Схема наноманитного устройства в различных состояниях, переключаемых полем входной наночастицы. Зеленым цветом отмечены частицы, намагниченность которых отвечает логическому нулю, а красным — наночастицы, соответствующие логической единице¹

¹ Anisul Haque S. et al. Magnetic logic gate for binary computing // Science and Technology of Advanced Materials. 2004. № 5. S. 79—82.

Например, на рис. 3.2.3а изображены три входящие частицы (отмечены зеленым, соответственно направлению их намагниченности) и одна «выходящая» частица, отмеченная красным. Намагниченность последней направлена против внешнего магнитного поля, создаваемого остальными тремя частицами, т.е. противоположно их намагниченности. Если теперь менять намагниченности внешних частиц, «выходящая» частица тоже станет переключаться. Если это рассуждение перевести на язык логических нулей и единиц, получится табличка (рис. 3.2.4), резюмирующая одноразрядное состояние выхода системы в зависимости от трехразрядного состояния входа. Легко заметить (см. разд. 2), что комбинации сигналов соответствуют выполнению этим ансамблем команд *NOR* и *NAND*.

A	B	C	Z	
0	0	0	1	<i>N</i>
0	0	1	0	<i>O</i>
0	1	0	0	<i>R</i>
0	1	1	0	
1	0	0	1	<i>N</i>
1	0	1	1	<i>A</i>
1	1	0	1	<i>N</i>
1	1	1	0	<i>D</i>

*Рис. 3.2.4. Таблица состояний частиц, отвечающих логическим операциям *NOR* и *NAND*¹*

¹ Anisul Haque S. et al. Magnetic logic gate for binary computing // Science and Technology of Advanced Materials. 2004. № 5. S. 79—82.

Таким образом, выполнение логических операций оказывается возможным с помощью совершенно иных физических реализаций, в отсутствие полевых транзисторов и при этом достигается выигрыш в энергии, приходящийся на одну операцию. Он составляет много порядков величины. Это говорит о том, что конструкция современных компьютеров далека от совершенства и будет модифицироваться с учетом эффективности энергосбережения. Конечно, представленная здесь модель наноманитного процессора громоздка и наивна. Она предполагает использование очень медленного атомного силового микроскопа, практически никак не позволяет создать гальванический отклик на перевероты намагниченностей частиц и даже по размерам оказывается не сильно меньше обычного логического элемента *FET*. Однако и первый паровоз или самолет были смешны и едва умели ездить и летать. Целью же нашей работы было получение представлений об альтернативных неклассических способах компьютеринга. Заметим также, что физические основы описанного устройства все же подчиняются классической физике (это для информатики они «не классичны»). Далее в следующих разделах мы затронем и такие принципы реализации логических операций, в основе которых будет уже лежать и неклассическая физика.

3.3. Лабораторная работа. Симулятор однобитной ячейки памяти

1. Зарегистрируйтесь на сайте <http://www.docircuits.com>.
2. Загрузите симулятор 1—битной ячейки памяти <http://www.docircuits.com/circuit-editor/565/sr-latch> (рис. 3.2.5).
3. Нажмите *Run Digital Simulation* и получите отклик на разных выходах ячейки на входящий прямоугольный сигнал (последовательность логических нулей и единиц).
4. Пользуясь результатом симуляции (см., например, рис. 3.2.6), составьте табличку, подобную той, что приведена на рис. 3.2.5 для исследуемого элемента.
5. Сопоставьте полученную таблицу с реализацией логических элементов с той, что возникает в наноманитном процессоре (рис. 3.2.5). Сделайте выводы.

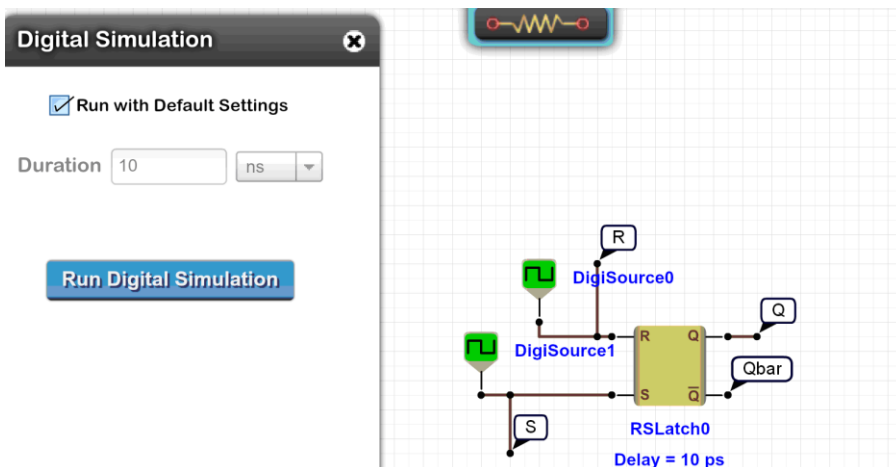


Рис. 3.2.5. Панель симулятора однобитной ячейки памяти

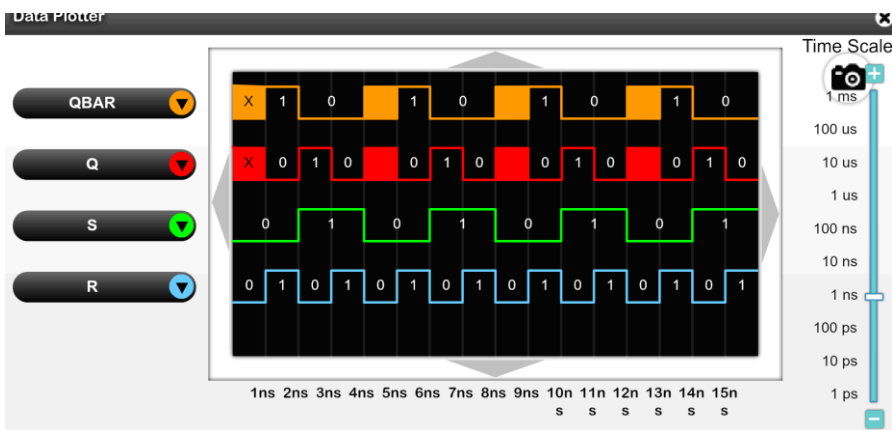


Рис. 3.2.6. Результат симуляции входящей последовательности нулей и единиц на входе $QBAR$ в виде выходящих сигналов Q , R , S

Контрольные вопросы

1. Что такое нанотехнологии?
2. Почему нанотехнологии приближают рассеяние энергии при логических операциях к пределу Ландауэра?

3. Опишите наномангнитный процессор. Какими достоинствами и недостатками он обладает?
4. Каким образом время переключения логического состояния наночастицы связано с энергозатратами на этот процесс.
5. Подчиняется ли наномангнитный процессор принципу Ландауэра?
6. Придумайте и опишите логический элемент, состоящий из двух наночастиц. Какую операцию можно реализовать таким элементом?

Раздел 4. Спинтроника

Спин элементарной частицы (мы будем в основном подразумевать электроны) — это вращательный момент (как у волчка), присутствующий ему наравне с массой $\sim 10^{-31}$ килограмм и зарядом $\sim 10^{-19}$ Кулон. Можно представлять, что электрон — это шарик, который вращается вокруг собственной оси. При этом электроны могут иметь только два направления вращения с одинаковым и неизменным вращательным моментом. Об этом говорят и иначе: имеется лишь две проекции спина электрона. В любом энергетическом состоянии могут находиться два электрона, но только с разными спинами. Обнаружилось, что через намагниченный проводящий металл могут проходить электроны со спином, направленным в ту же сторону, куда намагничен этот металл. Электроны с противоположным спином рассеиваются и проходят через намагниченный металл плохо. Пожалуй, только этих сведений достаточно, чтобы разобраться, как работают современные приборы для обработки информации, основанные не на передаче заряда, а на передаче спина элементарных частиц. Одним из решений проблемы избыточной энергии является создание приборов спинтроники, в которых для управления их свойствами помимо заряда электрона используется его спин. Впервые о спинтронике заговорили в 1988 г. в связи с открытием гигантского магнетосопротивления Альбертом Фертом и Петером Грюнбергом. Эффект гигантского магнетосопротивления обусловлен зависимостью рассеяния электронов от ориентации их спинов, которая, в свою очередь управляется намагниченностью той среды, в которой электроны двигаются. Такая ситуация реализуется в ферромагнитных материалах. В качестве наиболее практичного примера, реализуемого в промышленности, можно привести сплав FeCoB. Понимание физической природы спинового транспорта позволило создать магниторезистивные приборы на основе гигантского магнетосопротивления.

На рис. 4.1.1 приведена схема элемента памяти, построенного на вышеописанном принципе. Верхний и нижний слои этого элемента ферромагнитны. Они состоят из сплава FeCoB. Нижний слой при этом сделан так, чтобы внешнее магнитное поле не могло переключать его намагниченность. Он как бы обеспечивает опорное магнитное поле в прослойке между слоями. Верхний (обычно тонкий) слой

FeCoB способен перемагничиваться. Таким образом, у этой гетероструктуры есть два принципиально разных состояния: когда слои намагничены в одну и ту же сторону и когда направления их намагниченности противоположны. Этим состояниям, которые в отсутствие внешних воздействий могут сохраняться очень долго, можно сопоставить логический ноль и логическую единицу. Возникает вопрос, как считывать эти состояния, получая быстрый доступ к сохраненному состоянию ячейки.

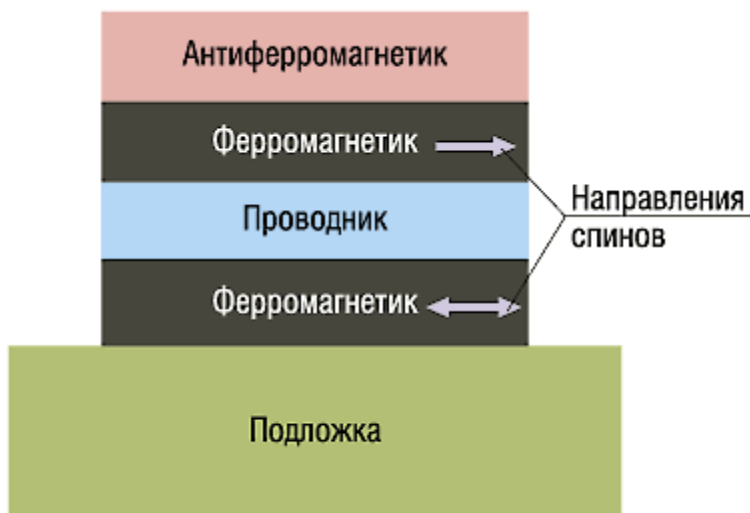


Рис. 4.1.1. Элемент спиновой памяти

На этом этапе физика заканчивается, и начинается инженерия. Немагнитный проводник, разделяющий два ферромагнитных слоя, создает условия для избирательного пропускания электронов, для которых он является барьером. Преодолеть этот барьер электроны могут только тогда, когда верхний и нижний ферромагнитные слои намагничены в одном и том же направлении. Следовательно, такое состояние (назовем его логической единицей) характеризуется хорошей электрической проводимостью через разделяющий слой. В противном случае (если ферромагнитные слои намагничены встречно) наблюдается сильное рассеяние электронов на границе ферромагне-

тик—проводник. В результате это состояние (логический ноль) характеризуется отсутствием электрической проводимости через слоистую структуру. В этой конструкции запись осуществляется путем подачи в течение какого-то времени (времени записи) одинаковых по направлению, но слабых магнитных полей от системы скрещенных проводов (рис. 4.1.2) в результате протекания по ним тока. Этот импульс (которого было бы недостаточно при пропускании тока лишь по одному проводу) заперемагничивает верхний слой и изменяет проводимость элемента памяти. Обнуление информации (инициализация, приведение к исходному состоянию элемента памяти) осуществляется аналогично, только теперь магнитные поля и токи в токопроводах должны быть направлены в противоположную сторону.

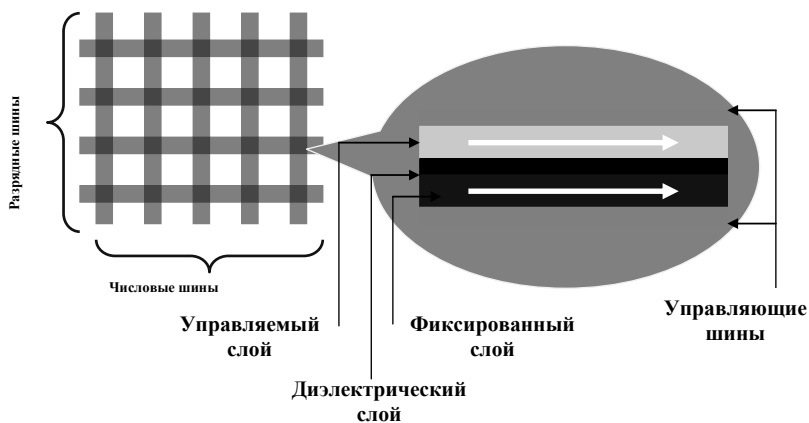
Описанный эффект используют для создания полевых транзисторов, которые основаны на «спинтронном» эффекте. Ферромагнетики и являются истоком и стоком транзистора, и они всегда намагничены одинаково и постоянно. Еще один ферромагнетик между ними является проводящим каналом. Путем подачи на него поля с разным направлением можно варьировать проводимость канала и добиваться разного значения тока.

Основной недостаток приборов спинтроники — использование разных источников переменного магнитного поля и создание активных областей, соответствующих различным ферромагнетикам, с быстрым реагированием на переключение магнитного поля. В результате очень непросто создать области с малыми размерами, а в случае получения таких ферромагнитных областей — очень не просто изолировать их от влияния не своих полей. Например, подавая магнитное поле на верхний ферромагнетик, мы частично влияем и на нижний ферромагнетик, зачастую не позволяя переключиться ему в нужное состояние. Чтобы избежать таких паразитных переключений нужно создавать очень сложные изолирующие конструкции, что заметно увеличивает размеры элементов и самих приборных структур.

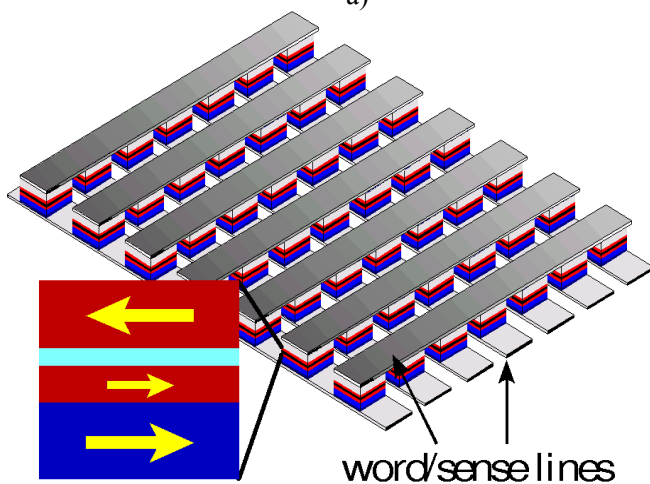
Но главной проблемой описанного устройства является необходимость пропускать токи по проводам, которые неизбежно приводят к энерговыделению и неприемлемо высокой мощности, требуемой для совершения единичной логической операции.

Устройства памяти, инженерия MRAM, SRAM. Прогресс в средствах обработки информации непосредственно связан с техноло-

гией хранения данных (рис. 4.2.1). К революционным достижениям в этой области можно отнести создание *MRAM*-чипов (*Magnetic random-access memory* — магнитная память с произвольным доступом) (рис. 4.2.1). В *MRAM* каждая ячейка может быть битом памяти и представляет собой область ферромагнитного материала размером в несколько нанометров. Если этот элемент намагничен в одном направлении, то это соответствует «1», если в другом, то — «0». Элементы *MRAM* расположены в виде квадратной сетки. Данные записываются путем пропускания тока через проводящие линии («нанопровода»), расположенные выше и ниже этих элементов. Эти линии называют, соответственно, разрядными и числовыми шинами. Ток, протекающий по плоскому проводнику, создает магнитное поле, направленное в плоскости проводника перпендикулярно направлению тока. Чтобы перемагнитить участок ферромагнитного материала, отвечающий одному биту, недостаточно величины электрического тока в одной из управляющих шин. Лишь одновременное воздействие магнитного поля от обеих шин приводит к перемагничиванию. Таким образом, одновременное использование двух управляющих шин позволяет направлять запись информации в определенную область, обходясь без механически движущихся частей и устройств позиционирования. Считывание информации в *MRAM* основано на эффекте гигантского магнетосопротивления. К управляющим шинам прикладывается разность потенциалов. Если направление намагниченности управляемого ферромагнитного слоя совпадает с направлением намагниченности фиксированного ферромагнитного слоя, то туннельный ток через диэлектрический слой течет.



а)



б)

Рис. 4.2.1. Схема MRAM (word line — линия слов, sense line — опорная линия): а — вид сверху, б — вид сбоку. На врезке показано направление намагниченности внутри каждого элемента памяти

Если направление намагниченности управляемого слоя противоположно намагниченности фиксированного слоя ферромагнетика, то туннельный ток отсутствует. Таким образом, регистрируя электрический ток, можно считывать биты информации. *MRAM* значительно превосходит все существующие типы памяти по плотности записи и скорости записи/считывания. Определенные требования предъявляются к записывающей среде, которая в лучшем варианте должна представлять собой независимые, геометрически разделенные ячейки памяти.

Соответствие логических нулей и единиц состояниям намагниченности гетероструктуры показано на рис. 4.2.2. Из этого рисунка видно, что о состоянии намагниченности двуслойной системы можно судить по ее электрическому сопротивлению (проводимости), которое измеряется почти мгновенно. Причем в схеме *MRAM* отсутствуют движущиеся части, и любой элемент оказывается одинаково доступным.

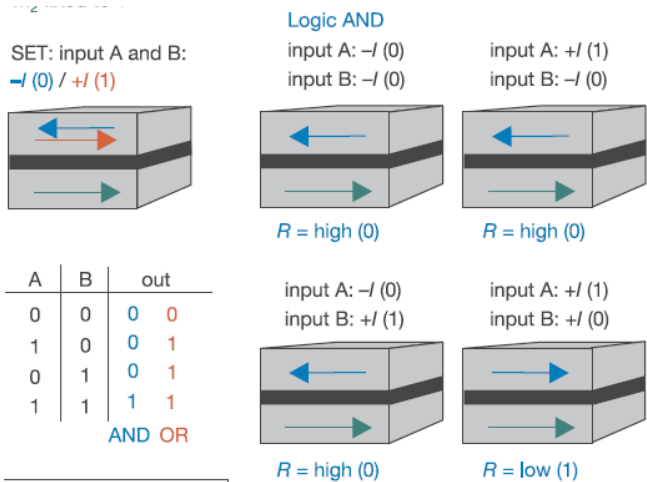


Рис. 4.2.2. Реализация логических операций *AND* и *OR* с помощью гетероструктуры с гигантским магнитосопротивлением¹

¹ Ney A. et al. Magnetic-field-controlled reconfigurable semiconductor logic // Nature. Vol. 425, 485(2003).

Существует значительно улучшенная конструкция спиновой ячейки памяти, которая называется *SRAM* (*Spin Random Access Memory*). В отличие от вышеописанной ситуации, в такой ячейке переключение намагниченности слоев достигается не магнитным полем токопроводящих шин, а так называемым спин-поляризованным током. Это такое движение заряженных частиц, при котором у электронов, движущихся внутрь гетероструктуры, спины направлены в одну сторону, а у электронов, которые движутся в противоположном направлении, — в другую. При этом электрический ток, который отвечает переносу заряда, может быть равен нулю. Это большой успех, потому что в результате отсутствует нагрев проводов и потери энергии. При этом информация теперь переносится не зарядами, а спинами. Электроны «дарят» свою намагниченность ферромагнитному слою, и если она отвечает противоположному направлению намагниченности по сравнению с исходным, то могут привести к опрокидыванию магнитного момента слоя (к его перемагничиванию). Поскольку теперь этот процесс происходит не в результате действия магнитного поля, а в результате потока частиц с ориентированными спинами, в слове *MRAM* первая буква *M* и была заменена на *S*.

Устроена такая ячейка памяти несколько сложнее. Она, разумеется, включает в себя те элементы, что представлены на рис. 4.1.1, однако теперь ячейка снабжена дополнительным ферромагнитным слоем для создания спин-поляризованного тока и дополнительными разделительными и покровными слоями, технологически необходимыми для поддержания вышеописанного процесса (рис. 4.2.3).

Теперь представьте, что над носителем информации закреплен вышеописанный магниторезистивный «сэндвич», и он чувствует намагниченность носителя, т.е. реагирует своим сопротивлением на логические нули и единицы, закодированные в магнитном слое, как на рис. 4.2.4. Что за устройство тогда получится? Правильно! Это бесконтактная высокочувствительная, и притом сверхбыстрая, головка для чтения информации с жестких дисков. Именно такая головка стоит сегодня в вашем компьютере и серийно производится фирмой *IBM*.

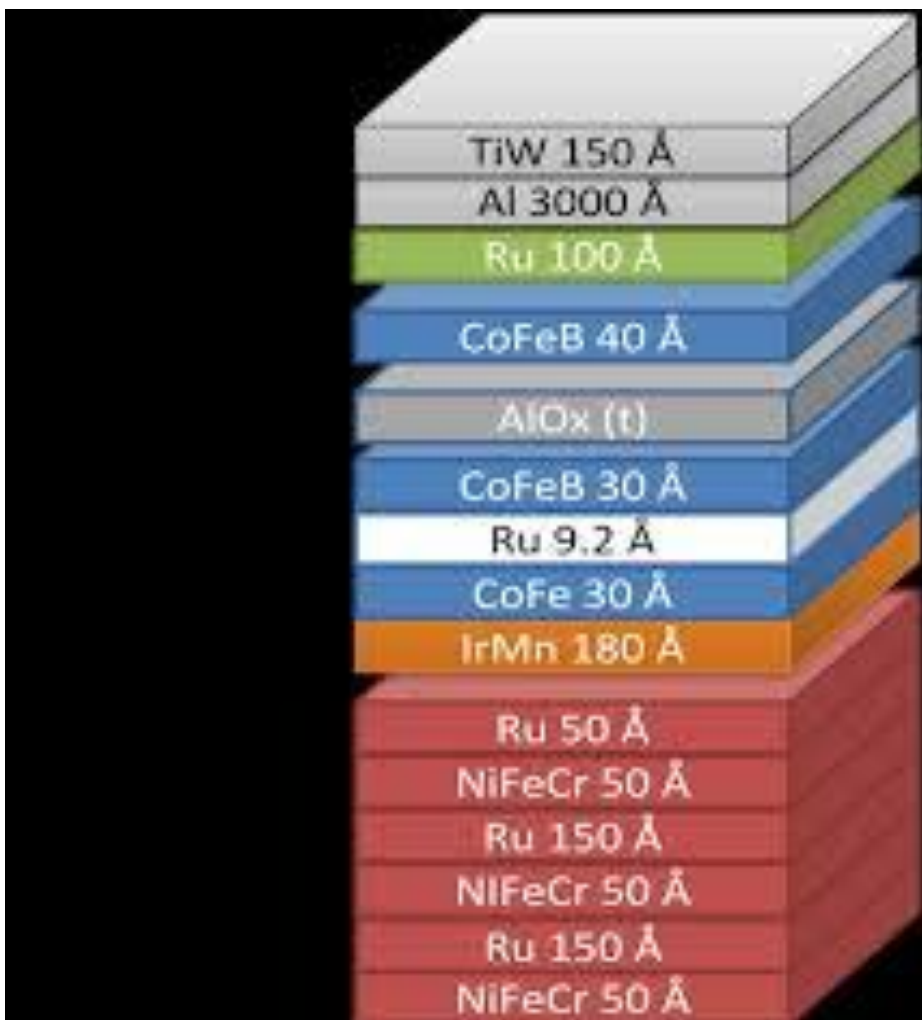


Рис. 4.2.3. Элемент памяти SRAM

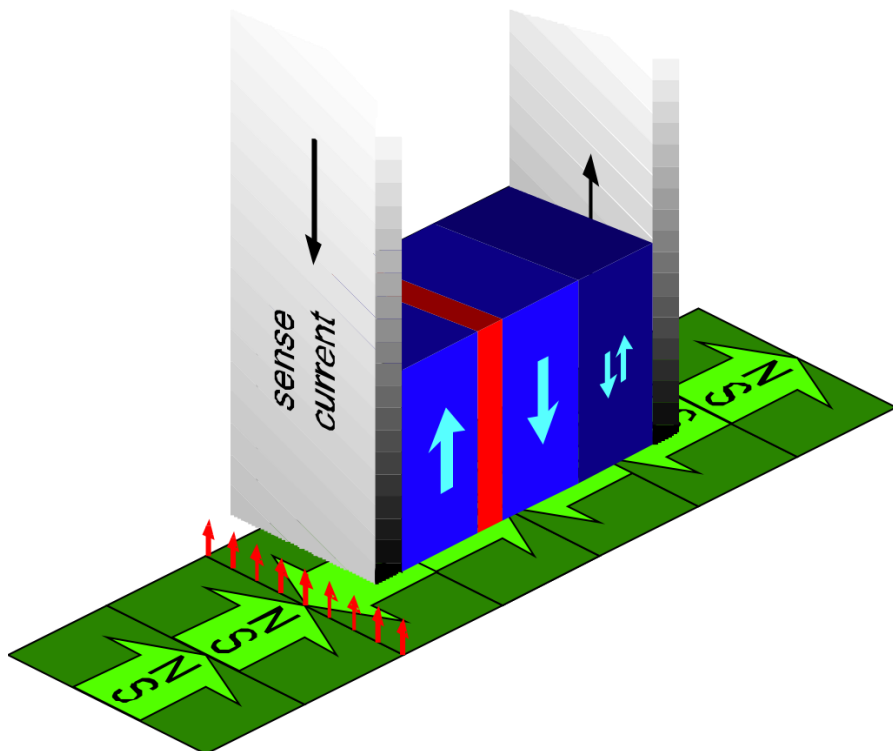


Рис. 4.2.4. Принципиальная схема магниторезистивной головки для считывания информации

Не следует думать, что технология изготовления вышеописанных элементов является завершенной и оконченной. Хотя описанные элементы и используются в промышленности, например, процесс перемагничивания слоя не так тривиален. Он происходит расширением областей, намагниченных в одну сторону за счет поглощения остальных областей, намагниченных в противоположную сторону (рис. 4.2.5).

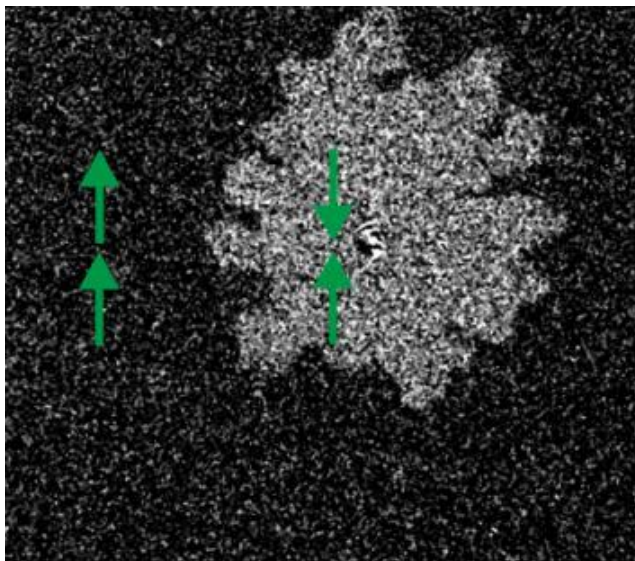


Рис. 4.2.5. Поверхность магниторезистивной ячейки памяти FeCoV в поляризационном микроскопе. Темные области намагничены в одну сторону, а серые — в другую. Намагниченности показаны стрелками

Поэтому управлять намагниченностью слоев, сделать ее достаточно быстрой и надежной все еще сложно, а готовые приборы стоят все еще дорого. Однако это практически не является препятствием для серийного выпуска считывающих головок, элементов памяти, электронного компаса, который чувствует магнитное поле Земли и уже заменяет гироскоп в самолетах и ракетах. Таким образом, практические применения вышеописанных технологий выходят далеко за пределы вычислительной техники и развиваются с высокой скоростью.

Контрольные вопросы

1. Что такое *MRAM*?
2. Каковы недостатки магниторезистивной ячейки памяти, переключаемой магнитным полем?
3. Чем отличается *MRAM* от *SRAM*?

4. Каким образом время переключения логического состояния наночастицы связано с энергозатратами на этот процесс?
5. Подчиняется ли спиновый переключатель принципу Ландауэра?
6. Придумайте и опишите логический элемент, состоящий из двух наночастиц. Какую операцию можно реализовать таким элементом?
7. Что такое ферромагнетик?
8. Каково происхождение слов «спин-поляризованный ток»?
9. Может ли спиновый ток быть равен нулю, а ток заряда отличен от нуля? Может ли быть наоборот?
10. В каких еще областях, кроме вычислительной техники, применяется эффект гигантского магнитосопротивления?

Раздел 5. Принципы квантовых вычислений

5.1. Физические квантовые вычислительные устройства

Предел плотности логической памяти для наноустройств электроники в настоящее время уже почти достигнут. Электронные логические вентили состоят из частиц размерами ~ 40 нм (сравнимой с длиной волны Де Бройля), и дальнейшее понижение их размера не обещает сильного уплотнения информации, давая при этом огромные технические осложнения. Стремление записать один бит в один атом продвинет классические компьютеры не более чем на 1—2 порядка величины на пути уплотнения информации и скорости ее обработки. Старый подход, в чем-то исчерпав себя, одновременно переходит в новый физический принцип, связанный с уменьшением размеров приборов — создание квантовых битов (кубитов). Для их реализации приходится изменить сами принципы хранения, записи и обработки информации, математику, управляющую этим процессом. На смену арифметическим операциям Булевой алгебры приходит математический аппарат квантовой механики, богатый чудесами интерференции волновых функций микрочастиц, иногда не позволяющими признать, что $2 \times 2 = 4$. Например, две частицы со спинами представляют собой простейший ансамбль с волновыми спиновыми функциями одного синглетного — S и трех триплетных (T) состояний (т.е. способны хранить четыре бита). Будучи изолированными друг от друга, эти частицы способны хранить лишь по одному биту информации. Таким образом, увеличение числа единиц информации с числом частиц растет как $\sim 2^N$, что значительно быстрее, чем в классической памяти, где число сохраненных бит пропорционально числу ячеек $\sim N$.

Различие между логическими кодами современных компьютеров и способом квантовой записи информации напоминает разницу между алфавитом и иероглифами. В случае квантового подхода волновые функции многих частиц образуют единую волновую функцию ансамбля. При этом «запоминание» происходит не по очереди, в виде отдельных разрядов — частиц, изменяющих свое состояние, а одновременно во всем ансамбле. Сохраняемая информация затрагивает волновые функции всех частиц одновременно и представляет собой квантовый штрих-код. Ситуация сходна с рисунком (иероглифом),

обозначающим какое-либо японское слово, например «магнит» (рис. 5.1.1). Элементы этого рисунка не читаются по-отдельности, а информация, переносимая иероглифом, считывается глазом одновременно и включает в себя значительно больше, чем то, что можно передать с помощью одной буквы. Иероглиф — исторически сложившийся штрих-код, комбинация первоначальных рисунков-представлений о мире и, своего рода, «волновая функция» человеческого сознания, компилирующая простые образы.



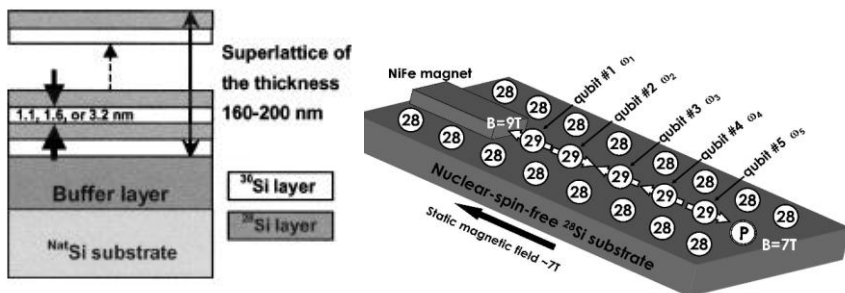
Рис. 5.1.1. Японский иероглиф (Jishyaku — магнит)

Иероглиф объединяет два иероглифа: «магнитный» (первый иероглиф) и «камень» (второй иероглиф). Первый иероглиф участвует во всех словах, сообщающих о магнетизме (магнитная восприимчивость, стрелка компаса и др.).

Создание квантовой памяти требует конкретной материальной реализации. Можно ли использовать ядерные спины, управляющие химической реакцией, для сохранения и обработки квантовой информации? Как осуществить эту идею на практике? Главным претендентом на пути создания ядерных спин-селективных кубитов оказался кремний, легко совместимый с уже существующими кремниевыми устройствами стандартной Булевой логики. (Эта интригующая совместимость аналогична японскому языку, который комбинирует алфавит и иероглифы.) Кремний состоит из трех изотопов ^{28}Si , ^{29}Si и ^{30}Si с естественной распространенностью: 92,2% ^{28}Si , 4,7% ^{29}Si и 3,1% ^{30}Si .

Вопреки широко распространенному мнению о том, что изменение физических и химических свойств кремния, вызванное обогащением изотопами, незначительно, имеется ряд экспериментальных доказательств, опровергающих эту точку зрения.

По иронии судьбы именно японские ученые (в соавторстве с американскими) первыми создали полностью кремниевые кубиты. Сначала было показано, что различие между ядерными спинами I изотопов кремния ($I=1/2$ для ^{29}Si , но $I=0$ для ^{28}Si и ^{30}Si) сильно изменяет физико-химические свойства наноструктур. Впервые в мире были созданы образцы, в которых тонкие слои (1—3 нм), обогащенные различными изотопами (^{28}Si , ^{29}Si и ^{30}Si), чередовались в очень аккуратной сверхрешетке, выполненной с нанометровой точностью (рис. 5.1.2).



а)

б)

Рис. 5.1.2. а) гетероструктура, состоящая из чередующихся слоев кремния магнитного и немагнитного изотопов, б) схема ядерного спинового процессора на изотопах кремния.

Слева указаны номера кубитов и их рабочие частоты ω_i

Такие структуры позволили обнаружить чудесные особенности химических реакций и их зависимость от наличия магнитного изотопа, т.е. магнитный изотопный эффект, а также увеличить точность дизайна логического симулятора CMOS-процессора (*complementary metal-oxide-semiconductor*), нужного для создания кремниевой микросхемы памяти. Это повлекло за собой развитие изотопной спиновой инженерии кремния и новых инструментов для спинтроники.

Идея применения одиночного атома кремния с магнитным ядром в качестве бита и процессора была предложена японскими учеными. Реализация этой концепции показана на рис. 5.1.26. В этом устройстве ориентация и фаза каждого спина ядра ^{29}Si в цепочке эквивалентна биту информации. Ключом к считыванию является атом фосфора с электронным спином, помещенный на конце цепочки и «считывающий» информацию, записанную в цепочке ядерных спинов кремния. Реализация такого квантово-механического устройства оказалась возможна благодаря выполнению следующих условий:

1) изотопно-очищенный кремний без парамагнитных примесей обладает гигантским временем удержания квантовой информации — при комнатной температуре спины ядер ^{29}Si остаются ориентированными 25 секунд. Этого времени вполне достаточно, чтобы произвести миллиарды логических операций;

2) спиновая когерентность электронного спина фосфора допускает передачу информации о состоянии кубитов через сверхтонкое взаимодействие с ядрами ^{29}Si . Таким образом, электронный спин фосфора на конце цепочки дирижирует работой всех кубитов в ней;

3) цепочки ядер кремния ^{29}Si удалось создать и доказать их согласованную работу;

4) инициализация прибора (ориентирование всех спинов в одном направлении) осуществлялась оптической накачкой, а также динамической ядерной поляризацией с помощью электронного спина фосфора. Манипуляции спинами ядер кремния была произведена с помощью ядерного магнитного резонанса;

5) в качестве метода считывания с ансамбля ядерных спинов ^{31}P были использованы два метода: оптический и электрический. В качестве электрического метода считывания (разработанного при участии российских ученых из Физико-технического института РАН им. А. Ф. Иоффе в Санкт-Петербурге) был использован факт передачи информации от цепочки кремниевых ядер к ядру фосфора. А для считывания информации с ядра фосфора использовали электродипольный магнитный резонанс в небольшом магнитном поле (~ 200 Э), который создавал запутанные состояния Бэлла между ядерной подсистемой и электронным спином, т.е. под действием микроволнового электрического поля возникали квантовые иероглифы — штрих-коды, нужные

для обработки информации и установления одновременно и электронных, и ядерных спиновых состояний.

Таким образом, создание реального устройства квантовой памяти требует выполнения целого ряда вышеперечисленных высокотехнологичных условий, реализуемых только в рамках нанотехнологий и оборудования, пригодного для манипулирования одиночными атомами. Одним из главных условий записи-считывания информации (общения с кубитной памятью) является надежное и сильное влияние ядерных спинов на состояния электронных спинов в считывающем устройстве. Этот элемент (называемый «ключом») предлагалось выполнить в виде концевго атома фосфора, обладающего электронным и ядерным спинами, связанными эффективным сверхтонким взаимодействием (рис. 5.1.2б).

Отметим, что и здесь продолжается аналогия с японским языком. «Ключом» в первом иероглифе на рис. 5.1.1 называют изображение второго иероглифа («камень»), включенного также и в первый иероглиф («магнит»). Именно по нему находят это слово в словаре, относят к определенной группе иероглифов. «Ключ» одинаково открывает двери в семантический анализ информации в японском языке и при считывании квантового «слова». В заключение заметим, что аналогия между иероглифами (продуктом человеческого сознания) и квантовой памятью (объектом, способным существовать вне сознания) нами выбрана не только потому, что кислород способен проявить рисунок на поверхности кристалла, задаваемый состоянием кубитов. Фантазия подсказывает, что принципы работы компьютера постепенно приближаются к неизвестным до сих пор принципам работы человеческого мозга.

5.2. Квантовые логические операции

Современные компьютеры выполняют вычисления и обрабатывают информацию, используя классическую модель вычислений, которая восходит к Тьюрингу и фон Нейману. В этой модели вся информация сводится к биту, который может принимать значения 0 или 1 — и вся обработка может быть выполнена с помощью простых логических вентилей (*AND*, *OR*, *NOT*, *NAND*), действующих на один или два бита. В любой момент работы классического компьютера его со-

стояние полностью определяется состояниями всех его битов, так что компьютер с n битами может существовать в одном из возможных состояний, от 000...00 до 111...11.

Мощность квантового компьютера основана на значительно большем богатстве состояний. Квантовый компьютер также имеет биты, такие же, как любой компьютер. Но вместо 0 и 1, его квантовые биты (кубиты) могут представлять собой 0, 1 или оба значения сразу в определенной пропорции (или в суперпозиции). Это само по себе не очень помогает делу, так как компьютер, чьи биты могут принимать промежуточные значения между 0 и 1, — это обычный аналоговый компьютер, едва ли более мощный, чем современная цифровая вычислительная машина. Преимущество квантового компьютера заключается в том, что он использует суперпозиции, что позволяет обрабатывать экспоненциально много логических состояний одновременно. Это мощный шаг вперед, и ни один классический компьютер не может достичь этого. Подавляющее большинство квантовых состояний такого компьютера «запутаны» — они являются состояниями всего компьютера, и не соответствуют ни одному заданию цифровых или аналоговых состояний отдельных кубитов. Квантовый компьютер является значительно более мощным, чем любой классический компьютер — будь то детерминированный, вероятностный или аналоговый. В нескольких известных проблемах (таких, как разложение больших чисел на множители (факторизация)) квантовый компьютер, очевидно, является победителем над классическим компьютером. Квантовый компьютер способен за день решать такие задачи, для которых классическому компьютеру потребуются миллионы лет. Большая часть таких задач лежит в социогуманитарной, финансовой, банковской сферах, а также в задачах управления производством.

Приведем примеры таких задач.

1. В курсе «Методы оптимальных решений» изучаются задачи линейного программирования, которые решаются для ограниченного количества переменных. Это и *транспортные задачи*, и задача коммивояжера, и задачи рационального расходования ресурсов. Достаточно представить себе многотысячную сеть супермаркетов, в каждом из которых имеется многотысячный набор товаров, которые нужны рационально распределить по магазинам в зависимости от спроса, времени суток, ограничений на хранение и т.п. Решение такой

задачи даже упрощенными методами линейного программирования в настоящее время невозможно, не говоря уже о многих нелинейных экономических задачах.

2. Необходимо передать секретную информацию (банковскую, военную) по защищенной линии связи. Какие бы классические принципы передачи информации мы не использовали, эти же принципы позволяют незаметно подключиться к линии связи и раскрыть секрет. Лишь квантовые состояния частиц обладают фундаментальным свойством их обязательного изменения в процессе считывания. Уберечь информацию от считывания мы не можем даже в квантовых линиях связи, но именно такие линии обладают свойством ставить нас в известность, если информация считана кем-либо другим. Это происходит потому, что квант (света, радиоволна и т.д.) передается в смешанном состоянии. Его взаимодействие с любым считывающим прибором немедленно переводит его в чистое состояние, о чем сразу же узнают владельцы секретной линии связи. Наука о создании защищенных квантовых линий связи, а также о кодировании информации в квантовой форме (запутанные состояния, о которых вы подробно узнаете из лабораторных работ) называется **квантовой криптографией**. Главными персонажами в научных статьях на эту тему являются Алиса и Боб, все время желающие обмениваться секретами, а также злодей Ив, который все время крадет информацию на линиях связи (рис. 5.2.1).

3. **Задачи небесной механики**, решение которых важно для расчета траекторий ракет и предсказания сближений Земли с метеоритами. Прогнозирование столкновений должно учитывать очень много планет и малых небесных тел, чтобы точно рассчитать траектории.

4. **Задачи биоинформатики**, расшифровка кодов ДНК, которые представляют собой миллиарды комбинаций боковых групп, аминокислот и молекулярных мостиков, формирующих генетический код. В настоящее время эта задача считается плохо разрешимой именно в силу низкой скорости расшифровки классическими суперкомпьютерами.

5. **Квантовые деньги**. Возможность создания защищенных линий связи и передачи запутанных состояний позволяет ввести криптовалюту. Современные биткоины являются классическими деньгами, зашифрованными классическими алгоритмами, а потому могут

быть расшифрованы или «взломаны» за сравнительно короткое время. Настоящий расцвет криптовалют связывают с применением квантовых компьютеров.

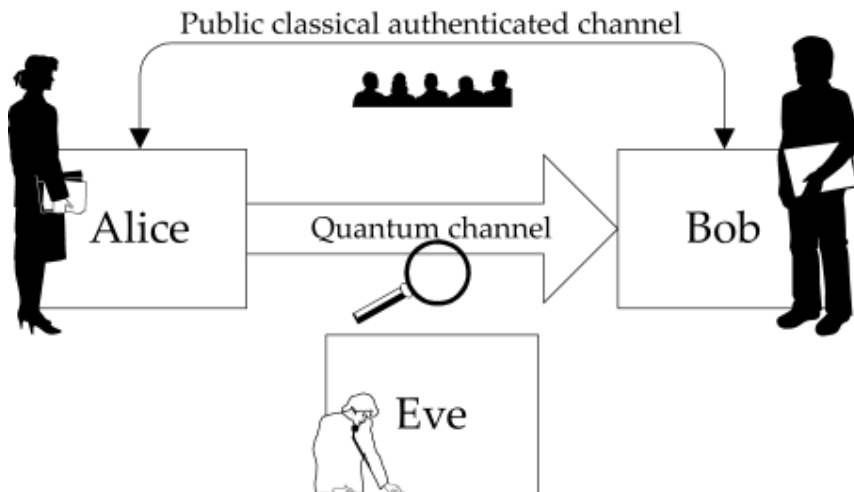


Рис. 5.2.1. Алиса и Боб стараются сообщить друг другу конфиденциальную информацию, сделав так, чтобы ее не узнал злодей Ив

6. Поиск данных в несортированных массивах (алгоритм Гровера). В лабораторных работах будет предложена реализация этого алгоритма. Он заключается в использовании квантовых перепутанных состояний для быстрого определения расположения объекта в массиве себе подобных. Особенно актуальной кажется такая задача в гуманитарных науках для анализа текстов, поиска выражений, семантического анализа, получения статистических закономерностей текстов. Все эти операции крайне затруднены неупорядоченностью массивов данных, требующих найти в них желаемый элемент. Квантовый алгоритм Гровера решает эту проблему потому, что не перебирает поочередно все элементы, а мгновенно определяет наличие его в перепутанной волновой функции и затем лишь остается правильно провести измерение, чтобы узнать, где такой элемент находится.

7. Разложение целых чисел на множители, используемое в криптографии (Алгоритм Шора). Банковские и военные секреты ко-

дируются путем записи множителей, на которые можно разложить целое число. Некоторые числа мы можем разложить устно. Например, не трудно определить, что число 8 может быть представлено, как $1 \times 2 \times 4$. Но попробуйте так же быстро это определить для числа 98. Классический компьютер делает это сравнительно быстро для двузначных чисел. Однако уже 3—5-значные числа требуют существенного времени работы классического компьютера. Эта проблема полностью решается алгоритмом Шора.

8. Квантовая телепортация — наиболее интригующее свойство квантовых приборов, которое заключается в том, что объект можно передать мгновенно (т.е. без трудностей, связанных с преодолением ограничений на скорость света) на любое расстояние. Это становится возможным (и уже реализовано физиками в мире микрочастиц) благодаря тому, что квантовые состояния одновременно присутствуют в контролируемых точках и не требуют для перемещения затрат энергии.

Можно было бы подумать, что трудности в понимании квантовых вычислений или квантовая физика лежит в «жесткой математике», но на самом деле математически квантовые понятия являются лишь немного более сложными, чем курс алгебры средней школы. Квантовая физика носит ореол «трудной», потому что, как и теория относительности Эйнштейна, она требует усвоения небольшого количества идей, которые являются простыми, но иногда противоречит повседневному классическому здравому смыслу. В теории относительности такая странная идея состоит в том, что пространство и время взаимосвязаны, в то время как здравый смысл подсказывает нам, что они должны действовать независимо друг от друга.

В квантовой физике нужно принять «на веру» следующие интуитивно-неприемлемые идеи:

1) физическая система в совершенно определенном состоянии может вести себя случайным образом; 2) две системы, которые слишком далеки друг от друга, чтобы влиять друг на друга, могут, тем не менее, вести себя таким образом, они так или иначе сильно коррелируют. Этих знаний, которые прямо следуют из многочисленных физических экспериментов, достаточно, чтобы освоить принципы работы квантового компьютера.

Кубит. Здесь мы рассмотрим понятие «кубит». Мы начнем использовать немного математических обозначений, в том числе некоторые понятия линейной алгебры.

Кубит представляет собой квантовую систему, состоящую из двух уровней, обозначенных $|0\rangle$ и $|1\rangle$ (здесь мы используем так называемые обозначения Дирака). Эти уровни представлены двумерным векторным пространством C^2 над комплексными числами. Это означает, что необходимы всего два комплексных числа, чтобы полностью описать кубит. Так называемые базисные состояния, на которых можно построить любые другие смешанные состояния, очень важны для нас. Они называются еще чистыми состояниями, в отличие от смешанных, которые на них могут быть построены. Чистые состояния — это именно то, что обнаруживает прибор при попытке определить состояние системы, хотя исходно она может пребывать в смешанном состоянии. Базис соответствует двум уровням и следующим векторам: $|0\rangle$ (0; 1) и $|1\rangle$ (1; 0).

Суперпозиция этих чистых состояний запишется в виде:

$$\psi = \alpha |0\rangle + \beta |1\rangle.$$

Причем, хотя α и β , которые называют амплитудами соответствующих состояний, могут быть и комплексными, их квадраты являются действительными числами и имеют простой смысл:

$|\alpha|^2$ — вероятность пребывания системы в состоянии $|0\rangle$;

$|\beta|^2$ — вероятность пребывания системы в состоянии $|1\rangle$.

Поскольку никаких других состояний нет, условие нормировки требует выполнения равенства:

$$|\alpha|^2 + |\beta|^2 = 1.$$

Интересные вещи происходят, когда квантовые системы измеряются. Квантовое измерение описывается правилом Борна. В частности, если кубит, который находится в некотором состоянии, измеряется в стандартном базисе, результат $|0\rangle$ получается с вероятностью $|\alpha|^2$, а результат $|1\rangle$ получается с дополнительной вероятностью $|\beta|^2$.

То, что мы описали здесь, является абстрактным описанием любого кубита. Прототип квантового компьютера, который вы используете в *IBM Quantum Experience*, основан на сверхпроводящем кубите *transmon*, который сделан из сверхпроводящих материалов, таких как

ниобий и алюминий, расположенных на узорчатой кремниевой подложке. Чтобы он работал, а вы имели возможность использовать его для расчетов, кубит охлаждают до очень низких температур. В *IBM Quantum Lab* поддерживают 15 миллиКельвин в холодильнике, где нет окружающего теплового шума. Чтобы привести систему в это состояние, требуется несколько дней. Для того чтобы получить представление о том, как работает симулятор, попробуйте запустить первый файл в режиме моделирования (<https://quantumexperience.ng.bluemix.net/qstage/#/editor?codeId=14ffabde6401dc85c78c655e3d5cf301>) (рис. 5.2.2). Здесь кубит изначально приготовлен в основном состоянии $|0\rangle$, затем над ним проводят стандартное измерение. Из результатов выполнения видно, что в идеальном случае с очень высокой вероятностью кубит все еще находится в состоянии (рис. 5.2.3). Если же вы воспользуетесь реальным процессором, то сможете заметить, что есть некоторые ошибки, которые получаются в результате теплового шума даже при сверхнизких температурах. Чем выше температура, тем с большей вероятностью вы будете находить кубит не только в состоянии $|0\rangle$, но и в состоянии $|1\rangle$.

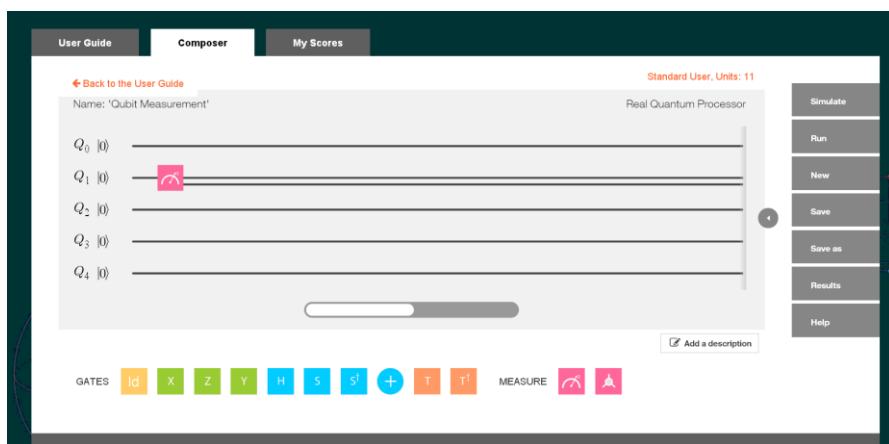


Рис. 5.2.2. Схема для измерения исходного состояния кубита Q_1



Рис. 5.2.3. Результаты квантовых вычислений:
по горизонтальной оси отложены возможные значения
вычисляемой величины, по вертикальной оси отложены
вероятности их реализации

При запуске каждой операции внизу странички вы каждый раз будете видеть физические характеристики реального прибора, который выполнял ваше квантовое вычисление (рис. 5.2.4).

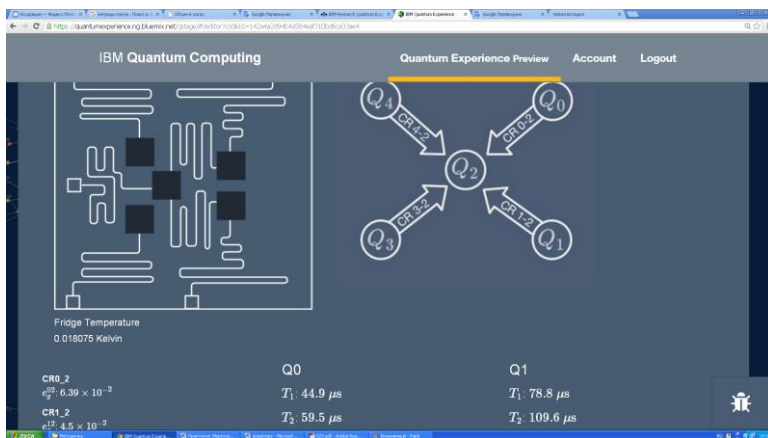


Рис. 5.2.4. Физические характеристики кубитов и квантового процессора в процессе конкретного вычисления
(изменяются во времени и при смене процесса)

Здесь показана схема расположения кубитов (черными квадратами слева) в схеме процессора, показана взаимосвязь между кубитами (стрелками между состояниями на схеме справа), а также температура процессора (на рис. 0.018075 Кельвин) и времена сохранения когерентности амплитуды T_1 и фазы T_2 для кубитов Q_0 и Q_2 , выраженные в микросекундах. Как вы догадались, времена сохранения когерентности — ключевые параметры устройства. Чем они больше, тем больше операций можно совершить за это время, тем меньше шум и тем ближе результаты вычислений к идеальному процессору. На рис. 5.2.5 представлена зависимость времен когерентности квантовых состояний кубитов в фирме *IBM* в разные годы. Из этой зависимости видно, что все время продолжается улучшение системы.

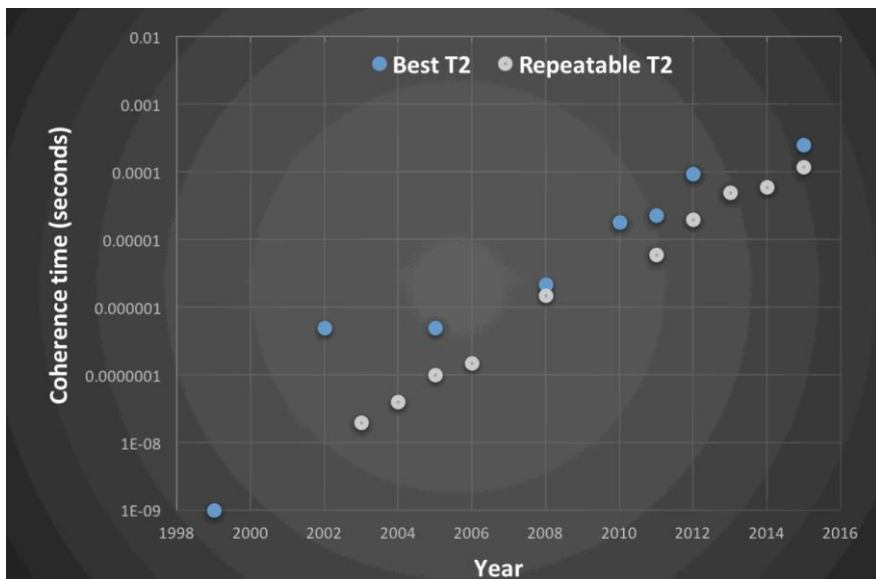


Рис. 5.2.5. Улучшение времени когерентности системы с годами

Опишем точнее времена T_1 и T_2 . Время T_1 соответствует длительности процесса самопроизвольной релаксации вектора возбужденного состояния $|1\rangle$ в исходное состояние $|0\rangle$. Этот процесс сопровождается выделением энергии, и он влияет на чистые состояния.

Дефазировка — другой процесс декогеренции, и в отличие от релаксации энергии, она влияет только на состояния суперпозиции и характеризуется временем T_2 . Хотя это может быть понято только в квантовых терминах, попытаемся привести и классический аналог. Если относить состояния $|1\rangle$ и $|0\rangle$ к ориентации магнитного момента электрона, то можно представить два процесса, которые развиваются со временем: 1) происходит переориентация спина из состояния против поля в состояние по полю; 2) происходит сбой фазы вращения спина вокруг магнитного поля. Именно этим процессам (которые в физике называют спин-решеточной и спин-спиновой релаксациями, соответственно) отвечают времена T_1 и T_2 , если компьютер спиновый.

Постоянная времени включает в себя эффект дефазировки, а также релаксации энергии и является еще одним важным достоинством термоэлектрического охладителя.

Контрольные вопросы

1. Что такое квантовый компьютер?
2. Почему спин частиц является квантовым, а не классическим объектом?
3. Каковы основные положения квантовой механики, отличающие ее от классической механики?
4. Каким образом время переключения логического состояния квантовой частицы связано с энергозатратами на этот процесс?
5. Подчиняется ли квантовый процессор принципу Ландауэра?
6. Чем отличается работа квантового симулятора от работы реального квантового компьютера?
7. Что такое квантовый симулятор?
8. Что такое «кот Шредингера»?
9. Дайте определение декогеренции.
10. Почему квантовый компьтинг близок именно задачам гуманитарной сферы?

Раздел 6. Лабораторные работы по квантовым вычислениям

К1. Визуализация квантовых вычислений

Цель работы: освоение приемов визуализации квантовых вычислений.

Порядок выполнения работы

1. Изучить теоретическую часть.
2. Выполнить задания.
3. Представить файл для проверки преподавателю.

Предлагается почувствовать взаимосвязь между квантовыми состояниями и реализацией образов в системе из 22 кубит в онлайн симуляторе <http://www.quantumplayground.net/#/home>. Этот симулятор позволяет реализовывать также и квантовые алгоритмы Шора и Гровера на множестве квантовых операций, конструируемых пользователем. Разумеется, выполнение этих операций производит классический компьютер, однако вычисления, которые он производит, находятся в строгом соответствии с квантовыми алгоритмами.

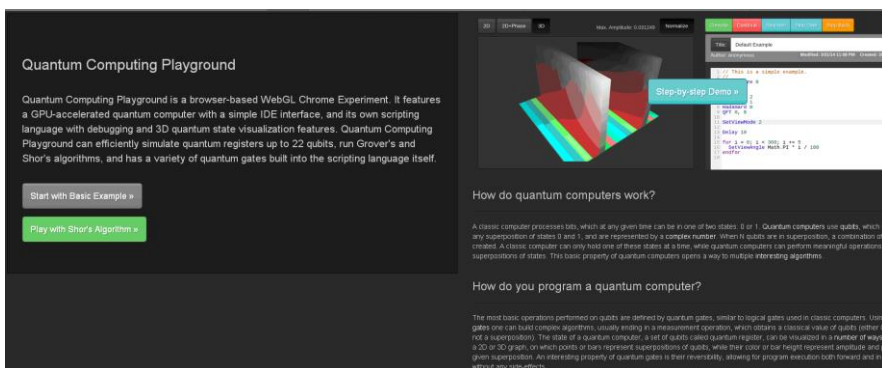


Рис. 6.1.1. Главная страница симулятора

На рис. 6.1.1 показана главная страница симулятора. В ней есть краткий учебник с основами квантовых вычислений, о которых было сказано выше. Описано, как программировать квантовый компьютер. Начнем с кнопки *Start with Basic Example*. После нажатия на нее откроется страничка, где справа будет записана программка в формате C++, а слева будет темный экран, готовый к выполнению действий (рис. 6.1.2).

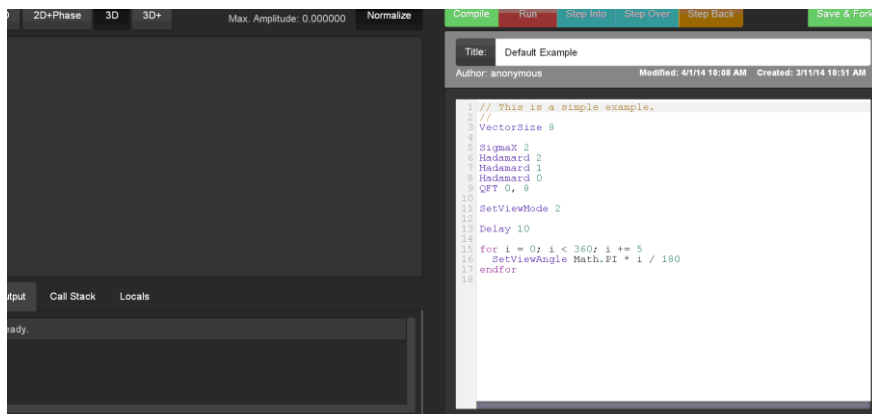


Рис. 6.1.2. Панель для выполнения преобразований Адамара

Попытаемся визуализировать преобразование Адамара. Для этого нажмем сначала *Compile* (на правой панели вверху в ряду кнопок слева). Программа начнет работать и показывать ступенчатое вращение балки (рис. 6.1.3). Это и есть многократные преобразования Адамара. В программе можно менять параметры процесса, если вписывать в ячейки разные значения (рис. 6.1.4).

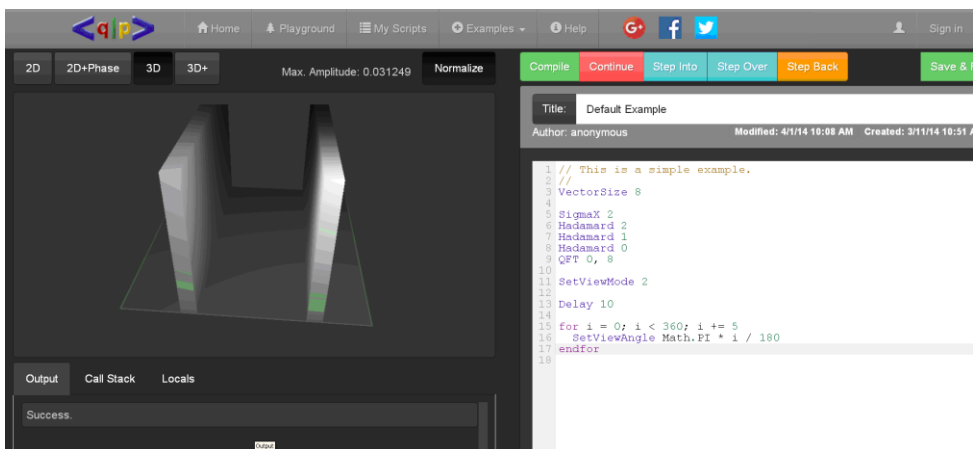


Рис. 6.1.3. Вращение объекта с помощью квантового преобразования Адамара

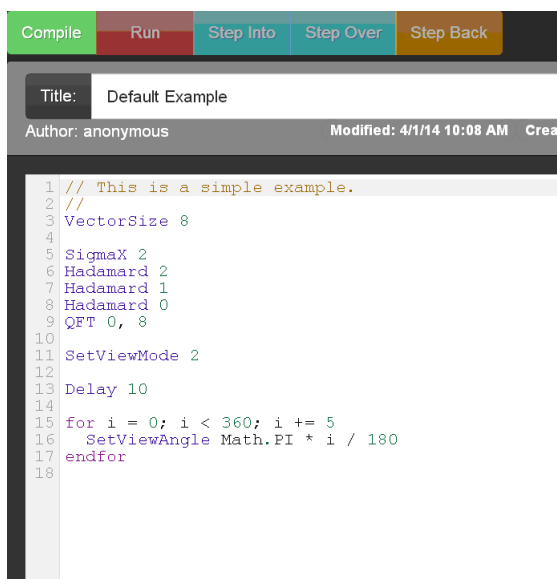


Рис. 6.1.4. Программа визуализации преобразования Адамара

Посмотрим теперь, что представляет собой алгоритм Гровера. Мы ищем некоторый объект в неструктурированной базе данных. Для этого нажмем клавишу *Play with Shor's algorithm*. Снова появится изображение меню и пустой экран, где будут отображаться операции (рис. 6.1.5). Как обычно, нажмем *Complile* (актуализация введенных параметров) и *Run* (пуск). Станет видно, как программа перебирает столбики, останавливаясь на выборе одного из них (рис. 6.1.5).

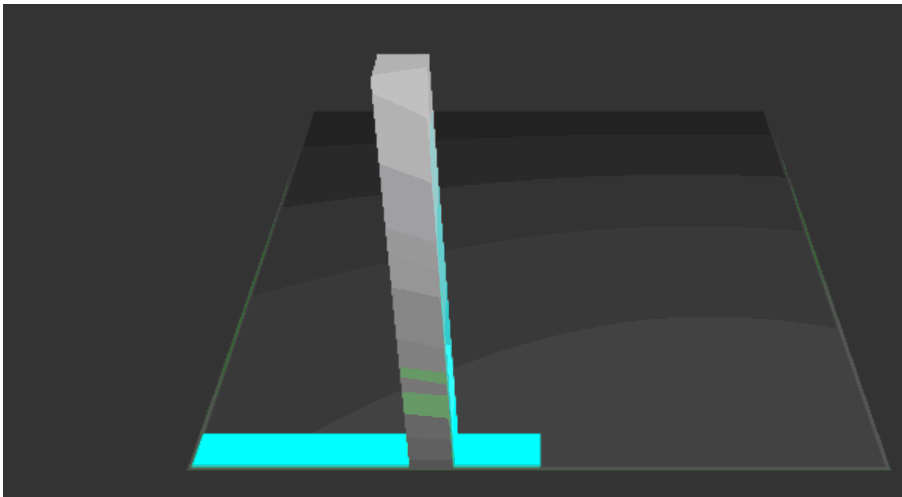


Рис. 6.1.5. Объект «столбик», найденный в результате реализации алгоритма Гровера

Таким образом, мы можем, слегка модифицируя логическую схему, стартовать из разных квантовых состояний и следить за тем, что получается.

Задания

Порядок выполнения работы:

1. Визуализация преобразований Адамара (*Hadamard gate*). Нажимаем «*Start with basic example*». Появляется панель с программой справа и пустым окном слева (рис. 6.1.6).

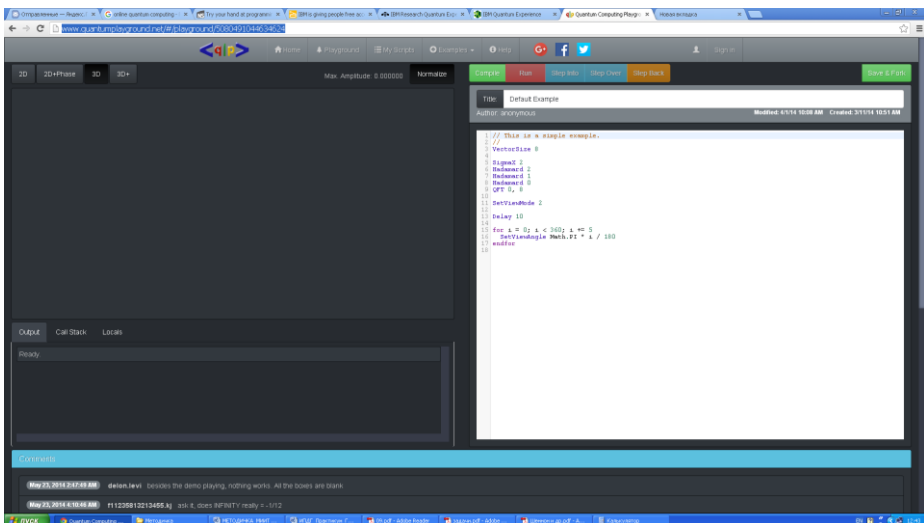


Рис. 6.1.6. Панель для визуализации преобразования Адамара

2. Далее нажимаем кнопку *Compile* и *Run* и запускаем действие программы, осуществляющей вращение Адамара для объекта (рис. 6.1.7).

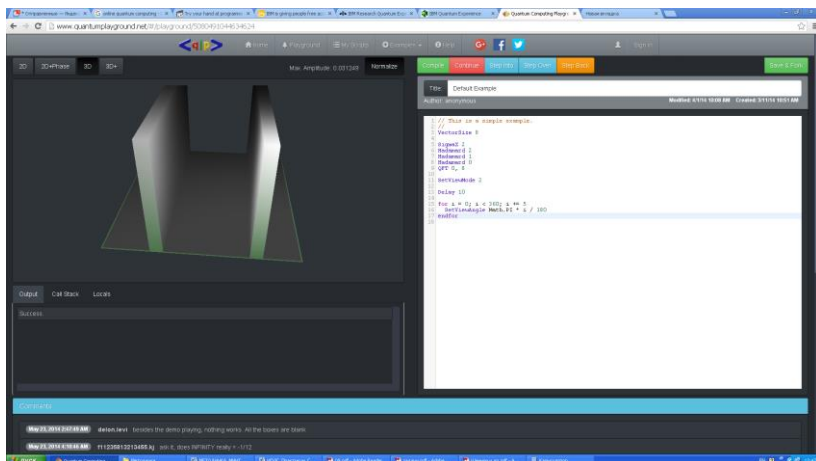


Рис. 6.1.7. Визуализация преобразования Адамара, примененного к трехмерному объекту

3. Полюбовавшись на вращение балки, приступаем к работе.

В правом окошке по очереди заменяем количественные параметры и определяем, какой из них, за что отвечает. Преподавателю должен быть представлен список команд и описан результат, к которому приводит изменение количественных значений параметров.

Контрольные вопросы

1. Перечислите практические применения квантового компьютера.

2. Что такое алгоритм Гровера? Для чего он используется.

3. Что такое алгоритм Шора? Какие задачи он позволяет решать?

4. Каковы цели персонажей, используемых для иллюстрации квантовой криптографии?

5. В чем причина принципиальной невозможности украсть квантовую информацию незаметно для Ива?

6. Можно ли украсть информацию из квантовой линии связи?

7. Квантовые состояния каких частиц использует квантовый центр Сколково для создания квантовой связи в банке «Газпром-банк»? См. <http://www.rqc.ru/>

8. Во сколько раз быстрее квантовый компьютер справится с поиском объекта среди подобных по сравнению с классическим, если общее число объектов 30?

9. Почему квантовые алгоритмы в большей степени востребованы гуманитарными науками, чем техническими?

10. Приходится ли перебирать разные квантовые состояния при реализации алгоритма Гровера?

К2. Квантовый алгоритм Гровера

Цель работы: изучение квантового алгоритма Гровера.

Порядок выполнения работы

1. Изучить теоретическую часть.

2. Выполнить задания.

3. Представить файл для проверки преподавателю.

Работа посвящена изучению алгоритма Гровера и амплитуды усиления. Как уже было сказано, одно из многих преимуществ квантового компьютера по сравнению с классическим компьютером заключается в превосходных возможностях поиска в базах данных. Это как раз та задача, которая трудна для реализации «закона Яровой» о сборе информации по электронным письмам и телефонным сообщениям. Нет никакой проблемы в том, чтобы хранить петабайты информации об электронных коммуникациях, но есть проблема использования этой гигантской базы данных и поиска в ней нужного письма. Если идти классическим путем, то количество времени, необходимое для поиска нужной информации, увеличивается пропорционально числу объектов N . Использование квантового алгоритма Гровера приводит к росту времени лишь как функция $N^{1/2}$. Алгоритм Гровера (и не только) демонстрирует эту возможность. Этот алгоритм может ускорить неструктурированный поиск квадратично по отношению к числу объектов (например, для базы данных, содержащей 10 объектов, квантовый компьютер будет справляться в 100 раз быстрее, чем классический). Но прежде чем мы начнем симуляции, давайте посмотрим на задачи неструктурированного поиска. Перейдем по ссылке: <https://quantumexperience.ng.bluemix.net/qstage/#/tutorial?sectionId=8443c4f713521c10b1a56a533958286b&pageIndex=2>. Перед вами панель для реализации алгоритма Гровера (рис. 6.2.1).

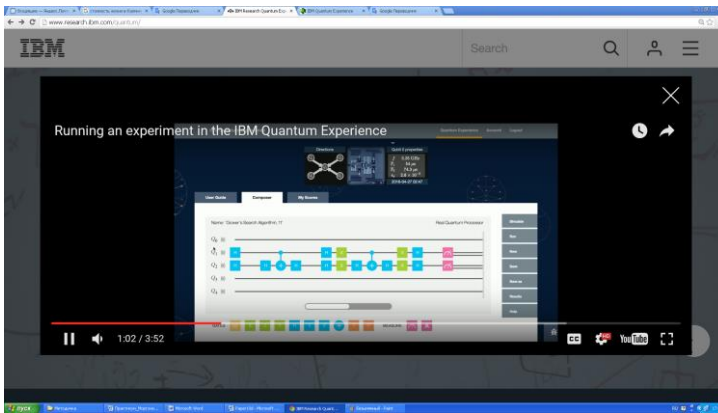


Рис. 6.2.1. Панель для реализации алгоритма Гровера

Задачу кратко можно сформулировать следующим образом: имеется база данных, состоящая из 32 элементов (которые кодируются двумя кубитами), один из которых помечен. Число элементов можно менять в программе, как и число кубитов. Можно также изменить помеченный элемент. Задачей алгоритма является определение номера помеченного элемента. В вашем распоряжении имеется возможность указать, сколько раз необходимо применить операторы контролируемой фазы и диффузии (второй шаг алгоритма Гровера), а также увидеть теоретические результаты измерений (распределение амплитуд вероятностей по элементам) и результаты того, что может получиться на практике (гистограмма результатов измерений ансамбля одинаковых состояний базы). Начнем с реализации алгоритма в том виде, как он предложен на сайте (рис. 6.2.2).

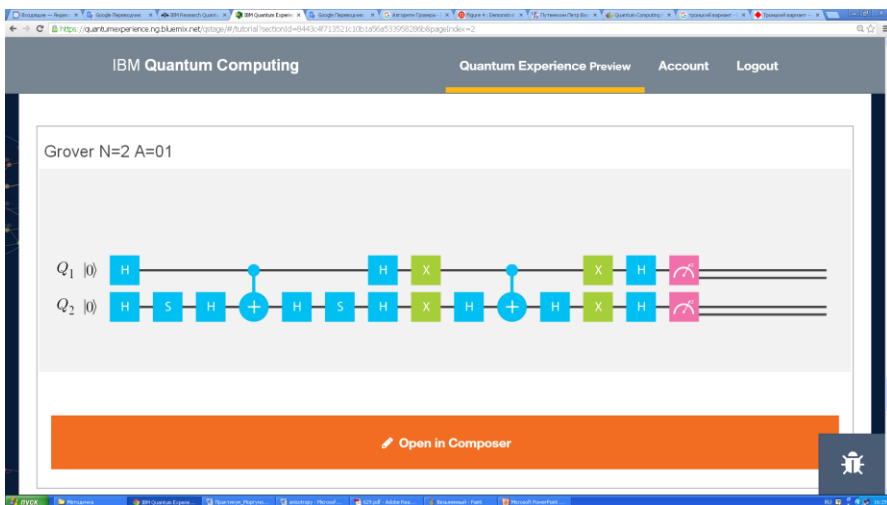


Рис. 6.2.2. Логическая схема реализации квантового алгоритма Гровера из исходного состояния 00 в двухкубитной схеме

Нажимая *Run* или *Simulate* в зависимости от того, хотим ли мы пользоваться реальным компьютером или симулятором, получим ответ в виде распределения вероятностей состояний (рис. 6.2.3).

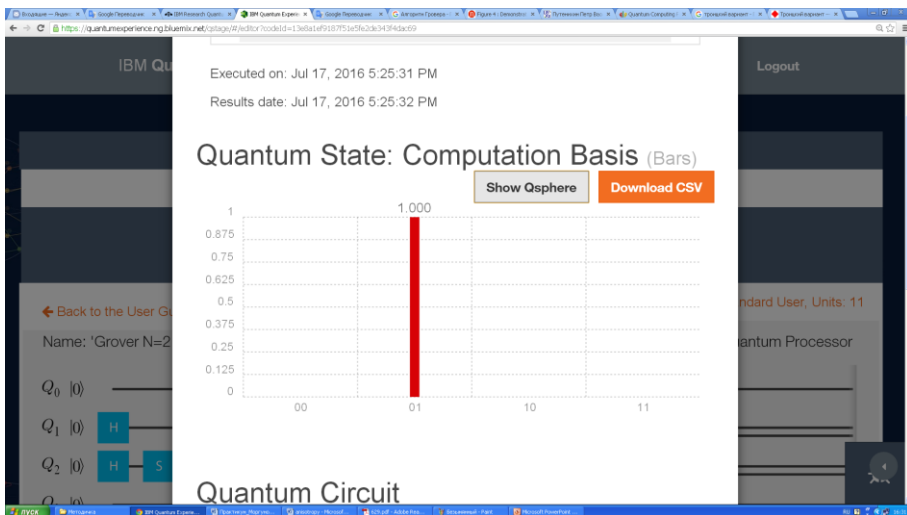


Рис. 6.2.3. Результаты работы двухкубитного алгоритма Гровера

Этот же результат можно изобразить точкой на сфере Блоха (рис. 6.2.4).

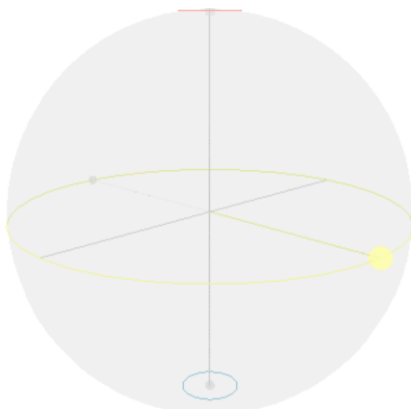


Рис. 6.2.4. Результаты реализации двухкубитного алгоритма Гровера на сфере Блоха.

Желтой крупной точкой отмечено найденное алгоритмом помеченное состояние

Совершенно аналогично можно составить программу, которая будет стартовать из состояния 01. Как уже было отмечено, мы не можем изменить в симуляторе исходные состояния кубитов, но мы можем их менять оператором X . Поэтому, желая стартовать из иного исходного состояния, мы просто добавляем этот оператор на «нотную доску» сразу после кубита. Это будет вполне эквивалентным старту из состояния 01 вместо 00. Полученный результат и измененная схема показаны на рис. 6.2.4 и 6.2.5. Иногда полученные результаты в двухкубитном процессоре можно представить в наглядном виде столбчатой диаграммы (рис. 6.2.5).

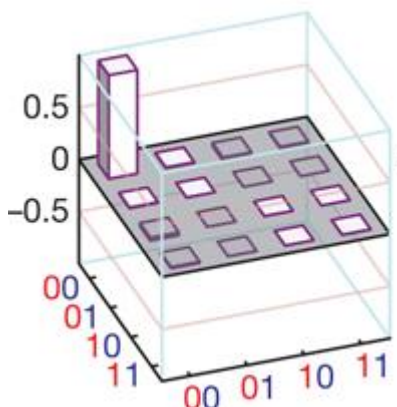


Рис. 6.2.5. Заполнение квантовых состояний в алгоритме Гровера для двухкубитного случая

Задания

1. В алгоритме Гровера получить результаты симуляции для всех четырех исходных состояний, представить результаты в виде распределения вероятностей и Блоховских сфер. Объяснить сходства и различия.
2. В алгоритме Гровера поменять количество объектов и запустить программу, убедившись, что она правильно находит нужный объект.
3. Построить двумерную столбчатую диаграмму квантовых состояний в *MS Excel* для разных стартовых состояний.

Контрольные вопросы

1. Каково назначение элементов в алгоритме Гровера?
2. Каким образом в рамках симулятора *Composer* можно изменять исходные квантовые состояния кубитов?
3. Будет ли влиять на результат направление, в котором присоединяется элемент «+».
4. Чем обусловлена разница между вычислениями реального и идеального квантовых компьютеров?
5. Чем отличаются алгоритмы Гровера и Шора?
6. Можно ли представить результаты расчета пятикубитного процессора в виде столбчатой диаграммы?
7. Как в алгоритме Гровера избегают последовательного просмотра всех объектов?
8. Может ли алгоритм Гровера работать на одном кубите?
9. Можно ли использовать сигнал на линии кубита, после того, как уже было произведено измерение? Что получится в результате?
10. Возможно ли реализовать алгоритм Гровера не на двух, а на большем количестве кубитов? Как будет расти эффективность алгоритма по мере увеличения числа кубитов?

К3. Квантовый «композитор»

Цель работы: освоение приемов работы с симулятором квантовых вычислений.

Порядок выполнения работы

1. Изучить теоретическую часть.
2. Выполнить задания.
3. Представить файл для проверки преподавателю.

Quantum Composer — графический пользовательский интерфейс для программирования квантового процессора, предложенный фирмой *IBM* бесплатно для всех желающих пользователей. Чтобы начать работу с ним, достаточно зарегистрироваться на сайте <http://www.research.ibm.com/quantum/> (рис. 6.3.1). Для этого вам понадобится ваша личная электронная почта. Далее нужно перейти по ссылке Explore the IBM Quantum Experience.

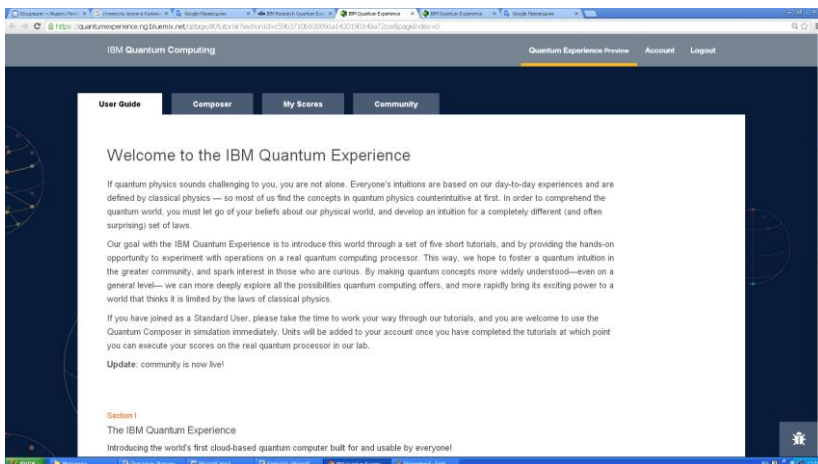


Рис. 6.3.1. Меню сайта, в котором нужно выбрать *Composer*

При первом нажатии на вкладку «Composer» вам будет предложено определить, хотите ли вы запустить идеальный квантовый процессор или реальный квантовый процессор (рис. 6.3.2).

New Experiment
✕

Please, enter a name for your experiment:

Experiment Name

Ideal Quantum Processor

Topology independent

Real Quantum Processor

Based in real Topology

Рис. 6.3.2. Выбор симулятора или реального квантового процессора

Для наших целей будет достаточно симулятора, поэтому выполнение лабораторных работ потребует лишь использования раздела *Ideal Quantum Computer*.

В *идеальном* процессоре преобразования (*gates*) могут быть размещены в любом месте, в то время как в *реальном* процессоре топология задается физическим устройством, которое в настоящее время работает в лаборатории *IBM* в режиме *on line* (именно это ограничивает возможности использования некоторых из двух кубитов). В *идеальном* процессоре мы по-прежнему имеем дело с симулятором, а не с реальным квантовым компьютером. Однако он является инструментом для построения квантовых схем с использованием библиотеки четко определенных команд (преобразований) и соответствующих измерений. А вот реальный процессор — это уже ваши первые шаги на настоящем квантовом компьютере!

После того, как вы окажетесь на вкладке *Composer*, вы можете начать делать собственные квантовые схемы (рис. 6.3.3).

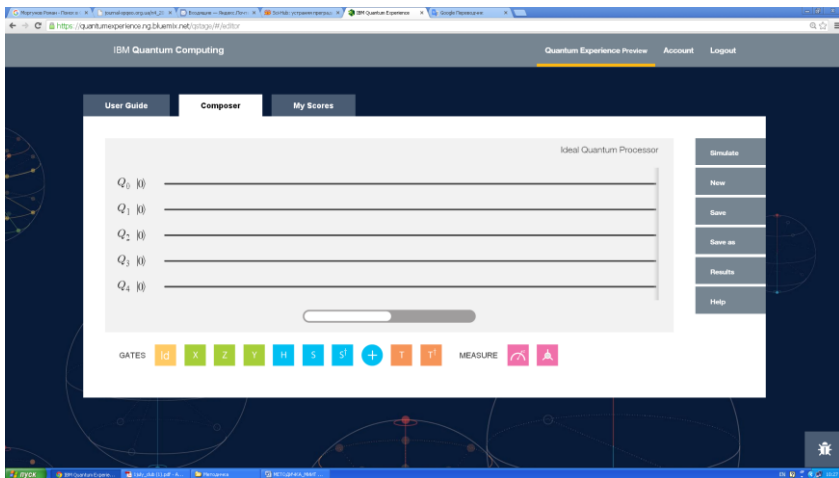


Рис. 6.3.3. Вкладка *Composer* с панелью для перетаскивания операций *X*, *Y*, *Z*, *H* и другие на линии кубитов *Q*₁, *Q*₂...

Мы называем это изображение «квантовый счет» (*quantum score*), потому что оно напоминает музыкальный временной ряд. Время от-

считывается слева направо. Каждая строка представляет собой перечисление событий, которые происходят с данным кубитом с течением времени. Каждый кубит работает на другой частоте, как и строка другой музыкальной ноты. Квантовые операции (*quantum gates*) представлены цветными квадратиками внизу, и они преобразуют длительности, амплитуды и фазы волновых функций. Они называются однокубитными операциями (командами). Пример панели с введенными (перемещенными мышкой) операциями выглядит так, как изображено на рис. 6.3.4. Здесь собрана схема для генерации случайных чисел с помощью квантового компьютера. Этот алгоритм имеет аналог и в классическом компьютере.

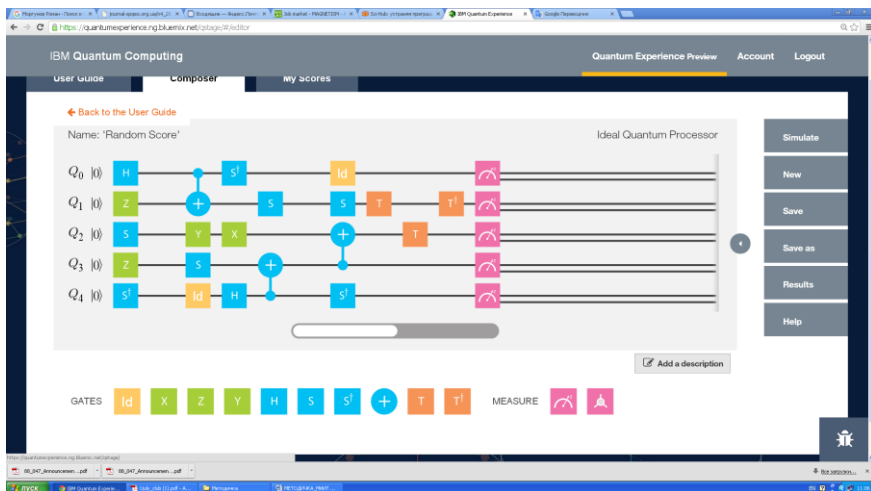


Рис. 6.3.4. Квантовая схема, собранная для генерации случайных чисел

Команда $\oplus$ оснащена вертикальными линиями, которые соединяют два кубита вместе и известна как команда *CNOT*; эта двухкубитная команда аналогична исключающему ИЛИ в обычной цифровой логике. Некоторые команды, такие как *CNOT*, имеют аппаратные ограничения; набор допустимых соединений определяется в схеме устройства, расположенного ниже.

Квантовый композитор (Quantum Composer). Библиотека квантового композитора (расположена внизу) содержит четыре класса команд, каждый из которых обозначается своим цветом. Нажатие на кнопку справки на правой панели дает краткий обзор всех команд. Первый класс команд (желтые кубики) представляет собой операции холостого хода (задержки) кубита в течение времени, равного длительности команды. Второй класс команд (зеленый) представляет собой группу, известную как операторы Паули X , которые представляют собой повороты объекта (бит-флип соответствует классическому NOT), фазовые перевороты Y (фаза-флип) и комбинированные бит-флип и фаза-флип перевороты Z . Третий класс (синий) представляет собой команды Клиффорда H , S , $S+$ для создания квантовых наложений, т.е. наряду с важными операциями $CNOT$ и двухкубитными операциями, нужны команды, которые осуществляют «запутывание». Последний класс (оранжевый) представляет собой команды, которые необходимы для обеспечения контроля.

Квантовый алгоритм (схема) начинается с подготовки системы кубитов и переводу их в четко определенные состояния. Эта процедура называется инициализацией и в предлагаемой программе она совершается автоматически. За исходные состояния принимаются $|0\rangle$. Далее идет выполнение ряда одно- и двухкубитных команд во времени, а затем измерение состояния кубитов. Измерения изображены розовыми квадратиками с нарисованным измерительным прибором на них. Есть два варианта измерения: стандартное измерение, которое является простой проекцией, или измерение Блоха, которое представляет проекцию состояния на сферу Блоха, чтобы указать конечное состояние кубита, проецируемое вдоль осей X , Y , Z . После измерения, состояние кубита становится классическим (или $|0\rangle$ или $|1\rangle$); это представлено двойной линией, которая появляется после операции измерения. Все эти элементы и совершаемые ими действия станут более ясными, когда вы начнете запускать программу и по очереди испытывать работу команд.

Для использования «композитора» нужно просто перетащить квадратики-команды на отметки, которые стоят на временной линии каждого кубита. Чтобы удалить или перетащить их в корзину нужно дважды шелкнуть по ним. Для операции $CNOT \oplus$ нужно перетащить ее на отметку и указать целевой кубит, а затем снова нажать на па-

нель управления кубита. Обратите внимание, что как только вы произвели измерения, вы не можете далее осуществлять квантовые операции, потому что информация в этой точке стала классической.

Как выглядит решение в квантовом компьютере? Если вы выбрали «идеальный квантовый процессор» (*Ideal quantum processor*), далее единственная опция — это запустить счет в моделировании. Идеальный процессор позволяет больше операций и соединений, чем доступны с реальным устройством, которое ограничено физическими процессами и их скоростью. Если вы нажимаете клавишу «Имитировать», программа выполняется немедленно, и ваши результаты изображаются в виде распределения вероятностей по различным состояниям. Например, программа случайного счета в пятибитной системе выдала результат, показанный на рис. 6.3.5. Этот результат не должен вас удивлять, поскольку квантовый компьютер выдает результаты с некоторой вероятностью. Так что даже расчет 2×2 он покажет в виде вероятностей различных состояний. В этом причина того, что использовать квантовый компьютер для тривиальных вычислений неразумно — он всегда проиграет классическому компьютеру в точности и скорости выполнения операций.

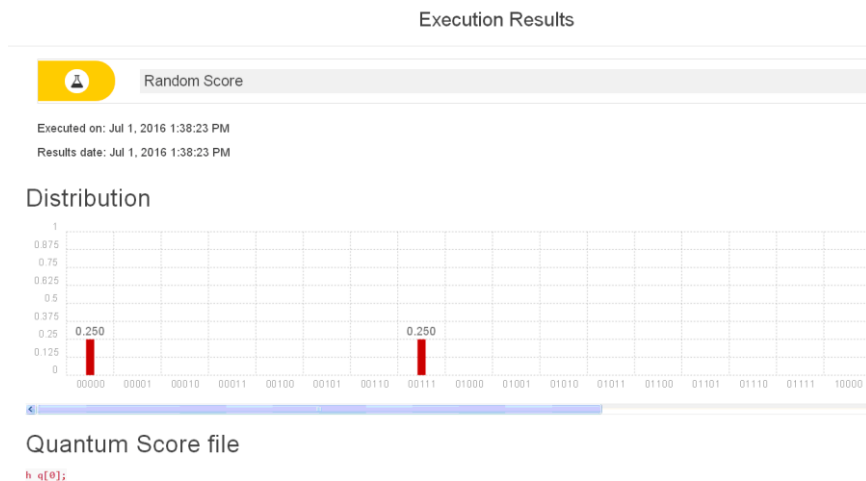


Рис. 6.3.5. Результаты квантовых вычислений для программы случайного квантового счета

Результаты расчетов могут быть представлены в двух формах: в виде уже упомянутого распределения по состояниям (рис. 6.3.5), а также в виде точек на сфере Блоха, которые также символизируют заполнение состояний при считывании, т.е. визуализируют информацию, даваемую распределением в пространстве (рис. 6.3.6).

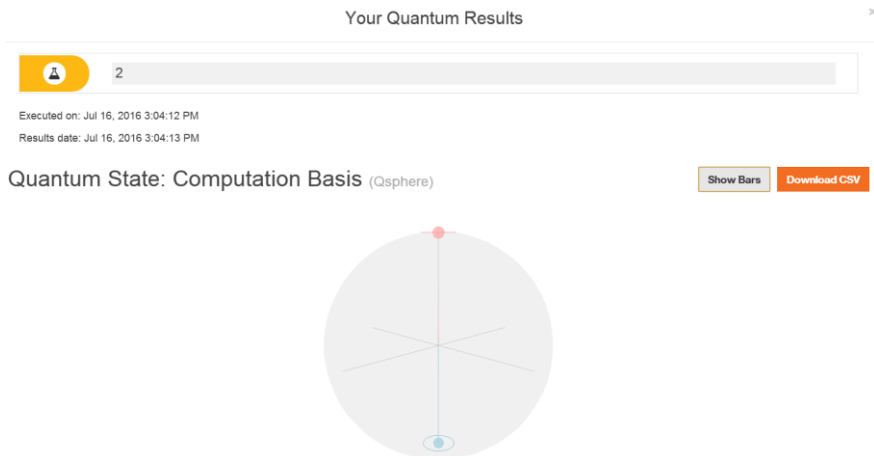


Рис. 6.3.6. Представление результата расчета с помощью сферы Блоха

Если вы выбрали *Real Quantum Processor*, вы можете заполнить вашу панель командами и измерениями, а затем нажмите кнопку «Выполнить», чтобы запустить процесс на фактическом квантовом компьютере. Выглядит от так, как показано на рис. 6.3.7.

Этот реальный компьютер находится в фирме *IBM* на другом континенте при сверхнизкой температуре в несколько миллиКельвин (обратите внимание на белые цилиндрические криостаты слева и справа на рисунке — именно там располагаются квантовые процессоры). Доступ к онлайн вычислениям на этом компьютере предоставляется по очереди всем заинтересованным пользователям. Эта опция в настоящее время требует использования крипто-валюты, для того чтобы управлять очередью пользователей квантового процессора. Сразу после регистрации в качестве стандартного пользователя на сайте *IBM Quantum Experience* у вас не будет каких-либо единиц ва-

люты. Однако вы можете получить слоты до 10 единиц. Изучая учебник в руководстве пользователя, вы можете получить эти 10 единиц. Когда вы получите эти единицы, при нажатии кнопки *Run* на экране отобразится новый квадратик с вопросом о том, сколько «выстрелов» или повторов вашего вычисления вы хотели бы отработать. В зависимости от вашего ответа будет определена «стоимость» в условных единицах.

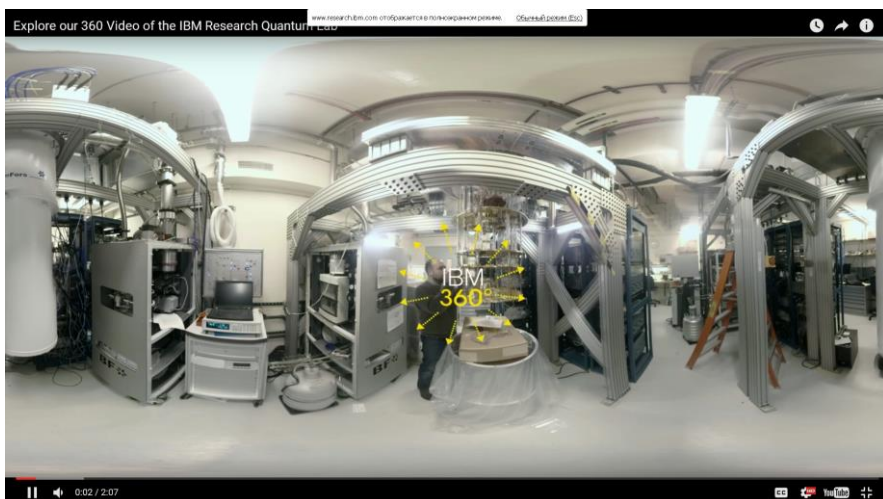


Рис. 6.3.7. Квантовый компьютер IBM

После нажатия *Ok* вы получите возможность либо запускать новый процесс и получить необходимые единицы, либо немедленно получить результаты от предыдущего запуска из одного и того же процесса, из КЭШа фирмы бесплатно. Вы заметите, что многие процессы из учебника используют ряд предыдущих процессов, на которых они могут быть построены, и потому позволяют вам наблюдать и учиться. Запуск совершенно нового исполнения помещает ваш процесс в очередь, и вы получите уведомление по электронной почте, когда ваш счет был запущен. В качестве стандартного пользователя вы можете пополнить ваш счет по запросу «Счет» в верхней части экрана. Все простейшие операции и некоторые их комбинации, а также все логические схемы, которые предлагаются в данных лабораторных работах, уже рассчитаны реальным квантовым компьютером. Поэтому

при выборе реального квантового компьютера и запуске вычислений клавишей *Run* в дальнейшем вам будет предложено не тратить бит-коины, а получить готовое решение, уже произведенное машиной до вашего запроса и хранящееся в памяти удаленного сервера. Для этого нужно выбрать *Result from Cache*.

После запуска процесс расчета будет виден во вкладке *My counts*, где все процессы будут отсортированы по дате их выполнения. Когда ваши результаты будут готовы, вы сможете просматривать их с этой вкладки. Вы также можете повторно отредактировать ваш процесс и снова запустить его на тренажере в ожидании результатов.

Задания

1. Начать нужно с того, чтобы по очереди помещать все операторы на линии кубитов и устанавливать после них измерительный прибор. Например, так, как показано на рис. 6.3.8.

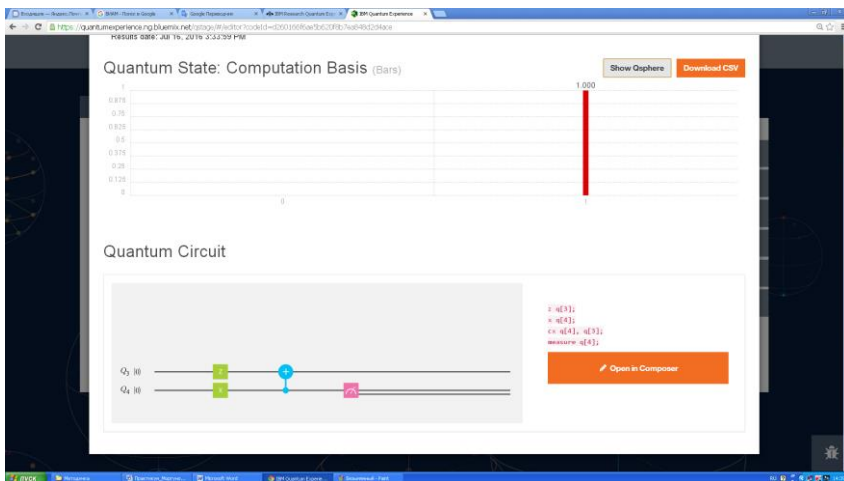


Рис. 6.3.8. Логическая схема (внизу) и результаты измерения (вверху) после выполнения преобразований Паули X и Z и смешивания получившихся состояний операцией «+», а также измерения состояний 4-го кубита. Исходные состояния 3-го и 4-го кубитов были чистыми состояниями 0

Таким образом, первое задание заключается в том, чтобы составить отчет и таблицы вероятностей для всех типов операций, которые предлагаются симулятором.

2. Вы можете заметить, что сами по себе элементы не изменяют исходного состояния кубитов. Наличие ненулевой вероятности заполнения состояния 1 возникает только в случае смешивания сигналов кубитов разных линий с помощью операции «+».

Контрольные вопросы

1. Чем смешанные состояния отличаются от чистых?
2. Какие состояния получаются в результате измерения?
3. Зависит ли результат измерения от способа измерения?

К4. Элементарные квантовые алгоритмы

Цель работы: изучение основных однокубитных квантовых логических алгоритмов.

Порядок выполнения работы

1. Изучить теоретическую часть.
2. Выполнить задания.
3. Представить файл для проверки преподавателю.

Как вы уже догадались, кубит делает нечто больше, нежели просто находится в основном состоянии. Для того чтобы поместить его в возбужденное состояние, нам нужны квантовые команды. В этом разделе мы представим некоторые команды и как их использовать в *Composer*. Квантовые команды, или операции, как правило, представлены в виде матриц. Оператор (*gate*), который действует на один кубит, представлен 2×2 матрицей. Поскольку квантовые операции должны быть обратимыми и сохранять амплитуды вероятности, матрицы должны быть унитарными. Результат действия квантового вентилля определяется путем умножения матрицы, представляющей квантовое состояние.

Логический элемент NOT. Рассмотрим квантовый логический элемент отрицания *NOT*, обозначив его через *X*. Сначала посмотрим, как работает соответствующая математика и физика. Определим дей-

ствие этого оператора на базисные векторы (так говорят математики), или посмотрим, что делает этот квантовый процесс с чистыми состояниями системы (так говорят физики о том же самом). Потребуем, чтобы оператор X переводил $|0\rangle$ в $|1\rangle$, а $|1\rangle$ в $|0\rangle$:

$$X|0\rangle = |1\rangle,$$

$$X|1\rangle = |0\rangle.$$

В этом случае квантовый оператор NOT станет обобщением классического оператора NOT , ведь он действует точно так же в «чистых» случаях, да еще способен изменять смешанные состояния $|\psi\rangle$. Поэтому квантовый оператор «богаче» классического. Запишем действие оператора X на произвольный кубит с амплитудами вероятностей состояний α и β :

$$X|\psi\rangle = X(\alpha|0\rangle + \beta|1\rangle) = \alpha X|0\rangle + \beta X|1\rangle.$$

В результате получим:

$$X|\psi\rangle = \alpha|1\rangle + \beta|0\rangle = \beta|0\rangle + \alpha|1\rangle.$$

Таким образом, оператор X меняет местами коэффициенты при базисных векторах 0 и 1. Если перейти на язык матричной алгебры, то оператор можно записать в виде матрицы. Тогда его матричные элементы X_{mn} запишутся в виде:

$$X_{00} = 0, X_{10} = 1, X_{01} = 1, X_{11} = 0.$$

Запишем матрицу X оператора X_{mn} :

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

Подеиствуем этой матрицей на исходное состояние, в котором находится кубит. Тогда получим результат действия оператора X :

$$\begin{pmatrix} \bar{\alpha} \\ \bar{\beta} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$$

Он просто переставляет местами амплитуды вероятностей в столбце.

Такие операторы называются *унитарными* (от слова «единица» — *unit*) потому что для них выполняется простое свойство:

$$X^+X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^+ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I.$$

Выполним теперь эту же операцию с помощью квантового симулятора.

Сначала построим схему (рис. 6.4.1), затем нажмем *Simulate* на правой панели:

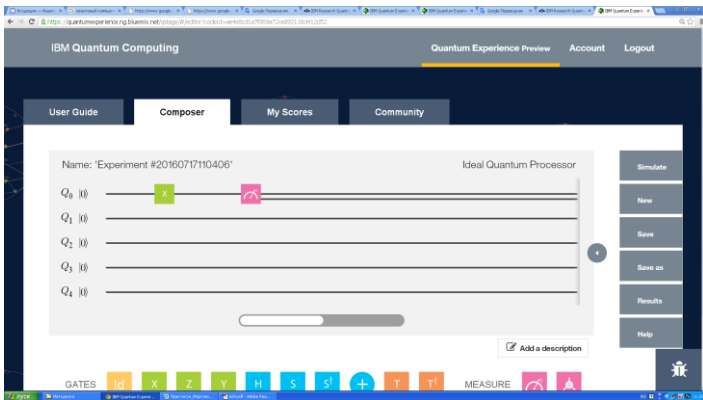


Рис. 6.4.1. Панель с одним оператором и измерителем

Полученный результат будет выглядеть так, как показано на рис. 6.4.2.



Рис. 6.4.2. Результат действия оператора X (NOT), представленный в виде вероятности состояний

Полученный результат можно представить иначе. В общем случае коэффициенты α и β являются комплексными числами. Поэтому

они не имеют определенного физического смысла и размерности. Однако их модули или квадраты такой смысл имеют — это величины связанные простым тождеством $\alpha^2 + \beta^2 = 1$. Таким же тождеством связаны вероятности полной группы событий (вспоминаем про «орел» и «решку» в теории вероятностей). Поэтому и здесь смысл этих коэффициентов прост — их квадраты являются вероятностями заполнения соответствующих состояний. Если коэффициенты комплексные, то им можно сопоставить точку на сфере, поскольку комплексное число может быть представлено в тригонометрическом виде $\alpha = \cos\psi + i \sin \psi$. Аналогично для β можно записать: $\alpha = \cos\varphi + i \sin \varphi$. Углы ψ и φ однозначно задают положение точки на сфере единичного радиуса. Применительно к рассматриваемой нами задаче она называется *сферой Блоха* или *Q-sphere* (сфера кубита). В трехмерном пространстве рассуждать всегда тяжело, поэтому, если представить, что мы имеем дело с вещественными числами α и β , то вместо сферы можно будет рассматривать единичную окружность.

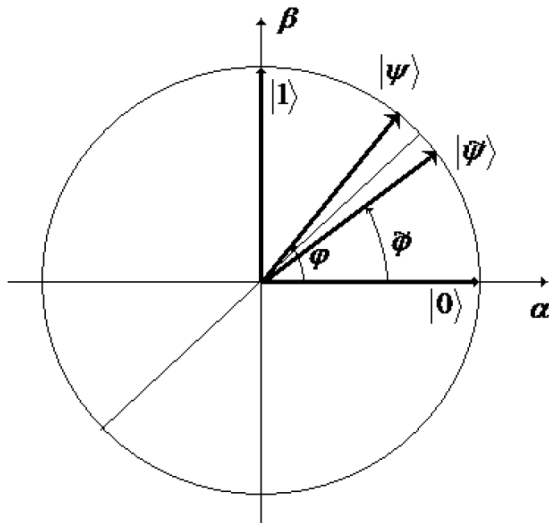


Рис. 6.4.3. Результат действия оператора $\hat{X}$ для случая вещественных значений α и β
 Вектор состояния $|0\rangle$ преобразован в вектор состояния $|1\rangle$ путем изменения угла ψ на 90 градусов

В симуляторе предусмотрена возможность построения состояний на сфере Блоха в трехмерном случае. Для этого после получения решения нужно нажать клавишу *Show Qsphere*. Таким образом, полученный результат (изменение вероятности состояний) можно представить графически в виде перемещения точки по сфере Блоха на противоположный полюс (рис. 6.4.4).

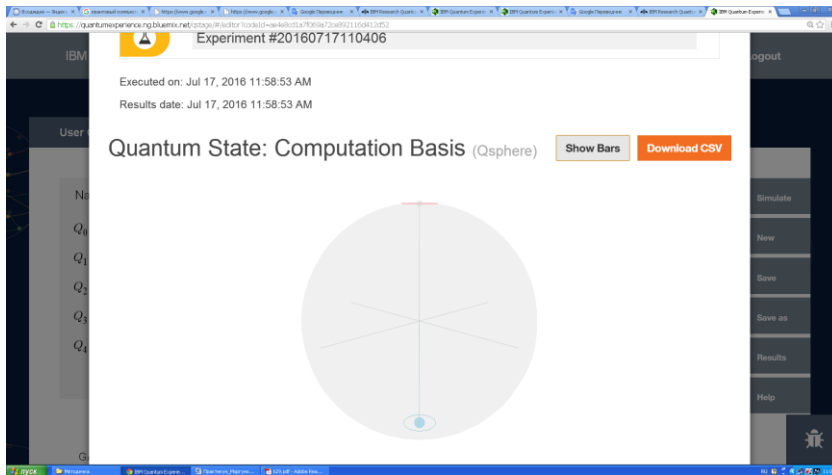


Рис. 6.4.4. Графическое представление действия оператора X (NOT)

Если теперь запустить реальный процессор и посмотреть, что получится в результате действия оператора X , то обнаружится расхождение с идеальным процессором (рис. 6.4.5).

Из рис. 6.4.5 видно, что в отличие от идеального симулятора, реальный компьютер никогда не дает результат, совпадающий точно с математическими вычислениями. Как уже говорилось, причиной этого поведения являются ошибки, вызванные температурным шумом. Даже огромных криостатов и очень низких температур не хватает, чтобы полностью подавить этот шум. Поэтому маленький столбик ненулевого значения α соответствует этому шуму и той доли компьютера, которая под действием тепла перестает быть квантовой.

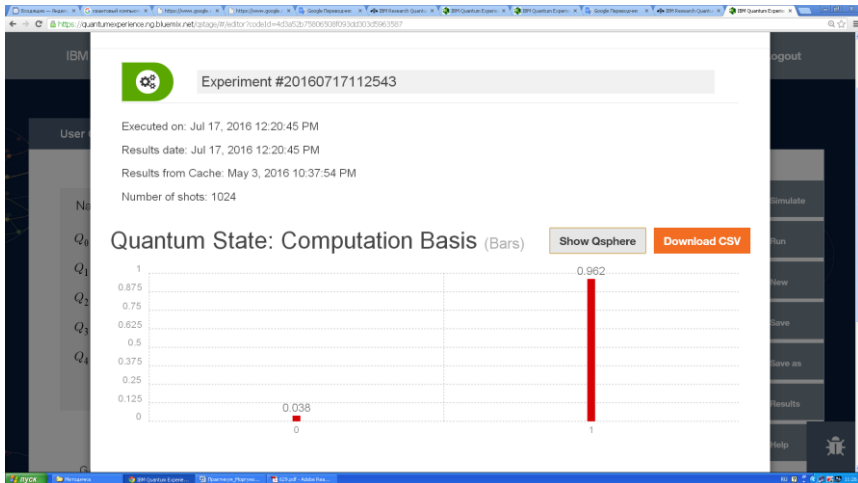


Рис. 6.4.5. Результаты выполнения операции X реальным квантовым компьютером

Не углубляясь в детали, приведем матрицы оператора Z . Этот оператор, действуя на базисные векторы, не изменяет $|0\rangle$, но $|1\rangle$ переводит в $-|1\rangle$:

$$Z|0\rangle = |0\rangle,$$

$$Z|1\rangle = -|1\rangle.$$

Матрица этого оператора выглядит так:

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

Если этим оператором воздействовать на произвольное смешанное (небазисное) состояние, получится следующий результат:

$$\begin{pmatrix} \tilde{\alpha} \\ \tilde{\beta} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ -\beta \end{pmatrix}$$

Для вещественных коэффициентов на единичной окружности результат действия оператора Z выглядит так:

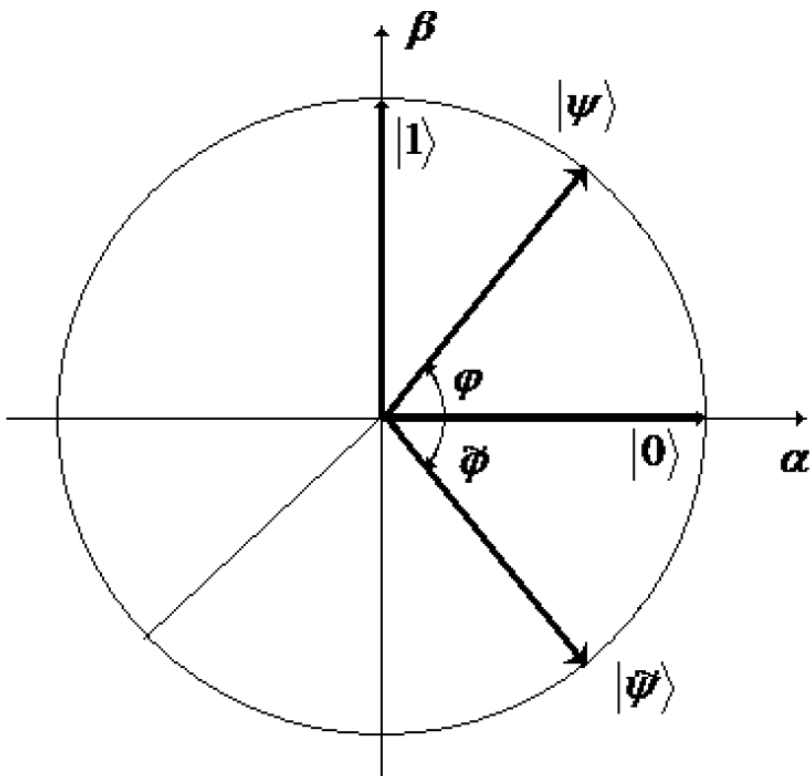


Рис. 6.4.6. Результат действия оператора Z на базисный вектор $|0\rangle$ и на произвольный вектор смешанного состояния $|\psi\rangle$

Наконец, еще один оператор Y запишется в виде матрицы:

$$Y = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}.$$

Все перечисленные операторы имеют важный физический смысл, который прямо указывает на их физическую реализацию. Эти операторы в физике называются *матрицами Паули* и соответствуют математическим действиям, которые описывают переориентацию механического или магнитного момента элементарной частицы (электрона). Каждый электрон можно представлять себе как маленький волчок

(юлю), чистые состояния которого отвечают направлению оси вращения только вдоль или против внешнего магнитного поля. Можно также думать, что электрон — это дискретная стрелка компаса, которая может быть направлена только вдоль или только против магнитного поля. Смешанное состояние соответствует примеси состояний «вдоль» и «против» в большом коллективе частиц. Квадраты коэффициентов α и β являются вероятностями найти стрелку компаса в соответствующих состояниях. О кремниевых устройствах, использующих для квантовых вычислений эти свойства спинов, говорилось выше. Однако в настоящей лабораторной работе к производимым вычислениям можно также относиться как к предсказаниям ориентации спинов при действии на них радиочастотных импульсов. Операторы X , Y , Z реализуются на практике облучением коллектива спинов сверхвысокочастотным электромагнитным полем.

Задания

1. Произвести квантовую симуляцию для операций X , Y , Z (по схемам, изображенным на рис. 6.4.7) и представить результаты в виде распределения вероятностей и перемещения точки на сфере Блоха.

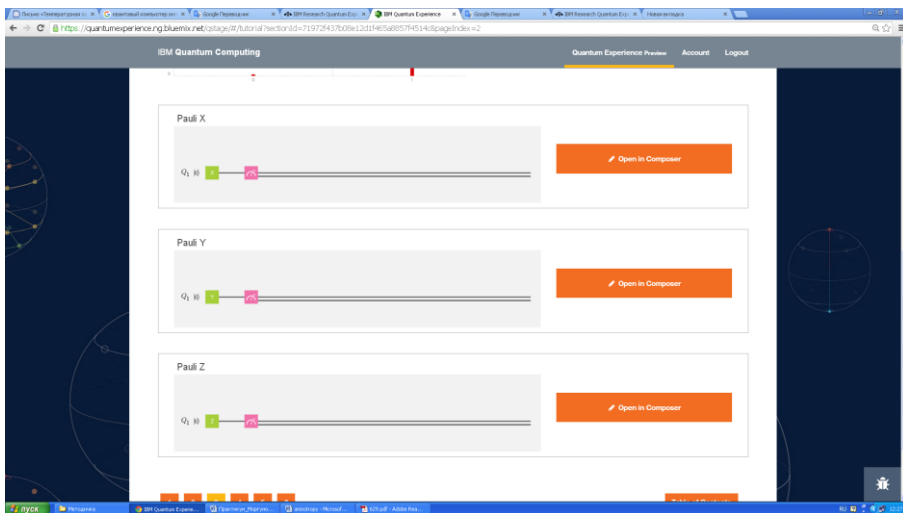


Рис. 6.4.7. Схемы для испытаний однокубитных квантовых операций X , Y , Z

2. Создайте последовательные цепочки однокубитных команд XU , XYZ , XZ и получите результаты вычислений в виде распределения вероятностей на сфере Блоха.

3. Получите те же результаты в идеальном компьютере (из кэш-памяти) и сравните с предсказаниями реального компьютера.

4. Пользуясь матричным представлением операторов X , Y , Z , запишите матрицы, которые отвечают операторам XU , XYZ , XZ , и рассчитайте результат их действия на состояния $|0\rangle$ и $|1\rangle$. Сравните полученные результаты с результатами вычисления идеального и реального квантовых компьютеров.

Контрольные вопросы

1. Что будет с вероятностями α^2 и β^2 в результате расчетов оператора X , если в фирме *IBM* закончится жидкий гелий?

2. Какие способы представления однобитной информации вам известны?

3. Что такое «сфера Блоха»?

4. Что такое «унитарный оператор»?

5. Каков смысл величин α и β ?

6. Что такое матрицы Паули?

7. Какому реальному физическому процессу соответствуют матрицы Паули?

8. Рассчитать, как действует оператор Y на векторы чистых состояний $|0\rangle$ и $|1\rangle$, а также на смешанное состояние, описываемое амплитудами вероятностей α и β .

9. Убедиться в унитарности операторов X и Z .

10. Каким изменениям ориентации спинов электронов отвечает каждая из матриц Паули?

K5. Квантовый оператор Адамара (*Hadamard gate*)

Цель работы: изучение однокубитных квантовых логических алгоритмов, смешивающих состояния.

Порядок выполнения работы

1. Изучить теоретическую часть.

2. Выполнить задания.

3. Представить файл для проверки преподавателю.

В предыдущей работе были рассмотрены операторы, которые превращали одно чистое квантовое состояние в другое чистое состояние, не смешивая их. Теперь рассмотрим элементы квантовой логики, действие которых приводит к смешиванию квантовых состояний.

Квантовый логический элемент Адамара записывают в виде матрицы:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Его действие на смешанное состояние с амплитудами вероятностей запишется в виде

$$\begin{pmatrix} \tilde{\alpha} \\ \tilde{\beta} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} \alpha + \beta \\ \alpha - \beta \end{pmatrix}$$

Видно, что в результате действия оператора Адамара получаются так называемые «симметричное» и «антисимметричное» смешанные квантовые состояния.

Для того чтобы проверить, как работает такой оператор, построим элементарную схему (рис. 6.5.1), затем нажмем *Simulate* на правой панели:

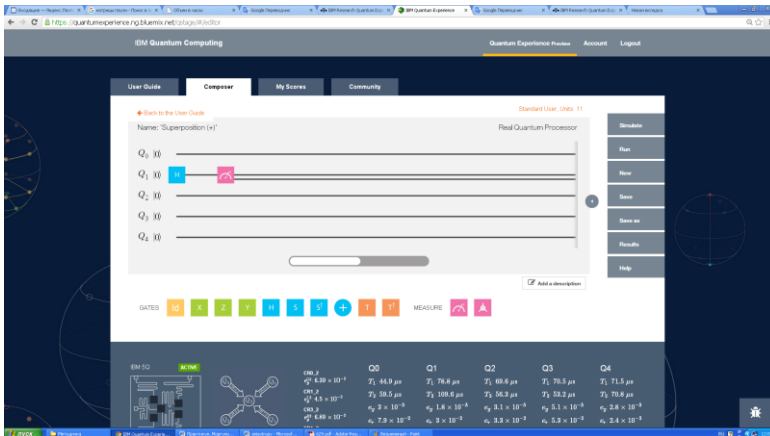


Рис. 6.5.1. Логическая схема для испытания оператора Адамара Н

Полученный результат будет выглядеть так, как показано на рис. 6.5.2.

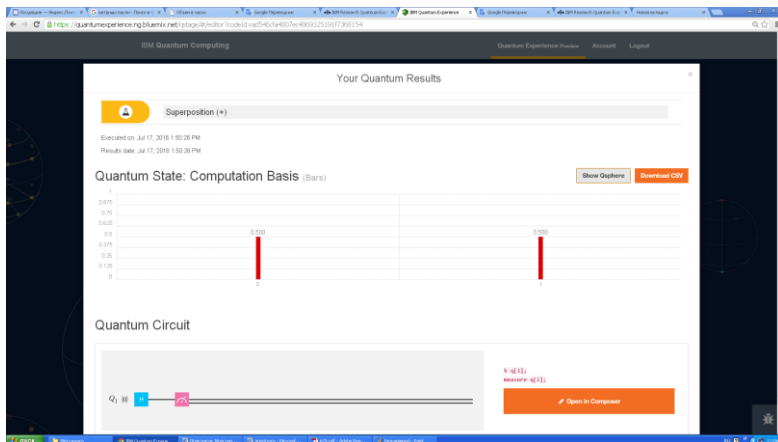


Рис. 6.5.2. Результат действия оператора Н (Hadamard gate), представленный в виде вероятности состояний

Нетрудно также получить результат вычисления Н в реальном квантовом компьютере (рис. 6.5.3).

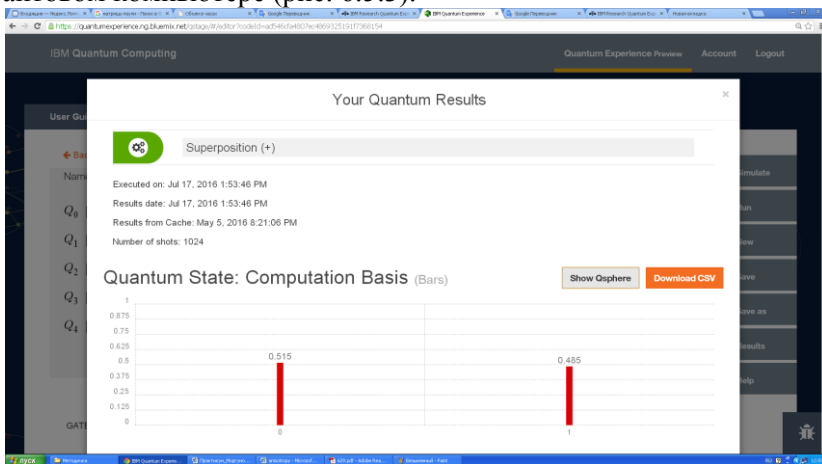


Рис. 6.5.3. Результаты выполнения операции Н реальным квантовым компьютером

Он отличается от реального компьютера во втором знаке после запятой в записи амплитуд вероятностей состояний. В чем причина получения такого результата, который свидетельствует о том, что система одновременно находится с равной вероятностью в состояниях $|0\rangle$ и $|1\rangle$? Исходное состояние кубита $|0\rangle$ было преобразовано в состояния $(|0\rangle + |1\rangle)/2^{1/2}$ и $(|0\rangle - |1\rangle)/2^{1/2}$, называемые симметричным и антисимметричным, соответственно.

В практической части вы убедитесь, что смешивание операции Адамара с операциями Паули не приводит к разнице в измерениях, хотя эти результаты отвечают разным состояниям системы. Для того чтобы обнаружить различие между этими состояниями, мы должны провести измерения в диагональном базисе. В наших экспериментах мы не можем физически изменить измерение, однако мы можем сделать некоторый аналог, если изменим измерение с помощью операции, вставленной перед измерением. Для измерения в диагональном базисе стандартный базис (Z) поворачивается по отношению к (X) с помощью операции Адамара.

Попробуйте рассчитать суперпозицию (+) X (см. рис. 6.5.4), а также суперпозицию (-) X, показанную на схеме далее.

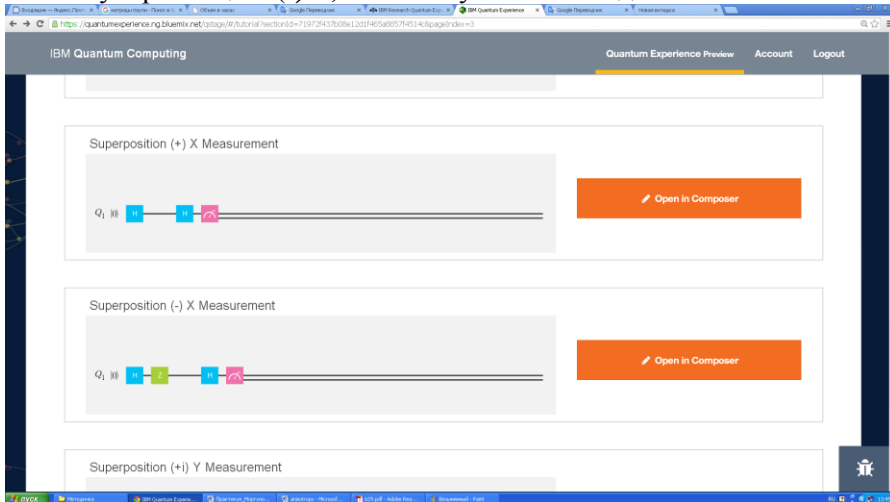


Рис. 6.5.4. Реализация смешанных операций (+) X и (-) X

Вы должны найти, что почти в 100% случаев распределение вероятностей станет равно 0 и 1, соответственно. Другими словами, если мы делаем измерение в стандартном базисе, результат совершенно однородный (50% на 50%), но в диагональном базисе, состояние оказывается детерминированным. Измерения не могут различать все четыре вида состояний $|0\rangle, |1\rangle, |+\rangle$ и $|-\rangle$. Это не является ограничением измерения или каким-то дефектом квантового компьютера. Здесь впервые в обыденной жизни вы сталкиваетесь с фундаментальным следствием принципа неопределенности. Именно это ограничение приводит к возможности квантовых денег и квантовой криптографии.

Задания

1. Произвести квантовую симуляцию для операций HX , HU , HZ (по схемам, изображенным на рис. 6.5.5) и представить результаты в виде распределения вероятностей и перемещения точки на сфере Блоха.

Убедиться, что эта схема приводит к результатам измерений, которые не отличаются от применения одной только операции Адамара. Разные квантовые состояния, к которым приводят операции Z и HZ , дают одинаковый результат их измерений на выходе.

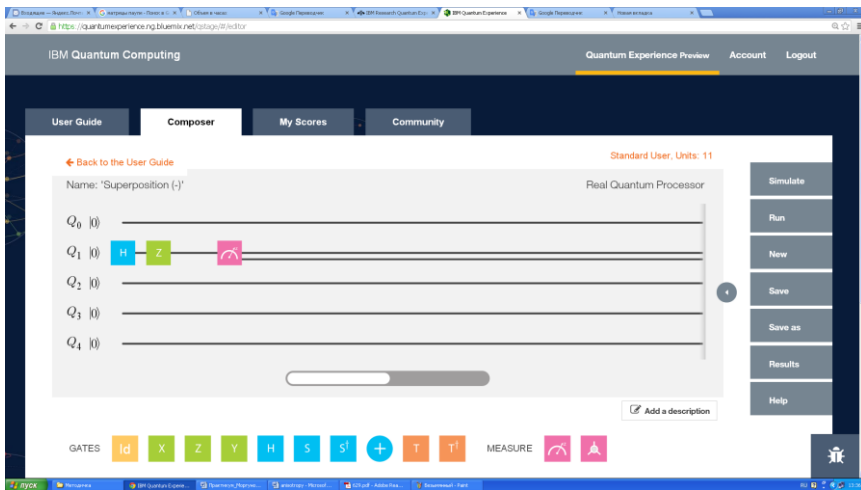


Рис. 6.5.5. Схема для испытаний однокубитных двойных квантовых операций HZ

2. Получите те же результаты в идеальном компьютере (из кэш-памяти) и сравните с предсказаниями реального компьютера.

3. Пользуясь матричным представлением операторов $-X$, $-Y$, $-Z$, запишите матрицы, которые отвечают этим операторам и рассчитайте результат их действия на состояния $|0\rangle$ и $|1\rangle$. Сравните полученные результаты с результатами вычисления идеального и реального квантовых компьютеров.

Контрольные вопросы

1. Рассчитать из условия нормировки коэффициенты α и β для преобразования Адамара.

2. Как убедиться в унитарности оператора Адамара?

3. Почему в результате измерения разных квантовых состояний может получиться один и тот же результат?

4. В чем проявляется квантовая неопределенность при изучении состояний, генерируемых операцией Адамара?

5. Что такое «квантовая криптография» и «квантовые деньги»?

6. Опишите процесс, заменяющий переход к новому базису в программе *Composer*.

7. Где в природе встречаются «симметричные» и «антисимметричные» состояния реальных физических объектов?

8. Будет ли заметно изменение в вычислениях операции Адамара, если квантовый компьютер неожиданно нагреется?

9. Как выглядит преобразование Адамара на сфере Блоха?

10. Для чего используется преобразование Адамара?

К6. Двухкубитные квантовые схемы

Цель работы: изучение простейших двухкубитных квантовых логических схем.

Порядок выполнения работы

1. Изучить теоретическую часть.

2. Выполнить задания.

3. Представить файл для проверки преподавателю.

Рассмотрим состояния квантовой системы, состоящей из двух кубит:

$$\begin{aligned} |\psi_1\rangle &= \alpha_1 |0_1\rangle + \beta_1 |1_1\rangle, \\ |\psi_2\rangle &= \alpha_2 |0_2\rangle + \beta_2 |1_2\rangle \end{aligned}$$

Состояния такой системы будем обозначать символами

$$|0\rangle_1, |1\rangle_1, |0\rangle_2, |1\rangle_2$$

Амплитуды вероятностей этих состояний будем записывать в виде:

$$\alpha_1, \beta_1, \alpha_2, \beta_2$$

Конечно, можно рассматривать векторные пространства каждого кубита по отдельности, т.е. считать кубиты невзаимодействующими. Однако наш главный интерес как раз заключается в том, чтобы с помощью нескольких кубитов научиться приготавливать новые полезные квантовые состояния.

Для двух кубитов эти состояния запишутся так:

$$\begin{aligned} |00\rangle &= |0_1\rangle \otimes |0_2\rangle, \quad |01\rangle = |0_1\rangle \otimes |1_2\rangle \\ |10\rangle &= |1_1\rangle \otimes |0_2\rangle, \quad |11\rangle = |1_1\rangle \otimes |1_2\rangle. \end{aligned}$$

Всего получается четыре чистых состояния двухкубитной системы.

Если базисные векторы однокубитных систем представить в уже знакомом нам виде,

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

то двухкубитные состояния можно выразить через них таким символическим способом:

$$|00\rangle = |0_1\rangle \otimes |0_2\rangle = \begin{pmatrix} 1 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 0 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$|01\rangle = |0_1\rangle \otimes |1_2\rangle = \begin{pmatrix} 1 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 0 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},$$

$$|10\rangle = |1_1\rangle \otimes |0_2\rangle = \begin{pmatrix} 0 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 1 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix},$$

$$|11\rangle = |1_1\rangle \otimes |1_2\rangle = \begin{pmatrix} 0 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 1 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Можно также показать, что для произвольного смешанного состояния в двухкубитной системе матрица U , которая соответствует применению однокубитных операторов, может быть получена с помощью вот такого громоздкого выражения:

$$U = U^{(1)} \otimes U^{(2)} = \begin{pmatrix} U_{00}^{(1)}U_{00}^{(2)} & U_{00}^{(1)}U_{01}^{(2)} & U_{01}^{(1)}U_{00}^{(2)} & U_{01}^{(1)}U_{01}^{(2)} \\ U_{00}^{(1)}U_{10}^{(2)} & U_{00}^{(1)}U_{11}^{(2)} & U_{01}^{(1)}U_{10}^{(2)} & U_{01}^{(1)}U_{11}^{(2)} \\ U_{10}^{(1)}U_{00}^{(2)} & U_{10}^{(1)}U_{01}^{(2)} & U_{11}^{(1)}U_{00}^{(2)} & U_{11}^{(1)}U_{01}^{(2)} \\ U_{10}^{(1)}U_{10}^{(2)} & U_{10}^{(1)}U_{11}^{(2)} & U_{11}^{(1)}U_{10}^{(2)} & U_{11}^{(1)}U_{11}^{(2)} \end{pmatrix}.$$

Из приведенного примера ясно, что с увеличением числа кубитов, задействованных в вычислениях, будет нарастать и сложность расчетов, позволяющих предсказать квантовые вычисления. Ведь число состояний для трех кубитов будет уже 8, а для четырех кубитов - 16. Другими словами, количество чистых квантовых состояний растет с числом кубитов как 2^N , в то время как в классическом компьютере число состояний прямо пропорционально N . Это и есть главное

различие классического и квантового компьютеров, которое позволяет иметь в компьютере не миллиарды ячеек памяти, а всего небольшое ограниченное число кубитов. Специалисты считают, что для удовлетворения всех ближайших нерешаемых классическим компьютером задач и интересов человечества, достаточно иметь 1000 кубитов, чтобы решить любую такую задачу в конечное время. На рис. 6.6.1 приведена квантовая схема с исходным состоянием двухкубитной системы $|00\rangle$ и реализация квантового вычисления при наличии смешивания состояний оператором «+».

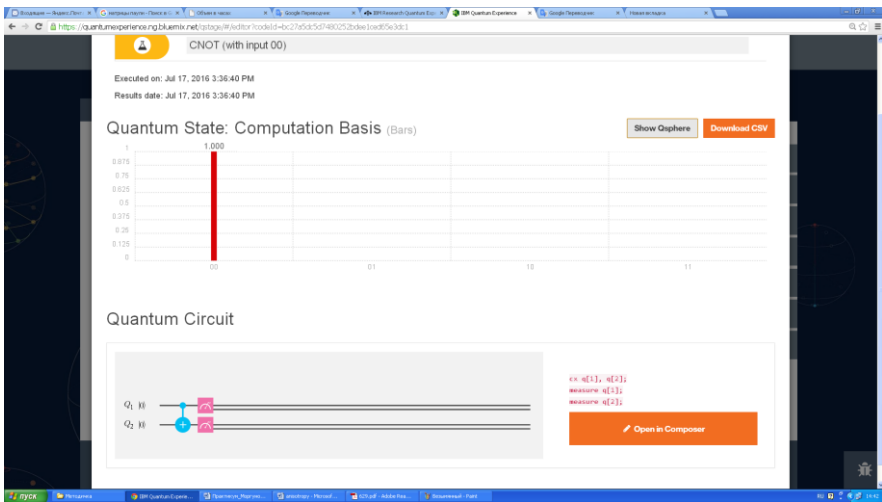


Рис. 6.6.1. Квантовая схема с исходным состоянием двухкубитной системы $|00\rangle$ и реализация квантового вычисления при наличии смешивания состояний оператором «+»

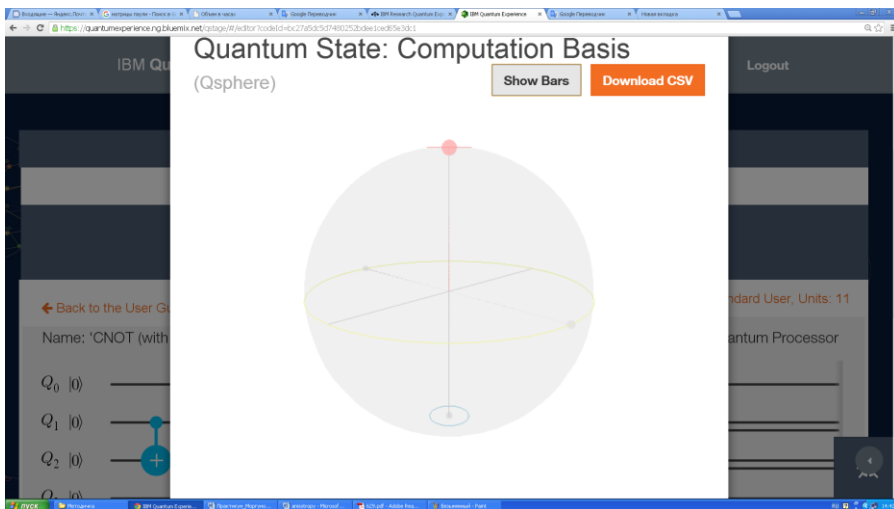


Рис. 6.6.2. Результаты вычисления квантовой схемы с исходным состоянием двухкубитной системы $|00\rangle$ на сфере Блоха

Задания

1. Собрать квантовую схему, используя квантовые логические элементы *CNOT*, *X*, *H* и *Z*, и подать на вход цепочки элементов двухкубитное состояние кубит, получить выходное двухкубитное состояние и, используя матричное представление схемы, сравнить результаты теоретических расчетов с полученными экспериментальными данными (рис. 6.6.3).

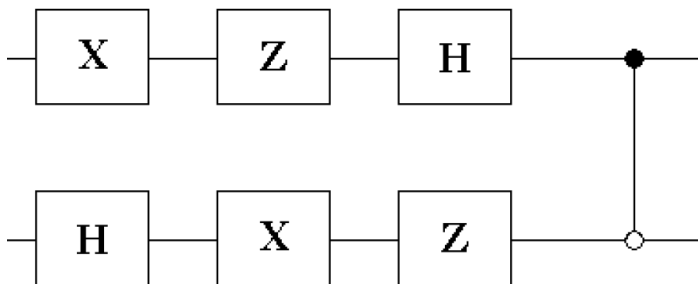


Рис. 6.6.3. Схема квантового оператора *CNOT*

2. Убедиться в правильности оператора $CNOT$, сравнив выполняемые им действия в виде матриц с результатами расчета квантового компьютера.

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

3. Проверить работу квантового генератора случайных чисел. Результаты наблюдений записать в таблицу.

Контрольные вопросы

1. Как зависит количество чистых состояний от числа кубитов в системе?

2. Имеется ли смысл в независимых кубитах?

3. Можно ли представить состояние, полученное в результате действия оператора $CNOT$, как комбинацию исходных чистых состояний системы?

4. Что такое «Бэлловские запутанные состояния системы»?

5. Кто такие Алиса и Боб в информатике?

6. Для каких задач планируется применять квантовый компьютер?

7. Какую роль играет тепловой шум в квантовых вычислениях?

8. Что такое «смешанные и чистые квантовые состояния»?

9. Почему квантовые состояния не учитываются в наномagnetном процессоре?

10. Сравните квантовый и классический компьютер. Какой из них лучше?

Рекомендуемая литература

Боумейстер, Д. Физика квантовой информации / Д. Боумейстер, А. Экерт, А. Цайлингер. — М. : Постмаркет, 2002.

Валиев, К. А. Квантовые компьютеры: надежда и реальность / К. А. Валиев, А. А. Кокин. — Ижевск : НИЦ «Регулярная и хаотическая динамика», 2001.

Нильсен, М. Квантовые вычисления и квантовая информация / М. Нильсен, И. Чанг ; пер. с англ. — М. : Мир, 2006.

Холево, А. С. Введение в квантовую теорию информации / А. С. Холево. — М. : МЦНМО, 2002.

Чивилихин, С. А. Квантовая информатика : учеб. пособие / С. А. Чивилихин. — СПб., 2009.

Яковлев, В. П. Элементы квантовой информатики / В. П. Яковлев, М. П. Кондрашин. — М. : МИФИ, 2004.