МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное

учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**В. А. Нестеренко**

**Квантовые вычисления**

**Методическое пособие**

Ростов-на-Дону

2019

**Содержание**

1. Квантовые вычисления……………………………………………………..3
   1. Физические основы квантовых вычислений…………………………..3
   2. Математическая модель квантовых вычислений……………………...7
   3. Квантовая запутанность и квантовый параллелизм…………………...14
   4. Алгоритм Дойча…………………………………………………………16
   5. Система кубитов. Алгоритм Дойча-Джоза……………………………19
   6. Алгоритм Гровера………………………………………………………21
2. Приложение B. Эмуляция квантовых вычислений………………………..24
3. Литература…………………………………………………………………..34

***1. Квантовые вычисления***

***1.1. Физические основы квантовых вычислений.***

*«Если вам кажется, что вы понимаете квантовую теорию… то вы не понимаете квантовую теорию.» – Ричард Фейнман.*

Квантовые вычисления [3] относительно новая и быстро развивающаяся область программирования. Результаты теоретических разработок и лабораторных исследований в этой области находятся в тесной взаимосвязи. Возможно в ближайшее время произойдёт прорыв в области создания экспериментальных образцов квантовых компьютеров.

Развитие вычислительных систем сопровождается существенным уменьшением размеров элементной базы. Размеры отдельных компонент современных процессоров приближаются к размерам атомов и поведение элементов вычислительных систем начинает подчиняться законам квантовой физики. В некоторых случаях поведение квантовых систем могут существенным образом отличаться от поведения систем, подчиняющихся законам классической физики. Из-за новых квантовых свойств вычислительных систем открываются возможности разработки и реализации новых алгоритмов, существенно отличающихся от алгоритмов для классических вычислительных систем. Таким образом, интерес к квантовым вычислениям можно объяснить следующими причинами:

* Уменьшение элементной базы до таких размеров, для которых начинают работать законы квантовой физики.
* Квантовые свойства систем позволяют реализовывать алгоритмы принципиально отличные от классических.

Основным отличием квантовой системы от классической является принципиальная невозможность исключить зависимость системы от наблюдателя, исключить влияние процесса измерения на состояние системы. В классической физике при определении физических характеристик классического объекта влияние измерительных приборов на объект может быть пренебрежимо мало и сам процесс измерения может быть исключён из описания состояния физической системы. В квантовой механике из-за малых размеров изучаемых объектов, когда энергия, импульс, и другие физические характеристики изучаемых объектов сопоставимы с соответствующими характеристиками взаимодействия измерительного прибора с объектом, невозможно исключить влияние процесса измерения на состояние изучаемой системы. Так, например, при определении положения объекта в пространстве можно посмотреть на объект и сопоставить его положение с некоторой измерительной шкалой. Но процесс "посмотреть" подразумевает освещение объекта и наблюдение за светом, отраженным от предмета. В классическом случае интенсивность (энергию) светового пучка можно сделать пренебрежимо малой по сравнению с энергией объекта и влиянием процесса отражения света на объект не принимать во внимание. В квантовом случае следует учитывать, что пучок света состоит из фотонов, при уменьшении энергии фотона увеличивается его длина волны, фотон "размазывается" в пространстве, его размер может превысить размер измеряемого объекта и, как следствие, пропадает возможность локализации объекта в пространстве.



Рис 8.

Наиболее наглядно влияние процесса измерения на объект можно наблюдать в опыте с интерференцией электронных пучков представленном на Рис8. Электронная пушка C испускает электроны, которые пройдя через две щели S1 и S2 попадает на экран S. Детектор D позволяет фиксировать число частиц, попадающую в определённую область экрана. График плотности G, полученной в результате эксперимента, приведён на рисунке и имеет вид интерференции, что свидетельствует о волновой природе электрона. Полученный результат можно попытаться объяснить взаимодействием электронов прошедших через разный щели. Однако, интерференционная картина сохраняется, даже если пушка испускает электроны не непрерывным потоком, а по одному через некоторый интервал времени. Дело выглядит так, будто один электрон проходит через две щели и после них взаимодействует сам с собой, формируя интерференционную картину.

В случае одиночных электронов немного изменим эксперимент: поставим детектор D1 возле одной из щелей и попытаемся определить через какую щель пролетел электрон. Результат резко изменился, см. Рис.9: интерференционная картина пропала, исчезли волновые свойства и электроны начали вести себя как частицы.



Рис. 9.

Результаты приведённого эксперимента свидетельствуют о том, что свойства квантового объекта электрона зависят от процесса измерения: если мы ставим эксперимент для определения волновых свойств, то электрон ведёт себя как волна, если мы измеряем характеристики типичные для частицы, то электрон ведёт себя как частица.

Таким образом, в квантовой физике, в отличие от классической, мы не можем определить в каком состоянии находилась система до процесса измерения. В процессе измерения мы получаем новую систему, созданную под воздействием измерительного прибора. Если в классическом случае при корректном измерении мы находим бит в определённом состоянии, то с уверенностью можно утверждать, что бит находился в этом состоянии и до процесса измерения. В квантовом случае процесс измерения влияет на состояние системы, и если мы определим некоторое состояние квантового бита, то до измерения квантовый бит мог находиться в этом состоянии, а мог находиться в другом состоянии.

Точка зрения, заключающаяся в том, что до процесса измерения состояние квантовой системы не определено, а система с некоторой вероятностью может находиться в любом допустимом состоянии, в своё время вызвал большую дискуссии в научном мире между Альбертом Эйнштейном и Нильсом Бором. Н. Бор утверждал, что до процесса измерения состояние квантовой системы не определено, и процесс измерения с некоторой вероятностью создаёт одно из возможных состояний системы. Противоположная точка зрения, сформулированная и поддерживаемая А. Эйнштейном, заключается в том, что состояние системы определено всегда («Вы действительно считаете, что Луна существует только когда вы на неё смотрите?» – спросил А. Эйнштейн), но существуют некоторые скрытые параметры, которые не позволяют нам точно описать систему и влияние этих параметров устраняется в процессе измерения.

Большинство учёных приняли точку зрения Н. Бора (так называемая Копенгагенская интерпретация квантовой механики) однако полной ясности в этом вопросе нет до сих пор. Позиция Н. Бора в отношении основ квантовой механики более привлекательна, так как приводят к появлению новых свойств системы в квантовом случае: суперпозиция состояний, квантовые запутанные состояния и квантовый параллелизм. Эти свойства квантовых систем позволяют реализовывать алгоритмы для квантовых вычислительных систем принципиально отличные от алгоритмов классических вычислительных систем.

***1.2. Математическая модель квантовых вычислений***

Основой цифровых вычислительных систем является элементарная ячейка – бит.

В классической физике бит может быть представлен системой с двумя устойчивыми (или квазиустойчивыми) состояниями: триггер, конденсатор, магнитный домен, …. Одно из этих состояний условно обозначается «0», другое – «1». В математической модели бит может быть представлен как элемент множества {0,1}. В классической физике состояние бита не зависит от наблюдателя и определяется лишь предшествующими состояниями системы – вычислительным процессом, зависящим от реализуемого алгоритма.

В квантовой физике бит это квантовая система с двумя возможными состояниями: направление спина электрона в магнитном поле, поляризация фотона, энергетические уровни атома или молекулы, …. Линейная алгебра может служить основой при построении математической модели квантовой системы. В математической модели квантовая система с двумя состояниями описывается вектором в 2-мерном пространстве, этот вектор нормирован и называется вектором состояния. Если ввести базисные вектора и , то вектор состояния можно представить в виде . Здесь α, β – комплексные числа, называемые амплитудами состояния и =1. Потенциально, вектору соответствует бесконечное множество состояний системы задаваемых значениями амплитуд α, β. При попытке определить состояние посредством измерения, происходит взаимодействие квантовой системы с классическим прибором, и система переходит в одно из базисных состояний или с вероятностями и соответственно. Такая квантовая система, представляющая бит, называется квантовый бит или кубит – qbit. Кубит представляет булеву переменную в вычислительной системе. В дальнейшем, для базисных векторов пространства состояний кубита мы будем использовать обозначения и ассоциированные с состояниями «0» и «1» соответственно.

*Отметим ключевой для квантовых вычислений момент: кубит (квантовый бит) представляет не одно фиксированное значение («0» или «1» как в случае классической физики), а набор возможных значений, вероятность которых определяется амплитудами α и β вектора состояний.*

Изменение состояния кубита может быть выполнено посредством применения оператора *U*:

Из условий нормировки

и

следует, что оператор *U* – унитарный оператор (), так как

Оператор *U* задаёт изменение кубита в процессе квантовых вычислений в соответствии с заданным алгоритмом.

Состояние квантового компьютера, как и любой квантово-механической системы, описывается уравнением Шредингера [1]

где – оператор Гамильтона.

Р. Фейнман доказал [2], что изменение вектора состояния квантовой системы

может быть связано с уравнением Шредингера и показал каким образом по заданному оператору *U* можно построить гамильтониан физической системы. Тем самым он доказал принципиальную возможность физической реализации процесса квантовых вычислений.



Рис 10.

Общую схему процесса квантовых вычислений условно можно представить в следующем виде Рис10:

* A – инициализация квантового регистра: установка начальных значений кубитов. Например, в качестве начального значения регистра все кубиты устанавливаются в состояние , затем применяется преобразование Адамара (определение этого преобразования будет приведено ниже), в результате чего система переходит в равновероятную смесь возможных состояний.
* - последовательность преобразований составляющих алгоритм квантового вычислительного процесса. Унитарные операторы действуют на вектор состояния квантового регистра и переводят его в новое состояние, шаг за шагом приближаясь к решению задачи. При реализации квантового компьютера «в железе» каждому оператору может соответствовать отдельное физическое устройство или некое универсальное устройство типа программируемого контроллера.
* B – определение результата вычислений. Так как в соответствии с общими принципами квантовой физики вектор состояния регистра не определён, то необходимо зафиксировать (с некоторой вероятностью) результат вычислений путём измерения состояния регистра.

Для представления n булевых переменных используется регистр (система) n кубитов. Регистру n кубитов соответствует вектор состояния в -мерном векторном пространстве:

Здесь и далее знак обозначает операцию тензорного произведения.

Регистр это не просто набор n независимых кубитов, а единая квантово-механическая система и вектор состояния описывает систему n кубитов как целое. Состояние регистра и состояния отдельных кубитов до процесса измерения не определено. Измерение состояния регистра не всегда означает измерение состояний образующих его кубитов (см. далее о «запутанных» состояниях).

При рассмотрении квантовых алгоритмов и моделировании процесса квантовых вычислений мы будем представлять вектора в *p*-мерном пространстве состояний в виде совокупности *p* компонент:

Здесь и далее , *n* – число кубитов образующих регистр.

Если в качестве базисных состояний регистра мы выберем вектора

, |, …,

то компоненты вектора будут амплитудами состояний: вероятность обнаружить регистр в состоянии () будет равна , и .

Свяжем базисные вектора регистра с базисными векторами образующих его кубитов следующим образом:

, |, …,

|1, |1

т. е., если последовательность нулей и единиц, представляющих базисные состояния кубитов в регистре рассматривать как число в двоичном представлении, то это число является номером единственной ненулевой (единичной) компоненты базисного вектора в -мерном пространстве.

Для системы произвольных n кубитов

, 1

с базисными состояниями вектор состояния регистра определяется следующими рекуррентными соотношениями (верхний индекс в скобках обозначает число кубитов образующих регистр):

. . .

Или для компонент вектора состояния:

В квантовой механике любая наблюдаемая величина (энергия, импульс, поляризация, …) описывается эрмитовым оператором: . Если и – собственные значения и собственные вектора оператора *E*:

то измерение физической величины, соответствующей оператору E эквивалентно использованию собственных векторов в качестве базисных в пространстве состояний системы:

И в результате измерения мы будем получать одно из собственных значений с вероятностью , где – амплитуды состояний, проекции вектора состояния системы на базисные вектора.

Вектор, используемый в математической модели для объекта, описывающего состояние квантовой системы, не вполне соответствует математическому вектору. В математике вектор это набор компонент, которые изменяются определённым образом при преобразовании системы координат. В квантовой физике преобразование координат пространства состояний это новый процесс измерения. Процесс измерения переводит систему из неопределённого состояния в состояние с фиксированными характеристиками. Последующие наблюдения при использовании заданного метода измерений будут давать тот же самый результат. Однако, изменение метода измерения соответствует изменению системы координат в пространстве состояний, не преобразует координаты вектора как в математике, а переводит систему в неопределённое состояние относительно нового процесса измерения. Новый процесс измерения снова фиксирует одно из возможных состояний системы с некоторой вероятностью.

***1.3. Квантовая запутанность и квантовый параллелизм***

Следует отметить, что в силу квантовых свойств системы регистр это не просто набор n независимых кубитов как это было бы в классическом случае, а связанная квантово-механическая система. Также как при измерении характеристик регистра формируется его состояние, также при измерении характеристик некоторых кубитов формируется состояние остальных кубитов входящих в состав регистра.

В математической модели регистр представлен как тензорное произведение образующих его кубитов:

Например, для регистра двух кубитов с базисными состояниями

и

базисными состояниями регистра будут

Такое математическое представление вполне адекватно и объяснимо в классическом случае: имеется однозначное соответствие между базисными состояниями кубитов и регистра. В квантовом случае ситуация осложняется тем, что кубиты не различимы как физические объекты. Состояния регистров и отличаются друг от друга и могут быть зафиксированы в процессе измерения. Однако комбинации кубитов и в квантовой механике неразличимы и при измерении состояния одного из кубитов регистра невозможно определить какой кубит мы измеряем первый или второй. Таким образом, задача определения состояний кубитов по состоянию регистра не всегда может быть решена. Подобное поведение носит название «квантовая запутанность» и является одним из принципиальных отличий в поведении квантовых и классических систем.

Предположим, что регистр из двух кубитов находится в состоянии «1». Возможными состояниями кубитов в этом случае будут «0», «1» и «1», «0». Если измерение первого кубита фиксирует его в состояние «0», то состояние второго кубита будет зафиксировано в «1». Подобное поведение не связано с физическим взаимодействием кубитов и не зависит от расстояния между ними. После создания регистра и измерения его характеристик кубиты можно разнести сколь угодно далеко и тем не менее измерение характеристик одного кубита будет влиять на характеристики другого. Такое «дальнодействие» квантовых объектов изначально составляющих единую систему вызвало критику А. Эйнштейна квантовой механики в трактовке Н. Бора.

Ещё одним особым свойством квантово-механических систем является квантовый параллелизм. Изменение состояния квантовой системы может быть представлено через действие унитарного оператора *U*:

При этом, если начальное состояние системы не определено – система находится во всех потенциально возможных состояниях, то действие оператора распространяется на все состояния системы и в результате получаем набор потенциально возможных конечных состояний . Предположим, что мы используем квантовый компьютер для вычисления значений булевой функции n переменных: : . Возможные значения аргументов функции представим в виде регистра n кубитов, нахождение значений функции через действие соответствующего оператора. Так как без процесса измерения начальное состояние системы не определено, и система потенциально находится во всех возможных состояниях, то вычисление значений функции (действие оператора) выполняется одновременно для всех возможных значений аргумента. Таким образом, неопределённость состояния квантовой системы позволяет одновременно выполнять вычисления всех возможных значений функции одновременно. Это свойство носит название «квантовый параллелизм». На первый взгляд «квантовый параллелизм» бесполезен, так как измерение фиксирует лишь одно состояние системы, а остальные состояния при этом теряются и для получения других значений функции потребуется провести новые вычисления и измерения для других значений аргументов. Свойство параллелизма может быть полезно если в процессе квантовых вычислений нас интересуют не значения функции, а её свойства. К примеру, это может быть решение уравнения как в алгоритме Гровера или класс функции (константа или сбалансированная функция) как в алгоритме Дойча-Джоза.

***1.4. Алгоритм Дойча***

В качестве первого примера квантовых вычислений рассмотрим один из простейших квантовых алгоритмов – алгоритм Дойча. На примере этого алгоритма можно продемонстрировать специфическую особенность квантовых вычислений – квантовый параллелизм.

Возьмём булеву функцию одной переменной . Существует всего четыре таких функции:

1.

2.

3. : ,

4. : ,

Первые два случая соответствуют функции константе. Случаи 3 и 4 – не константа. Чтобы определить к какому типу (константа или не константа) принадлежит функция, при использовании классической системы вычислений потребуется выполнить два запроса *f(0)* и *f(1)* на вычисление её значения. Квантовый алгоритм Дойча позволяет определить тип функции через один запрос.

Возьмём систему из одного кубита в базисном состоянии, соответствующем логическому значению «0». Получить такое состояние довольно просто: берём кубит в произвольном состоянии, которое является суперпозицией базисных и производим измерение в выбранном базисе. Если результат измерения «0», то переходим к следующему шагу, если нет – берём другой кубит и повторяем процесс измерения. Отобранный кубит с вектором состояния используем для дальнейших вычислений.

Как указывалось ранее, для базисных состояний кубита мы будем полагать:

К исходному кубиту в состоянии применим оператор Адамара H:

В результате преобразования Адамара мы получили кубит в состоянии

- равновероятной смеси базисных состояний и . Этот этап вычислений можно отнести к «приготовлению» начального состояния системы.

Теперь, для нахождения значений функции на возможных значениях аргумента, применим оператор фазового запроса :

Отметим, что оператор фазового запроса и оператор Адамара являются унитарными операторами.

И, в заключение, к полученному результату применим ещё раз преобразование Адамара:

Таким образом, цепочка вычислений

→ → →

приводит к результату

где

В результате последовательного применения преобразования Адамара, фазового запроса и ещё одного преобразования Адамара к кубиту в состоянии , мы получаем суперпозицию базисных состояний и с амплитудами α и β.

Обратим внимание на тот факт, что использование одного запроса к системе – применение оператора фазового запроса , позволило вычислить функцию для двух значений аргумента.

Для получения результата квантовых вычислений проведём измерение состояния кубита с конечным вектором состояния . Если искомая функция является константой: или , то в этом случае , и измерение определит состояние с вероятностью 100%. Если функция не константа: , или , , то измерения мы всегда найдём кубит в состоянии .

***1.5. Система кубитов. Алгоритм Дойча-Джозы***

Алгоритм Дойча-Джозы (Deutsch–Jozsa), иногда упоминается как алгоритм Дойча-Йожи, является обобщением алгоритма Дойча на случай функции n переменных:

Как и в алгоритме Дойча будем считать, что является либо функцией константа, и её значение равно «0» или «1» для всех вариантов возможных значений аргументов, или сбалансированной функцией: в половине случаев она принимает значение «0» и принимает значение «1» – в другой половине случаев. Другие варианты распределения значений функции среди комбинаций значений «0» и «1» её аргументов не рассматриваются. Алгоритм Дойча-Джоза позволяет посредством одного запроса к квантовой системе определить, к какому типу (константа или сбалансированная) принадлежит искомая функция.

Возможный набор значений аргументов функции будем хранить в регистре из n кубитов. В качестве базисных векторов пространства состояний регистра мы будем использовать базисные вектора … , введённые ранее в разделе «Математическая модель квантовых вычислений».

Начальное состояние системы зададим в виде регистра n кубитов в состоянии "0":

Для приготовления равновероятной смеси возможных состояний регистра применим n-мерное преобразование Адамара:

Здесь символ *H* обозначает оператор Адамара для двумерного случая

- оператор Адамара для n-мерного случая:

Далее, применим преобразование n-мерный фазовый запрос:

Оператор фазового запроса имеет вид:

И ещё раз преобразование Адамара:

Проведя вычисления аналогичные тем, которые мы проводили в предыдущем разделе для алгоритма Дойча, мы придём к результату для конечного состояния системы:

При этом, амплитуда α принимает значение ±1 в том случае, если функция является константой, и α=0 для сбалансированной функции. Теперь, проведя измерение состояния регистра после вычислительного процесса мы можем определить тип функции: если мы обнаружим регистр в состоянии , то это означает функцию – константу, любое другое состояние отличное от соответствует сбалансированной функции.

Следует отметить, что алгоритм Дойча-Джоза позволяет определить тип функции лишь в том случае, если функция может быть только константой или сбалансированной. Если возможен третий тип функции, то результат алгоритма не будет столь однозначен.

В классическом случае нам понадобилось бы раз вычислять значение функции для разных значений аргументов чтобы определить является ли функция константой или сбалансированной. Использование алгоритма квантовых вычислений позволяет получить ответ при однократном применении оператора фазового запроса. Как и в предыдущем примере алгоритма Дойча использование квантового параллелизма позволяет изменить и упростить алгоритм решения задачи.

***1.6. Алгоритм Гровера***

Рассмотренные алгоритмы Дойча и Дойча-Джоза давали определённый ответ о классе исследуемой функции. Это было связано с тем, что амплитуды конечного вектора состояния системы кубитов имели значения ±1 или 0, что приводило к вероятности 100% обнаружения системы в одном из состояний. Не все квантовые алгоритмы могут давать столь определённый результат, в общем случае вероятность получить правильный результат равна – квадрату амплитуды базисного вектора состояния, соответствующего решению задачи. Примером такого квантового алгоритма является алгоритм Гровера.

Алгоритм Гровера это квантовый алгоритм решения уравнения , где булева функция n переменных.

Как и в предыдущих примерах алгоритмов начнём с системы нулевых n кубитов: . Затем применим оператор Адамара и приготовим равновероятную смесь состояний:

Несколько раз применим унитарный оператор *G*:

где: , – введённый ранее оператор фазового запроса, *R* – оператор диффузии:

В заключении ещё раз оператор Адамара:

В итоге получаем вектор состояния:

Вектор состояния регистра n кубитов будет находиться в суперпозиции базисных состояний:

Среди набора базисных векторов один вектор соответствует решению исходного уравнения: *.* Алгоритм Гровера обеспечивает наибольшее значение для амплитуды, соответствующей вектору решения относительно других амплитуд. Величина не обязательно должна быть равна 1, при измерении мы получим результат, который будет решением уравнения с вероятностью . Можно показать, что максимальная вероятность получить правильный ответ достигается если *k* – число применений оператора равно:

(квадратные скобки обозначают целую часть числа). Вероятность получить неправильный результат в алгоритме Гровера оценивается как .

В приложении «Эмуляция квантовых вычислений» будет рассмотрен пример четырёх кубитов *n*=4 для которых решением уравнения является комбинация , т.е. . В этом случае оптимальное число итераций *k*=3 и полученные значения амплитуд равны

(0.05 0.98 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05).

Вероятность получать правильное решение в этом случае составляет .

***11. Приложение B. Эмуляция квантовых вычислений.***

В этом разделе мы рассмотрим реализацию программы эмуляции квантовых вычислений на обычном классическом компьютере. Эмуляция будет основываться на математической модели квантовых вычислений, рассмотренной ранее в разделе «Математическая модель квантовых вычислений». В этой модели вектора состояний и операторы преобразования состояний представлены векторами и унитарными матрицами в пространстве соответствующей размерности, а измерительный прибор и процесс измерения состояния квантовой системы – набором базисных векторов пространства состояний и проекциями вектора состояния системы на базисные вектора. Всё упомянутые объекты и перечисленные операции могут быть реализованы в рамках классического подхода к вычислениям. Подобная эмуляция квантовых вычислений будет полезна для более глубокого понимания предмета, изучения и разбора квантовых алгоритмов.

Для реализации программной эмуляции квантовых вычислений мы будем использовать объектно-ориентированное программирование и язык программирования C++. Для операций с комплексными числами мы будем использовать реализацию класса комплексных чисел, рассмотренную ранее в разделе «Объектно-ориентированное программирование».

В основе системы квантовых вычислений лежит квантовый бит – кубит, двухкомпонентный вектор состояния квантовой системы . В общем случае , где *α, β* – комплексные числа, удовлетворяющие условию ; и – базисные вектора, соответствующие двум устойчивым состояниям системы, условно обозначаемым «0» и «1».

Сопоставим квантовому биту класс:

class QBIT {

public:

COMPLEX a;

COMPLEX b;

QBIT (COMPLEX A, COMPLEX B);

void print() {printf("(%5.2f %5.2f)",a.re,b.re);}

};

Данными этого класса будут два комплексных числа, соответствующие амплитудам базисных состояний *α* и *β.* Кроме конструктора класса QBIT мы используем метод print для вывода на экран текущих значений амплитуд. Для наблюдения за изменениями состояния бита в процессе эмуляции квантовых вычислений нам будет достаточно следить за значения вещественных частей амплитуд *α* и *β*.

Определим конструктор класса QBIT:

QBIT::QBIT (COMPLEX A, COMPLEX B) {

double norm = sqrt (A.mod2() + B.mod2());

if ( norm == 0.0 ) {

a.re = M\_SQRT1\_2; a.im = 0.0;

b.re = M\_SQRT1\_2; b.im = 0.0;

}

else {

a = COMPLEX(1.0/norm,0.0) \* A;

b = COMPLEX(1.0/norm,0.0) \* B;

}

}

Так как амплитуды состояния нормированы: , то при создании объекта класса конструктором производится нормировка двух комплексных аргументов с целью использования их в качестве амплитуд состояния.

Теперь мы можем определить объекты e0 и e1 соответствующие векторам базисных состояний

и

QBIT e0(COMPLEX(1,0),COMPLEX(0,0));

QBIT e1(COMPLEX(0,0),COMPLEX(1,0));

Используя класс QBIT построим класс QBITN – регистр N кубитов. Так как число кубитов образующих регистр не фиксировано и может меняться от задачи к задаче, то при реализации класса QBITN используем средство template (шаблон) языка C++:

template <int N> class QBITN {

COMPLEX s[1<<N];

public:

QBITN (QBIT V[N]); // регистр из N кубитов

void gH (); // преобразование Адамара

void Of (); // Фазовый поворот

void R (); // Диффузия

void print ();

};

Параметр класса N определяет число кубитов в регистре. Поле s в классе QBITN – массив размерностью , предназначен для хранения компонент вектора или амплитуд вектора состояния регистра. Методы класса: print – вывод компонент вектора на экран, gH – оператор Адамара, Of – оператор фазового поворота, R – оператор диффузии.

Конструктор класса QBITN, строит регистр из N кубитов задаваемых в виде массива V[N]:

template <int N> QBITN<N>::QBITN (QBIT V[N]) {

int P = 1<<N;

int i, j, k, n;

for (i=0; i<P; i++) {

s[i] = COMPLEX (1.0, 0.0);

}

for (k=2, n=0; k<=P; k\*=2, n++) {

for (i=0; i<P/k; i++) {

for (j=0; j<k/2; j++) {

s[i + 2\*j\*P/k] = s[i + 2\*j\*P/k] \* V[n].a;

s[i+P/k + 2\*j\*P/k] = s[i+P/k + 2\*j\*P/k] \* V[n].b;

}

}

}

}

При реализации конструктора класса QBITN были использованы рекуррентные соотношения, связывающие компоненты векторов состояний отдельных кубитов и регистров кубитов:

Метод print:

template <int N> void QBITN<N>::print () {

int p = 1<<N;

int i;

printf("( ");

for (i=0; i<p; i++) {

printf("%5.2f ", s[i].re);

}

printf(")\n");

};

Метод print используется для просмотра амплитуд вектора состояния регистра. Отметим ещё раз: просматривать значения амплитуд возможно лишь при эмуляции квантовых вычислений на классическом компьютере. При выполнении реальных квантовых вычислений результат можно получить лишь в виде вероятности появления того или иного базисного состояния.

Методы унитарных преобразований системы кубитов.

Преобразование Адамара:

template <int N> void QBITN<N>::gH () {

double U[1<<N][1<<N];

int p, q, r, x;

int i;

COMPLEX s1[1<<N];

for (p=0; p<(1<<N); p++) {

for (q=0; q<(1<<N); q++) {

r = p&q;

x = 0;

for (i=0; i<N; i++) {

x = x + (r&1);

r >>= 1;

}

U[p][q] = (x&1 ? -1 : 1) \* pow (M\_SQRT1\_2, N);

}

}

for (p=0; p<(1<<N); p++) {

s1[p] = s[p];

}

for (p=0; p<(1<<N); p++) {

s[p] = COMPLEX (0,0);

for (q=0; q<(1<<N); q++) {

s[p] = s[p] + COMPLEX(U[p][q],0.0) \* s1[q];

}

}

}

Преобразование диффузии:

template <int N> void QBITN<N>::R () {

COMPLEX z[1<<N];

int i, j;

for (i=0; i<(1<<N); i++) {

z[i] = s[i];

s[i] = COMPLEX(-1.0, 0.0) \* z[i];

}

for (i=0; i<(1<<N); i++) {

for (j=0; j<(1<<N); j++) {

s[i]=s[i]+COMPLEX(1.0/(1<<(N-1)),0.0)\*z[j];

}

}

}

Фазовый запрос:

template <int N> void QBITN<N>::Of () {

// для примера Дойча-Джоза

int i;

for (i=0; i<(1<<N)/2; i++)

s[i] = COMPLEX(-1.0,0.0)\*s[i];

// для примера Гровера

// s[1] = COMPLEX(-1.0,0.0)\*s[1];

}

Матрица, соответствующая оператору фазового запроса, является диагональной:

Действие оператора заключается в изменении знака амплитуды состояния в зависимости от значения функции для соответствующего аргумента. Если значение функции на компонентах вектора равно 1, то амплитуда меняет знак и сохраняет знак в противном случае. В приводимом примере оператора фазового запроса используется функция, у которой половина значений — 1, а другая половина — 0.

Используя введённые классы QBIT и QBITN выполним вычисления, соответствующие алгоритмам Дойча-Джоза и Гровера. Для определённости будем использовать систему 4-х кубитов (соответствующий вектор состояния будет иметь 16 компонент). Значение параметра класса будет N=4.

Алгоритм Дойча-Джоза:

В качестве исходного состояния системы возьмём 4 кубита в состоянии :

QBIT v[4] = {e0, e0, e0, e0};

QBITN <4> q (v);

Применяя преобразование Адамара приготовим равновероятную смесь состояний:

q.gH();

затем преобразование фазового запроса:

q.Of();

и ещё раз преобразование Адамара:

q.gH();

В случае константной функции или независящей от значений аргументов на выходе получаем вектор состояния

(1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00)

или

(-1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00)

Этот вектор состояния с вероятностью 100% свидетельствует о том, что в операторе фазового запроса использована константная функция.

При использовании сбалансированной функции вектор состояния на выходе алгоритма будет иметь вид:

(0.00 x.xx … x.xx)

Величины x.xx зависят от того, как значения функции связаны с конкретными значениями её аргументов. Так, например, для сбалансированной функции которая принимает значение 1 для аргументов

0,0,0,0; 0,0,0,1; 0,0,1,0; 0,0,1,1; 0,1,0,0; 0,1,0,1; 0,1,1,0; 0,1,1,1;

и значение 0 для аргументов

1,0,0,0; 1,0,0,1; 1,0,1,0; 1,0,1,1; 1,1,0,0; 1,1,0,1; 1,1,1,0; 1,1,1,1;

результат вычислений по алгоритму Дойча-Джоза даст для вектора состояния следующий результат:

(0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 -1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00)

Этот вектор состояния с вероятностью 100% свидетельствует о том, что исследуется сбалансированная функция.

Алгоритм Гровера:

Для определённости возьмём функцию с единственным ненулевым значением при . Этому случаю соответствует вектор состояния:

Это приводит к следующему оператору фазового запроса:

template <int N>

void QBITN<N>::Of () {

s[1] = COMPLEX(-1.0,0.0)\*s[1];

}

Как и в предыдущем случае приготовим «смешанное» начальное состояние:

QBIT v[4] = {e0, e0, e0, e0};

QBITN <4> q (v);

q.gH();

Затем применим несколько раз преобразования фазового запроса и диффузии:

for (i=0; i<iterations; i++) {

q.Of();

q.R();

}

Для системы 4-х кубитов число оптимальное число итераций будет равно:

Проследим, как меняются амплитуды состояния системы после каждой итерации.

начальное состояние:

(0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25)

1-я итерация:

(0.19 0.69 0.19 0.19 0.19 0.19 0.19 0.19 0.19 0.19 0.19 0.19 0.19 0.19 0.19 0.19)

2-я итерация:

(0.08 0.95 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08)

3-я итерация (лучший результат):

(0.05 0.98 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05)

4-я итерация:

(-0.16 0.76 -0.16 -0.16 -0.16 -0.16 -0.16 -0.16 -0.16 -0.16 -0.16 -0.16 -0.16 -0.16 -0.16 -0.16)

Из приведённых результатов видно, наилучшее значение для «правильной» амплитуды получается после 3-й итерации и равно 0.98. В этом случае вероятность получить правильный результат составляет .

В настоящий момент квантовые вычисления, основанные на использовании квантовых свойств элементной базы вычислительных систем, находятся в начальной стадии развития. Дальнейший прогресс в этой области будет зависеть от решения технических задачи, связанных с созданием элементной базы квантовых компьютеров. Кроме этого, потребуется разработка универсальных или многофункциональных физических устройств, позволяющих реализовать действия, соответствующие различным унитарным операторам.

**13. Литература**

1. Ландау Л.Д., Лифшиц Е.М. «Квантовая механика (нерелятивистская теория)», — Издание 6-е, исправленное. — М.: Физматлит, 2004. («Теоретическая физика», том III).
2. Р. Фейнман «Квантовомеханические ЭВМ», УФН, 1986г., том 149, вып 4, стр. 671-688
3. М. Вялый «Квантовые алгоритмы: возможности и ограничения», http://www.lektorium.tv/course/?id=22805 (дата обращения 01.06.2019)