Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

Квантовые вычисления

Выполнил:

студент группы 382003-1

Ивлев А.Д.

Научный руководитель:

Линёв А.В.

Нижний Новгород

2022

Содержание

Введение

Квантовый компьютер - вычислительное устройство, которое использует для вычислений различные свойства квантовых состояний, например суперпозицию или квантовую запутанность.

Современные квантовые компьютеры пока что достаточно малы и далеки от совершенства, чтобы превзойти классические компьютеры на практике, но из-за своих свойств они способны решать определенные вычислительные задачи (например, целочисленная факторизация, которая лежит в основе шифрования RSA) асимптотически быстрее, чем классические компьютеры, часто сводя экспоненциальную сложность к полиномиальной.

Разработка квантовых алгоритмов для таких задач ведётся параллельно разработкам по созданию и совершенствованию квантовых компьютеров. Но для проверки работоспособности данных алгоритмов пока недостаточно доступных квантовых компьютеров по нескольким причинам: малое количество кубитов (квантовых битов), топология их связей представляет собой не полный граф, достаточная большая погрешность при вычислениях. Конечно, над решением этих проблем ведутся работы, например в современных моделях имеется некоторый максимально допустимый относительный размер ошибки, до которого погрешности корректируются.

Постановка задач

Модель идеального квантового компьютера представляет собой систему кубитов, где каждый из них связан со всеми другим и все гейты (квантовые вентили) выполняются без ошибок. Именно такая модель лучше всего подходит для первоначальной проверки квантовых, алгоритмов. Второй этап тестирования, алгоритма представляет собой проверку его устойчивости к ошибкам, возникающим во время вычислений. На третьем этапе алгоритм нужно проверить на работоспособность при топологии связей кубитов приближенной к реальной.

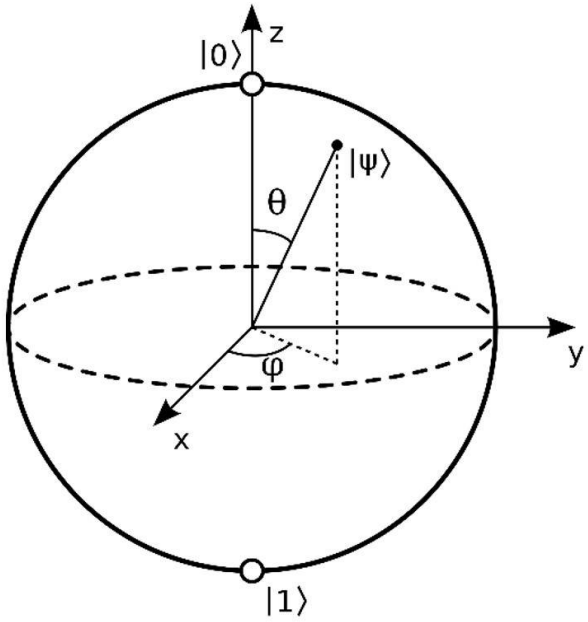
Задача состоит в создании на обычном компьютере модели квантового компьютера, на которой возможно составлять из базовых гейтов квантовые алгоритмы и проверять их работоспособность при наличии ошибок выполнения базовых гейтов. В данной работе за генерацию ошибок будет отвечать один из самых часто используемых гейтов, гейт фазы (gate P - гейт фазового сдвига), который будет описан ниже.

Также будут продемонстрированы и описаны некоторые квантовые алгоритмы, а классическое представление квантовой схемы алгоритма Шора будет протестировано на устойчивость к ошибкам.

Кубит и однокубитные квантовые гейты

Кубит - наименьшая единица измерения количества информации в квантовом компьютере. В отличии от классического бита может находится в базисных состояниях $|0\rangle$и $|1\rangle$, а также их суперпозиции $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, где $\alpha, \beta \in \mathbb{C}, |\alpha|^2 + |\beta|^2 = 1$, то есть состояние кубита описывается нормированным вектором $\binom{\alpha}{\beta}$. Условие нормировки должно сохраняться при всех действиях с кубитом.

Также исходное состояние кубита может быть эквивалентным образом представлено с помощью всего лишь двух вещественных параметров — углов φ и θ: $|\phi\rangle = cos\frac{\theta}{2}|0\rangle + e^{i\phi}sin\frac{\theta}{2}|1\rangle$. Таким образом можно отобразить состояние кубита на сфере Блоха.



Базовая операция над кубитом это его измерение, но невозможно измерить состояние кубита, не повлияв на него (особенность квантовой механики, не имеющая аналогов в классической физике). Измеряя кубит мы переводим его в одно из базисных состояний $|0\rangle$или $|1\rangle$ с вероятностями $|\alpha|^2 = |\langle 0|\phi\rangle|^2, |\beta|^2 = |\langle 1|\phi\rangle|^2$соответственно.

Любая логическая операция с кубитами называется гейтом (от английского gate – ворота). Они (по аналогии с классическими логическими

элементами) являются базовыми блоками для построения квантовых схем. По числу задействованных кубитов гейты делятся на однокубитные и многокубитные. Для демонстрации действия гейта на кубиты используют его матричную запись.

Применение произвольного однокубитного гейта u, который задаётся матрицей $u = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix}$, к произвольному кубиту $|\phi_0\rangle = \begin{pmatrix} \phi_{10} \\ \phi_{20} \end{pmatrix}$ можно описать $|\phi_1\rangle = u * |\phi_0\rangle$. При этом матрица, задающая гейт такая, что после его применения не нарушается условие нормировки. Далее кратко описаны основные однокубитные гейты.

Единичный гейт (Identity gate) I

Гейт I описывается единичной матрицей $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. Легко заметить, что при применении данного гейта состояние кубита не меняется и его реализация не имеет смысла, но он очень важен при математическом описании квантовых вычислений.

Гейты Паули (Pauli gates) X (NOT), Y, Z

Данные гейты задаются унитарными матрицами $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$. Можно заметить, что с помощью гейтов Паули можно отобразить поворот состояния кубита на сфере Блоха вокруг осей X, Y, Z соответственно.

Также гейт X называю гейтом NOT, потому что после применения данного гейта меняются местами весовые коэффициенты состояний $|0\rangle$ и $|1\rangle$. ($\alpha|0\rangle + \beta|1\rangle \to \beta|0\rangle + \alpha|1\rangle$)

Для кубита можно построить неограниченное число гейтов. Однако, в силу полноты системы состоящей из матриц Паули и единичной матрицы I, любая матрица 2 на 2 может быть разложена на комбинацию этих матриц. Поэтому для использования представляют интерес сами матрицы Паули и некоторые их специальные (часто использующиеся) комбинации, описанные ниже.

Гейт Адомара (Hadamard gate) H

Матрица данного гейта: $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ ($H = \frac{X+Z}{\sqrt{2}}$). На сфере Блоха данный гейт поворачивает состояние на π вокруг оси $\frac{(x+z)}{\sqrt{2}}$.

Гейт фазового сдвига (Phase shift gate) P

Матрица: $P = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$ ($P\left(\frac{\pi}{\psi}\right) = \sqrt[\psi]{Z}$, $P(0) = I$). Данный гейт не изменяет вероятности $|0\rangle$и $|1\rangle$, но изменяет состояние кубита. Данное изменение можно описать на сфере Блоха, как поворот на φ радиан в плоскости XY.

Система кубитов и многокубитные квантовые гейты

Набор из n кубитов составляет систему кубитов (квантовый регистр). Важно, что этот регистр также подчиняется принципу суперпозиции и находится одновременно сразу во всех своих базовых классических состояниях, число которых равно 2^n. Произвольное состояние регистра записывается в виде: $|\phi\rangle = \sum_{i=1}^{2^n} a_i * |i\rangle$, $a_i \in \mathbb{C}$ (в правой части выражения i записывается в двоичном виде). Для системы также должно выполняться условие нормировки. Значит состояние системы можно описать нормированным вектором, состоящим из a_i.

Чтобы объединить состояния двух квантовых регистров нужно взять их тензорное произведение $|\phi\psi\rangle = |\phi\rangle \otimes |\psi\rangle$. Но не всегда можно выделить состояния отдельных кубитов (групп кубитов) из системы обратив эту процедуру. Говорят, что кубиты сцеплены или запутаны (entangled), если их совместное состояние невозможно разложить в произведение индивидуальных состояний. Например, состояние Белла, которое можно задать с помощью гейтов H и CNOT (подробнее ниже).

Применение однокубитного гейта U к произвольному j кубиту в системе из n кубитов можно описать матрицей, полученной из тензорного произведения: $I \otimes \ldots \otimes I \otimes U \otimes I \otimes \ldots \otimes I$, где U стоит на j месте, а единичных матриц n-1. Также можно объединять применение нескольких однокубитных гейтов к разным кубитам в системе в одну матрицу преобразования, например, $X \otimes H$, где к 0 кубиту мы применили X, а к 1 H.

Основными многокубитными гейтами являются контролирующие (управляющие) гейты, которые применяют однокубитный гейт U к j кубиту,

если все контролирующие кубиты находятся в состоянии $|1\rangle$. Например, для 2-х кубитной системы матрица преобразования, где 1 управляющий гейт, а 0 управляемый можно записать в виде: $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{11} & u_{12} \\ 0 & 0 & u_{21} & u_{22} \end{pmatrix}$, если же наоборот, то $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & u_{11} & 0 & u_{12} \\ 0 & 0 & 1 & 0 \\ 0 & u_{21} & 0 & u_{22} \end{pmatrix}$. В первом случае состояния $|00\rangle$и $|01\rangle$останутся без изменений, а к паре состояний $|10\rangle$и$|11\rangle$применится гейт U. Во втором $|00\rangle$и $|10\rangle$ без изменений, к $|01\rangle$ и $|11\rangle$ применятся данный гейт. По аналогии строятся матрицы преобразований для систем большей размерности и большего количества контролирующих гейтов. Далее покажем основные гейты с контролем CNOT и CCNOT.

Гейт контролируемого отрицания CNOT (XOR)

Данный двухкубитный гейт применяет гейт X (NOT), если управляющий гейт находится в состоянии $|1\rangle$. Легко по описанным выше правилам построить матрицу данного преобразования, где 0 кубит управляемый, а 1 управляющий: $$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Если внимательней посмотреть на результат применения данного гейта, то можно заметить, что он эквивалентен логической операции XOR над 2 кубитами, где результат записывается в управляемый кубит.

Гейт дважды контролируемого отрицания CCNOT (Toffoli gate)

Данный трёхкубитный гейт применяет гейт X (NOT), если управляющие гейты находится в состоянии $|1\rangle$. По анологии его матрица: $$CCNOT = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

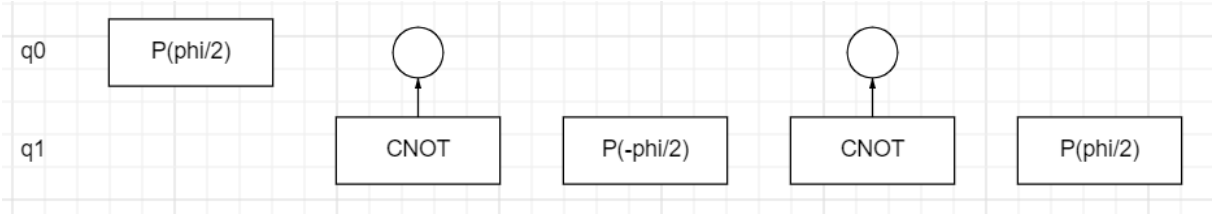
В общем случае, гейт n раз контролируемого отрицания задаёт операцию $u = u \oplus (c_1 \wedge \ldots \wedge c_n)$, где u - управляемый кубит, а c_i управляющие.

В теории дискретной математики доказано, что любую логическую операцию можно представить как совокупность некоторого числа стандартных, базовых операций, например системы AND, XOR и 1 (базис Жегалкина). В квантовых вычислениях аналогично можно любую обратимую унитарную операцию на кубитах можно представить как совокупность некоторых базовых операций. Базисом квантовой логики может служить один трехкубитный гейт (например CCNOT или CSWAP (гейт Фредкина)) или один однокубитный и один двухкубитный гейт (например, X (NOT) и CNOT (XOR)). Обычно рассматривают последний вариант базиса: X и CNOT.

Учитывая всё выше сказанное описанных в данной работе гейтов более чем достаточно, чтобы составлять любые квантовые схемы. И большинство из них представлены явно только потому что они часто используются при составлении, алгоритмов.

Другие гейты, которые будут использованы ниже можно составить из комбинации данных выше гейтов, например гейт контролируемого фазового сдвига $CP(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\phi} \end{pmatrix}$, где 0 кубит управляемый, а 1 управляющий можно представить, как последовательное применение гейтов P (0, $\frac{\varphi}{2}$), CNOT(0, 1), P (1, $\frac{-\varphi}{2}$), CNOT (0, 1), P (1, $\frac{\varphi}{2}$). Данное представление можно считать простым квантовым алгоритмом.

Любые квантовые алгоритмы принято представлять в виде квантовых схем, например представление гейта CP на схеме в общем вид:



Подробнее про алгоритмы и квантовые схемы их изображающие будет рассказано далее.

Реализация системы кубитов (квантового регистра) и основных гейтов. Общие принципы работы с системой.

Для моделирования системы кубитов нам необходимо хранить информацию о всех состояниях системы, то есть основу реализации будет составлять vector<complex<T>> длины 2^n, где T - тип с плавающей точкой, а n - число кубитов в системе. В данной реализации в i элементе вектора должен храниться весовой коэффициент состояния $|i\rangle$, где i представляется в двоичном виде, записанном справа налево, например для системы из 3-х кубитов 3-ий элемент хранит информацию о $|110\rangle$(в теории запись идёт слева на право), то есть младший кубит хранит информацию о младшем бите представления числа i.

Данный метод хранения приводит к экспоненциальному росту памяти, а также, применяя гейты (изменяя состояния системы), мы должны изменить состояния экспоненциально большого числа коэффициентов в векторе, что приводит к экспоненциальному росту числа операций, то есть нельзя моделировать квантовую вычислительную систему за полиномиальное время. Это не позволяет нам моделировать большие системы кубитов на домашних компьютерах, но для этого мы можем использовать суперкомпьютеры. Пока не существует известного способа эффективного моделирования квантового компьютера с помощью классического компьютера, но, что интересно, можно эффективно моделировать классический компьютер с помощью квантового компьютера.

В данной реализации при составлении квантовых алгоритмов подразумевается, что после задания начального состояния квантового регистра мы работаем с ним, только посредством гейтов, то есть перед работой с вектором пользователь должен, либо задать его весовые коэффициенты так, чтобы вектор был нормализован, либо после выставления коэффициентов вызвать для системы функцию normalization, которая нормализует вектор за него.

После задания начального состояния система готова, для применения к ней гейтов. Если мы вернёмся к теории, то для применения однокубитного гейта U к j кубиту нам необходимо умножить вектор состояний на матрицу преобразования, полученной из тензорного произведения: $I \otimes \dots \otimes I \otimes U \otimes I \otimes \dots \otimes I$, но если мы проделаем данные вычисления несколько раз, то заметим, что данное преобразование сводится к применению гейта U ко всем парам состояний, где отличается только состояние j-го кубита. Например, для системы из 3-х кубитов применение гейта U к 1 кубиту сведется к применению его к 4 парам состояний: $\begin{pmatrix} |000\rangle \\ |010\rangle \end{pmatrix}$

, $\begin{pmatrix} |100\rangle \\ |110\rangle \end{pmatrix}$, $\begin{pmatrix} |001\rangle \\ |011\rangle \end{pmatrix}$, $\begin{pmatrix} |101\rangle \\ |111\rangle \end{pmatrix}$. То есть нам нет необходимости проводить тензорные произведения и составлять матрицу преобразования достаточно, только находить данные пары и напрямую применять к ним данный гейт. Поскольку данные пары не пересекаются это позволяет нам провести парализацию применения однокубитных гейтов. Для парализации вычислений была выбрана библиотека openMP. Так, например, выглядит реализация гейта X (NOT):

```
    void X(size_t n) // (NOT)
    {
        int nbit = (1 << n);

#pragma omp parallel for
        for (int i = 0; i < data.size(); i++)
        {
            if (!(i & nbit))
            {
                swap(data[i], data[i + nbit]);
            }
        }
    }
```

//как явно находить r в алгоритме Шора

//связь квантовых регистров в КК