

2. Кляйн Кевин, Кляйн Дэниел, Хант Бренд SQL.Справочник, 3-е издание. – Пер. с англ. – СПб: Символ-Плюс, 2010. – 656с.
3. Хендерсон Кен Профессиональное руководство по Transact - SQL – СПб.: Питер,2005. - 558 с.

47.35.01.

ИСПОЛЬЗОВАНИЕ ЭМУЛЯТОРА ДЛЯ НАПИСАНИЯ КВАНТОВЫХ ПРОГРАММ НА ПЕРСОНАЛЬНОМ КОМПЬЮТЕРЕ

Автор: Солод Борис Евгеньевич, студент первого курса Филиала «Протвино» государственного университета «Дубна»

Научный руководитель: к.т.н., доцент кафедры Кульман Татьяна Николаевна

Аннотация

В работе рассматриваются квантовые компьютеры и квантовые вычисления. Кратко изложены основные понятия квантовых компьютеров, рассмотрены языки квантового программирования, эмуляторы квантовых вычислений. Реализован вариант программы для персонального компьютера (ПК) с помощью эмулятора квантового компьютера.

Annotation

The paper deals with quantum computing and quantum computers. The basics of quantum computing are briefly described, programming languages for quantum computer, emulators of quantum computing are considered. Implemented a version of the program for the quantum computer emulator.

Ключевые слова: Квантовый компьютер, кубит, условия существования кубита, языки квантового программирования, язык Q#, эмулятор квантового компьютера.

Keywords: *Quantum computer, qubit, condition of existence qubit, quantum programming languages, language Q#, emulator quantum computer*

Целью работы является изучение квантовых компьютеров и языков квантового программирования, установка эмулятора квантового компьютера на домашнем компьютере, создание программы на языке квантового программирования Q#.

Квантовый компьютер

Квантовый компьютер – вычислительное устройство, которое использует явления квантовой механики (квантовая суперпозиция, квантовая запутанность) для передачи и обработки данных. Не все операции можно выполнить с помощью квантового компьютера из-за ограничений квантовой физики.

Qubit (кубит) – квантовый бит. Классический бит, используемый в привычных всем компьютерах, может принимать значения либо "0", либо "1". У квантового бита в принципе нет определенного значения до того момента, как мы его измерили. Принято говорить, что он находится в определенном "состоянии". У кубита два состояния, $|0\rangle$ и $|1\rangle$ (эти скобки ученые договорились использовать, чтобы отличать состояния от обычных чисел 0 и 1). В общем случае кубит находится не в одном из этих основных состояний, а в некоторой их суперпозиции $\alpha|0\rangle + \beta|1\rangle$, которая и дает изображенную на рисунке 1 сферу.

То есть в момент измерения кубит может быть равен любому числу, удовлетворяющему условию $a^2 + b^2 = 1$. Например $0,64 + 0,36 = 1$.

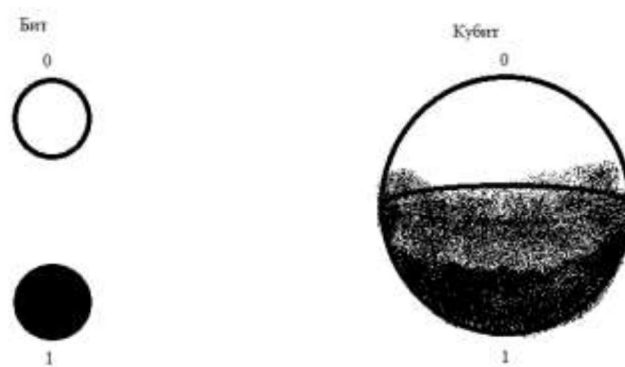


Рисунок 1 Изображение кубита

Одна из главных сложностей квантовых вычислений состоит в том, что для кубитов необходимы специфические условия. Первые квантовые компьютеры напоминают старые громоздкие вычислительные системы, они поставлялись в больших шкафах высотой в 3 м и объёмом около 20 куб. м. Размеры самого квантового чипа достаточно небольшие и сопоставимы с размерами ногтя большого пальца.

Большую часть остального пространства компьютера занимают системы охлаждения и экранирования. Они предназначены для создания необходимых условия функционирования компьютера и устранения внешних воздействий. Благодаря применению системы охлаждения на базе жидкого гелия, температура квантового чипа находится на уровне -273°C , иначе они могут исказить информацию и разрушаться.

Языки программирования для квантовых компьютеров

В настоящее время разработаны следующие языки для программирования для квантовых компьютеров:

- Q#
- Haskell
- Quantum Computation Language (QCL)
- Quipper

Большинство языков программирования для квантовых компьютеров состоит из синтеза теории квантовых вычислений и обычных языков программирования, это обеспечивает моделирование и полное выполнение квантовых алгоритмов.

Q#

Высокоуровневый язык программирования Q# нивелирует необходимость иметь глубокие знания в квантовой физике. Впервые был представлен в составе Quantum Development Kit. Стал доступен в конце 2017 года. Базовая возможность языка: создание и использование кубитов для алгоритмов.

Q# синтаксически похож на C#, с некоторыми отличиями:

- функции определяются словом `function`;
- операторы для работы над квантами определяются словом `operation`;
- отсутствуют многострочные комментарии;
- использование проверок вместо обработки ошибок;

Haskell

В Haskell есть так называемые ленивые (отложенные) вычисления. Суть их сводится к тому, что вычисления не производятся, пока они не будут необходимы. Haskell используют в Facebook для фильтров спама и он успешно справляется со своей задачей.

Quantum Computation Language (QCL)

QCL создан в 1998 году, он заимствует синтаксис C и Java. Квантовая система служит «оракулом», отвечающим на вопросы, которые могут быть заданы в формате, подходящем для вычислений кубитов.

Quipper

Quipper предназначен для тех же задач программирования, что и QCL, но имеет другую структуру и внешний вид. Язык реализован как расширение Haskell.

Существуют и другие языки квантового программирования.

Проект IBM Quantum Experience предоставляет возможность каждому запустить экспериментальную программу на реальном квантовом компьютере. Работа с языком программирования IBM похожа на процесс написания музыки с помощью приложения. Программист может просто перетащить квантовые объекты в определенную область, чтобы написать программу.

Эмулятор квантового компьютера

Эмулятор квантового компьютера – это компьютерная программа, которая позволяет эмулировать работу квантового компьютера на обычном персональном компьютере.

В работе использована программа Microsoft Visual Studio (VS) с инструментарием Quantum Development Kit для квантовых вычислений, написания программы на языке квантового программирования Q# и эмуляции работы квантового компьютера на ПК. Данный инструментарий был скачан с [сайта](#) и внедрен в VS.

Quantum Development Kit также содержит другие ресурсы, облегчающие разработку софта для квантовых компьютеров людям без знаний в квантовой физике.

Эмулятор, входящий в состав пакета средств квантовой разработки, позволяет эмулировать около 30 логических кубитов (при наличии 30 Гб оперативной памяти) квантовой вычислительной мощности, выполнять отладку квантового кода и тестировать программы на ПК. Кроме того, возможна эмуляция более 40 логических кубитов квантовой мощности в облаке Azure. Если кубитов более 30, то количество необходимой оперативной памяти растет экспоненциально.

Фрагмент программы квантовой телепортации представлен на рисунке 2. Программа позволяет увидеть явление квантовой телепортации кубитов. Сначала программа создает квантовую запутанность (связывает два числа). Затем, если изменяем один кубит, то изменяется второй кубит без операций над ним. Это еще раз дает возможность убедиться в том, что работают не математические законы, а законы квантовой физики. Результат выполнения программы предоставлен на рисунке 3.

```
operation bellTest (count : Int, initial: result) : (Int, Int)
{
    mutable numOnes = 0;
    using (qubit = qubit())
    {
        for (test in 1..count)
        {
            set (initial, qubit);

            let res = M (qubit);

            // Считаем количество увиденных:
            if (res == one)
            {
                set numOnes = numOnes + 1;
            }
        }
        Set(Zero, qubit);
    }

    // подсчитывает сколько раз как мы видели |0> и сколько видели |1>
    return (count-numOnes, numOnes);
}
```

Рисунок 2 Фрагмент программы квантовой телепортации

```
Init:Zero 0s=1000 1s=0
Init:One 0s=0 1s=1000
Press any key to continue...
```

Рисунок 3 Результат выполнения программы

Выводы

Результаты изучения квантовых компьютеров, эмуляторов квантовых компьютеров и языков квантового программирования открывают большие перспективы перед рядовыми пользователями персональных компьютеров. Это дает возможность понять структуру, принципы работы квантового компьютера, написать программу и интерпретировать результат.

Список использованных источников

1. Манин Ю. И. Вычислимое и невычислимое. – М.: Советское радио, 1980, – 128 с.
2. Манин Ю. И. Доказуемое и недоказуемое. – М.: Советское Радио, 1979, – 88 с.
3. А. Г. Грозин. Квантовый компьютер. ИЯФ Будкера СО РАН, 2004 – 24 с.
4. В. М. Соловьев, “Квантовые компьютеры и квантовые алгоритмы. Часть 2. Квантовые алгоритмы”, Изв. Саратов. ун-та. Нов. сер. Сер. Математика. Механика. Информатика, 16:1(2016), 104–112 с.
5. К.А.Валиев, А.А.Кокин. Квантовые компьютеры: надежды и реальность. Ижевск: РХД, 2001, 352 с.
6. Облачный компьютер. <https://quantumexperience.ng.bluemix.net/qx/editor> квантовый компьютер.
7. Инструментарий Quantum Development Kit. <https://www.microsoft.com/en-us/quantum/development-kit>

004.853

ПРИМЕНЕНИЕ НЕЙРОСЕТЕЙ ДЛЯ РАСПОЗНАВАНИЯ ТИПА СТРУЙ ЧАСТИЦ

Автор: Узунян Александр Сергеевич, студент 3 курса направления «Информатика и вычислительная техника» филиала «Протвино» государственного университета «Дубна»

Научный руководитель: Нурматова Елена Вячеславовна, к.т.н., доцент кафедры информационных технологий филиала «Протвино» государственного университета «Дубна»

Аннотация

Нейросети можно не объяснять правила игры в го — она сама их выучит.

Annotetion

Neural networks can not explain the rules of the game in go - she will learn them.

Ключевые слова: Сверточная нейронная сеть, машинное обучение, частицы.

Keywords: Convolutional neural network, machine learning, particles.

Проблема: В данной работе рассматривается вариант решения задачи построения собственного нейросетевого алгоритма классификации кварковых и