

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

СИМУЛЯТОР КВАНТОВЫХ ВЫЧИСЛЕНИЙ
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Яцулевича Владимира Владимировича

Научный руководитель
доцент, к. т. н.

В. М. Соловьев

Заведующий кафедрой
доцент, к. ф.-м. н.

С. В. Миронов

Саратов 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Ядро симулятора квантовых вычислений	4
2 Веб-приложение	7
ЗАКЛЮЧЕНИЕ	12

ВВЕДЕНИЕ

Целью данной дипломной работы является разработка полноценного симулятора квантовых вычислений. Для достижения данной цели были поставленные следующие задачи:

- разработать ядро симулятора квантовых вычислений;
- разработать веб-приложение с интерактивным конструктором квантовой схемы;
- разработать систему связи между ядром симулятора и веб-приложением.

В результате должно быть реализовано веб-приложение, с помощью которого можно будет конструировать различные квантовые схемы, формируя таким образом квантовые алгоритмы, которые можно будет запустить с использованием ядра симулятора квантовых вычислений. Симулятор будет написан на языке C#. Выбор в пользу данного языка был осуществлен по причине того, что итоговое приложение будет представлять из себя веб-сервис. А веб-сервис будет реализован с помощью технологии ASP.NET, в основе которой лежит язык C#.

Также стоит отметить, что в настоящее время в русскоязычном сегменте Интернета практически нет подобных сервисов. Поэтому реализация этого приложения является крайне актуальной. Помимо этого данный сервис в дальнейшем можно совершенствовать, добавляя в него, к примеру, элементы обучения квантовым вычислениям.

В данной работе представлено две главы: «Ядро симулятора квантовых вычислений» и «Веб-приложение». В первой главе описываются квантовые алгоритмы, необходимые для функционирования симулятора квантовых вычислений. Во второй главе описывается реализация веб-приложения, а также система связи между веб-приложением и ядром симулятора квантовых вычислений.

1 Ядро симулятора квантовых вычислений

Ядро симулятора квантовых вычислений представляет из себя набор библиотек, в которых реализованы такие составляющие квантовых вычислений, как кубит и квантовый регистр, квантовые операторы, квантовая схема, а также непосредственно симулятор, который собирает воедино блоки квантового регистра и схемы, а затем производит их запуск.

Кубит — минимальная единица квантовой информации. Ее особенность заключается в том, что данная единица информации обладает всеми необычными квантовыми свойствами. В частности, кубит может находиться в состоянии суперпозиции. Это означает, что кубит может находиться в двух состояниях одновременно. В программе кубит будет представлен вектором из двух комплексных чисел.

Для описания различных алгоритмов необходимо сформировать регистр кубитов. Регистр кубитов является более сложной структурой, по сравнению с регистром классических битов. Это объясняется тем, что несколько квантовых объектов могут находиться в состоянии квантовой запутанности. Это означает то, что при изменении состояния одного квантового объекта, состояние второго будет также изменено. В программе регистр кубитов представляет из вектор, полученный в результате тензорного произведения векторов состояния кубитов, входящих в этот регистр.

Квантовый алгоритм будет получать на вход последовательность кубитов, обрабатывать их, и на выходе выдавать последовательность кубитов. Последовательность кубитов будет записана в квантовом регистре. А обработка кубитов будет осуществляться с помощью применения квантовых операторов. Квантовые операторы — это унитарные линейные преобразования гильбертова пространства, которое описывает состояние кубита. Поскольку каждый оператор представляет из себя линейное преобразование, то его можно представить в виде матрицы. В зависимости от решаемой задачи, последовательность применяемых операторов будет различной. Определение конкретной последовательности — это уже математическая задача.

Обработка регистра кубитов будет осуществляться с помощью квантовой схемы. В программе квантовая схема состоит из двух функциональных блоков: шаблон квантовой схемы и квантовая схема в матричной форме.

Для описания квантовой схемы часто используется графический способ

представления. Напротив каждого кубита регистра рисуется длинная полоса. На этой полосе могут быть различные операторы. Также существуют операторы, которые могут занимать область нескольких кубитов. Это бинарные, тернарные и другие операторы. Таким образом, графическое представление квантовой схемы образует прямоугольную таблицу. Именно по этой причине, шаблон квантовой схемы в программе будет представлен двумерным массивом, элементами которого будут названия применяемых операторов, а также дополнительные параметры, необходимые для функционирования схемы.

Далее в программе происходит преобразование шаблона квантовой схемы в матричное представление. Сначала каждому названию оператора из таблицы шаблона ставится в соответствие матричная форма данного оператора. Затем происходит построение цельного оператора в матричной форме. Для этого определяется два правила построения оператора.

1. Если операторы располагаются последовательно, то общий оператор — обычное матричное произведение, произведенное в порядке справа налево.
2. Если операторы располагаются параллельно, то общий оператор — тензорное произведение матриц, произведенное в порядке снизу вверх.

Объединяя по этим правилам все операторы квантовой схемы, в результате получается квантовая схема в матричной форме.

Последний функциональный блок ядра — это непосредственно симулятор. Симулятор получает на вход подготовленный регистр кубитов в виде вектора, а также подготовленную квантовую схему в виде матрицы. При запуске симулятора происходит перемножение матрицы квантовой схемы на вектор состояния регистра кубитов. Этот процесс и есть симуляция квантовых вычислений. Но помимо просто изменения состояния симулятор способен производить измерение состояния отдельного кубита в квантовом регистре. Алгоритм измерения состояния выглядит следующим образом.

1. Вычисление вероятностей измерения выбранного кубита в состоянии «1» и в состоянии «0».
2. Проведение испытания Бернулли с определенными на предыдущем шаге вероятностями.
3. Изменение вектора состояния. Если в результате проведения испытания Бернулли была получена, к примеру, единица, то все вероятности, свя-

занные с измерением состояния «0» обнуляются. А оставшиеся коэффициенты нормируются. Аналогично происходит изменение вектора при получении нуля в результате измерения.

Таким образом, первая задача данной работы была выполнена. В результате был разработан симулятор квантовых вычислений, с помощью которого можно собирать произвольную квантовую схему. А данная возможность позволяет описать любой квантовый алгоритм. В случае, если квантовый алгоритм требует наличия особых операторов, то существует два способа решения этой задачи. Либо этот оператор можно представить в виде композиции базовых операторов, либо этот оператор можно добавить в программу, внося соответствующие изменения в классы, описывающие квантовый оператор, шаблон квантовой схемы и квантовую схему в матричной форме.

2 Веб-приложение

Веб-приложение будет состоять из нескольких страниц. Главная (или по-другому стартовая) страница будет содержать приветствие и краткую информацию о возможностях сервиса. А вторая страница — это интерактивный конструктор квантовых схем. Пользователю не нужно будет изучать особенности объектов из ядра симулятора. Ему будет достаточно просто размещать нужные ему операторы в нужных местах с помощью мыши.

Также стоит отметить, что этот сервис является расширяющимся. То есть с течением времени будет происходить расширение и наполнение этого сервиса разнообразным контентом. К примеру, в данном приложении можно реализовать интерактивное обучение квантовым вычислениям.

Веб-приложение разработано на основе технологии ASP.NET. ASP.NET (Active Server Pages для .NET) представляет из себя платформу для разработки веб-приложений. Данная платформа состоит из нескольких частей: веб-сервисы, программная инфраструктура и модель программирования. ASP.NET основан на исполняющей среде CLR (Common Language Runtime), что позволяет разрабатывать приложения на языках, доступных в комплекте .NET Framework. Таким образом, данная платформа позволяет разрабатывать код для веб-приложений и устанавливать связь между сервером и браузером. В платформе ASP.NET существует несколько программных моделей, предназначенных для создания веб-приложений. Одна из этих моделей — модель MVC, которая и была использована для разработки веб-приложения.

MVC (Model-View-Controller) — модель разделения данных приложения и управляющей логики на отдельные компоненты.

- Модель (Model) — компонент, который передает данные и изменяет свое состояние на основе команд контроллера.
- Представление (View) — компонент, который производит отображение данных пользователю.
- Контроллер (Controller) — компонент, который реагирует на действия пользователя.

Весь проект симулятора квантовых вычислений можно разделить на три блока: ядро симулятора квантовых вычислений QuantumCore, веб-приложение WebQuantum, а также связка между двумя этими частями QLogic.

Ядро симулятора квантовых вычислений состоит из двух блоков: матема-

тический и квантовый. В математическом блоке реализованы классы `Complex`, `Vector` и `Matrix`. В квантовом блоке реализованы классы `Qubit`, `QubitReg`, `Operator`, `Template`, `Circuit` и `Simulator`. Реализация всех этих классов была описана в предыдущей главе.

Веб-приложение `WebQuantum`, исходя из описания модели MVC, состоит из 3 основных частей: `Controllers`, `Models` и `Views`. Текущая версия веб-приложения содержит три страницы: главная страница, интерактивный конструктор квантовых схем и раздел «Privacy». Для их функционирования создано два контроллера: `HomeController`, который отвечает за главную страницу и раздел «Privacy», и `CircuitController`, отвечающий за конструктор квантовых схем.

В блоке `Models` реализован один класс `ErrorViewModel`. Данный класс создается автоматически при создании проекта MVC.

Последний блок `Views` содержит в себе несколько разделов и несколько дополнительных файлов. Раздел `Circuit` содержит файл, в котором описан вид страницы на языках HTML и JavaScript. Раздел `Home` содержит файлы, описывающие оформление главной страницы и раздела «Privacy». Все остальные разделы и файлы были созданы автоматически при создании проекта.

Помимо основных блоков, в проекте содержатся файлы, которые производят запуск веб-приложения, а также папку со стилевыми файлами, изображениями и прочими библиотеками.

Связка `QLogic` представляет из себя один класс с несколькими методами.

Внешний вид главной страницы можно посмотреть на изображении 1. Страница состоит из нескольких частей: заголовок, содержимое страницы и подвал. Изначально код описания заголовка и подвала страницы генерируются автоматически при создании проекта. Затем вносятся лишь небольшие корректировки. К примеру, добавляются новые кнопки.

Внешний вид раздела `Privacy` можно посмотреть на изображении 2.

При переходе с главной страницы на интерактивный конструктор квантовой схемы открывается страница следующего вида (см. изображение 3).

При выборе пользователем оператора из списка доступных операторов он выделяется красным цветом (см. изображение 4).

При наведении на область, в которую можно разместить оператор, происходит выделение этой области (см. изображение 5).

Пример расположения операторов на квантовой схеме представлен на

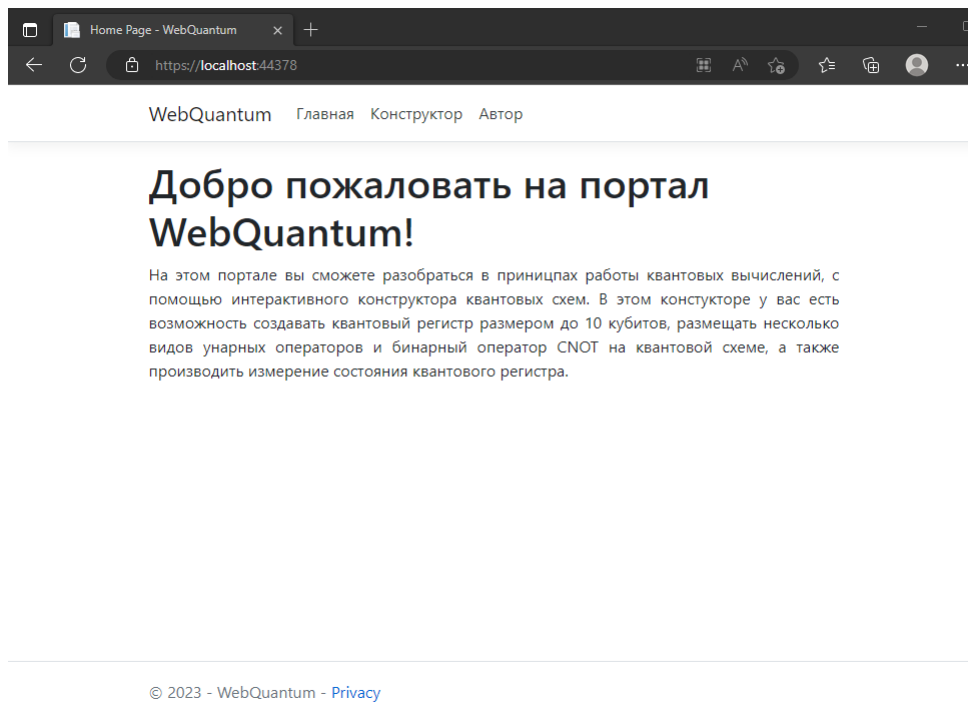


Рисунок 1 – Внешний вид главной страницы

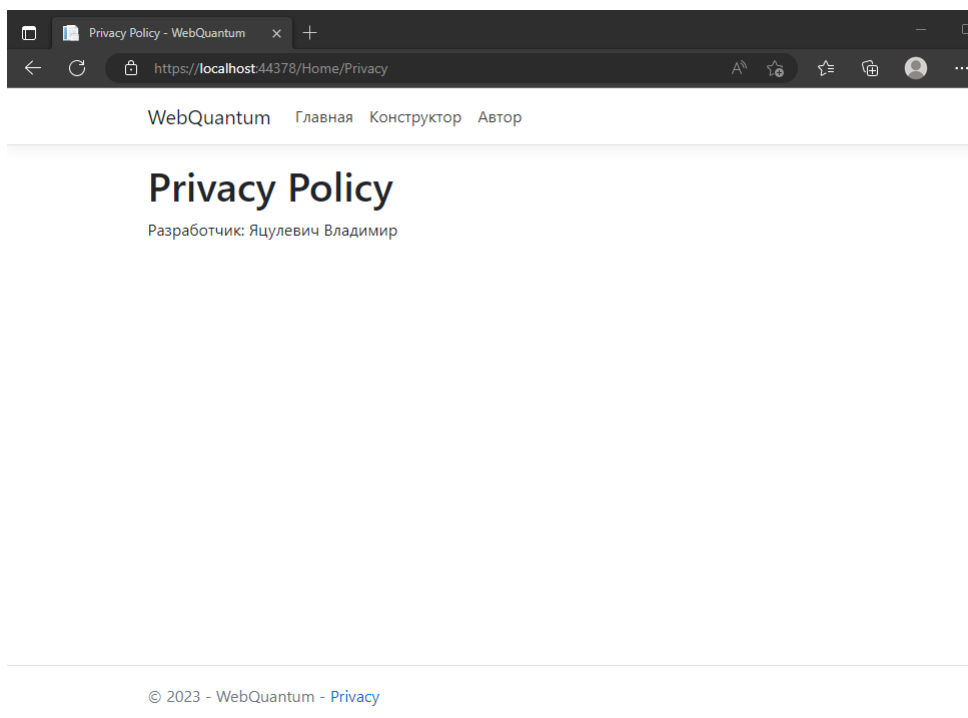


Рисунок 2 – Внешний вид раздела Privacy

изображении 6.

Еще один пример расположения операторов на квантовой схеме с большим количеством кубитов представлен на изображении 7.

В качестве иллюстрации, демонстрирующей работу квантовой схемы, представлен оператор Белла. Это оператор, действующий на регистр из двух

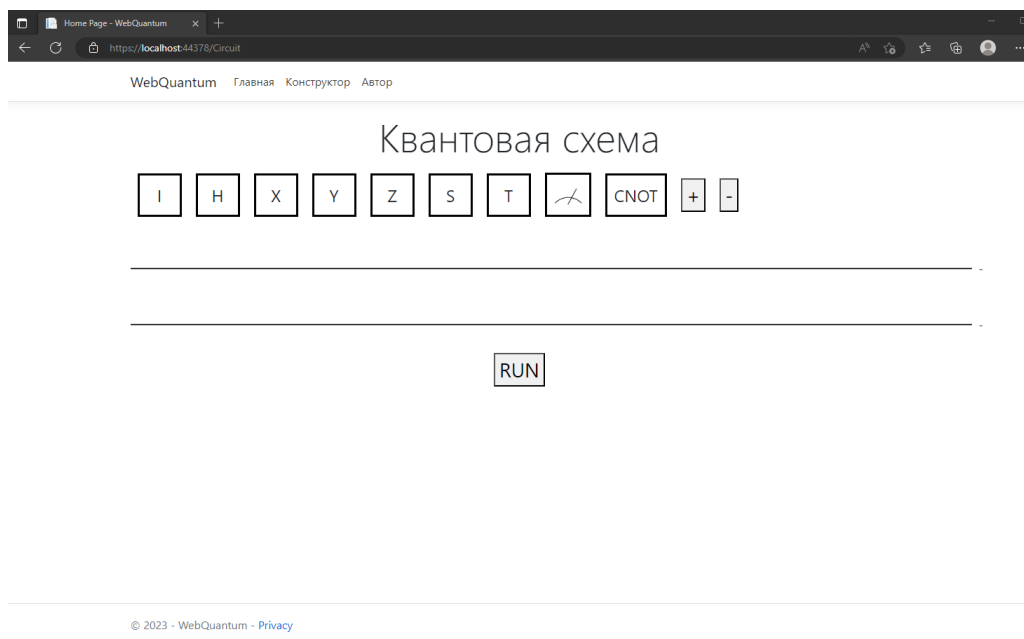


Рисунок 3 – Внешний вид интерактивного конструктора квантовой схемы



Рисунок 4 – При нажатии на оператор из списка доступных операторов происходит его выделение красным цветом



Рисунок 5 – Выделение области размещения оператора при наведении курсора

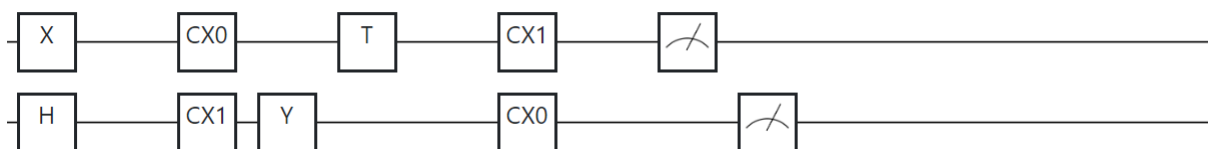


Рисунок 6 – Пример размещения операторов

кубитов, который переводит регистр в состоянии квантовой запутанности. Если один кубит примет значение нуля, то и второй примет значение ноль. Аналогично, если первый примет состояние один, то и второй примет это же состояние. Нулевое измерение можно наблюдать на изображении 8, а единичное измерение можно видеть на изображении 9.

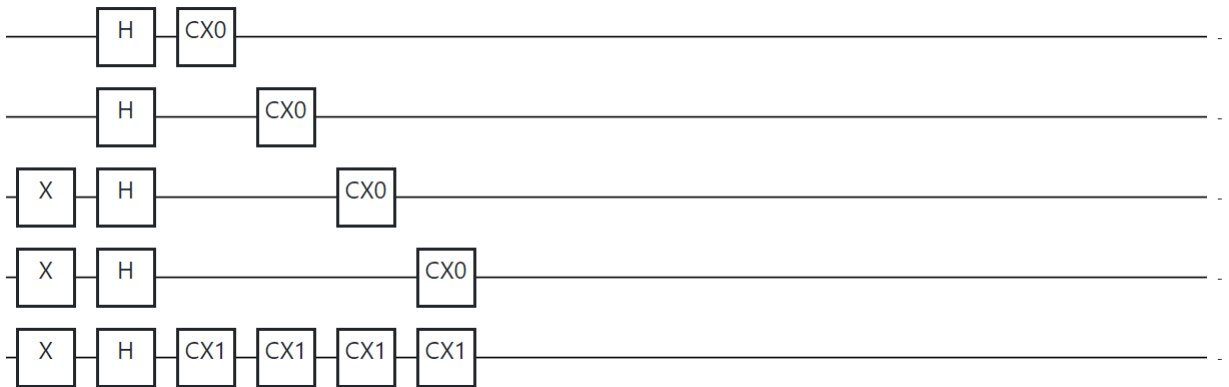


Рисунок 7 – Пример размещения операторов с большим количеством кубитов

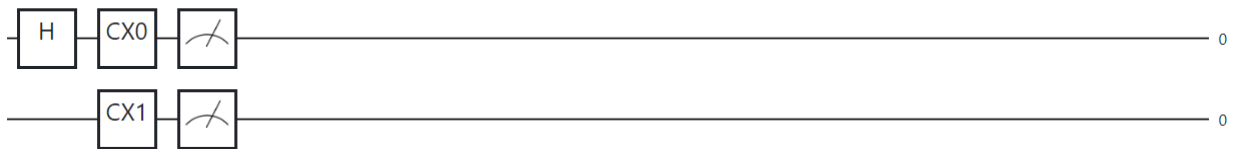


Рисунок 8 – Нулевое измерение состояния Белла

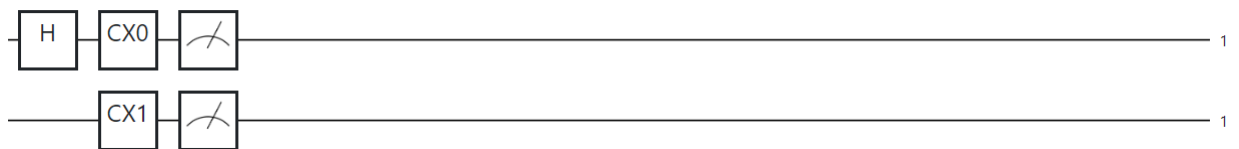


Рисунок 9 – Единичное измерение состояния Белла

ЗАКЛЮЧЕНИЕ

В ходе работы поставленная цель была выполнена. То есть результатом работы является полноценный симулятор квантовых вычислений. Для достижения этой цели были выполнены все поставленные задачи. А именно, было разработано ядро симулятора квантовых вычислений и было разработано веб-приложение, осуществляющее связь между пользователем и ядром симулятора.

В разработанном симуляторе можно конструировать квантовые схемы, состоящие из унарных и бинарных операторов, а также из оператора измерения. Симулятор способен обрабатывать квантовые регистры, содержащие менее 10 кубитов.

Стоит отметить перспективы развития разработанного симулятора. Во-первых, данный сервис можно опубликовать в сети Интернет. Во-вторых, этот сервис можно расширять, добавляя новые операторы или добавляя обучающие разделы на сайт. Таким образом, разработанный демонстрационный симулятор можно преобразовать в полноценный обучающий портал. Эта возможность является крайне актуальной, поскольку в русскоязычном сегменте Интернета подобных сервисов не имеется. А полноценного доступа к англоязычным аналогам в настоящее время нет. Именно по этой причине сервисы, подобные реализованному, сейчас крайне необходимы.