УДК 531: 530.145 ISSN 1995-5499
DOI: https://doi.org/10.17308/sait/1995-5499/2022/4/12-22
Поступила в редакцию 03.10.2022
Подписана в печать 05.12.2022

МОДЕЛИРОВАНИЕ ПРЕОБРАЗОВАНИЙ КВАНТОВЫХ РЕГИСТРОВ

© 2022 С. А. Запрягаев✉, Е. А. Килигин, И. М. Косенко, К. А. Турченко

Воронежский государственный университет
Университетская пл., 1, 394018 Воронеж, Российская Федерация

Аннотация. Квантовые информационные системы рассматриваются в качестве одного из направлений развития современных IT-систем. Настоящая статья посвящена разработке программной оболочки для выполнения цепи преобразований многокубитовых квантовых регистров с использованием набора стандартных квантовых гейтов. Алгоритмы таких преобразований реализуются и тестируются на симуляторах, имитирующих поведение квантовых компьютеров. Разработанный в работе программный модуль позволяет визуализировать исполнение квантовых операторов на группе кубитов, задать и исполнить конкретный квантовый алгоритм, а также использовать данный модуль в качестве ядра специализированной платформы квантовых вычислений. Приложение разработанного программного средства определяется современным повышенным интересом к проблеме детального анализа различных квантовых алгоритмов и протоколов передачи квантовой информации без непосредственного использования квантовых компьютеров. Представленные в настоящей работе примеры применения программного модуля демонстрируют возможность его гибкого использования в широком спектре анализа практических приложений алгоритмов для различных квантовых информационных систем.
Ключевые слова: квантовые вычисления, кубит, квантовый регистр, квантовый алгоритм Шора, сверхплотное кодирование.

ВВЕДЕНИЕ

Повышенный интерес к проблеме применения квантовых компьютеров и иных квантовых информационных систем в различных приложениях способствует росту числа программных средств для моделирования и проведения реальных преобразований квантовых регистров по заданным квантовым схемам. Широко известной системой моделирования квантовых схем является платформа Qiskit [1]. В данной платформе представлены многочисленные известные квантовые алгоритмы, выполнение которых можно проводить как модельно, так и с использованием подключения к реальному квантовому компьютеру IBM. Существуют и иные примеры программных оболочек для выполнения квантовых вычислений, особенно на базе зарубежных гигантов IT-индустрии, занятых решением проблемы создания реальных квантовых компьютеров [2]. Разработаны программные оболочки, «превращающие» настольные персональные компьютеры в «квантовые», что подразумевает определенное моделирование работы квантового компьютера. Так, например, реализовано веб-приложение Quantum Computing Playground для Chrome [3]. Приложение позволяет выполнять известные квантовые алгоритмы и писать собственные квантовые программы на языке QScript с имитацией системы с квантовым регистром до 22 кубитов.

Еще один пример облачной службы выполнения квантовых вычислений — это Azure Quantum [4] от Microsoft. Служба предоставляет среду разработки при создании квантовых алгоритмов для нескольких платформ, что позволяет уделить пристальное внимания программированию на уровне алгоритмов. В работе [5] предложен высокопроизво-

✉ Запрягаев Сергей Александрович
e-mail: zsa@cs.vsu.ru

Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.

дительный векторный эмулятор квантовых процессоров, разработанный на языке программирования Rust. Данный язык поддерживает язык программирования OpenQASM 2.0 для квантовых схем и имеет удобный API на основе Python. Представлен широкий ассортимент числовых тестов эмулятора и ожидается, что эмулятор будет использоваться для проектирования и проверки квантовых алгоритмов

В настоящее время имеется много причин, по которым необходимо создание национальных платформ, продвигающих реальные и модельные вычисления квантовых цепей различных уровней, самостоятельно созданных исследователями.

В данной работе представлено описание ядра программной оболочки, которая решает проблему анализа группы различных квантовых цепей со стандартными одно- и многокубитовыми операторами (гейтами). В качестве объекта исследования выполнено моделирование семикубитового квантового компьютера при реализации алгоритма Шора.

1. КВАНТОВЫЙ РЕГИСТР

В квантовой теории информации единицей квантовой теории информации является кубит (qubit = квантовый бит) [6]. Формальное определение кубита — это суперпозиция двух произвольных квантовых состояний. Для связи с классической теорией информации два выбранных квантовых состояния обычно обозначаются символами $|0\rangle$ и $|1\rangle$.

В данных обозначениях кубит $|q\rangle$ — это квантовое состояние вида $|q\rangle = c_0|0\rangle + c_1|1\rangle$. Здесь c_0 и c_1 — комплексные числа, удовлетворяющие условию: $|c_0|^2 + |c_1|^2 = 1$, что соответствует физической нормировке кубита $|q\rangle$ и интерпретации результатов его измерения. При этом базисные состояния кубита $|0\rangle$ и $|1\rangle$ ортонормированы $< i \mid j >= \delta_{ij}$, где $i, j = 0,1$.

В так называемом представлении «вычислительного базиса» [7], квантовым состояниям $|0\rangle$, $|1\rangle$ и их суперпозиции $|q\rangle$ ставятся в соответствие векторы $|0\rangle = (1,0)^T$ и $|1\rangle = (0,1)^T$ (здесь T — операция транспонирования). При этом кубит равен $|q\rangle = c_1|0\rangle + c_2|1\rangle \to (c_1,\ c_2)^T$.

Квантовый регистр представляет собой совокупность группы кубитов. А вычислительный базис регистра кубит образуется прямым произведением базисных состояний отдельных (i-ых) кубитов $|q_i\rangle = (c_1^{(i)}, c_2^{(i)})^T$. Например, регистр из двух кубитов $|Q\rangle_2$ имеет, по определению, следующий вид:

$$|Q\rangle_2 = |q_1\rangle \otimes |q_2\rangle = \sum_{k_1 k_2 = 0,1} C_{k_1 k_2} |k_1 k_2\rangle.$$

Здесь $C_{k_1 k_2}$ — произведение амплитуд базисных однокубитовых состояний пары кубитов, а $|k_1 k_2\rangle$ — четыре базисных состояния так называемого двухкубитового вычислительного базиса $|0\rangle \otimes |0\rangle = |00\rangle$, $|0\rangle \otimes |1\rangle = |01\rangle$, $|1\rangle \otimes |0\rangle = |10\rangle$, $|1\rangle \otimes |1\rangle = |11\rangle$.

Можно ввести тождественные десятичные обозначения для вычислительного базиса двухкубитовых состояний, используя бинарное представление десятичного числа $|0\rangle_2 = |00\rangle$, $|1\rangle_2 = |01\rangle$, $|2\rangle_2 = |10\rangle$, $|3\rangle_2 = |11\rangle$. Таким образом, алгебраическое выражение для произвольного двухкубитового регистра можно представить в более компактном виде:

$$|Q\rangle_2 = \sum_{k=0}^{3} a_k |k\rangle_n .$$

Очевидным обобщением выражения для регистра из *n* кубитов является формула

$$|Q\rangle_n = \sum_{k_1, k_2, \ldots k_n \in 0,1} C_{k_1, k_2, \ldots k_n} |k_1 k_2 \ldots k_n\rangle.$$

Если использовать матричные представления $|0\rangle = (1,0)^T$ и $|1\rangle = (0,1)^T$ для векторов вычислительного базиса однокубитовых состояний, то векторы вычислительного базиса *n*-кубитового регистра содержат $2n$ компонент и в нормированном виде есть столбцы:

$$|0\rangle_n \equiv |0\rangle \otimes |0\rangle \ldots \otimes |0\rangle \otimes |0\rangle \to (100\ldots00)^T$$

$$|1\rangle_n \equiv |0\rangle \otimes |0\rangle \ldots \otimes |0\rangle \otimes |1\rangle \to (000\ldots01)^T$$

$$\ldots$$

$$|2^n - 1\rangle_n \equiv |1\rangle \otimes |1\rangle \ldots \otimes |1\rangle \otimes |1\rangle \to (111\ldots11)^T.$$

Соответственно, в десятичных обозначениях регистр можно представить в виде

$$|Q\rangle_n = \sum_{k=0}^{2^n - 1} a_k |k\rangle_n .$$

Преобразования отдельных кубит и их регистров осуществляются унитарными операторами (обратный оператор совпадает с эрмитово-сопряженным) [6]. Символически преобразование регистра предполагает выполнение последовательности действий: исходный регистр → оператор (гейт, вентиль) → преобразованный регистр.

Простейший регистр состоит из одного кубита. Так как матричное описание однокубитового состояния определяется двухкомпонентным вектором, то матрица однокубитового унитарного оператора (квантового гейта) является матрицей размерности 2×2. Так как в классе матриц данной размерности матрицы Паули σ_x, σ_y, σ_z (и единичная матрица) образуют полный набор матриц, то имеется бесконечный набор однокубитовых операторов, которые выражаются через комбинации матриц Паули. В квантовых информационных системах матрицы Паули принято обозначать заглавными буквами $\sigma_x = X$, $\sigma_y = Y$, $\sigma_z = Z$.

В алгебраическом виде действие оператора Паули X можно представить следующим образом:

$$X : (c_1 \left|0\right\rangle + c_2 \left|1\right\rangle) = c_2 \left|0\right\rangle + c_1 \left|1\right\rangle,$$

что по смыслу определяет квантовый гейт отрицания (отрицаются базисные состояния).

При вычислении действия X на j-й кубит произвольного n-кубитового регистра, результат его действия может быть представлен следующим образом:

$$\sigma_x^{(j)} : \left|Q\right\rangle_n = $$

$$= \sum_{k_1,k_2,\ldots k_n \in 0,1} C_{k_1\ldots\overline{k_j}\ldots k_n} \left|k_1 \ldots k_j \ldots k_n\right\rangle,$$

где $\overline{k_j} = \mathrm{NOT}(k_j)$. Аналогичные выражения для $\sigma_y^{(j)}$ и $\sigma_z^{(j)}$ есть (i — мнимая единица):

$$\sigma_y^{(j)} : \left|Q\right\rangle_n =$$

$$= i \sum_{k_1,k_2,\ldots k_n \in 0,1} (-1)^{\overline{k_j}} \; C_{k_1\ldots\overline{k_j}\ldots k_n} \left|k_1 \ldots k_j \ldots k_n\right\rangle,$$

$$\sigma_z^{(j)} : \left|Q\right\rangle_n =$$

$$= \sum_{k_1,k_2,\ldots k_n \in 0,1} (-1)^{k_j} \; C_{k_1\ldots k_j\ldots k_n} \left|k_1 \ldots k_j \ldots k_n\right\rangle.$$

Фактически перечисленные выше три выражения достаточны для определения действия любого однокубитового оператора. Например, в квантовой теории информации важную роль играет оператор Адамара H, который является суперпозицией $H = (X + Z)/\sqrt{2}$.

В силу того, что оператор Адамара является часто используемым однокубитовым оператором, ниже приведен результат действия оператора Адамара на j-й кубит регистра

$$H_j : \left|Q\right\rangle_n =$$

$$= \frac{1}{\sqrt{2}} \sum_{k_1,k_2,\ldots k_n \in 0,1} A_{k_1\ldots k_j\ldots k_n} \left|k_1 \ldots k_j \ldots k_n\right\rangle,$$

где $A_{k_1\ldots k_j\ldots k_n} = (-1)^{k_j} C_{k_1\ldots k_j\ldots k_n} + C_{k_1\ldots\overline{k_j}\ldots k_n}$.

Используемые фазовые однокубитовые операторы (типа S, T [6, 7] и т.п.) всегда могут быть выражены через операторы I, X, Y, Z в силу общей теоремы алгебры матриц размерности 2×2.

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \lambda_0 I + \sum_{k=1}^{3} \lambda_k \sigma_k \; .$$

Здесь

$$\lambda_0 = \frac{a_{11} + a_{22}}{2}, \;\; \lambda_1 = \frac{a_{12} + a_{21}}{2},$$

$$\lambda_2 = i\frac{a_{12} - a_{21}}{2}, \;\; \lambda_3 = \frac{a_{11} - a_{22}}{2}.$$

Среди многокубитовых операторов для преобразования регистров основную роль играют операторы CNOT и CCNOT. Определение этих операторов можно представить в символическом виде следующим образом:

$$\mathrm{CNOT} : \left|i, j\right\rangle = \left|i, i \oplus j\right\rangle.$$

Здесь $\left|i, j\right\rangle \equiv \left|i\right\rangle \otimes \left|j\right\rangle$ — прямое произведение базисных состояний вычислительного базиса пары кубитов, где $i, j \in 0,1$. Другими словами, оператор осуществляет отрицание базисного состояния кубита $\left|j\right\rangle$ (управляемый кубит) только при условии, если базисное состояние кубита $\left|i\right\rangle$ (управляющий кубит) находится в состоянии $\left|1\right\rangle$. В противном случае управляемый кубит остаётся без изменений. Тождественные графические обозначения логической диаграммы оператора CNOT приведены на рис 1.

Преобразование произвольного регистра в результате действия гейта $\mathrm{CNOT}_{i,j}$, для которого первый индекс (i) определяет номер управляющего кубита, а второй индекс (j) —

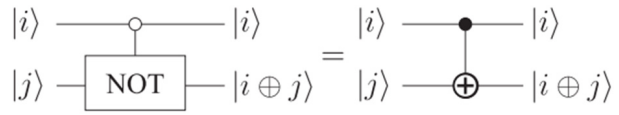

Рис. 1. Условные обозначения CNOT
[Fig. 1. CNOT symbol]

номер управляемого кубита, приводит к результату:

$$\mathrm{CNOT}_{i,j} : \left|Q\right\rangle_n = \\ = \sum_{k_1,..,k_n \in 0,1} C_{k_1..k_i..k'_j..k_n} \left|k_1..k_i..k_j..k_n\right\rangle.$$

Здесь $k'_j = k_j \otimes k_i$.

Соответственно, определение оператора CCNOT имеет вид:

$$\mathrm{CCNOT} : \left|i, j, k\right\rangle = \left|i, j, i \wedge j \oplus k\right\rangle.$$

Здесь $\left|i, j, k\right\rangle = \left|i\right\rangle \otimes \left|j\right\rangle \otimes \left|k\right\rangle$ и кубиты $\left|i\right\rangle$, $\left|j\right\rangle$ являются управляющими, а кубит $\left|k\right\rangle$ — управляемым и для него выполняется операция отрицания только в случае, если оба управляющих кубита находятся одновременно в базисных состояниях вычислительного базиса, равных $\left|1\right\rangle$. В противном случае состояние управляемого кубита не меняется. Тождественные графические обозначения логической диаграммы оператора CCNOT приведены на рис 2.

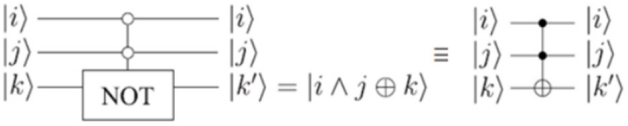

Рис. 2. Условные обозначения CCNOT
[Fig. 2. CCNOT symbol]

Наконец, действие оператора $\mathrm{CCNOT}_{i,j,l}$ на произвольный регистр, при котором индексы i и j определяют номера управляющих кубитов, а l — номер управляемого кубита, приводит к следующему алгебраическому результату:

$$\mathrm{CCNOT}_{i,j,l} : \left|Q\right\rangle_n = \\ = \sum_{k_1,k_2,...k_n \in 0,1} C_{k_1...k_i...k_{j...}k'_l...k_n} \left|k_1 k_2 \ldots k_n\right\rangle;$$

$$k'_l = k_l \otimes \left(k_i \wedge k_j\right).$$

Выведенные выше алгебраические правила преобразования квантового регистра достаточны для реализации моделирования квантовых цепей различных квантовых алгоритмов. Перечисленный комплекс правил преобразований произвольного многокубитового регистра реализован в программной оболочке, представленной в данной работе.

Основной принцип, заложенный в данной работе, к алгебраическому определению регистра кубитов состоит в последовательном перечислении суперпозиции базисных состояний регистра, начиная от состояния $\left|0\right\rangle_n$ и закачивая состоянием $\left|2^n - 1\right\rangle_n$.

2. ОПИСАНИЕ РАЗРАБОТАННОГО ПРИЛОЖЕНИЯ

Программная оболочка для выполнения преобразований квантового регистра представляет собой одно окно, включающее блоки управления и передачи информации. В левой верхней части приложения приведены окна таблиц **Input** и **Output**, в которых соответственно указываются (задаются) входные и отображаются выходные значения амплитуд регистров. Значения амплитуд регистров из n кубитов последовательно определяются (в общем случае) комплексными числами или символами для базисных состояний, начиная с состояний $\left|0\right\rangle_n$, $\left|1\right\rangle_n$ и до $\left|2^n - 1\right\rangle_n$ состояния. В каждом из окон приведены по два столбца: вещественные и мнимые части амплитуд. Количество строк равно 2^n, где n — число кубитов, которое задаётся в блоке «Number of qubits». Помимо метода ручного задания значений амплитуд, для упрощения работы имеются следующие кнопки для автоматического задания амплитуд регистров:

- Set zero — установить значение соответствующей таблицы в ноль;
- Set symbolic — заполнить таблицу комплексными символьными переменными c1, c2, … для каждого регистра;
- Set symbolic by Re and Im — заполнить таблицу символьными переменными r1, i1, r2, i2, … для каждого регистра отдельно для вещественной и мнимой части.

В начале работы пользователь задает необходимое число кубитов, заполняет значения амплитуд входных регистров, а затем (в режиме выполнения одной операции преобразования) применяет (выбором кнопки

Apply) к заданному регистру один из квантовых вентилей, представленных в правом блоке основного окна программы. Результат преобразования выводится в таблице выходных значений. Если запись во входных данных окажется ненормированной, программа предложит это сделать. Поскольку в таком режиме можно совершить только одну операцию, кнопка «Copy result to input» позволяет перенести результат во входную таблицу для дальнейшего преобразования, если это необходимо. В настоящее время в программе реализованы следующие гейты:

• Матрицы Паули (требуют указания целевого кубита).

• Гейт Адамара (требует указания целевого кубита).

• Гейт Уолша — Адамара (применяется ко всему регистру).

• Управляемое «Нет» (CNOT), с указанием управляющего и управляемого кубитов.

• Вентиль Тоффоли (CCNOT), с указанием двух управляющих кубитов и одного управляемого.

• Гейт управляемой фазы, с указанием управляющего и управляемого кубитов.

• Перестановка пары кубитов в регистре (Swap), с указанием целевых кубитов.

При однократном применения гейта — нажатие **Apply** на кнопку при соответствующем операторе, его краткое имя и параметры добавляются в строку формирования квантовой цепи (под таблицами для регистров). Выбираются гейты по очереди (кнопками **Apply**), а в окне формирования квантовой цепи автоматически последовательно создается текстовая строка с задаваемой к исполнению квантовой схемой, Данную строку можно редактировать, копировать, очищать и выполнять с помощью кнопки «Apply» у строки формирования квантовой цепи. В данном режиме все вентили строки будут выполнены за

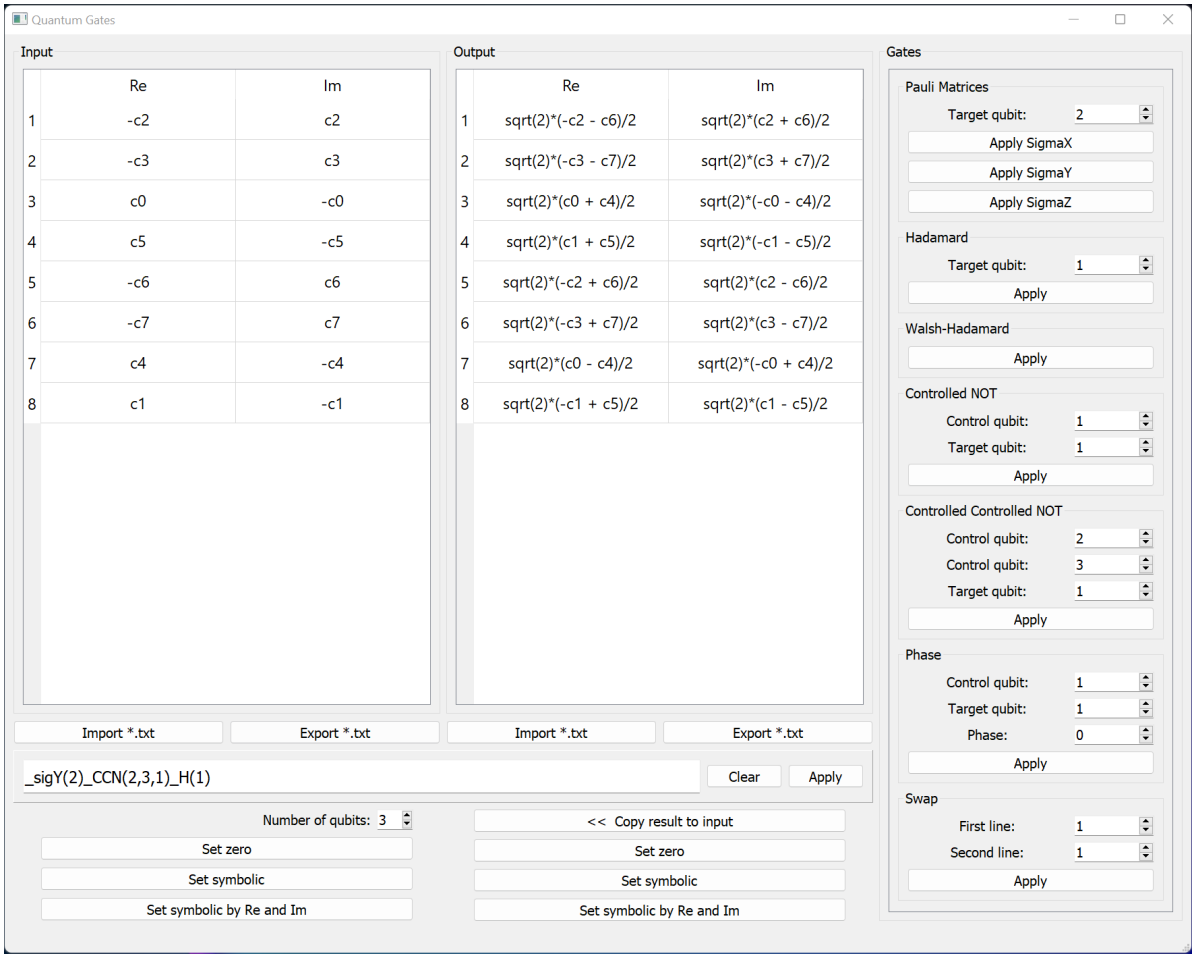

Рис.3. Интерфейс программной оболочки
[Fig. 3. Shell window interface]

один раз в той последовательности, как они записаны в строке (слева направо), а результат сформируется в таблице **Output**.

Следует отметить, что в данном режиме в окне **Output** возникает результат действия выбранного оператора на фиксированный начальный регистр при каждом добавлении оператора в окно формирования цепи

Формат используемого текстового файла — это набор строк, где каждая строка последовательно отвечает амплитуде соответствующего базисного состояния регистра в таблице и записывается в виде комплексного числа. Например, если в программе в третьей строке в колонках для вещественной и мнимой части стояли единицы, то при экспорте в той же строке будет записано «1+ I».

3. СВЕРХПЛОТНОЕ КОДИРОВАНИЕ И ТЕЛЕПОРТАЦИЯ

Квантовые каналы связи — физическая система для передачи информации, в которой в качестве носителя информации используются кубиты. В настоящее время различные формы квантовой коммуникации рассматриваются как перспективное направление в квантовой теории информации и методах её обработки. Для описания квантовых каналов с использованием разработанной программной оболочки рассмотрены сверхплотное кодирование и квантовая телепортация.

Сверхплотное кодирование — это технология использования квантового канала связи для прямой пересылки группы кубитов от абонента А к абоненту Б, при которой одним кубитом может быть передано два бита классической информации (хотя передаётся только один физический объект). В своей основе данный метод использует запутанное состояние пары кубитов [6] и специальный протокол передачи информации от одного абонента другому (без взаимного обмена).

Метод предполагает, что абоненты предварительно создали одно из четырёх возможных состояний Белла $\left|q_{i,j}\right\rangle_2$ [6, 7] и затем пространственно разъединились, забрав с собой один из кубитов созданной ими пары:

$$\left|q_{i,j}\right\rangle_2 = \left(\left|0,j\right\rangle + \left(-1\right)^i \left|1,\bar{j}\right\rangle\right)/\sqrt{2},$$

где $i, j \in 0,1$.

Протокол предусматривает возможность передачи одного, предварительно обработанного, кубита из разъединенной пары от передающего информацию абонента А к принимающему информацию абоненту Б по идеальному квантовому каналу. При этом абонент Б своей «дешифрующей квантовой машиной» обрабатывает полученный кубит совместно со своим кубитом и проводит измерение полученного двухкубитового состояния по приведённой на рис. 4 квантовой цепи.

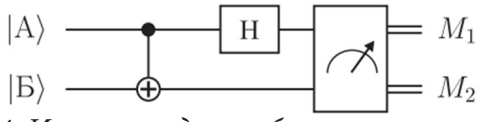

Рис.4. Измерение двухкубитового состояния [Fig. 4. Measuring of two-qubit state]

Результатом измерения являются два бинарных значения M_1 и M_2, равные либо 0, либо 1.

Конкретный результат измерения абонентом Б зависит от типа выбранного запутанного состояния (которые в литературе эквивалентно называются EPR- пары [6, 7]) и способа обработки абонентом А своего кубита перед отправкой его абоненту Б. Четыре возможных результата измерения у абонента Б всегда будет содержать какие-то два бита информации (M_1, M_2). Ими могут быть только $(0,0)$, $(0,1)$, $(1,0)$ или $(1,1)$.

В табл. 1 приведены базисные состояния регистра кубитов перед процессом измерения абонентом Б в зависимости от действия абонентом А на свой кубит до его отправки абоненту Б. В таблице указаны однокубитовые операторы, которые мог использовать А. Такими операторами являются – оператор тождественного преобразования и комбинации матриц Паули

Данные таблицы получены исполнением команд в окне формирования квантовой цепи

$$\_H(1)\_CN(1,2)\_G(1)\_CN(1,2)\_H(1).$$

Здесь гейт $G(1)$ предполагает одно из четырёх действий абонента А ($G = I$, или X, или Z, или произведение $Z \cdot X$).

Таблица 1. Состояния регистров до измерения
[Table 1. Register states before measurement]

		1	2	3	4
EPR пара		$\|q_{00}\rangle_2$	$\|q_{00}\rangle_2$	$\|q_{00}\rangle_2$	$\|q_{00}\rangle_2$
		Регистр Б пере измерениями			
абонент А	I	|00>	|01>	|10>	|11>
	X	|01>	|00>	–|11>	–|10>
	Z	|10>	|11>	|00>	|01>
	(ZX)	–|11>	–|10>	|01>	|00>

Для пересылки целого текста от А к Б абоненты должны запастись набором EPR-пар. В общем случае абонент А кодирует текст битовой последовательностью, а затем для каждой пары битовых значений абонент А выбирает кубит из набора EPR-пар, имеющихся и у А, и у Б. Далее А преобразует свой кубит по протоколу и посылает его абоненту Б. Абонент Б измеряет каждый переданный ему кубит и получает последовательность двухбитовых значений, которая и составляет передаваемый текст.

Результаты передаваемой информации по такому каналу связи с применением разработанной программной оболочки при использовании восьми различных EPR-пар в этом случае (для примера) представлены в табл. 2

Таблица 2. Кодирование передаваемой информации
[Table 2. Encoding of transmitted information]

Типы пары	Абонент А	Измерения Б	Передаваемое число
$\|q_{00}\rangle_2$	| XZ | X	00 01 10 00 01	97
$\|q_{01}\rangle_2$	| XZ | X	01 00 11 01 00	308
$\|q_{10}\rangle_2$	| XZ | X	10 11 00 10 11	715
$\|q_{11}\rangle_2$	| XZ | X	11 10 01 11 10	926
$\|q_{00}\rangle_2$	X | ZX (ZX)	01 00 10 01 11	295
$\|q_{01}\rangle_2$	X | ZX (ZX)	00 01 11 00 10	114
$\|q_{10}\rangle_2$	X | ZX (ZX)	11 10 00 11 01	909
$\|q_{11}\rangle_2$	X | ZX (ZX)	10 11 01 10 00	728

Представленная табл. 2 демонстрирует использование программной оболочки для моделирования передачи информации произвольной длины по квантовому каналу.

Аналогичное моделирование может быть выполнено и для явления квантовой телепортации, в которой используется логическая диаграмма вида:

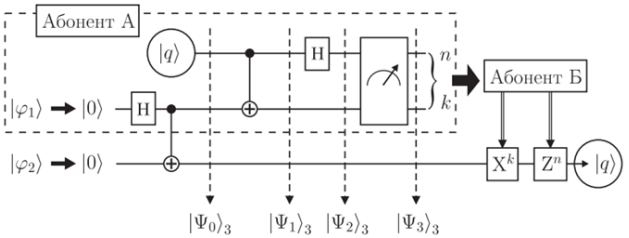

Рис. 5. Цепь квантовой телепортации
[Fig. 5. Quantum teleportation chain]

Модельное вычисление состояний кубита у абонента Б, до того, как абонент А сообщил по классическому каналу связи результат измерений, представлен в табл. 3 (для всех возможных вариантов выбора начальных состояний кубитов φ_1 и φ_2 (рис. 5) с использованием следующей последовательности операторов в окне формирования квантовой цепи программной оболочки:

$$\_H(2)\_CN(2,3)\_CN(1,2)\_H(1).$$

4. МОДЕЛИРОВАНИЕ АЛГОРИТМА ШОРА

Квантовый алгоритм Шора решает классическую задачу разложения числа на простые множители с экспоненциальным ускорением. Задача факторизации целых чисел обычно состоит в определении простых множителей p и q для заданного целого числа $N = p \cdot q$. Классическое решение такой задачи может опираться на алгоритм нахождения порядка числа в арифметике по модулю. Порядком числа x по модулю N называется наименьшее натуральное число r, для которого выполняется сравнение $x^r mod N = 1$.

В качестве примера использования разработанной программной оболочки рассмотрен квантовый алгоритм Шора на семи-кубитовой квантовой машине. Выберем для примера $x = 7$ и $N = 15$. В общем случае при известном периоде r множители N определяются как наибольшие общие делители чисел $2^{r/2}+1,\ 2^{r/2}-1,\ N$. Для справки $7^k mod 15$ при $k = 0,1,2,3,4,\ldots$ образуют последовательность

Таблица 3. Состояние кубита абонента Б
[Table 3. Subscriber B qubit state]

Варианты исходных кубит φ_1, φ_2	1 вариант	2 вариант	3 вариант	4 вариант
	|00>	|01>	|10>	|11>
Измерения А	Состояние кубита абонента Б до получения сообщения от А			
|00>	a |0>+b |1>	a |1>+b |0>	a |0>–b |1>	a |1>+b |0>
|01>	a |1>+b |0>	a |0>+b |1>	a |1>+b |0>	a |0>–b |1>
|10>	a |0>–b |1>	a |1>+b |0>	a |0>+b |1>	a |1>+b |0>
|11>	a |1>–b |0>	a |0>–b |1>	a |1>+b |0>	a |0>+b |1>

1, 7, 4, 13, 1, 7, 4, 13, … (то есть $r = 4$ для данного примера).

Квантовый алгоритм Шора в данном примере можно реализовать квантовой цепью для вычисления степеней числа 7 по модулю 15 и цепью обратного квантового преобразования Фурье, представленными на рис. 6.

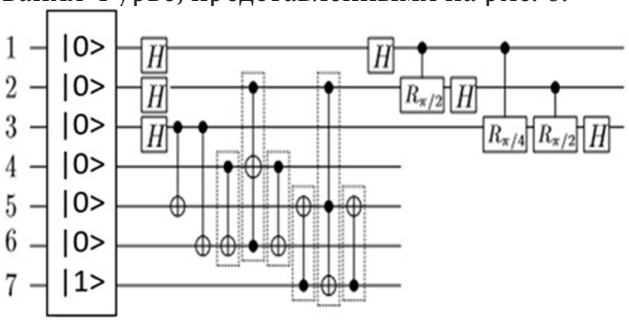

Рис.6. Квантовая цепь алгоритма Шора
[Fig. 6. Shor algorithm quantum chain]

Задание исходного базисного состояния $|0000001\rangle$ соответствует значению, равному 1 в окне действительной части **Input** во второй строке. Все остальные коэффициенты равны 0. В соответствии с последовательностью операторов алгоритма Шора (рис. 6) в строку формирования квантовой цепи необходимо записать (данная последовательность автоматически вводится при нажатии **apply** гейтов):

$$H(1)\_H(2)\_H(3)\_CN(3,5)\_$$
$$CN(3,5)\_CN(3,6)\_CN(4,6)\_$$
$$CCN(2,6,4)\_CN(4,6)\_CN(7,5)\_$$
$$CCN(2,5,7)\_CN(7,5)$$

Исполнение описанной выше последовательности (выбором **apply** у строки формирования квантовой цепи) приведет к вычислению состояния квантового регистра после гейта CNOT(7,5) в виде

$$[|000,0001\rangle + |001,0111\rangle + |010,0100\rangle +$$
$$|011,1101\rangle + |100,0001\rangle + |101,0111\rangle +$$
$$|110,0100\rangle + |111,1101\rangle] / \sqrt{8}.$$

Здесь для наглядности базисные состояния первых трёх кубитов отделены запятой. Бинарное представление первыми тремя кубитами определяет возможные значения степени k, а остальные четыре кубита отражают бинарное представление функции $f(k) = 7^k \, mod 15$. В десятичных обозначениях полученный регистр имеет вид:

$$|0, f(0) = 1\rangle + |1, f(1) = 7\rangle + |2, f(2) = 4\rangle +$$
$$|3, f(3) = 13\rangle + |4, f(4) = 1\rangle + |5, f(5) = 7\rangle +$$
$$|6, f(6) = 4\rangle + |7, f(7) = 13\rangle.$$

В соответствии с логикой Алгоритма Шора с данным регистром выполняется процедура измерения функции f. Результатом измерения может быть одно из чисел $1, 7, 4, 13$. Пусть для примера измерение f (регистр кубитов с 4 по 7)) оказалось равно 7. Тогда регистр значений k примет вид $[|001\rangle + |101\rangle] / \sqrt{2}$.

Дальнейшие преобразования с данным регистром представлены на рис. 3 и являются обратным квантовым преобразованием Фурье. При введении соответствующих данных в программную оболочку регистр кубитов k (после исполнения Фурье-преобразования) в бинарном представлении будет иметь вид: $[|000\rangle + |100\rangle + i|010\rangle - i|110\rangle] / 4$. Это числа 0, 4, 2, 6. На основании классической части алгоритма Шора [7] можно сделать вывод об определении $k = 4$. Таким образом, сомножитель $C_1 = 7^2 + 1 = 50$, $C_2 = 7^2 - 1 = 48$. Их

наименьшие общие делители с $N=15$ равны 5 и 3. В результате установлено, что число N состоит из $3\cdot5$, то есть выполнена факторизация числа N.

5. ВЫЧИСЛЕНИЕ СТЕПЕНИ ЧИСЛА ПО МОДУЛЮ

Квантовые алгоритмы вычисления целой степени числа по модулю $y(x)=x^k mod N$ определяются последовательностью квантовых гейтов, которая зависит от выбора числа и модуля вычислений. Задание диапазона числа k определяется количеством используемых кубитов регистра, что даёт возможность сформировать значения k в диапазоне $[0,\ 2^n-1]$.

Рассмотрим для примера вычисление функции $2^k mod 15$ на примере семикубитового регистра. В регистре X (первые три кубита рис. 7) формируются все десятичные значения k от 0 до 7. В четвертом — девятом кубитах формируются значения искомой функции $2^k mod 15$.

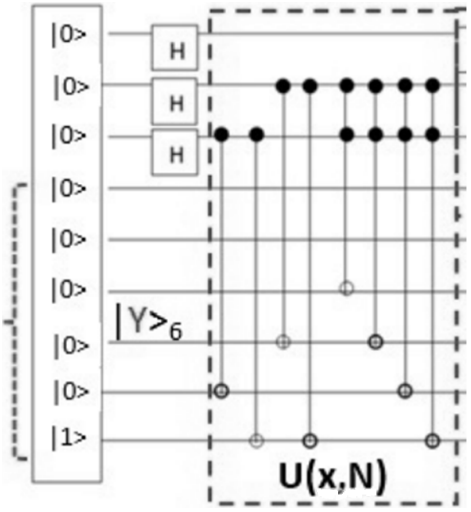

Рис.7. Вычисление степени числа по модулю [Fig. 7. Calculating the degree of a number modulo]

Результат вычисления текущей последовательности преобразований, заданной в окне формирования квантовой цепи программной оболочки

$$\_H(1)\_H(2)\_H(3)\_CN(3,8)\_CN(3,9)\_$$
$$CN(2,7)\_CN(2,9)\_CCN(2,3,6)\_$$
$$CCN(2,3,7)\_CCN(2,3,8)\_CCN(2,3,9)$$

по представленной программе имеет вид (в десятичных обозначениях):

$$|Q\rangle_9=\frac{\sqrt{2}}{4}[|1\rangle_9+|66\rangle_9+|132\rangle_9+|200\rangle_9+$$
$$+|257\rangle_9+|322\rangle_9+|388\rangle_9+|456\rangle_9].$$

Если представить слагаемые регистра с указанием k и y: $|k,y\rangle_9$, то приведённая выше суперпозиция регистров будет иметь вид

$$|Q\rangle_9=\frac{\sqrt{2}}{4}[|0,1\rangle_9+|1,2\rangle_9+|2,4\rangle_9+$$
$$+|3,8\rangle_9+|4,1\rangle_9+|5,2\rangle_9+|6,4\rangle_9+|7,8\rangle_9].$$

ЗАКЛЮЧЕНИЕ

В настоящей статье разработана программная оболочка, позволяющая выполнять преобразования квантовых регистров с любыми однокубитовыми и известными многокубитовыми операторами. Приложение может представлять интерес при анализе и моделировании различных квантовых алгоритмов.

КОНФЛИКТ ИНТЕРЕСОВ

Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

СПИСОК ЛИТЕРАТУРЫ

1. Изучение квантовых вычислений с помощью Qiskit. – Режим доступа: https://qiskit.org/textbook/preface.html. – (Дата обращения: 26.11.2022).

2. Яркие моменты IBM Quantum Summit 2022. – Режим доступа: https://www.ibm.com/quantum. – (Дата обращения: 26.11.2022).

3. Площадка для квантовых вычислений. – Режим доступа: http://www.quantumplayground.net/#/home. – (Дата обращения: 26.11.2022).

4. Документация по Azure Quantum. Что такое Azure Quantum? – Режим доступа: https://learn.microsoft.com/ru-ru/azure/quantum/overview-azure-quantum. – (Дата обращения: 26.11.2022).

5. *Лучников, А. И.* Высокопроизводительный векторный эмулятор вентильного квантового процессора, реализованный на языке программирования Rust [Электронный ресурс] / А. И. Лучников, О. Е. Татаркин, А. К. Фёдоров // Режим доступа: https://arxiv.org/abs/2209.11460. – (Дата обращения: 26.11.2022). DOI: 10.48550/arXiv.2209.11460.

6. *Нильсен, М.* Квантовые вычисления и квантовая информация / М. Нильсен, И. Чанг. – М. : Мир. 2006. – 824 с.

7. *Запрягаев, С. А.* Введение в квантовые информационные системы. Учебник Воронежского госуниверситета / С. А. Запрягаев. – Воронеж : Издательский дом ВГУ, 2015. – 219 с.

Запрягаев Сергей Александрович — д-р физ.-мат. наук, проф., профессор кафедры цифровых технологий Воронежского государственного университета.
ORCID iD: https://orcid.org/0000-0001-8695-5382
E-mail: zsa@cs.vsu.ru

Килигин Егор Александрович — магистрант 2-го года обучения кафедры цифровых технологий Воронежского государственного университета.
ORCID iD: https://orcid.org/0000-0003-4606-6497
E-mail: egor.kiligin@yandex.ru

Косенко Иван Михайлович — магистрант 2-го года обучения кафедры цифровых технологий Воронежского государственного университета.
ORCID iD: https://orcid.org/0000-0001-7641-1672
E-mail: kv1tr4vn@gmail.com

Турченко Константин Анатольевич — магистрант 2-го года обучения кафедры цифровых технологий Воронежского государственного университета.
ORCID iD: https://orcid.org/0000-0003-4860-4076
E-mail: t.costya@yandex.ru

DOI: https://doi.org/10.17308/sait/1995-5499/2022/4/12-22 ISSN 1995-5499
Received 03.10.2022
Accepted 05.12.2022

SIMULATION OF TRANSFORMATIONS FOR QUANTUM REGISTERS

© 2022 S. A. Zapryagaev✉, E. A. Kiligin, I. M. Kosenko, K. A.Turchenko

Voronezh State University
1, Universitetskaya Square, 394018 Voronezh, Russian Federation

Annotation. Quantum information systems are considered as a development of modern IT systems. This article is devoted to the development of a software shell for performing a chain of transformations of multi-qubit quantum registers using standard quantum gates. The use of this software tool is determined by the interest in the problem of analyzing various quantum algorithms and protocols for the transmission of quantum information without direct using of quantum computers.
Keywords: quantum computing, qubit, quantum register, quantum Shor algorithm.

✉ Zapryagaev Sergey A.
e-mail: zsa@cs.vsu.ru

CONFLICT OF INTEREST

The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

REFERENCES

1. Learn Quantum Computation using Qiskit, available at: https://qiskit.org/textbook/preface.html (accessed 26 November 2022).

2. Highlights of the IBM Quantum Summit 2022, available at: https://www.ibm.com/quantum (accessed 26 November 2022).

3. Quantum Computing Playground, available at: http://www.quantumplayground.net/#/home (accessed 26 November 2022).

4. Azure Quantum documentation. What is Azure Quantum?, available at: https://learn.microsoft.com/ru-ru/azure/quantum/overview-azure-quantum (accessed 26 November 2022).

5. *Luchnikov A. I., Tatarkin O. E., Fedorov A. K.* High-performance state-vector emulator of a gate-based quantum processor implemented in the Rust programming language, available at: https://arxiv.org/abs/2209.11460. DOI: 10.48550/arXiv.2209.11460

6. *Nielsen A., Chuang L.* (2006) Quantum Computation and Quantum Information. *Moscow, The World*. 824 p.

7. *Zapryagaev S. A.* (2015) Vvedenie v kvantovye informacionnye sistemy. Uchebnik Voronezhskogo gosuniversiteta [Introduction to quantum information systems. Textbook of Voronezh State University]. *Voronezh, Izdatel'skij dom Voronezhskogo gosudarstvennogo universiteta*. 219 p.

Zapryagaev Sergey A. — Doctor of Physico-mathematical Sciences Professor at the Department of Digital Technologies, Voronezh State University.
ORCID iD: https://orcid.org/0000-0001-8695-5382
E-mail: zsa@cs.vsu.ru

Kiligin Egor A. — Second-year master's student at the Department of Digital Technologies, Voronezh State University.
ORCID iD: https://orcid.org/0000-0003-4606-6497
E-mail: egor.kiligin@yandex.ru

Kosenko Ivan M. — Second-year master's student at the Department of Digital Technologies, Voronezh State University.
ORCID iD: https://orcid.org/0000-0001-7641-1672
E-mail: kv1tr4vn@gmail.com

Turchenko Konstantin A. — Second-year master's student at the Department of Digital Technologies, Voronezh State University.
ORCID iD: https://orcid.org/0000-0003-4860-4076
E-mail: t.costya@yandex.ru