

А. А. МУХАНБЕТ, М. Т. НАКИБАЕВА, Б. С. ДАРИБАЕВ*

Казахский национальный университет имени аль-Фараби, Алматы, Казахстан

e-mail: mukhanbetaksultan0414@gmail.com, m.nakibaeva@gmail.com,

beimbet.daribayev@gmail.com

РАЗРАБОТКА КВАНТОВОГО КАЛЬКУЛЯТОРА НА ПЛАТФОРМЕ IBM QUANTUM EXPERIENCE И ЕГО ПРИЛОЖЕНИЯ ДЛЯ ПРЕОБРАЗОВАНИЯ ДЕСЯТИЧНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДВОИЧНУЮ

В этой статье демонстрируется квантовый калькулятор, экспериментально моделируя и выполняя основные арифметические операции, а именно – сложение, вычитание, умножение, деление на реальных квантовых устройствах с использованием платформы IBM Quantum Experience (IBM QE). Рассмотрены два алгоритма сложения, классическая версия, с помощью вентиль Тоффоли, и с использованием квантового преобразования Фурье (QFT). В частности, суммирование с QFT экономит память и время. Задача работы состоит в том, чтобы построить квантовую схему, выполняющую арифметические операции между двоичным представлением двух целых чисел. Затем мы предлагаем квантовый алгоритм для преобразования десятичного числа в двоичное. Этот метод также допускает массовое распараллеливание во время его выполнения. Функции сложения, вычитания, умножения и деления на основе QFT улучшены с некоторыми изменениями. Предлагаемые операции сравниваются с ближайшими операциями квантовой арифметики. Следовательно, мы можем реализовать практический квантовый калькулятор, выполняя операции на реальном квантовом компьютере. Схемы определены как разновидность квантового сумматора с использованием квантового преобразование Фурье, реализована в Qiskit и продемонстрирована на одном-двух экземплярах.

***Ключевые слова:** квантовый компьютер, квантовые операции, квантовое преобразование Фурье, арифметические операции, qiskit.*

Введение. В последние годы наблюдается огромный прогресс в области квантовых вычислений. IBM через свою платформу IBM Q Experience предоставила доступ к реальным квантовым компьютерам и симулятору, который является отличным ресурсом в области экспериментов. С использованием этой платформы был проведен ряд экспериментов: тестирование существующих квантовых алгоритмов [1,2], моделирование модели Изинга [3], изучение динамики, далекой от равновесия [4], наблюдение парадокса Клейна [5], измерение топологической фазы [6], моделирование квантового туннелирования [7], квантовый искусственный интеллект [8], квантовое машинное обучение [9], разработка новых алгоритмов для сложных задач, решение квантовых игр, проектирование квантовых устройств [10], телепортация квантовых состояний и врат, распознавание квантовых состояний, квантовая информация, квантовая коррекция ошибок, и это лишь некоторые из них.

Калькулятор в первую очередь включает в себя четыре основных арифметических операции: сложение, вычитание, умножение и деление. Для реализации вышеуказанных операций были разработаны квантовые алгоритмы. Однако их реализация на квантовом компьютере или на квантовом симуляторе до настоящего времени не была

* E-mail корреспондирующего автора: mukhanbetaksultan0414@gmail.com

выполнена. Здесь мы приводим краткий обзор существующих алгоритмов и имитируем схемы сложения, вычитания и умножения на симуляторе классической топологии IBM. Хотя классические компьютеры могут эффективно выполнять арифметические операции, необходимость в квантовом алгоритме возникает потому, что они необходимы для других алгоритмов, например, алгоритм Шора требует модульных арифметических операций [30].

2. Создание калькулятора. Прежде всего, реализуем операцию сложения так же, как это реализовано на классическом компьютере (конструкция полного сумматора) и с помощью вентиля Тоффоли. Далее мы обрисуем недостатки этих подходов и проанализируем преимущества использования вместо них квантового преобразования Фурье (QFT).

Затем мы продолжим отчет, реализуя, исходя из соображений, написанных о QFT, все другие операции, такие как вычитание, умножение и деление. Каждая операция будет протестирована как на автономном симуляторе Qiskit, так и на онлайн-симуляторе, предоставленном IBM.

2.1 Сложение. Мы начнем с перевода суммы двух чисел в квантовую схему. На самом деле существует много способов решить эту проблему, и мы покажем два из них. Первый называется «суммой Тоффоли», поскольку в нем используются вентили Тоффоли и CNOT. Это более простая схема, поскольку она имитирует хорошо известную классическую схему суммирования (сумматор с половинным сумматором и сумматор с переносом импульса).

Второй подход заключается в использовании квантового преобразования Фурье (QFT), которое преобразует между двумя базисами, вычислительным (Z) базисом и базисом Фурье. Используя этот метод, мы подсчитываем различные вращения вокруг оси Z .

Прежде чем углубляться в сам код, важно понять основные принципы.

Сложение Тоффоли основано на классических схемах двоичной суммы: половинном сумматоре и сумматоре с переносом импульса (RCA).

Давайте сначала взглянем на схему полусумматора, которая показана на рисунке 1.

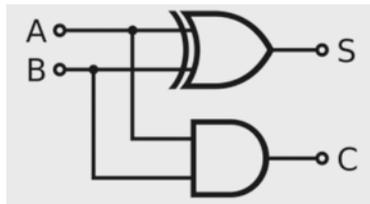


Рисунок 1 – Классическая схема полусумматора

Схема полусумматора выполняет суммирование двух входных битов A и B (где они могут быть 0 или 1), выводя сложение на выходе S .

Кроме того, он вычисляет бит переноса в выводе C . Бит переноса равен единице, когда A и B равны единице.

Однако схема не учитывает возможный бит переноса. Эта проблема решается в схеме полного сумматора.

Схема полусумматора состоит из элемента И и элемента XOR. Два входных сигнала передаются как на логический элемент XOR, который будет отвечать за вычисление частичной суммы, так и на логический элемент AND, который будет отвечать за вычисление переноса. Таблица истинности полусумматора показана на таблице 1.

Таблица 1 – Таблица истинности полусумматора

ВВОД		ВЫВОД	
A	B	Сумма (S)	Перенос (Cout)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Из таблицы мы можем сделать вывод, что функция добавления может быть описана уравнениями (1)-(2):

$$S = A_i \otimes B_i \quad (1)$$

$$C_{out} = A_i \cdot B_i \quad (2)$$

Как упоминалось ранее, половинный сумматор не учитывает возможный переносной бит, что делает схему пригодной только для добавления двух одиночных битов. Чтобы рассмотреть бит переноса, мы должны построить схему, которая принимает 3 входа: A, B и Cin, где первые два – это биты для добавления, а последний – бит переноса, вычисленный предыдущей операцией. Эта схема называется полным сумматором.

Логическими выражениями для полного сумматора являются следующие:

$$S = (A_i \otimes B_i) \otimes Cin_i \quad (3)$$

$$C_{out} = (A_i \cdot B_i) + ((A_i \otimes B_i) \cdot Cin_i) \quad (4)$$

При добавлении битов мы используем больше одной пары. Для этого мы можем использовать сумматор с переносом пульсаций(RCA), т.е. множество полных сумматоров в каскаде.

2.1.2 Добавление с квантовым преобразованием Фурье. Прежде чем мы начнем обсуждать реализацию сложения с использованием квантового преобразования Фурье (QFT), мы представим математическую формулу и то, как можно реализовать схему, которая выполняет эту операцию. Как упоминалось ранее, QFT представляет собой квантовую реализацию дискретного преобразования Фурье по амплитудам волновой функции и преобразует вычислительный базис в базис Фурье. Давайте посмотрим пример с одним кубитом: мы знаем, что с одним кубитом у нас есть состояния ноль ($|0\rangle$) и состояние один ($|1\rangle$), так что есть два основных состояния. Для

двух кубитов у нас есть $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, т.е. 2^2 возможных базовых состояний. Обобщая этот процесс, мы будем иметь с n кубитами 2^n возможных состояний. Итак, для простоты давайте рассмотрим $N = 2^n$ и $|\tilde{x}\rangle = \text{QFT}|x\rangle$, где \tilde{x} – упомянутый ранее базис Фурье, а x – наша вычислительная основа. Итак, у нас есть формула QFT:

$$\text{QFT}|x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{2\pi ixy}{N}} |y\rangle \quad (5)$$

в случае одного кубита мы имеем:

$$\begin{aligned} \text{QFT}|x\rangle &= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{2\pi ixy}{N}} |y\rangle = \frac{1}{\sqrt{2}} \sum_{y=0}^{2-1} e^{\frac{2\pi ixy}{2}} |y\rangle \geq \frac{1}{\sqrt{2}} \left[e^{\frac{2\pi ix0}{2}} |0\rangle + e^{\frac{2\pi ix1}{2}} |1\rangle \right] \\ &= \frac{1}{\sqrt{2}} [|0\rangle + e^{i\pi x} |1\rangle] = \frac{1}{\sqrt{2}} [|0\rangle + |1\rangle] = |+\rangle, \text{ когда } x = 0 \\ &= \frac{1}{\sqrt{2}} [|0\rangle - |1\rangle] = |-\rangle, \text{ когда } x = 1 \end{aligned} \quad (6)$$

Теперь, если мы рассмотрим это состояние одного кубита $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ и рассчитаем QFT, как это делалось ранее, эта операция будет в точности результатом применения оператора Адамара (H) к кубиту:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (7)$$

QFT можем переписать, как:

$$\text{QFT}_N |x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \omega_N^{xy} |y\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi ixy/2^n} |y\rangle, \text{ где } \omega_N^{xy} = e^{\frac{2\pi ixy}{2^n}} \text{ и } N = 2^n \quad (8)$$

перепишав в дробной двоичной записи $y = y_1 \dots y_n$, $\frac{y}{2^n} = \sum_{k=1}^n y_k / 2^k$, получим:

$$\frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i \left(\sum_{k=1}^n y_k / 2^k \right) x} |y_1 \dots y_n\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \prod_{k=1}^n e^{2\pi i y_k x / 2^k} |y_1 \dots y_n\rangle \quad (9)$$

после перестановки суммы и произведений и расширения

$$\begin{aligned} &\frac{1}{\sqrt{N}} \otimes_{k=1}^n \left(|0\rangle + e^{\frac{2\pi ix}{2^k}} |1\rangle \right) \sum_{y=0}^{N-1} = \sum_{y_1=0}^1 \sum_{y_2=0}^1 \dots \sum_{y_n=0}^1 \\ &= \frac{1}{\sqrt{N}} \left(|0\rangle + e^{\frac{2\pi ix}{2^1}} |1\rangle \right) \otimes \left(|0\rangle + e^{\frac{2\pi ix}{2^2}} |1\rangle \right) \otimes \dots \otimes \left(|0\rangle + e^{\frac{2\pi ix}{2^{n-1}}} |1\rangle \right) \end{aligned} \quad (10)$$

Для построения схемы QFT нам нужны следующие вентили:

- вентиль Адамара: вентиль Адамара применяется к конкретному кубиту x_k , поэтому x_k равно нулю или единице имеет два возможных результата, которые можно выразить в качестве $H|x_k\rangle \geq |0\rangle + e^{\frac{2\pi ix_k}{2}} |1\rangle / \sqrt{2}$.

• двухкубитное управляемое вращение $CROT_k = \begin{bmatrix} I & 0 \\ 0 & UROT_k \end{bmatrix}$, где $UROT_k =$

$$= \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(\frac{2\pi i}{2^k}\right) \end{bmatrix}.$$

С учетом этих двух вентилях показана схема, реализующая 4-кубитную QFT.



Рисунок 2 – Схема, реализующая 4-кубитную QFT

2.2 Вычитание квантового преобразования Фурье. Если мы подумаем о том, как происходит вычитание между двоичными числами, то в классических вычислениях мы знаем, что мы можем получить результат, используя «одно дополнение» сложения, т.е. инверсируем отрицательное число и добавим единицу, затем находим сумму полученного результата и первым исходным числом и, наконец, удаляем первую цифру суммы. Если у нас нет лишней цифры, мы пытаемся вычесть большее число из меньшего.

То есть, если мы приведем простой пример, у нас будет:

$$1101_2 - 0111_2 = 1101_2 + 1000_2 = 1101_2 + (1000_2 + 0001_2) = 1101_2 + 1001_2 = 10110_2 = \cancel{1}0110_2 = 0110_2$$

И снова использование QFT позволяет нам пропустить множество шагов и сохранить кубиты.

Алгоритм, которому мы следуем для вычитания, по существу идентичен алгоритму, использованному для сложения, но вместо выполнения положительных вращений в функции, преобразует предыдущее кодированное состояние $|F(\psi(\text{reg}_a))\rangle$ первого числа в $|F(\psi(\text{reg}_a + \text{reg}_b))\rangle$, мы выполняем отрицательные вращения.



Рисунок 3 – QFT: схема вычитания 4-кубитов

2.3 Умножение квантового преобразования Фурье. Даже в десятичной системе счисления произведение между a и b может быть получено путем добавления первого множителя, a , столько раз, сколько указывает b .

Поскольку это свойство не зависит от системы счисления, мы используем его для умножения чисел в двоичном формате.

Давайте приведем пример, умножив $6_{10} \cdot 3_{10}$:

$$\begin{aligned} 6_{10} \cdot 3_{10} &= 0_{10} + 6_{10} = 6_{10} \\ &= 6_{10} + 6_{10} = 12_{10} \\ &= 12_{10} + 6_{10} = 18_{10} \end{aligned}$$

где 18_{10} – это результат умножения.

Чтобы доказать, что это работает даже при использовании двоичных чисел, смотрите следующий пример:

$$\begin{aligned} 6_{10} \cdot 3_{10} &= 110_2 \cdot 011_2 \\ &= 0_2 + 110_2 = 110_2 \\ &= 110_2 + 110_2 = 1100_2 \\ &= 1100_2 + 110_2 = 10010_2 \end{aligned}$$

где 10010_2 – это 18_{10} в десятичной системе счисления.

В начале умножения мы кодируем только a в кубитах, и поскольку значение b используется только для того, чтобы знать, сколько раз нам нужно повторить сложение a , мы решили не преобразовывать его в квантовый регистр, а сохранить как обычное целое число.

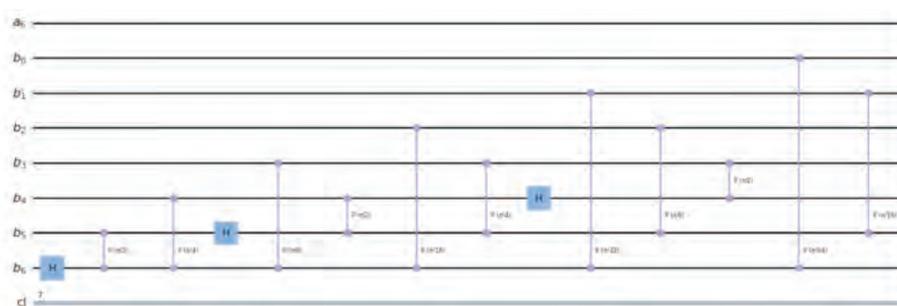


Рисунок 4 – Схема QFT умножения

2.4 QFT Деление. Для реализации последней элементарной операции нам нужны два квантовых регистра для кодирования деления и делителя и дополнительный квантовый регистр для использования в качестве аккумулятора. Давайте посмотрим на процесс выполнения целочисленного деления в следующем примере. Предположим, мы хотим разделить число $6_{10} = 110_2$ на $2_{10} = 010_2$, что мы собираемся сделать, это продолжать вычитать из делимого делитель до тех пор, пока вычитание, реализованное с помощью QFT, не приведет кубит в состояние, которое будет соответствовать числу (в двоичной системе счисления), большему, чем сам делимый. Например:

$$\begin{aligned}
 & accumulator = 0 \\
 6 - 2 &= 0110 - 0010 = 4 \text{ (0100) } accumulator + = 1 \\
 4 - 2 &= 0100 - 0010 = 2 \text{ (0010) } accumulator + = 1 \\
 2 - 2 &= 0010 - 0010 = 0 \text{ (0000) } accumulator + = 1 \\
 0 - 2 &= 1110 \quad accumulator + = 1
 \end{aligned}$$

На этом этапе мы выходим из цикла, и если мы проверим значение счетчика, оно будет равно 4, но мы вычли слишком много, поэтому мы вычитаем 1, чтобы получить 3, что является правильным результатом. Мы замечаем, что интуитивно то, что мы делаем – это итеративное вычитание до тех пор, пока вращения по оси Z в базе Фурье не приведут к неправильному результату, на самом деле, как мы можем видеть, если мы попытаемся вычесть 2 из 0, мы получим недостаточный поток, то есть с поворотом от 0000_2 мы переходим к 1111_2 и с другим поворотом приходим к 1110_2 .

Перед началом цикла while мы создаем кубит d и устанавливаем ему значение 1, которое будет использоваться в качестве константы для добавления в накопитель на каждой итерации. Более того, чтобы знать, когда выходить из цикла, мы проверяем первую наиболее значимую цифру результата вычитания между делением и делителем, потому что, если в начале первая цифра не была 1, она наверняка никогда не станет 1, поскольку мы выполняем вычитания. Чтобы быть уверенным в начале цикла, мы заставляем первую цифру быть равной 0. Отметим, что, учитывая принцип неопределенности Гейзенберга, мы не можем

знать значение состояния кубита, пока не измерим его. По этой причине, чтобы узнать значение вычитания между делимым и делителем, необходимо выполнять измерения на каждой итерации. Результаты измерения затем используются для извлечения первой наиболее значимой цифры, чтобы узнать, следует ли продолжать цикл.

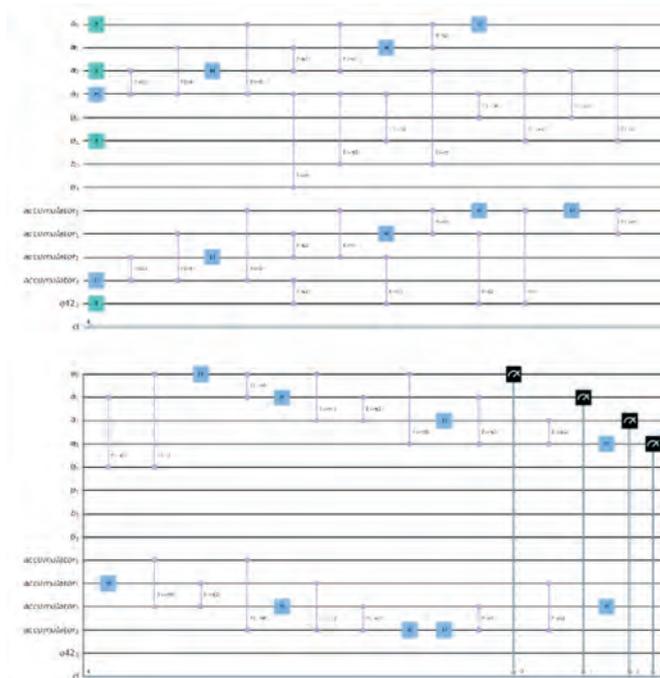


Рисунок 5 – Схема QFT деления

3. Результаты. После построения квантовой схемы был написан скрипт для проверки результатов. Сперва нужно выбрать операцию, которую мы хотим выполнить. Результат можно увидеть на 6 рисунке.

```
[92m#####<[0m
[92m##### Квантовый Настольный Калькулятор #####<[0m
[92m#####<[0m
[93m
Выберите один оператор [+ сложение, - вычитание, * умножение, / деление]: <[0m*
```

Рисунок 6 – Выбор арифметической операции

После выбора арифметической операции надо ввести первое и второе положительное целое число от 0 до 2047. Введенные числа преобразовываются в двоичную систему и выполняется арифметическая операция. Для вычисления используется qasm_simulator. Результат умножения можно увидеть на 7 рисунке.

```

Выберите один оператор [+ сложение, - вычитание, * умножение, / деление]: +[0m*
+[93mВведите первое положительное целое число от 0 до 2047:
+[0m5
+[93mВведите второе положительное целое число от 0 до 2047:
+[0m4

+[96m#####
#####+[0m
+[1m+[96mВы хотите выполнить следующую операцию:+[0m
+[1m+[96m5 * 4 --> 000101 * 100 = ...+[0m
+[1m+[96mПодключение к локальному симулятору...+[0m
+[1m+[96mПодключить!+[0m
+[1m+[96mЗапуск эксперимента на 100 выстрелы...+[0m
+[1m+[92m
000101 * 100 = 0010100 with a probability of 100%+[0m
+[96m#####
#####+[0m

```

Рисунок 7 – Результат квантового умножения

В результате можно увидеть умножение двух чисел. Это $101_2 * 100_2 = 10100_2$. Если перевести это десятичную систему, то результат будет $5_{10} * 4_{10} = 20_{10}$.

```

Выберите один оператор [+ сложение, - вычитание, * умножение, / деление]: +[0m/
+[93mВведите первое положительное целое число от 0 до 2047:
+[0m30
+[93mВведите второе положительное целое число от 0 до 2047:
+[0m6

+[96m#####
#####+[0m
+[1m+[96mВы хотите выполнить следующую операцию:+[0m
+[1m+[96m30 / 6 --> 11110 / 00110 = ...+[0m
+[1m+[96mПодключение к локальному симулятору...+[0m
+[1m+[96mПодключить!+[0m
+[1m+[96mЗапуск эксперимента на 100 выстрелы...+[0m
+[1m+[92m
11110 / 00110 = 000101 with a probability of 100%+[0m
+[96m#####
#####+[0m

```

Рисунок 8 – Результат QFT деление

В рисунке 8 можно увидеть, что деление двух целых чисел. $11110_2 : 00110_2 = 000101_2$. Если перевести это десятичную систему, то результат будет $30_{10} / 6_{10} = 5_{10}$.

Таблица 2 – Результаты выполнения арифметических операций по времени

Операции	Время выполнения
Сложение	1.65797615051269
Вычитание	1.67863130569458
Умножение	1.71801996231079
Деление	3.72493243217468

В таблице 2 можно увидеть результаты выполнения арифметических операций по времени. Результат времени деления занимает много времени, если сравнить с остальными, потому что там выполняется 2 QFT операции. Это сложение и вычитание.

Заключение. В заключение мы продемонстрировали основные арифметические операции, которые составляют основу калькулятора и квантового алгоритма преобразования десятичного числа в его двоичную форму. Большинство предыдущих работ по квантовым арифметическим операциям носят теоретический характер, и было предпринято несколько попыток моделирования или реализации на квантовом компьютере. Целью этой работы было моделирование и запуск алгоритмов на реальных квантовых устройствах с использованием доступных в настоящее время ресурсов. Таким образом, алгоритмы, смоделированные здесь, не самые эффективные, но самые простые для моделирования с использованием современных технологий. В случае разделения, как упоминалось ранее, даже самые простые подходы имеют ограничения, когда дело доходит до моделирования, мы предложили более общие и эффективные алгоритмы. Мы предложили квантовый алгоритм для преобразования десятичного числа в его двоичную форму. Мы надеемся, что с большей связью между кубитами и более универсальными операциями станет возможным реализовать эффективные алгоритмы для арифметических операций наряду с другими полезными операциями на реальных устройствах квантового компьютера, чтобы его можно было использовать в качестве квантового калькулятора.

Благодарность. Научно-исследовательская работа выполнена за счет гранта Комитета науки МОН РК по проекту № AP09260564.

ЛИТЕРАТУРА

- 1 Gangopadhyay, S., Manabputra, Behera, B. K., Panigrahi, P. K.: Generalization and Demonstration of an Entanglement Based Deutsch-Jozsa Like Algorithm Using a 5-Qubit Quantum Computer, *Quantum Inf. Process.* 17, 160 (2018)
- 2 Coles, P. J. et al.: Quantum Algorithm Implementations for Beginners, arXiv:1804.03719 (2018)
- 3 Cervera-Lierta, A.: Exact Ising model simulation on a Quantum Computer, *Quantum* 2, 114 (2018).
- 4 Zhukov, A. A., Remizov, S. V., Pogosov, W. V., Lozovik, Y. E.: Algorithmic simulation of far-from-equilibrium dynamics using quantum computer, *Quantum Inf. Process.* 17, 223 (2018)
- 5 Kapil, M., Behera, B. K., Panigrahi, P. K.: Quantum Simulation of Klein Gordon Equation and Observation of Klein Paradox in IBM Quantum Computer, arXiv:1807.00521 (2018)
- 6 Viyuela, O. et al.: Observation of topological Uhlmann phases with superconducting qubits, *npj Quantum Inf.* 4, 10 (2018)
- 7 Hegade, N. N., Behera, B. K., Panigrahi, P. K.: Experimental Demonstration of Quantum Tunneling in IBM Quantum Computer, arXiv:1712.07326 (2017)
- 8 Alvarez-Rodriguez, U., Sanz, M., Lamata, L., Solano, E.: Quantum Artificial Life in an IBM Quantum Computer, *Sci. Rep.* 8, 14793 (2018).
- 9 Zhao, Z., Pozas-Kerstjens, A., Rebentrost, P., Wittek, P.: Bayesian Deep Learning on a Quantum Computer, *Quantum Mach. Intel.* 1, 4 (2019).
- 10 Srinivasan, K., Satyajit, S., Behera, B. K., Panigrahi, P. K.: Efficient quantum algorithm for solving travelling salesman problem: An IBM quantum experience, arXiv:1805.10928 (2018)

REFERENCES

- 1 Gangopadhyay, S., Manabputra, Behera, B. K., Panigrahi, P. K.: Generalization and Demonstration of an Entanglement Based Deutsch-Jozsa Like Algorithm Using a 5-Qubit Quantum Computer, Quantum Inf. Process. 17, 160 (2018)
- 2 Coles, P. J. et al.: Quantum Algorithm Implementations for Beginners, arXiv:1804.03719 (2018)
- 3 Cervera-Lierta, A.: Exact Ising model simulation on a Quantum Computer, Quantum 2, 114 (2018).
- 4 Zhukov, A. A., Remizov, S. V., Pogosov, W. V., Lozovik, Y. E.: Algorithmic simulation of far-from-equilibrium dynamics using quantum computer, Quantum Inf. Process. 17, 223 (2018)
- 5 Kapil, M., Behera, B. K., Panigrahi, P. K.: Quantum Simulation of Klein Gordon Equation and Observation of Klein Paradox in IBM Quantum Computer, arXiv:1807.00521 (2018)
- 6 Viyuela, O. et al.: Observation of topological Uhlmann phases with superconducting qubits, npj Quantum Inf. 4, 10 (2018)
- 7 Hegade, N. N., Behera, B. K., Panigrahi, P. K.: Experimental Demonstration of Quantum Tunneling in IBM Quantum Computer, arXiv:1712.07326 (2017)
- 8 Alvarez-Rodriguez, U., Sanz, M., Lamata, L., Solano, E.: Quantum Artificial Life in an IBM Quantum Computer, Sci. Rep. 8, 14793 (2018).
- 9 Zhao, Z., Pozas-Kerstjens, A., Reberntrost, P., Wittek, P.: Bayesian Deep Learning on a Quantum Computer, Quantum Mach. Intel. 1, 4 (2019).
- 10 Srinivasan, K., Satyajit, S., Behera, B. K., Panigrahi, P. K.: Efficient quantum algorithm for solving travelling salesman problem: An IBM quantum experience, arXiv:1805.10928 (2018)

A. A. МУХАНБЕТ, М. Т. НАКИБАЕВА, Б. С. ДАРИБАЕВ

*Әл-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан
e-mail: mukhanbetaksultan0414@gmail.com, m.nakibaeva@gmail.com,
beimbet.daribayev@gmail.com*

IBM QUANTUM EXPERIENCE ПЛАТФОРМАСЫНДА КВАНТТЫҚ КАЛЬКУЛЯТОРДЫ ЖАСАУ ЖӘНЕ ОНЫҢ ОНДЫҚ САНАУ ЖҮЙЕСІН ЕКІЛІК САНАУ ЖҮЙЕСІНЕ ТҮРЛЕНДІРУГЕ АРНАЛҒАН ҚОЛДАНБАСЫ

Бұл мақала IBM Quantum Experience (IBM QE) платформасын пайдаланып нақты кванттық құрылғыларда негізгі арифметикалық амалдарды, атап айтқанда қосу, алу, көбейту, бөлуді эксперименттік модельдеу және орындау арқылы кванттық калькуляторды көрсетеді. Қосудың екі алгоритмі қарастырылады: классикалық нұсқасы, Тоффоли қақпасы және кванттық Фурье түрлендіруі (QFT). Атап айтқанда, QFT көмегімен жинақтау жады мен уақытты үнемдейді. Жұмыстың міндеті – екі бүтін санның екілік кескіні арасында арифметикалық амалдарды орындайтын кванттық схеманы құру. Содан кейін ондық санауды екілік санау жүйесіне түрлендірудің кванттық алгоритмін ұсынамыз. Бұл әдіс сонымен қатар оны орындау кезінде жаппай параллелизацияға мүмкіндік береді. QFT негізіндегі қосу, алу, көбейту және бөлу функциялары кейбір өзгерістермен жақсартылды. Ұсынылған амалдар ең жақын кванттық арифметикалық амалдармен салыстырылады. Сондықтан нақты кванттық компьютерде амалдарды орындау арқылы практикалық кванттық калькуляторды жүзеге асыра аламыз. Тізбектер Kiscuitte іске

асырылған және бір немесе екі көшірмеде көрсетілген кванттық Фурье түрлендіруін қолданатын кванттық қосқыштың бір түрі ретінде анықталады.

Түйін сөздер: кванттық компьютер, кванттық операциялар, кванттық Фурье түрлендіруі, арифметикалық операциялар, qiskit

A. A. MUKHANBET, M. T. NAKIBAYEVA, B. S. DARIBAYEV

Al-Farabi Kazakh National University, Almaty, Kazakhstan
e-mail: mukhanbetaksultan0414@gmail.com, m.nakibaeva@gmail.com,
beimbet.daribayev@gmail.com

DEVELOPMENT OF A QUANTUM CALCULATOR ON THE IBM QUANTUM EXPERIENCE PLATFORM AND ITS APPLICATION FOR CONVERTING THE DECIMAL NUMBER SYSTEM TO BINARY

This article demonstrates a quantum calculator by experimentally simulating and performing basic arithmetic operations, namely addition, subtraction, multiplication, division on real quantum devices using the IBM Quantum Experience (IBM QE) platform. Two addition algorithms are considered, the classical version, using the Toffoli gate, and using the quantum Fourier transform (QFT). In particular summation with QFT saves memory and time. The task of the work is to build a quantum circuit that performs arithmetic operations between the binary representation of two integers. We then propose a quantum algorithm for converting decimal to binary. This method also allows for massive parallelization during its execution. The addition, subtraction, multiplication, and division functions based on QFT have been improved with some changes. The proposed operations are compared with the closest quantum arithmetic operations. Therefore, we can implement a practical quantum calculator by performing operations on a real quantum computer. The circuits are defined as a kind of quantum adder using a quantum Fourier transform, implemented in Qiskit and demonstrated on one or two copies.

Keywords: quantum computer, quantum operations, quantum Fourier transform, arithmetic operations, qiskit.