

# ОПТИМИЗАЦИЯ ТРАДИЦИОННЫХ МЕТОДОВ ЭМУЛЯЦИИ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА ДЛЯ ВЫПОЛНЕНИЯ НА КВАНТОВОМ КОМПЬЮТЕРЕ

## OPTIMISATION OF TRADITIONAL METHODS OF EMULATING A CENTRAL PROCESSOR FOR EXECUTION ON A QUANTUM COMPUTER

S. Tyryshkin

*Summary.* Over the past few years, the field of quantum algorithm optimisation has demonstrated significant development and progress. Scientists have developed advanced methods and approaches that reduce the depth of quantum circuits, minimise the number of gates, and increase the overall performance of quantum computing. At the same time, quantum technology mostly functions at the level of logic and scenarios, where a significant part of the achievements made in the field of traditional computing are inaccessible and very difficult to implement. The aim of this article is to study the possibilities of optimising traditional methods of emulating a central processing unit for execution on a quantum computer. *Research methods* include comparative analysis, formal logic, structuring, grouping, generalisation, and a systematic approach. *Results:* The study analysed some traditional emulation methods, highlighting their advantages and disadvantages, as well as their potential and limitations. In addition, the performance and resource intensity indicators of these methods were formalised. Emphasis is placed on promising areas of development and opportunities for optimising traditional central processing unit instructions for execution on a quantum computer. *Conclusions.* The article demonstrates alternative methods for executing traditional central processing unit commands on a quantum computer. The results obtained allow us to conclude that the integration and combination of various emulation methods will make it possible to increase the accessibility and usability of quantum computers in the transition from traditional computing.

*Keywords:* quantum computing, emulation, central processing unit, traditional method, circuit, algorithm.

Тырышкин Сергей Юрьевич

кандидат технических наук,

ФГБОУ ВО Алтайский государственный технический  
университет им. И.И. Ползунова, г. Барнаул

service.vip-spe@yandex.ru

*Аннотация.* На протяжении последних нескольких лет сфера оптимизация квантовых алгоритмов демонстрирует значительное развитие и прогресс. Учеными были разработаны передовые методы и подходы, которые позволяют уменьшить глубину квантовых схем, минимизировать количество вентилей и повысить общую производительность квантовых вычислений. В то же время квантовая технология в большинстве своем функционирует на уровне логики и сценариев, где значительная часть достижений, сделанных в области традиционных вычислений, недоступна и сложна для реализации. С учетом отмеченного цель статьи заключается в изучении возможностей оптимизации традиционных методов эмуляции центрального процессора для выполнения на квантовом компьютере. *Методы исследования* включают в себя сравнительный анализ, формальную логику, структуризацию, группировку, обобщение, системный подход. *Результаты:* в процессе исследования проведен анализ некоторых традиционных методов эмуляции, что позволило выделить их достоинства и недостатки, возможности и ограничения использования. Кроме того, формализованы показатели эффективности и ресурсоемкости этих методов. Особый акцент сделан на перспективных направлениях развития и возможностях оптимизации традиционных команд центрального процессора для выполнения на квантовом компьютере. *Выводы.* В статье продемонстрированы общепринятые и альтернативные методы с использованием квантовых схем, которые могут использоваться для эмуляции центрального процессора на квантовом компьютере. Полученные результаты позволяют сделать заключение, что интеграция и комбинирование различных подходов к эмуляции даст возможность повысить доступность и удобство использования квантовых компьютеров при переходе от традиционных вычислений.

*Ключевые слова:* квантовые вычисления, эмуляция, центральный процессор, традиционный метод, схема, алгоритм.

## Введение

Квантовые вычисления на сегодняшний день вызывают большой интерес у инженеров-программистов, поскольку они намного быстрее традиционных подходов. Например, они способны осуществлять факторизацию и поиск в базах данных более эффективно, чем обычные компьютеры. Криптографические алгоритмы на базе квантовых схем повышают безопасность, предотвращая потерю или кражу информации при передаче данных через Интернет [1]. Продолжающаяся эволюция аппаратного обеспечения для квантовых компью-

теров и совершенствование квантовых схем решения задач достаточно убедительно продемонстрировали для конкретных приложений превосходство квантовых вычислений над классическими. Внутренний параллелизм, основанный на принципах суперпозиции квантовых состояний, в сочетании с эффектами интерференции приводит к экспоненциальному росту вычислительных возможностей, что открывает широкие перспективы в области различного рода исследований [2].

Моделирование и решение прикладных проблем с помощью квантовых алгоритмов может продемонстри-

ровать ряд потенциальных преимуществ по сравнению с существующими методами, к числу которых относятся:

1. Ускорение работы существующих классических алгоритмов оптимизации за счет использования специфических квантовых эффектов.
2. Некоторые классы задач могут быть закодированы и решены квантовыми вычислительными схемами более эффективно, чем в ходе применения классических подходов, например задачи неупорядоченного поиска или дискретные логарифмы.
3. Теория и традиционные симуляции показывают, что для некоторых сфер квантовые алгоритмы позволяют применять более эффективные методы к решению определенных классов сложных задач комбинаторной оптимизации.

В таблице 1 представлена статистика, наглядно отражающая преимущества квантовых алгоритмов в задачах оптимизации и анализа.

Современные подходы к квантовым вычислениям используют квантовые системы, такие как фотоны, захваченные ионы или сверхпроводящие цепи. Все эти подходы основаны на поддержании высококогерентного квантового состояния с помощью серии операций с вентилями, чтобы добиться вычислительного преимущества [3]. Подготовка и манипулирование такими системами является довольно сложной задачей, поскольку небольшие взаимодействия с окружающей средой быстро приводят к декогеренции состояния и, как следствие, к значительной потере производительности. Недавнее открытие классических аналогов квантовых систем, позволяет предположить, что эмуляция квантовых вычислений средствами классической архитектуры может быть практически реализуема, технически менее сложна и в значительной степени устойчива к эффектам декогеренции.

Таким образом, можно утверждать, что в отсутствие масштабируемых квантовых компьютеров, квантовые

симуляторы являются важными инструментами для проектирования и оценки квантовых алгоритмов и сравнения уровня шума результатов реальных вычислительных устройств с данными симуляции. С учетом отмеченного, очевидным является тот факт, что оценка квантовых алгоритмов, поддерживаемых квантовыми симуляторами, которые работают на классических машинах, является реальной возможностью для расширения участия исследовательских и экспертных сообществ в работе с квантовыми вычислениями.

В то же время необходимо акцентировать внимание на том, что квантовые системы, моделируемые на классических компьютерах, никогда не будут обладать всеми преимуществами реального физического квантового компьютера за полиномиальное время. В данном контексте цель создания симулятора квантового компьютера — сделать его более быстрым, надежным, способным обрабатывать несколько кубитов и моделировать квантовые алгоритмы с более высокой производительностью. Другими словами, задача заключается в создании комплекса средств разработки программного обеспечения, т.е. набора новых технологий, знаний и возможностей, которые будут использоваться для эмуляции традиционного центрального процессора (CPU) для выполнения на квантовом компьютере. Традиционный CPU очень хорош в низкоуровневой логике и математике, тогда как выполнение этих задач на квантовом компьютере чрезвычайно сложно по причине значительно отличающейся архитектуры и необходимости соблюдения квантовыми компьютерами теорем о невозможности клонирования и удаления. Ожидается, что эмуляция традиционного CPU будет способна обеспечить аргументированное доказательство точности и возможностей битовых, математических и логических операций, а также устранил и восполнит некоторые пробелы, присутствующие в конвейере квантовых вычислений.

Таким образом, изучение аспектов, связанных с ключевыми компонентами реализации потенциала кванто-

Таблица 1.  
Сравнение производительности квантовых и классических алгоритмов в задачах оптимизации и анализа

Тип задачи / Алгоритм	Квантовый подход	Классический подход	Преимущество квантового подхода
Булевы задачи с ограничениями второго порядка (до 156 переменных)	Решение с использованием квантового алгоритма на 156-кубитном процессоре IBM (время $\approx 0,5$ секунды)	Коммерческий решатель CPLEX (30–50 секунд); метод имитации отжига ( $\approx 1$ –2 секунды)	До 80 раз быстрее, чем CPLEX; в 3 раза быстрее, чем имитация отжига
Задачи кластеризации данных (свыше 1000 переменных)	Квантовый отжиг на квантовом процессоре D-Wave демонстрирует ускорение порядка $10^8$	Имитация отжига на классических компьютерах	Ускорение в миллионы раз в отдельных задачах кластеризации
Факторизация целых чисел	Квантовый алгоритм Шора решает задачу за полиномиальное время $O((\log M)^3)$	Классические методы (например, метод решёток) работают в сверхполиномиальном времени	Теоретическое экспоненциальное ускорение

Примечание: составлена автором на основе полученных данных в ходе исследования

вых вычислений и улучшением удобства использования квантовых компьютеров, которые позволят преодолеть разрыв в доступности между классическими алгоритмами и новой квантовой схемой решения задач, представляет собой актуальное направление научного поиска, которое и предопределило выбор темы данной статьи.

### Цель исследования

Изучение возможностей оптимизации традиционных методов эмуляции центрального процессора для выполнения на квантовом компьютере.

### Материал и методы исследования

Существующие на сегодняшний день исследования в основном сосредоточены на квантовых вычислениях и их приложениях и содержат беглый обзор квантовых симуляторов как одного из классов квантового программного обеспечения. Например, Mohammed Alghadeer, Eid Aldawsari [4], Пелешенко В.А. [5], Nikita Stroeve, Natalia G. [6] изучают и классифицируются языки квантового программирования.

Роль различных типов инструментов, таких как компиляторы и симуляторы для эмуляции рассматривают Tian Chen, Weixuan Zhang, Deyuan Zou [7], Karthik Krishnan, Saranyan Vijayaraghavan [8], Милай М.В. [9].

Перспективные подходы к разработке и усовершенствованию структуры новых методов, включающих квантовые схемы, а также возможности их объединения для эмуляции традиционного центрального процессора на квантовом компьютере, нашли свое отражение в публикациях Черемисина Д.Г., Мкртчян В.Р., Музловой А.Д. [10], Рахимов Р.Х. [11], Guillem Llodrà, Christos Charalambous [12].

Подходы, благодаря которым возможности квантовых компьютеров могут быть расширены с помощью традиционных логических функций, с фокусировкой внимания на максимально точной эмуляции традиционного центрального процессора, описывают Giuliana Siddi Moreau, Lorenzo Pisani [13], Ульянов С.В., Рябов Н.В., Зрелов П.В., Иванцова О.В., Кореньков В.В. [14].

Несмотря на активный интерес ученых к рассматриваемой проблематике и наличие широкого спектра публикаций, следует отметить, что задача эмуляции классических центральных процессоров на квантовых компьютерах является достаточно сложной и находится на ранних стадиях исследования, поскольку квантовые архитектуры значительно отличаются от традиционных. Это в свою очередь обуславливает наличие широкого спектра дискуссионных и недостаточно освещенных вопросов. Так, например, в дальнейшем развитии нужда-

ются подходы к созданию симулятора квантового компьютера, которые позволят сделать его более быстрым, надежным, способным обрабатывать несколько кубитов и моделировать квантовые алгоритмы с более высокой производительностью. Кроме того, дополнительного обоснования требуют критерии выбираемых методов квантовой реализации для каждой команды центрального процессора, что позволит пользователю сравнить исходные функциональные методы с новыми оптимизированными подходами.

Методы исследования: сравнительный анализ, формальная логика, структуризация, группировка, обобщение, системный подход.

### Результаты исследования и их обсуждение

Прежде всего отметим, что эмуляция центрального процессора для выполнения на квантовом компьютере заключается в выполнении реального приложения на платформе, которую можно изменить с помощью специального программного или аппаратного обеспечения, чтобы воспроизвести желаемые экспериментальные условия [15]. Часто бывает сложно провести эксперименты в реальной среде, которая соответствует потребностям экспериментатора: доступная инфраструктура может быть недостаточно большой или не обладать необходимыми характеристиками производительности или надежности. Кроме того, изменение экспериментальных условий часто требует административных прав, которые редко предоставляются обычным пользователям платформ. Поэтому эксперименты «in situ» часто имеют относительно ограниченный охват: они, как правило, не поддаются обобщению и предоставляют единственную точку данных, ограниченную данной платформой [7]. Использование эмуляторов может решить эту проблему, позволяя исследователю изменять характеристики производительности платформы. Принимая во внимание тот факт, что для всего эксперимента можно использовать одну и ту же платформу, легко сделать вывод о влиянии измененного параметра.

Методология эмуляции состоит из четырех основных этапов: предварительная настройка, настройка, тестовая платформа и эмуляция в цепи (ICE).

1. Перед началом настройки эмуляции необходимо выполнить предварительную подготовку. На этом этапе собираются все необходимые файлы управления, списки сетей, библиотеки и другие входные данные.
2. На этапе полной конфигурации CPU с помощью поставляемого поставщиком программного обеспечения для конфигурации создается эмуляционная база данных, доступная для загрузки. На этом этапе на основе результатов этапа пред-

варительной подготовки конфигурации генерируется битовый поток, необходимый для программирования эмуляционного оборудования.

3. Подготовка испытательного стенда включает проектирование, планирование и реализацию среды, в которой будет проходить эмуляция. Эта испытательная среда должна обеспечивать все необходимые интерфейсы и ресурсы реального оборудования.
4. Этап эмуляции в цепи объединяет тестовую среду и эмуляцию для создания системы, в которой можно проводить верификацию и отладку.

В настоящее время существует несколько методов и инструментов, позволяющих эмулировать CPU. Некоторые ученые сходятся во мнении относительно того, что традиционные, универсальные подходы эмуляции CPU могут быть также эффективны для выполнения на квантовом компьютере, как если бы стояла задача тестировать приложение или алгоритм в среде с настраиваемым и воспроизводимым уровнем гетерогенности процессоров [1, 6, 9]. При этом необходимо обратить внимание на тот факт, что традиционные способы эмуляции CPU различаются по многим параметрам.

На основании анализа имеющихся на сегодняшний день публикаций можно выделить несколько стандартных подходов к эмуляции, которые различаются на очень фундаментальном уровне, но не являются взаимоисключающими:

1. «Прожиг» CPU
2. Контроль над эмулируемыми процессами
3. Подход с аппаратной поддержкой
4. Подход с поддержкой планировщика

Первая и самая очевидная группа методов заключается в запуске приложения, которое потребляет нужную часть ресурсов CPU, оставляя остальные ресурсы эмулируемой среде. Обычно такая программа запускает цикл, интенсивно использующий ресурсы CPU, и периодически переходит в режим ожидания [5]. Одного этого недостаточно, поскольку нет гарантии, что планировщик не прервет работу приложения. Основной способ обеспечить это — использовать класс планирования реального времени для «программы, перегружающей CPU», чтобы программа отбирала все другие эмулируемые процессы, и сама не отбиралась. Другой, более радикальный подход — перегружать CPU на уровне ядра, чтобы иметь прямой контроль над планированием. Это дает большую свободу, но вызывает вопросы об обслуживаемости решения.

Следующий подход заключается в прямом управлении эмулируемыми процессами. Это можно сделать, запрашивая текущее использование CPU процессами и принимая решение о их остановке или возобновлении.

Для управления процессами, в частности, находит свое широкое применение следующий интерфейс: сигналы POSIX; управление приоритетами планирования; подсистема cgroup freezer.

Аппаратная группа методов использует возможности базового процессора. Эти методы опираются на способность некоторых процессоров контролировать собственную скорость выполнения. Однако у них есть некоторые ограничения: они зависят от модели процессора и допускают только ограниченный набор возможных уровней частоты [12].

Подходы с помощью планировщика используют некоторые расширенные возможности. Например, можно контролировать аффинность процессов к процессору или их долю времени процессора на очень высоком уровне, даже выше, чем сам планировщик.

Хотя описанные выше методы универсальны, широко известны и достаточно просты в использовании, они, по мнению автора, не подходят для выполнения на квантовом компьютере.

Во-первых, эти методы основаны на детерминированной последовательной логике и классических би-нарных операциях, которые не имеют прямых аналогов в унитарной, обратимой природе квантовых вычислений. Квантовые компьютеры оперируют с кубитами и квантовыми вентилями, что делает эмуляцию классических логических схем крайне неэффективной с точки зрения затрат ресурсов.

Во-вторых, классическая архитектура CPU требует частого ветвления, состояний и регистров, что весьма затруднительно масштабируется на квантовом уровне из-за ограничений глубины схем, числа кубитов и проблем декогеренции.

Кроме того, традиционная эмуляция имитирует все функции CPU, но выполняется в программном обеспечении, а не в аппаратном. Интерфейс CPU позволяет проводить параллельное сравнение физического аппаратного обеспечения с кодом эмуляции. Поэтому важно отличать традиционную эмуляцию от эмуляции, спроектированной для выполнения на квантовом компьютере, чтобы можно было проверить каждую команду/функцию в отдельности. Кроме того, вызовы API к квантовому компьютеру на порядок медленнее, чем локальное выполнение кода. Поэтому важно, чтобы как можно больше разработки и тестирования проводилось локально, прежде чем переопределять методы написания кода для выполнения на квантовом компьютере.

В ряде других публикаций описываются более целевые методы, которые разработаны специально для пря-



мого моделирования CPU с помощью квантовых схем. К числу таких методов относятся следующие.

1. Использование квантовой запутанности, формируемой при помощи вентиля Адамара, в качестве основы для построения схемы-защелки. Эта схема выполняет базовые логические операции над отдельными разрядами: проверку значения бита, установку бита в единицу и сброс в ноль. Такая схема может функционировать как квантовый аналог классической логики управления отдельными битами регистра [4]. Путём последовательного применения этих операций возможно формирование более сложных логических инструкций, например операции загрузки значения в регистр. Использование запутанности позволяет сохранить и управлять состояниями квантовых разрядов, обеспечивая потенциальное квантовое ускорение при выполнении логических и регистровых инструкций.
2. Создание квантовой схемы с недостаточным количеством элементов, которые объединяют необходимые логические вентили в одну схему. Для практической реализации более крупных 8- или 16-разрядных инструкций предлагается создать несколько 4-разрядных вариантов, в которых шум или количество доступных кубитов не позволяют реализовать полную версию. Проведенные испытания свидетельствуют о том, что этот метод значительно сократил время выполнения, поскольку большинство команд (за исключением повторяющихся, таких как CIPR, LDIR и т.д.) могут быть выполнены с использованием одной или двух схем [10,15]. Недостатком данного метода является то, что он требует больше кубитов, однако учеными в настоящее время предпринимаются меры, чтобы не допустить необоснованных накладных расходов.
3. Базовая версия комплекта средств разработки программного обеспечения реализуется с помощью квантовых схем, которые используются для представления традиционных логических вентилях. Эти квантовые логические вентили соединяются для построения более крупных схем, таких как сумматоры, вычитатели и защелки. Эксперты отмечают, что в ходе использования данного метода необходимо следить за тем, чтобы начальное состояние каждого кубита в квантовом эквиваленте логического вентиля было чистым, т.е. не было определено или чтобы входной сигнал вентиля был установлен на известное значение из другой схемы (при построении более крупных логических схем, таких как сумматор) [2]. Однако несмотря на то, что данная техника является удобным руководством и убедительным доказательством концепции выполнения битовых,

логических и арифметических функций на квантовом компьютере, она не использует основные преимущества квантовых вычислений, такие как возможность обработки нескольких состояний в одном кубите.

Систематизация достоинств и недостатков некоторых традиционных методов эмуляции CPU представлена в таблице 2.

Используя данные опытов и экспериментов с эмуляцией CPU для выполнения на квантовом компьютере, которые получены учеными в разных странах мира, на рис. 1 приведено сравнение эффективности традиционных методов.

На рис. 1 темно-серые столбцы представляют собой условную эффективность метода (%), другими словами, относительную способность метода реализовать функциональность классического процессора с минимальными затратами квантовых ресурсов. Эффективность выражена в процентном эквиваленте по следующим критериям:

- корректность выполнения логики (эмулируются ли инструкции ISA как есть?) — 40 %;
- объём требуемых ресурсов (кубиты, глубина схемы) — 30 %;
- скорость выполнения / время на схеме (в квантовых циклах) — 20 %;
- гибкость/масштабируемость (можно ли перенести на другую архитектуру?) — 10 %.

Общий балл по этим критериям нормируется на 100 %.

Серая кривая — это ресурсоёмкость метода (относительные единицы): чем выше значение, тем больше требуется квантовых ресурсов (кубитов, глубина схем и т.п.).

Приведенные на рис. 1 данные свидетельствуют о том, что квантовые виртуальные машины и LLVM/IR-трансляция показывают хорошее соотношение между эффективностью и потреблением ресурсов, в то же время, логическая эмуляция вентилях является наименее эффективной и наиболее ресурсоёмкой. Прямая трансляция машинного кода имеет значительный теоретический интерес, но крайне затратна по ресурсам.

Представленные выше результаты свидетельствуют об актуальной необходимости развития и улучшения традиционных методов эмуляции CPU для выполнения на квантовом компьютере. Развиваемые на сегодняшний день наработки включают оптимизацию эмуляции традиционных логических вентилях, методы упрощения нескольких вентилях, уменьшение количества необходимых кубитов и т.д. Ключевые направления исследований и перспективные способы решения проблем представлены в таблице 3.

Таблица 2.

Описание традиционных методов эмуляции центрального процессора для выполнения на квантовом компьютере

Метод эмуляции	Описание подхода	Преимущества	Ограничения
Трансляция машинного кода в квантовые логические элементы	Прямое преобразование инструкций CPU в эквивалентные квантовые схемы	Позволяет в теории воссоздать поведение конкретной архитектуры процессора	Требует большого количества кубитов и глубокой схемы, что ограничивает масштабируемость
Квантовые интерпретаторы	Создание логических структур, интерпретирующих команды на квантовом уровне	Гибкость: возможность адаптации под различные архитектуры набора команд	Высокая сложность реализации, значительные накладные вычисления
Моделирование логических вентилях («и», «или», «искл.или»)	Аппроксимация работы логических схем с помощью универсальных квантовых вентилях	Подходит для демонстрационных и образовательных целей	Эффективность крайне низкая для практического моделирования современных CPU
Использование схем обратной компиляции	Восстановление CPU-алгоритмов и их реализация через квантовые субрутины	Позволяет анализировать уязвимости и поведение процессоров на низком уровне	Метод нестандартизирован, требует большого объёма ручной работы
Квантовые виртуальные машины	Виртуальные программные среды, имитирующие CPU с использованием квантовой вычислительной платформы	Обеспечивает переносимость кода и абстракцию архитектуры	Необходим мощный классический сопроцессор, квантовая часть выполняет лишь ограниченные задачи
Трансляция промежуточного представления в квантовые инструкции	Преобразование промежуточного кода в квантовые схемы с использованием квантового промежуточного представления	Совместимость с современными компиляторами, возможность автогенерации квантовых программ	На данный момент применяется только для специализированных задач, не полноценно эмулирует CPU

Примечание: составлена автором на основе полученных данных в ходе исследования

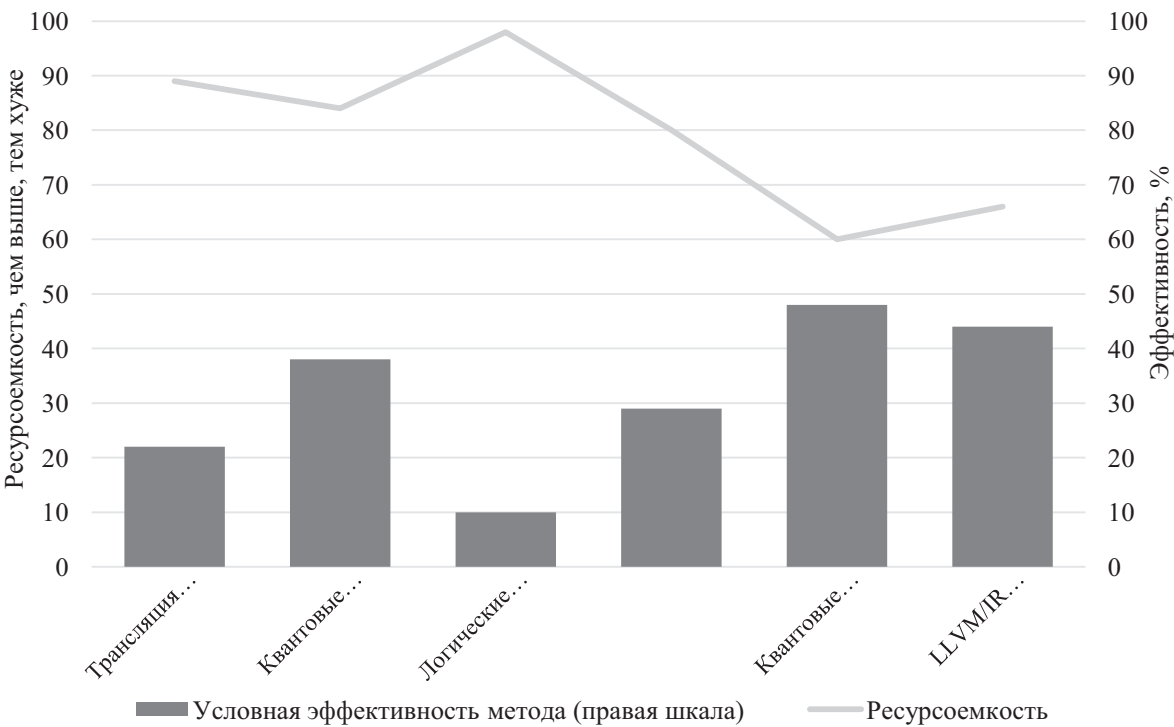


Рис. 1. Сравнение эффективности традиционных методов эмуляции центрального процессора для выполнения на квантовом компьютере (составлено автором)

Таблица 3.  
Систематизация перспективных подходов оптимизации традиционных методов эмуляции центрального процессора для выполнения на квантовом компьютере (составлено автором).

Аспект сравнения	Традиционные методы эмуляции	Адаптация / Проблемы для квантового компьютера	Методы оптимизации	Принцип работы
Основной принцип эмуляции	Интерпретация: пошаговое исполнение инструкций целевой архитектуры. Динамическая/JIT-компиляция: перевод блоков инструкций целевой архитектуры в нативный код хост-системы во время выполнения	— несоответствие архитектур: классические инструкции (последовательные, битовые) не имеют прямого квантового аналога; — проблема состояния: классический CPU оперирует детерминированными битовыми состояниями; квантовый — суперпозициями и запутанностью; — измерение/декогеренция: каждое чтение состояния кубита вызывает коллапс волновой функции, нарушая квантовые свойства;	1. Квантовая интерпретация: создание квантовых вентилях для эмуляции логики классических инструкций. 2. Квантовая компиляция (транспилиция): перевод классических инструкций в эквивалентные квантовые схемы (квантовые вентили и операции).	— квантовый вентиль как логическая операция: каждая классическая логическая операция («и», «или», «нет») может быть реализована с помощью обратимых квантовых вентилях; — ограниченные ресурсы: задача состоит в минимизации количества вентилях и глубины цепи для каждой эмулируемой инструкции; — избыточность кубитов: для представления N битов классической информации требуется N кубитов.
Представление данных и состояний	Битовые регистры, оперативная память как массив битов	Проблема: эффективное кодирование классических битов в кубиты и управление классической памятью на квантовой системе	1. Бинарное кодирование $1 \rightarrow  1\rangle$	$0 \rightarrow  0\rangle$ $1 \rightarrow  1\rangle$
Выполнение инструкций / операции	Последовательное выполнение микроопераций (чтение регистра, ALU-операция, запись в память).	1. Проблема: выполнение сложных арифметических и логических операций, циклов, ветвлений на квантовом уровне. 2. Обратимость: большинство классических операций необратимы, квантовые вентили должны быть обратимыми. 3. Условные операции: реализация условных переходов в квантовой логике сложна	1. Квантовые арифметические схемы: реализация сложения, умножения и других ALU-операций с помощью квантовых вентилях (например, квантовые сумматоры). 2. Квантовые управляющие вентили: для условных операций 3. Применение специализированных квантовых алгоритмов: для ускорения некоторых операций	— квантовая арифметика: например, схемы на основе QFT (квантового преобразования Фурье) для быстрого сложения/умножения; — условная логика: комбинация контролируемых вентилях для эмуляции классического ветвления. Требует копирования состояния для условного выполнения, что нарушает теорему о не клонировании, если не использовать вспомогательные кубиты или специфические подходы.

Окончание табл. 3

Аспект сравнения	Традиционные методы эмуляции	Адаптация / Проблемы для квантового компьютера	Методы оптимизации	Принцип работы
Управление памятью / ввод-вывод	Адресное пространство, кэш, стеки, аппаратные прерывания.	1. Проблема: эффективная адресация и доступ к квантовой памяти; синхронизация классического ввода-вывода с квантовыми вычислениями. 2. Доступ к данным: каждое измерение кубита коллапсирует его состояние, что затрудняет многократный доступ к одним и тем же данным без переинициализации	1. Квантовые схемы адресации: на основе QFT или специализированных методов для доступа к данным, закодированным в квантовых регистрах. 2. Вспомогательные кубиты: для временного хранения результатов или выполнения операций без коллапса основного состояния. 3. Гибридные подходы: использование классического компьютера для управления основной памятью и планирования квантовых вычислений для отдельных инструкций/блоков	— ограниченная квантовая память: текущие квантовые компьютеры имеют мало кубитов и короткое время когерентности, что делает эмуляцию больших объемов памяти непрактичной; — гибридные архитектуры: наиболее реалистичный подход, где квантовый компьютер выступает в роли копроцессора для выполнения специфических, — вычислительно сложных частей классической программы.
Масштабируемость и производительность	Конвейеризация инструкций и распараллеливание на уровне команд	1. Проблема: низкое количество кубитов (NISQ-эра), высокая частота ошибок, короткое время когерентности, отсутствие квантовой коррекции ошибок 2. «Квантовое ускорение»: для большинства классических задач квантовый компьютер не дает ускорения, а эмуляция классического CPU может быть медленнее.	1. Фокус на «квантово-ускоряемых» блоках: идентификация фрагментов кода, которые могут получить квантовое ускорение. 2. Кодирование высокой плотности: попытки закодировать более одного бита в один кубит 3. Оптимизация квантовых схем: минимизация глубины цепей вентилей для уменьшения влияния декогеренции.	— NP-полные проблемы: для задач, где классическое решение экспоненциально по времени, квантовое может быть полиномиальным, однако эмуляция CPU в целом не является такой задачей; — вариационный квантовый собственный решатель или алгоритм квантовой приближенной оптимизации: могут быть использованы для оптимизации квантовых схем эмуляции.



Итак, несмотря на значительные успехи, достигнутые за последнее десятилетие, масштабируемость симуляции больших квантовых схем (с точки зрения их ширины и глубины) на классических машинах остается основной проблемой. Поскольку ни один подход не является на сегодняшний день панацеей от проблемы масштабируемости, наиболее оптимальный результат позволяет получить сочетание несколько методов оптимизации.

### Выводы

В статье представлен обзор литературы и описание различных подходов к эмуляции центрального процессора для выполнения на квантовом компьютере. Взаимосвязь между классическим и квантовым программированием еще не получила своего должного освещения в современных публикациях, и, хотя научно-экспертное сообщество осознает потенциал повышения эффективности, который открывает использование квантовых компьютеров, построение простых схем/функций представляет на сегодняшний день сложную задачу.

В процессе исследования описаны некоторые традиционные методы эмуляции, выделены их достоинства

и недостатки, возможности и ограничения использования. Также приведены показатели эффективности и ресурсоемкости этих методов, которые систематизированы на базе проведенных учеными и экспертами экспериментов. Отдельное внимание уделено перспективным направлениям развития и возможным вариантам оптимизации традиционных инструкций CPU для выполнения на квантовом компьютере. По мнению автора, подход, основанный на использовании нескольких методов, улучшает доступность/удобство использования квантовых компьютеров при переходе от традиционных вычислений.

Полученные результаты позволяют сравнить традиционные методы с более эффективными квантовыми подходами к выполнению эмуляции CPU. Перспективы дальнейших исследований связаны с преодолением проблем, обусловленных ограниченной доступностью кубитов и непредсказуемым механизмом очередей. Предполагается, что это приведёт к созданию новых решений, которые позволят повысить эффективность работы квантовых систем даже при ограниченных ресурсах и сократить влияние задержек и очередей на процесс выполнения программ.

### ЛИТЕРАТУРА

1. Баскаков П.Е., Хабовец Ю.Ю., Пилипенко И.А., Кравченко В.О., Черкесова Л.В. Инструменты для выполнения и эмуляции квантовых вычислений // Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2020. Т. 18. № 2. С. 43–53. <https://doi.org/10.25205/1818-7900-2020-18-2-43-53>
2. Тырышкин С.Ю. Эмуляция квантовых вычислительных процессов для автоматизированных систем управления на классическом ПК // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2024. № 12. С. 130–134. <https://doi.org/10.37882/2223-2966.2024.12.36>
3. Hoa T. Nguyen, Muhammad Usman iQuantum: A toolkit for modeling and simulation of quantum computing environments // Software: Practice and Experience. 2024. Volume 54, Issue 6. P. 23–29. <https://doi.org/10.1002/spe.3331>
4. Mohammed Alghadeer, Eid Aldawsari Psitrum: An open-source simulator for universal quantum computers // IET Quantum Communication. 2024. Volume 5, Issue 4. P. 49–53. <https://doi.org/10.1049/qtc.12101>
5. Пелешенко В.А. Реализация квантовых вычислений на базе персональных квантовых компьютеров с использованием ядерно-магнитного резонанса // Мягкие измерения и вычисления. 2024. Т. 76. № 3. С. 14–34. <https://doi.org/10.36871/2618-9976.2024.03.002>
6. Nikita Stroeve, Natalia G. Berloff Analog Photonics Computing for Information Processing, Inference, and Optimization // Advanced Quantum Technologies. 2023. Volume 6, Issue 9. P. 105–113. <https://doi.org/10.1002/qute.202300055>
7. Tian Chen, Weixuan Zhang, Deyuan Zou Engineering Topological States and Quantum-Inspired Information Processing Using Classical Circuits // Advanced Quantum Technologies. 2025. Volume 8, Issue 6. P. 89–93. <https://doi.org/10.1002/qute.202400448>
8. Karthik Krishnan, Saranyan Vijayaraghavan Unraveling the Atomic Redox Process in Quantum Conductance and Synaptic Events for Neuromorphic Computing // Advanced Electronic Materials. 2022. Volume 8, Issue 11. P. 56–66. <https://doi.org/10.1002/aem.202200509>
9. Милай М.В. Квантовая телепортация сверхпроводящих кубитов в квантовых компьютерах открытого проекта IBM QUANTUM EXPERIENCE // Сборник избранных статей научной сессии ТУСУР. 2023. № 1–1. С. 131–135. EDN: DXGSNI
10. Черемисин Д.Г., Мкртчян В.Р., Музлова А.Д. Развитие квантовых компьютеров: перспективы для вычислительной мощности и решения сложных задач // Символ науки: международный научный журнал. 2024. № 1–2. С. 49–51. EDN: XVHDEE
11. Рахимов Р.Х. Потенциал импульсного туннельного эффекта (ИТЭ) для преодоления технических барьеров квантовых компьютеров // Computational Nanotechnology. 2024. Т. 11. № 3. С. 11–33. <https://doi.org/10.33693/2313-223X-2024-11-3-11-33>
12. Guillem Llodrà, Christos Charalambous Benchmarking the Role of Particle Statistics in Quantum Reservoir Computing // Advanced Quantum Technologies. 2022. Volume 6, Issue 1. P. 105–110. <https://doi.org/10.1002/qute.202200100>
13. Giuliana Siddi Moreau, Lorenzo Pisani Quantum Artificial Intelligence Scalability in the NISQ Era: Pathways to Quantum Utility // Advanced Quantum Technologies. 2025. № 197. P. 44–49. <https://doi.org/10.1002/qute.202400716>
14. Ульянов С.В., Рябов Н.В., Зрелов П.В., Иванцова О.В., Кореньков В.В. Оценка возможностей классических компьютеров при реализации симуляторов квантовых алгоритмов // Программные продукты и системы. 2022. № 4. С. 618–630. <https://doi.org/10.15827/0236-235X.140.618-630>
15. Кирилюк М.А., Бочаров Н.А., Парамонов Н.Б., Суминов К.А. Разработка и оценка эффективности метода организации вычислений на специализированных вычислительных системах с квантовым сопроцессором // Приборы. 2024. № 3 (285). С. 12–20. EDN: TPQGGM

© Тырышкин Сергей Юрьевич (service.vip-spe@yandex.ru)

Журнал «Современная наука: актуальные проблемы теории и практики»