

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
"МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
имени М. В. ЛОМОНОСОВА"

ФИЗИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА КВАНТОВОЙ ЭЛЕКТРОНИКИ

## МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

# «Анализ рандомизированных алгоритмов для реалистичных моделей квантовой вычислительной машины»

Выполнил студент  
Группы 227М  
Моисеевский А. Д.

---

Допущен к защите \_\_\_\_\_

Зав. кафедрой квантовой электроники  
д.ф.-м.н., профессор Панов В. И.

---

Научный руководитель  
Старший научный сотрудник  
к.ф.-м.н.  
Страупе С. С.

---

Москва  
2020

## Содержание

ВВЕДЕНИЕ.....	3
1. Актуальность, цели и задачи.....	3
ОСНОВНАЯ ЧАСТЬ.....	5
2. Теоретическая часть и литературный обзор.....	5
2.1 Историческая справка и краткий экскурс в теорию сложности.....	5
2.2. Современные исследования квантового превосходства.....	12
2.3. Кросс-энтропийный бенчмаркинг.....	22
2.4. Модель ошибок.....	26
2.5. Шумовые гейты и вероятностные ошибки.....	29
3. Численное моделирование.....	33
3.1. Детали реализации квантового симулятора.....	33
3.2. Тестирование симулятора с различными ошибками.....	37
3.3. Анализ распространения ошибки в кубитном регистре.....	42
3.4. Определение реалистичных параметров ошибок.....	45
3.5. Демонстрация квантового превосходства.....	47
3.6. Результаты.....	53
ВЫВОДЫ.....	54
ЗАКЛЮЧЕНИЕ.....	55
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	56

# ВВЕДЕНИЕ

## 1. Актуальность, цели и задачи

В 1981 году Ричард Фейнман призвал мировых исследователей к созданию квантового компьютера. В свойственной ему смелой манере он заявил: «Природа не классическая! <...> Если вы хотите построить её симуляцию, вам лучше сделать эту симуляцию квантово-механической, и это <...> замечательная задача, потому что она не кажется такой уж простой». Эта фраза стала, возможно, одним из первых, и уж точно одним из самых знаменитых публичных высказываний о существовании квантового превосходства.

Квантовое превосходство — качественное преимущество квантового вычислительного устройства над классическим, позволяющее ему эффективно справляться с задачами, на решение которых у классического вычислителя ушло бы асимптотически больше времени.

Ричард Фейнман очень верно заметил, что коль скоро законы мироздания носят квантово-механический характер, моделировать их будет эффективнее и удобнее в системе, собственные законы которой имеют ту же природу. Однако, смелое заявление обернулось вопросом, достоверно не разрешенным и по сей день. Несмотря на столь логичные предпосылки, строго доказать, что квантовый компьютер способен решать задачи, качественно недоступные для классического компьютера, оказалось невероятно сложно. Неясность наличия квантового превосходства обсуждается в научных кругах уже более трёх десятилетий.

Данная работа посвящена изучению методов определения границ практического квантового превосходства. Это упрощённая задача, призванная на примере частных случаев продемонстрировать, а не доказать

математически строго наличие искомого преимущества. Недавние успехи в этой области спровоцировали появление огромного количества новых публикаций, разбор которых будет проведён в основной части работы. Объём информации по данной теме сейчас растёт столь быстро, что можно со всей ответственностью утверждать – актуальность темы не вызывает сомнений.

Основная часть работы имеет следующую структуру. Историческая справка и экскурс в теорию сложности призваны осветить развитие исследуемого вопроса, его комплексность и глубокую взаимосвязь с математикой. Далее следует обзор современных работ по теме, с разбором появившихся в них уточнений понятия квантового превосходства, а также с освещением многочисленной критики. В завершение обзорной части производится описание используемых в дальнейшем методов. Оригинальная часть работы заключается в реализации симулятора квантового компьютера на основе холодных атомов, выработке реалистичной модели учёта ошибок, и в проведении математического моделирования с целью определения границы превосходства для перспективных квантовых компьютеров данного типа. Также интерес представляет исследование динамики состояния квантового регистра под действием ошибок разных типов.

Цель работы – моделирование рандомизированных квантовых алгоритмов в квантовом регистре с ошибками, характерными для архитектуры на основе одиночных холодных атомов.

В работе были поставлены следующие задачи: исследовать существующие методы анализа квантового превосходства, составить необходимую модель ошибок, реализовать программу-симулятор квантовых вычислений, проанализировать динамику состояния квантового регистра под действием различных ошибок, и на основе результатов моделирования сделать вывод о достижимости в обозримом будущем квантового превосходства для компьютеров на основе холодных атомов.

# ОСНОВНАЯ ЧАСТЬ

## 2. Теоретическая часть и литературный обзор

### 2.1 Историческая справка и краткий экскурс в теорию сложности

Квантовая информатика — область науки и технологии, занимающаяся изучением и описанием принципов, управляющих сложными квантовыми системами в приложении задач теории информации и теории алгоритмов. В самом общем случае, так или иначе, эти задачи могут быть сведены к хранению, получению, передаче и обработке информации [1].

Отправной точкой, от которой можно было бы отсчитывать начало истории квантовых вычислений как науки об обработке информации с использованием специфических квантовых систем, можно считать 1980-й год и публикацию работы об использовании квантовомеханической модели для классической машины Тьюринга [2]. Туда же можно отнести первые соображения об использовании квантового компьютера как более эффективного, по сравнению с классическим, симулятора физических процессов [3, 4]. Поворотным же моментом стала публикация работы [5] в 1985г., в которой была представлена универсальная теоретическая модель квантового вычислительного устройства, а также специфичная задача, квантовый алгоритм решения которой, известный теперь как квантовый алгоритм Дойча, превосходил по эффективности любой мыслимый классический алгоритм.

Таким образом была впервые наглядно продемонстрирована потенциальная возможность наличия преимущества квантового компьютера над классическим. Однако, насколько велико это потенциальное преимущество? Для уточнения этого вопроса необходимо обратиться к понятийному аппарату теории вычислительной сложности. Данная теория

изучает классификацию сложности различных вычислительных задач. И поскольку алгоритм решения любой задачи, независимо от его специфики, может быть разбит на отдельные элементарные операции, выводы теории сложности могут быть с равным успехом применены как к классическим, так и к квантовым алгоритмам. Фундаментальным понятием в данной теории является понятие класса сложности. Класс сложности может быть представлен как совокупность вычислительных задач, имеющих некоторое общее свойство по отношению к вычислительным ресурсам, требуемым для их решения. Два важных для рассматриваемого вопроса класса сложности — классы  $P$  и  $NP$ . Не углубляясь в математические детали положим, что класс  $P$  — множество вычислительных задач, которые могут быть решены с помощью классического алгоритма за время, зависящее от объёма входных данных как некоторый полином. Примерами задач данного класса могут служить такие задачи как перемножение матриц, сортировка численного массива и выяснение связности графа.

Класс  $NP$ , в свою очередь, включает в себя задачи, частное решение которых может быть проверено на корректность с помощью классического алгоритма за время, также зависящее от объёма данных полиномиально. Очевидно, что класс  $P$  является подмножеством класса  $NP$ , поскольку при возможности быстрого отыскания корректных решений задачи, корректность заданного частного решения может быть проверена тривиальным перебором. В качестве примера, иллюстрирующего задачи класса  $NP$  и потенциально большую общность этого класса по сравнению с классом  $P$ , часто приводят задачи разложения числа на простые множители и задачу коммивояжера. На сегодня неизвестны классические алгоритмы, способные в общем случае найти решение данных задач за полиномиальное время. Однако осуществить проверку корректности произвольного заданного решения в таких условиях вполне возможно. Нет, однако, и строгих доказательств того, что алгоритм

решения подобных задач за полиномиальное время не существует и не будет найден в будущем. Поэтому нет достаточных оснований утверждать, что данные задачи не являются представителями класса  $P$ .

Вопрос существования задач, принадлежащих классу  $NP$  и находящихся при этом за рамками класса  $P$ , или, проще говоря, вопрос равенства этих двух классов, является одной из важнейших нерешенных проблем теории алгоритмов и современной математики в целом уже на протяжении более чем трёх десятилетий [6]. Тем не менее, исходя из современного понимания теории алгоритмов, можно начать склоняться ко мнению, что  $NP$  всё же содержит задачи, не входящие в  $P$ . В частности, в классе  $NP$  выделяют важный подкласс задач, называемых  $NP$ -полными. Классическим примером такой задачи может служить упомянутая ранее задача коммивояжера.  $NP$ -полные задачи характеризуются тем, что они в некотором смысле эквивалентны между собой и являются не менее сложными, чем любая другая задача класса  $NP$ . Иначе говоря, решение любой задачи класса  $NP$  может быть сведено к решению какой-либо  $NP$ -полной задачи. Конечно, такое обстоятельство сильно увеличивает интерес к данному классу задач, поскольку их исследование ведёт к потенциальным прорывам в решении огромного количества практически важных проблем. И кроме того, отыскание полиномиально сложного решения любой  $NP$ -полной задачи мгновенно докажет равенство классов  $P$  и  $NP$ . Однако же, при современном развитии теории алгоритмов такие решения остаются неизвестными и вопрос об их существовании — открытым. Таким образом, никакая  $NP$ -полная задача не может быть сегодня в общем случае решена за полиномиальное время с использованием классического алгоритма. Помимо этого, существует целый ряд задач,  $NP$ -полнота которых не доказана, но классическое полиномиально-сложное решение которых также не найдено, и, тем самым, принадлежность их к классу  $P$  не определена. Примером может

служить также упомянутая ранее задача факторизации или разложения целого числа на простые множители [1, 7].

А что же касательно квантовых алгоритмов? Приблизиться к пониманию возможностей квантовых компьютеров позволит рассмотрение класса BPP. К этому классу относят задачи, которые могут быть эффективно (за полиномиальное время) решены с помощью вероятностного классического алгоритма, если в решении допускается ограниченная вероятность ошибки. Строгое отношение между классами BPP и P неизвестно, хотя опять же, исходя из современного понимания теории алгоритмов, можно ожидать, что BPP включает в себя класс P и даже выходит за рамки NP. Множество же задач, эффективно разрешимых с конечной вероятностью ошибки на квантовом компьютере принято обозначать как класс BQP. Работа Дойча 1985г. [5] продемонстрировала, что при текущем понимании математики существуют задачи, сложность решения которых на квантовом компьютере ниже, чем на классическом. Более того, через семь лет, в 1992г. была опубликована работа [8] и ещё через два года — работа [9], в которых была продемонстрирована возможность экспоненциального ускорения решения специфической задачи с помощью квантового алгоритма по сравнению с детерминированным классическим, а затем и вероятностным классическим алгоритмом.

Тут также нельзя не упомянуть отдельно, что в 1994г. Питером Шором был разработан квантовый алгоритм, обеспечивающий экспоненциальное ускорение решения задачи факторизации целых чисел [10]. Конкретно, данный алгоритм является усовершенствованием и практическим приложением квантового алгоритма поиска периода функции. Он позволяет разложить число  $M$  на простые множители за время  $O(\log^3 M)$  с использованием  $O(\log M)$  кубитов. Огромная значимость данного алгоритма заключается в том, что с его помощью становится потенциально

возможным взлом криптографических систем с открытым ключом за время, зависящее от размера ключа не экспоненциально (субэкспоненциально), как было бы для классического алгоритма, а линейно, что угрожает способности подобных криптосистем гарантировать необходимое время конфиденциальности шифруемой информации. Кроме того, публикация [10] также включает описание алгоритма для ускорения решения задачи дискретного логарифмирования, которая наравне с задачей факторизации благодаря своей вычислительной сложности используется в некоторых криптографических системах для обеспечения секретности [11].

Обобщение подобного результата на весь класс задач NP стало бы колоссальным прорывом, поскольку любая задача класса NP подразумевает возможность экспоненциально сложного поиска решения последовательным перебором. Соответственно, экспоненциальное ускорение такого перебора в результате дало бы полиномиальную сложность, аналогичную сложности решения задач класса P на классическом компьютере. Таким образом было бы показано, что класс BQP полностью включает в себя класс NP, а квантовые компьютеры обладают качественно большими вычислительными возможностями по сравнению с классическими (если только  $P \neq NP$ ).

Ускоренный квантовый алгоритм перебора был найден в 1996 году Ловом Гровером [12]. Задача для этого алгоритма была поставлена как поиск решения уравнения вида  $f(x)=1$ , где  $f$  – «чёрный ящик» или «оракул» — некая булева функция, истинная только для искомых  $x$  из неструктурированного, в общем случае, набора. Такое уравнение могло быть решено с помощью квантового алгоритма Гровера за  $R \leq \lceil \frac{\pi}{4} \sqrt{\frac{N}{M}} \rceil$  обращений к функции-оракулу  $f$ , где  $N$  — число элементов, среди которых производится поиск, а  $M$  — число искомых решений на данном множестве элементов. Сам по себе этот факт может показаться удивительным, так как

алгоритм Гровера, по сути, позволяет провести поиск по неструктурированной базе данных за число операций меньшее, чем потребовалось бы для простого обращения к каждому элементу этой базы. На самом деле, такая возможность оказывается реализуемой, поскольку алгоритм Гровера требует наличия квантовой адресации. Тогда, при реализации унитарного оператора, действующего по правилу  $|x\rangle|q\rangle \rightarrow |x\rangle|q \oplus f(x)\rangle$  и аналогичного функции-оракулу, такой оператор сможет быть применён к суперпозиции элементов базы данных. Получив таким образом доступ к информации о суперпозиции решений задачи поиска, пусть даже ограниченный возможностями работы с квантовым регистром, можно квадратично ускорить процесс нахождения необходимого элемента.

Алгоритм Гровера применим, вообще говоря, для ускорения решения NP-полных задач. Но в отличие от алгоритма Шора он обеспечивает только квадратичное, а не экспоненциальное ускорение решения. К сожалению оказывается, что квантовые компьютеры, по-видимому (опять же, если только  $P \neq NP$ ), не способны к решению NP-полных задач за полиномиальное время. В работе [13] демонстрируется потенциальная возможность ускорения поиска по базе данных до сложности  $O(N^{1/3})$ , но вместе с тем и показывается, что данная граница является оптимальной.

Таким образом, потенциально возможная картина иерархии классов сложности вычислительных задач отражена на рисунке 1. Прежде всего, можно ожидать, что все изображенные на рисунке 1 классы входят в класс #P — класс задач, сводящихся к подсчёту количества решений задачи класса NP. Допустим теперь, что классы P и NP не совпадают. В таком случае подмножество NP-полных задач точно находится вне класса P — в частности, задача коммивояжера не имеет классического полиномиально-сложного решения. Это необходимо в силу того, что любая задача NP может быть

сведена к произвольной NP-полной, поэтому если хоть одна NP-полная задача находится в классе P, то классы P и NP с необходимостью совпадают. (Выходить за рамки NP класс P также не может, поскольку возможность полиномиально сложного решения подразумевает возможность полиномиально сложной проверки решения.) Помимо NP-полных задач, вне класса P могут оказаться и другие задачи класса NP, например задача факторизации. Класс P, как частный случай, полностью входит в класс BPP — класс задач разрешимых с полиномиальной сложностью для вероятностного классического алгоритма, допускающего определённую долю ошибок. Внешние границы класса BPP оцениваются очень приблизительно. Нет строгих оснований исключать, что класс BPP может совпадать с классом NEXP, аналогичным классу NP, но проверка корректности решений задач которого требует не полиномиального, а экспоненциального времени [6].

Возвращаясь к вопросу превосходства квантовых компьютеров над классическими, мы можем видеть, что данный вопрос сводится соотношению классов BPP и BQP. Важно отметить, что класс BQP не может оказаться вложенным в класс BPP, поскольку квантовый компьютер в состоянии эффективно симулировать классический.

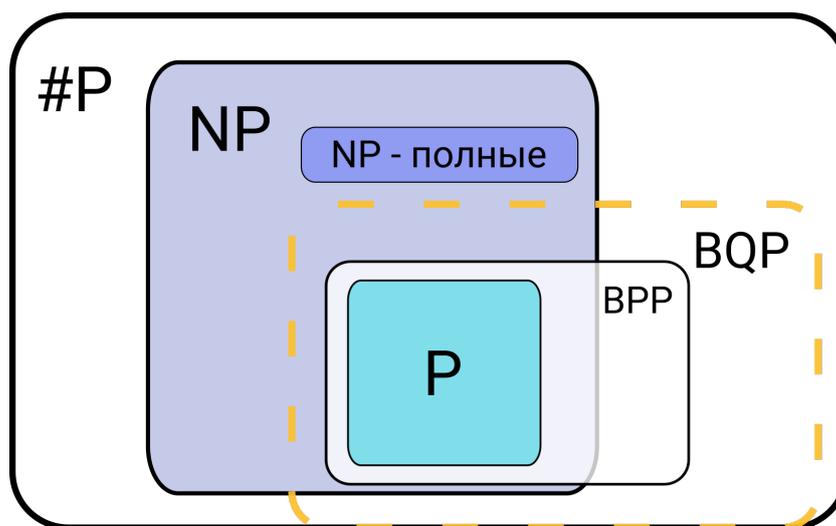


Рисунок 1: Потенциально возможная иерархия классов сложности вычислительных задач

## 2.2. Современные исследования квантового превосходства

Ещё одной важной, и уже более современной работой в данном направлении стало исследование [14], рассматривающее математическую сложность классического расчёта результатов работы линейно-оптической схемы, состоящей из множества линейных интерферометров. Эта работа показывает, что отталкиваясь от реалистичных предположений, а именно #P-сложности расчёта перманента матрицы с независимыми нормально распределёнными элементами, можно заключить, что не существует классического алгоритма, способного в общем случае за полиномиальное время сгенерировать битовую строку из распределения, совпадающего с распределением битовых строк, генерируемых заданной линейно-оптической сетью. В противном случае имеет место равенство классов P и VPP. При этом линейно-оптическая сеть не является полноценной моделью квантового вычислителя, так как не позволяет обычным образом реализовывать запутывающие двухкубитные гейты.

Однако, безусловно, самой громкой работой последнего времени в данной области стала работа [15], в которой делается однозначное заявление о достижении квантового превосходства на 53-кубитном квантовом компьютере. Для его демонстрации используется задача сэмплирования, то есть генерации выходных данных случайной квантовой схемы определённой глубины, составленной из фиксированных двухкубитных и однокубитных гейтов, случайно выбираемых из фиксированного множества согласно определённым правилам. Кроме того, делается предположение о том, что производительность квантовых вычислителей в дальнейшем будет расти с двойной экспоненциальной скоростью. Столь серьёзные заявления незамедлительно вызвали волну встречной обоснованной критики [16]. Основную критику в адрес работы [15] можно свести к тому, что авторы

работы конкретизировали понятие «квантового превосходства» таким образом, что оно перестало совпадать с пониманием этого термина рядом других специалистов. Действительно, мы видим, что результаты большинства работ с 1980 года до наших дней так или иначе могут быть сведены к некоторым выводам в рамках теории сложности, или, по крайней мере, к их предпосылкам. И сам вопрос квантового превосходства эквивалентен понятию соотношения классов BPP и BQP. В то время как работа [15] в своей основе апеллирует к тому факту, что при линейном росте числа кубитов, количество классической информации, необходимой для их описания, растёт экспоненциально. А значит, классическому компьютеру просто не хватает физических ресурсов для симуляции достаточно мощного квантового компьютера. Таким образом, работа [15] уделяет меньше внимания вопросам теории сложности и концентрируется на практической демонстрации способности реального квантового процессора быстро решить задачу, для решения которой классическому компьютеру в его нынешнем понимании, или, как выражаются авторы статьи, «State-of-the-art supercomputer», потребуется существенно больше времени. В конкретном случае, рассмотренном в [15], 10000 лет против 200 секунд. В критической заметке [16] авторы прямо указывают на то, что использование термина «квантовое превосходство» в таком значении может привести к непониманию, поскольку оно не соответствует общепринятому значению данного термина, в котором его уже использовали ранее.

Второе направление критики работы [15] основывается на том, что, как мы оговорили в самом начале, нельзя быть уверенным, что наше понимание математики совершенно, и используемые нами классические алгоритмы оптимальны. Так, авторы заметки [16] утверждают, что после должной оптимизации рассмотренная в статье [15] задача потребует для решения на классическом суперкомпьютере всего 2.5 дня, вместо первоначально

заявленных 10000 лет. Это всё ещё на порядок больше времени, которое требует квантовый компьютер для достижения той же цели, однако явно демонстрирует нашу неспособность делать математические строгие выводы на почве нынешнего понимания теории алгоритмов.

Очень интересным в этом контексте является также рассмотрение работы [17]. В ней авторы указывают, что несмотря на то что классическая симуляция идеального квантового компьютера требует экспоненциально растущего объёма вычислительных ресурсов, реальные квантовые компьютеры, вообще говоря, идеальными не являются. И за счёт этого симуляцию можно значительно оптимизировать. Демонстрируется класс классических алгоритмов, способных симулировать работу квантового компьютера с малой ошибкой двухкубитного гейта, требования которых к вычислительным ресурсам при увеличении числа кубитов или глубины квантового алгоритма растут лишь линейно. Показано, что с помощью данного метода может быть произведена полиномиально-сложная симуляция работы квантового компьютера при некотором пороговом значении ошибки двухкубитного гейта. Дальнейшее уменьшение ошибки, как и можно ожидать, сопряжено с экспоненциальным ростом сложности симуляции. Весомым выводом статьи становится то, что при параметрах работы квантового компьютера, близких к параметрам в эксперименте, описанном в статье [15] (по аналогии, авторы называют данные параметры характерными для «state of the art» экспериментов), результаты квантового компьютера могут быть за время порядка часов воспроизведены на одноядерном классическом компьютере при объёме доступной оперативной памяти менее 5Гб. На сегодняшний день данная статья ещё не получила публикацию в рецензируемом научном издании, однако можно ожидать, что в случае успешной публикации выводы данной статьи станут критическими аргументами против общности выводов статьи [15].

К похожим результатам приходит исследование [18]. В ней представляется новый метод симуляции, основанный на представлении работы квантовой схемы в виде тензорной сети (tensor network). Тензорная сеть имеет вид двоичного дерева, каждая вершина которого представляет собой операцию свёртки. Такое дерево имеет выраженный «ствол», операции на котором вносят основной вклад в сложность симуляции работы квантовой схемы. В работе предложен алгоритм оптимизации расчёта данной цепочки свёрток, получивший название стержневой оптимизации (stem optimization). Помимо прочего, данный алгоритм также позволяет производить расчёты параллельно, с последующим усреднением, что, в свою очередь, позволяет справиться с задачей сэмплинга случайной квантовой схемы из публикации [15] на классическом суперкомпьютере за время порядка минут. В целом, данное исследование вновь подтверждает вывод о том, что может потребоваться ещё значительное время, прежде чем можно будет однозначно говорить демонстрации квантового превосходства даже со всеми сделанными оговорками.

Ещё одна интересная работа, указывающая на необходимость дальнейшего совершенствования квантовых компьютеров, описана в статье [19]. В ней результаты статьи [15] теоретически аппроксимируются для области больших размеров кубитных регистров и больших глубин квантовых схем. Поскольку с ростом этих величин качество работы квантового вычислителя падает экспоненциально, и поддержание точности результатов требует растущего числа усреднений, возникает вероятность того, что за областью оговоренного квантового превосходства вновь следует область превосходства классических компьютеров. Именно такой вывод и демонстрируется в работе — для реалистичных моделей квантового вычислителя (noisy intermediate-scale quantum hardware, NISQ) квантовое превосходство ограничивается квантовыми схемами с глубинами порядка 200

слоёв. Однако, эта граница будет расти экспоненциально с падением характерных значений ошибок в квантовом вычислителе.

Для лучшего понимания того, какая задача используется во всех упомянутых статьях для выявления квантового превосходства, а также каким образом оно количественно характеризуется, помимо статьи [15] стоит обратиться к статьям [20] и [21]. В них описывается процесс сэмплирования — задача генерации битовой строки из распределения, характерного для результатов работы определённой квантовой схемы. Обсудим сначала в данном контексте понятие самой квантовой схемы, или, как говорят иначе, квантового алгоритма. Прежде всего, в рассмотрение обычно вводится двумерный массив кубитов, характерный для множества физических реализаций квантовых компьютеров, например для реализации на основе сверхпроводящих цепей или на основе ионов либо холодных атомов в ловушках. Все кубиты инициализируются одним состоянием вычислительного базиса. Далее, к кубитам данного регистра циклично применяется определённая последовательность однокубитных и двухкубитных гейтов. Поскольку физически невозможно к одному и тому же кубиту применять несколько гейтов одновременно, такая схема очевидным образом разбивается на слои. Гейты внутри каждого слоя физически могут выполняться параллельно. Количество слоёв в квантовой схеме определяет её глубину. После применения всех гейтов кубиты квантового регистра могут быть измерены в вычислительном базисе, результаты чего и представляются в виде искомой битовой строки. А поскольку одиночные измерения не могут определить состояние кубита однозначно, а также в следствие наличия шума и ошибок, многократное исполнение заданного квантового алгоритма приведёт к получению множества различных битовых строк, которое может быть охарактеризовано распределением вероятности. Типичное представление квантовой схемы изображено на рисунке 2.

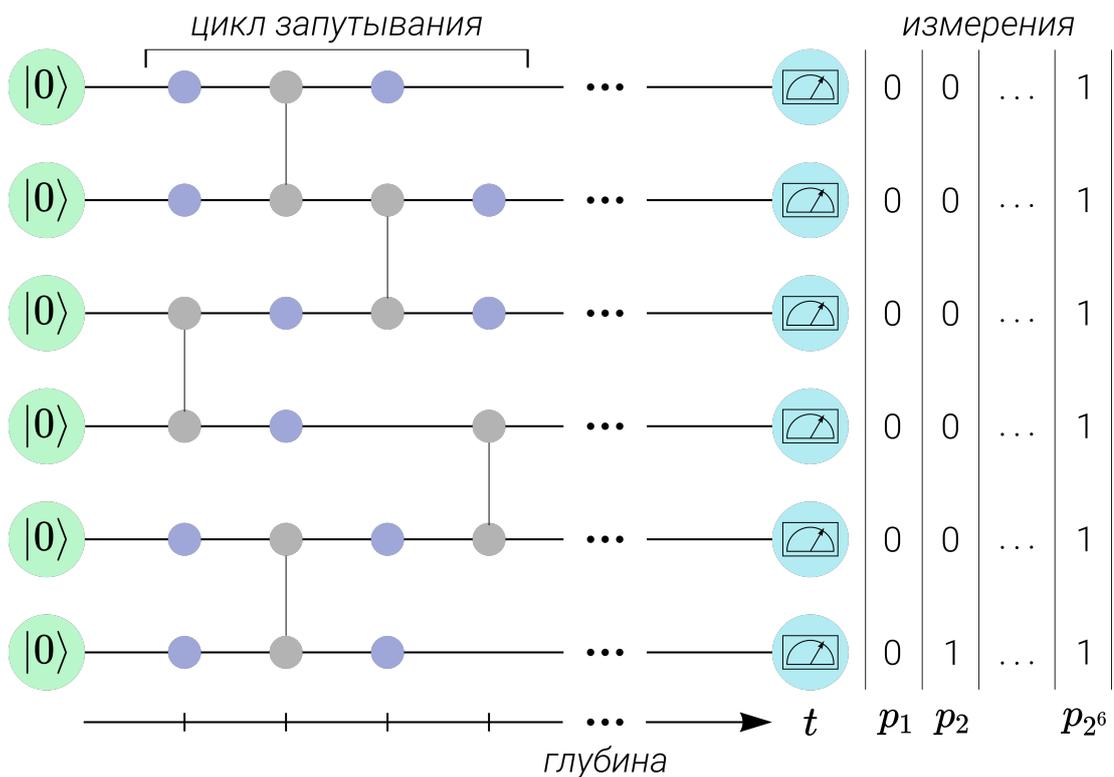


Рисунок 2: Схема случайного квантового алгоритма на примере одномерного шестикубитного регистра. Гейты выбираются случайным образом из заранее определённого набора. За один цикл каждый кубит запутывается со всеми соседями, после чего цикл повторяется до достижения нужной глубины. Многократное исполнение такого алгоритма породит множество различных битовых строк, характеризуемое вероятностью  $p_N$  наблюдения каждой строки.

Если говорить более строго, можно ввести в рассмотрение понятие архитектуры квантовой схемы как множества ориентированных графов  $\{G_n\}$ . Количество вершин для каждого графа  $G_n$  определяется как полином по  $n$ , причём каждая вершина имеет либо по одному, либо по два входящих и выходящих ребра. Тогда схема с данной архитектурой, действующая на  $n$  кубитов определяется путём указания для каждой вершины графа  $G_n$  гейта, действующего на кубиты, определённые входящими в неё рёбрами. Таким образом, архитектура определяет квантовую схему для каждого числа кубитов  $n$ , оставляя только выбор конкретных гейтов.

Пусть дана квантовая схема  $S$  с определённой архитектурой, гейты которой выбраны случайно с равномерным распределением согласно мере

Хаара, а также задан некоторый параметр ошибки  $\epsilon$ . Сэмплированием случайной схемы  $C$  на  $n$  кубитах называется задача генерации выборки битовых строк  $y \in \{0, 1\}^n$  с вероятностями  $Pr(y) = |\langle y|C|0^n \rangle|^2$  и максимальным разбросом  $\epsilon$  за время  $t$ , зависящее от  $n$  и  $\epsilon$  как некоторый полином [20].

Квантовые схемы демонстрируют высокую чувствительность к вносимым в них ошибкам, что является признаком хаоса [22]. Отсюда естественно предположить, что задача, требующая воспроизведения работы квантовой схемы с высокой точностью окажется неразрешимой эффективно на классическом компьютере. Примером такой задачи и выступает сэмплирование случайной квантовой схемы.

Однако, на практике генерация случайных гейтов, распределённых равномерно согласно мере Хаара, необходимая для создания случайной квантовой схемы, на квантовом компьютере оказывается экспоненциально сложной по числу кубитов. Поэтому, вместо генерации случайных гейтов используется, как правило, конечный их набор, конкретный гейт из которого выбирается уже случайно в процессе генерации схемы. Такой подход, позволяет аппроксимировать действие случайных гейтов, являясь при том экспоненциально более эффективным [23, 24].

#P-сложность задачи сэмплинга случайной схемы для общего случая демонстрируется в работе [20]. Важно отметить, что обеспечение вычислительной сложности требует сохранения общности на этапе выбора конечного набора гейтов. Так, теорема Готтесмана-Книлла утверждает, что схемы, составленные только из гейтов группы Клиффорда могут быть за полиномиальное время симулированы вероятностным классическим алгоритмом. Другими словами, сложность задачи достоверно падает до класса BPP [25].

В гильбертовом пространстве распределение состояний кубитного регистра, генерируемых случайными схемами, с увеличением глубины, стремится к равномерному согласно мере Хаара. Следствием этого является то, что распределение вероятностей встречаемости битовых строк стремится к экспоненциальному распределению или распределению Портера — Томаса [21].

Для вероятности  $p = |\langle x_0 | \psi \rangle|^2$  исхода  $|x_0\rangle$  измерения в базисе  $\{|x\rangle\}$  состояния  $|\psi_0\rangle$ , случайного по равномерному, согласно мере Хаара, распределению, распределение Портера-Томаса имеет вид

$$Pr(p) = (N-1)(1-p)^{N-2}, \quad (1)$$

где  $N = 2^n$  – размерность гильбертова пространства, а  $n$  – число кубитов. Для больших  $N \gg 1$  выражение (1) можно переписать как

$$Pr(p) = N e^{-Np}. \quad (2)$$

При добавлении к гейтам алгоритма вероятностных ошибок, наблюдаемое распределение быстро отклоняется от экспоненциального, что продемонстрировано на рисунке 3, повторяющем промежуточные выводы статьи [21]. В предельном случае с очень большой вероятностью ошибки для каждого гейта, на выходе схемы всегда будет наблюдаться результат измерения полностью смешанного состояния, когда вероятности возникновения всех возможных битовых строк равны. Тогда график вырождается в дельта-функцию в точке  $p = 1/N$ , где  $N = 2^n$  — число возможных битовых строк, а  $n$  — число кубитов в регистре. В реальности, однако, этого не происходит из-за ограниченности числа усреднений.

В качестве эффективного способа генерации случайных схем, способных при относительно небольшой глубине (20 - 40 слоёв) достаточно приблизиться к распределению Портера-Томаса, в работе [21] предлагается

следующий алгоритм. К кубитному регистру, первоначально инициализированному нулевым состоянием вычислительного базиса применяется оператор Адамара, переводящий каждый кубит в состояние равновесной суперпозиции. Затем слои состояются из операторов CZ и однокубитных гейтов, случайно выбираемых из множества  $\{X^{1/2}, Y^{1/2}, T\}$ .

Гейт T не является клиффордовским, что не позволяет эффективно симулировать данную схему на классическом вычислителе известным образом. Выбор однокубитного гейта производится так, чтобы первым гейтом, применённым к каждому кубиту был гейт T, а в дальнейшем исключалось последовательное применение к одному кубиту одного и того же гейта. Кроме того, если к данному кубиту был применён однокубитный гейт, применение нового гейта не производится до задействования данного кубита в работе гейта CZ.

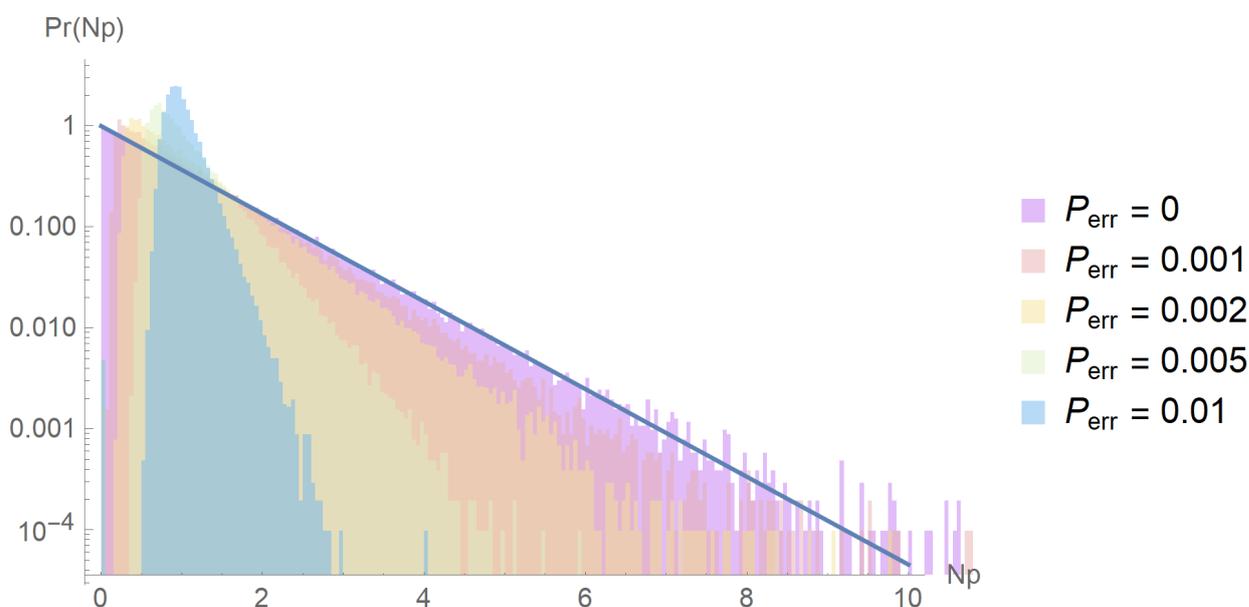


Рисунок 3: Гистограмма распределения перенормированных на общее число  $N$  вероятностей выходных битовых строк для случайного квантового алгоритма при разных значениях вероятности ошибки дефазировки после каждого гейта. Регистр состоит из 12 кубитов, глубина схемы 40, усреднение по 50 алгоритмам, и 100 усреднений по шумам. Сплошной линией обозначена зависимость  $\exp(-Np)$ , соответствующая распределению Портера-Томаса

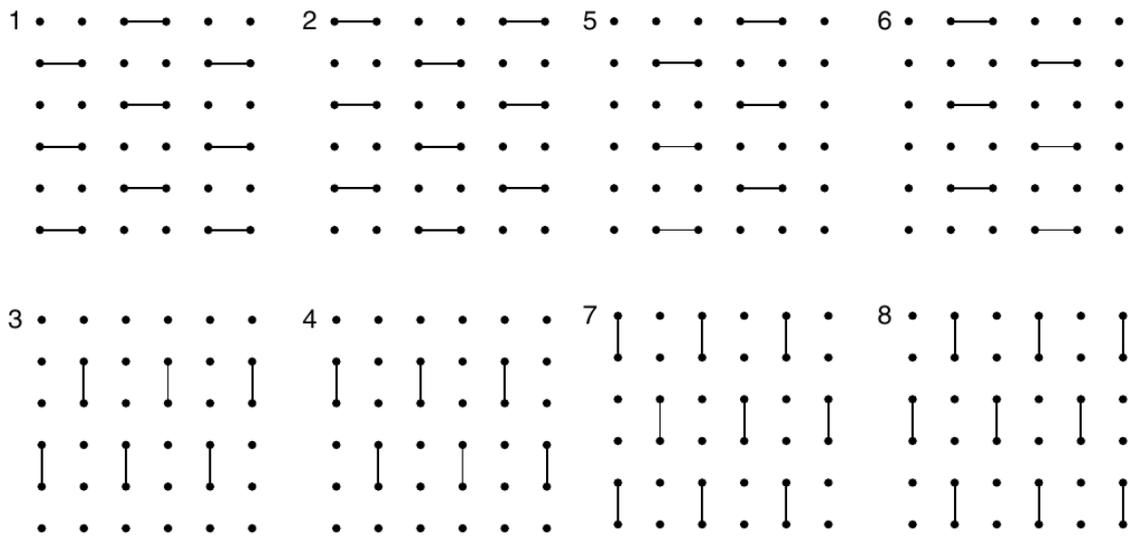


Рисунок 4: Послойная схема применения двухкубитных гейтов при генерации случайного квантового алгоритма на примере 36-кубитного регистра. Для генерации алгоритма произвольной глубины применяется циклично. Рисунок из статьи [21]

Схема применения гейтов CZ на примере квадратного регистра из 36 кубитов изображена на рисунке 4. Без труда можно заметить, что данная схема, даже будучи обобщённой на регистр произвольного размера и формы, обеспечивает запутывание каждого кубита со всеми соседями. Проведённое в данной работе численное моделирование подтверждает, что результаты алгоритмов, сгенерированных по изложенному методу уверенно сходятся к распределению Портера-Томаса уже при глубине схемы в 20 - 30 слоёв. Квантовое превосходство в статье [15] показано для схем глубиной в 20 слоёв.

### 2.3. Кросс-энтропийный бенчмаркинг

Как показано в статьях [21, 15], возможность количественно характеризовать способность какого-либо классического алгоритма симулировать работу квантовой схемы вытекает из различия распределений их выходных битовых строк. Используемый при этом метод получил название кросс-энтропийного бенчмаркинга (cross-entropy benchmarking).

Рассмотрим выборку из  $m$  битовых строк  $S = \{x_1, \dots, x_m\}$ , полученных в результате работы некоего квантового алгоритма  $U$ . Будем считать, что вероятности обнаружения  $p_U(x)$  для каждой строки независимы и подчиняются распределению Портера-Томаса. Рассмотрим теперь другую выборку  $S_{unc} = \{x_1^{unc}, \dots, x_m^{unc}\}$ , имеющую распределение  $p_{unc}(x^{unc})$  и некоррелированную с первой. Нас интересует вероятность  $Pr_U(S_{unc})$ , с которой строки из выборки  $S_{unc}$  могли возникнуть в результате работы заданного квантового алгоритма. Согласно центральной предельной теореме, можно утверждать (смотри статью [21] и дополнительную информацию к ней), что

$$\log(Pr_U(S_{unc})) = -m H(p_{unc}, p_U) + O(m^{1/2}), \quad (3)$$

где

$$H(p_{unc}, p_U) \equiv - \sum_{j=1}^N p_{unc}(x_j) \log(p_U(x_j)) \quad (4)$$

– кросс-энтропия между распределениями  $p_U(x)$  и  $p_{unc}(x)$ . Усреднение по случайным алгоритмам  $\{U\}$  может быть произведено независимо для  $p_{unc}(x)$  и  $\log(p_U(x))$ . При усреднении по случайным алгоритмам, если они имеют достаточную глубину, наблюдаемые вероятности обнаружения фиксированной битовой строки  $\{p_U(x)\}$  также подчиняются распределению Портера-Томаса. Это позволяет ввести величину

$$H_0 \equiv -\langle \log p_U(x_j) \rangle_U = \log N + \gamma . \quad (5)$$

Здесь  $N = 2^n$  – число всевозможных битовых строк, а  $\gamma$  — константа Эйлера.

Возможность произвольного алгоритма  $A$  с распределением вероятностей  $p_A$  для выходных строк генерировать строки из распределения  $p_U$  может быть численно охарактеризована величиной изменения кросс-энтропии:

$$\Delta H(p_A) \equiv H_0 - H(p_A, p_U) . \quad (6)$$

В работе [15] идентичная по смыслу, однако более удобная в отношении производимых вычислений величина вводится как

$$F_{\text{ХЕВ}} = 2^n \langle P(x_i) \rangle_i - 1 , \quad (7)$$

где  $n$  – число кубитов в квантовой схеме,  $P(x_i)$  — вероятность обнаружения битовой строки  $x_i$  на выходе идеальной квантовой схемы, а усреднение ведётся по полученным битовым строкам. Данная величина получила название кросс-энтропийной точности или фиделити (cross-entropy fidelity, ХЕВ-fidelity). Также как и величина изменения кросс-энтропии, кросс-энтропийная точность обращается в единицу при анализе результатов идеальной квантовой схемы, и в нуль при равномерном распределении вероятностей битовых строк. Ещё раз заострим внимание на том, что данные величины допускают усреднение по множеству случайных квантовых алгоритмов и, таким образом, могут являться независимым от конкретной квантовой схемы маркером способности классического алгоритма воспроизводить результаты квантовых вычислений.

Но как с помощью данных величин наглядно продемонстрировать границу квантового превосходства? Ведь они являются характеристикой алгоритма, а на классическом компьютере потенциально может быть

реализован любой алгоритм, в том числе точно сэмплирующий идеальную квантовую схему. Вопрос лишь в том, сколько ресурсов потребуется для выполнения этого алгоритма. Как кросс-энтропийный бенчмаркинг можно связать с мерой ресурсоёмкости сэмплирования квантовой схемы на классическом компьютере?

Для ответа на этот вопрос надо вспомнить тот факт, что выходная битовая последовательность квантового алгоритма является случайной величиной. Имея на руках результат единственного прогона квантовой схемы, об алгоритме нельзя сказать абсолютно ничего. Тогда можно ввести величину  $N_s$  – размер выборки выходных битовых строк, позволяющий оценить величину  $F_{ХЕВ}$  с точностью до стандартного отклонения. Нетрудно показать, что вследствие центральной предельной теоремы эта величина может быть оценена как  $N_s = O(F_{ХЕВ}^{-2})$  [19]. Значение  $F_{ХЕВ}$  для схем с шумами экспоненциально падает с ростом количества кубитов и глубины схемы. Таким образом, схема с большой глубиной, для которой влияние ошибки очень и очень значительно, требует генерации большого количества битовых строк для оценки величины  $F_{ХЕВ}$  и, тем самым, определения вклада первоначального (безошибочного) алгоритма в наблюдаемые результаты её работы.

Тогда, уже оперируя величиной  $N_s$  можно без труда определить границу квантового превосходства. Для этого необходимо реалистично оценить время выполнения заданного случайного алгоритма на квантовом и классическом компьютере. Для квантового компьютера эту оценку легко получить вне зависимости от конкретного сгенерированного алгоритма. Поскольку, как правило, время исполнения всех однокубитных гейтов отличается слабо, оценка времени исполнения случайного алгоритма зависит только от условных правил его генерации. Аналогично, для классических вычислений можно получить либо аналитическую оценку, основанную на

асимптотической сложности функций, составляющих симуляцию эволюции кубитного регистра с данным алгоритмом, либо получить эту же оценку экспериментально. Построив зависимости полученных времён исполнения алгоритма от количества кубитов или глубины схемы, в случае пересечения графиков можно говорить об экспериментальной демонстрации квантового превосходства. Естественно полагается, что в тривиальных случаях малого регистра или малой глубины схемы классический компьютер, ввиду простоты расчётов, будет иметь преимущество перед квантовым, вычисления на котором сопряжены со значительными по времени операциями инициализации и измерения кубитного регистра.

Однако, все изложенные выше рассуждения, ровно как и предпосылки к оценке квантового превосходства, изложенные в статье [21], предполагают, что классическая реализация алгоритма с высоким значением  $F_{ХЕВ}$  возможна только посредством полной и высокоточной численной симуляции квантовых вычислений. В то время как в работе [26] демонстрируется алгоритм, способный показать высокие значения величины  $F_{ХЕВ}$  для заданной квантовой схемы ограниченной глубины, двухкубитные гейты которой генерируются случайно согласно мере Хаара. При этом классическая сложность данного алгоритма полиномиальна по числу кубитов и по вводимой в рассмотрение величине светового конуса — максимальному количеству информации, влияющему на значение каждого выходного бита. Зависимость от величины светового конуса ограничивает глубину схемы, в то время как квантовое превосходство, как ожидается, будет иметь место для квантовых схем со значительными глубинами либо с большим количеством кубитов. Однако демонстрация подобного алгоритма может вновь указывать на несовершенство определения квантового превосходства методом кросс-энтропийного бенчмаркинга, поскольку демонстрация высоких оценок в

данном методе может быть достигнута с меньшими затратами, чем точная симуляция работы квантовой схемы.

Потенциальные возможности ускорения и повышения точности расчётов отнюдь не исчерпываются упомянутыми работами. Ещё можно было бы отметить методы шумоподавления в квантовых схемах [27] и использование схем, составленных из диагональных гейтов [28]. Однако, ввиду новизны этих работ, оставим их детальное изучение для дальнейших исследований.

## 2.4. Модель ошибок

Поскольку работа посвящена исследованию квантового превосходства для компьютеров на основе холодных атомов, важным её этапом было создание симулятора с реалистичной для данной архитектуры моделью ошибок. В качестве основы при определении этой модели послужили работы [29] и [30], а также дополнительная информация к ним. В них произведено обширное описание возможных реализаций двухкубитных гейтов на холодных атомах, осуществимых посредством возбуждения атомов в ридберговское состояние и использования явления ридберговской блокады, методов измерения атомных состояний, а также способов борьбы с некоторыми ошибками. Помимо этого, в дополнительной информации к работе [29] приведён исчерпывающий анализ различных типов ошибок, учёт которых необходим при работе с квантовым вычислителем на основе холодных атомов.

В результате анализа работ [29] и [30] было принято решение об использовании модели, включающей следующие типы ошибок:

1. Вероятностная ошибка обращения фазы при применении одно- и двухкубитных гейтов;
2. Дефазировка состояний кубитов со временем;
3. Динамическая ошибка унитарного поворота;
4. Затухание амплитуды (амплитудный демпинг);
5. Потеря атома;
6. Ошибки измерения и инициализации регистра (state preparation and measurement error, SPAM).

Динамическая ошибка унитарного поворота отражает физическую неточность продолжительности лазерного импульса и реализуется как случайная нормально распределённая поправка к углу поворота вектора квантового состояния.

Потеря атома — специфическая, характерная преимущественно для квантовых вычислений на основе холодных атомов или ионов в ловушках ошибка. Физически она заключается в том, что атом или ион может с конечной вероятностью покинуть ловушку. Поскольку измерения состояний атома производятся при помощи подачи на него лазерного импульса, с точки зрения вычислений это чаще всего будет эквивалентно мгновенному и перманентному коллапсу соответствующего кубита к состоянию  $|0\rangle$  или  $|1\rangle$ , в зависимости от выбора обозначений и метода измерений.

Измерения всегда могут быть вынесены за рамки квантовой вычислительной схемы, поэтому ошибка измерений и инициализации может учитываться единовременно при, непосредственно, работе с результатами

вычислений, через вероятность обращения бита при считывании состояний  $|0\rangle$  и  $|1\rangle$ . Однако, поскольку тут имеется дело с численным моделированием, для оптимизации симуляций может быть удобно вместо разрушающих состояние измерений прибегнуть к нефизичной процедуре расчёта вероятностей исхода измерения определённого кубита или квадратов модулей амплитуд вектора состояния сразу всего кубитного регистра. Особенно это удобно для быстрого получения исчерпывающей информации о распределении вероятностей выходных битовых строк квантовой схемы, необходимой для проведения кросс-энтропийного бенчмаркинга. Далее будет показано, что симуляции могут проводиться и с сохранением чистоты состояния регистра, так что операция получения квадратов модулей амплитуд оказывается очень важной и полезной. Необходимая для учёта SPAM-ошибки корректировка вероятности исходов измерения одного кубита тривиальна. Учёт же ошибки при расчёте квадратов модулей амплитуд заключается в возвращении значения для амплитуды с изменённым номером, причём поправка к номеру генерируется случайно и учитывает параметры ошибки. Кроме того, искомым модуль квадрата должен быть возвращён с коэффициентом, необходимым для сохранения нормировки. К рассмотрению деталей этого вопроса мы вернёмся в дальнейшем.

Ещё один важный тип ошибки — затухание амплитуды или амплитудный демпинг. Данная ошибка реализуется за счёт диссипации энергии в квантовых системах и заключается в смещении состояний к состоянию с наименьшей энергией, в то время как само состояние с наименьшей энергией остаётся неизменным. Для одиночного атома это отражается в его способности к спонтанному излучению и неупругому рассеянию фотонов. Можно показать, что преобразование, реализующее затухание амплитуды в квантовой системе, записывается как

$$\varepsilon_{AD}(\rho) = E_0 \rho E_0^\dagger + E_1 \rho E_1^\dagger, \quad (8)$$

где

$$E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \quad E_1 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}, \quad (9)$$

а  $\gamma$  — вероятность испускания фотона. Для состояний на сфере Блоха данное преобразование будет выглядеть как равномерное сжатие сферы к полюсу, отражающему состояние с меньшей энергией — обычно  $|0\rangle$ .

## 2.5. Шумовые гейты и вероятностные ошибки

Важным вопросом при разработке системы для численного моделирования сложных физических процессов является оптимизация использования вычислительных ресурсов. Один из эффективных методов оптимизации учёта шумов при моделировании квантовых вычислений описан в статье [31]. Он заключается в использовании для учёта шума специальных операторов, называемых шумовыми гейтами. Особенностью шумовых гейтов является то, что с их помощью можно избавиться от необходимости хранить матрицу плотности исследуемого состояния, ограничившись хранением вектора, соответствующего волновой функции. Обратной стороной такой особенности является то, что для получения корректных результатов, шумовые гейты требуют усреднения по учитываемым шумам. Таким образом, с практической точки зрения, использование данного метода позволяет значительно оптимизировать расходование памяти, однако увеличивает продолжительность вычислений, что, впрочем, может быть отчасти компенсировано разумным использованием многопоточности.

Учёт шумовых операторов производится по схеме, представленной на рисунке 5.

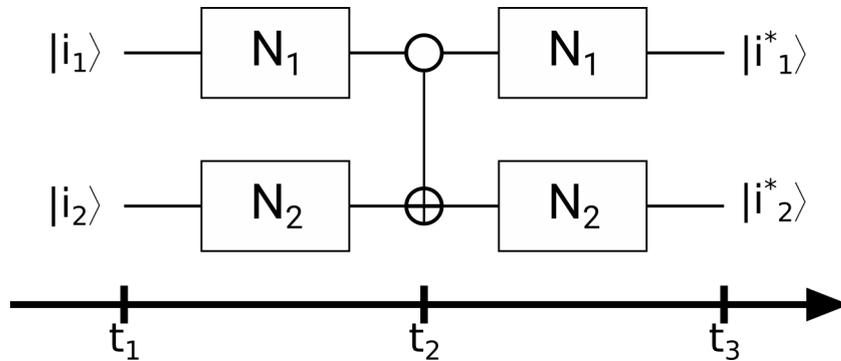


Рисунок 5: Учёт шума при помощи шумовых операторов на примере двухкубитного гейта CNOT

В оригинальной статье приводятся явные виды шумовых операторов  $N$  для различных каналов ошибки, таких как ошибка дефазировки, ошибка обращения бита, амплитудный демпинг. Подразумевается, что данные типы шумов реализуют независимую эволюцию квантового состояния с течением времени. Можно, однако, показать, что с использованием этого же метода возможна реализация и вероятностных ошибок, например сопряженных с применением каких-либо гейтов.

Шумовые гейты, реализующие каналы дефазировки и обращения бита на временном интервале  $(t_1, t_2)$ , записываются в виде

$$N(t_1, t_2) = \exp[i\sqrt{\gamma}\sigma(W_{t_2} - W_{t_1})] . \quad (10)$$

Здесь  $\gamma$  — постоянная связывания кубита со средой, обратная характерному времени дефазировки или обращения бита,  $\sigma$  — одна из матриц Паули, выбираемая в зависимости от того, какой канал ошибки требуется реализовать:  $X$  для обращения бита,  $Z$  для дефазировки,  $Y$  для учёта двух каналов одновременно, а  $(W_{t_2} - W_{t_1}) = \Delta W$  — приращение Винеровского

процесса на заданном интервале — случайная величина с нормальным распределением плотности вероятности.

Рассмотрим теперь, каким образом возможен учёт вероятностной Паули-ошибки. Действие ошибки такого типа на состояние  $\rho$  может быть выражено оператором

$$P_{\sigma}\rho=(1-p)\rho+p(\sigma\rho\sigma) , \quad (11)$$

где  $p$  — вероятность ошибки, а  $\sigma$  — оператор ошибки. В то же время выражение (10) может быть переписано как

$$N(t_1, t_2)=\cos(\sqrt{\mathcal{Y}}\Delta W)I-i\sin(\sqrt{\mathcal{Y}}\Delta W)\sigma , \quad (12)$$

где  $I$  — единичная матрица. Тогда, усредняя действие (12) на состояние  $\rho$ , получим

$$\begin{aligned} \langle N\rho N^{\dagger}\rangle &= \langle (\cos\theta I-i\sin\theta\sigma)\rho(\cos\theta I+i\sin\theta\sigma)\rangle = \\ &= \langle \cos^2\theta\rho+\sin^2\theta\sigma\rho\sigma+i\cos\theta\sin\theta\rho\sigma-i\cos\theta\sin\theta\sigma\rho\rangle = \\ &= \langle \cos^2\theta\rangle\rho+\langle \sin^2\theta\rangle\sigma\rho\sigma , \end{aligned} \quad (13)$$

причём угол поворота  $\theta=\sqrt{\mathcal{Y}}\Delta W=\sqrt{\mathcal{Y}}(W_{t_2}-W_{t_1})$ . Сопоставляя выражение (13) с выражением (11), можно видеть, что требование их тождественности записывается как

$$p=\langle \sin^2(\theta)\rangle=\langle \sin^2(\sqrt{\mathcal{Y}}(W_{t_2}-W_{t_1}))\rangle . \quad (14)$$

Таким образом, механизм шумовых гейтов позволяет учитывать возникновение ошибки с заданной вероятностью  $p$  как среднее влияние шума на соответствующем эффективном временном интервале  $(t_1, t_2)$ . Произведя усреднение и выражая величину эффективного интервала времени  $\Delta t$ , получим

$$p = \frac{1}{2} - \frac{1}{2} e^{-2\gamma \Delta t} , \quad (15)$$

$$\Delta t = -\frac{\ln(1-2p)}{2\gamma} . \quad (16)$$

В последней формуле на вероятность ошибки накладывается ограничение  $p < 0.5$ , что, впрочем, не мешает проведению симуляций.

Значение эффективного времени, полученное по формуле (16), должно быть подставлено в приращение Винеревского процесса в формуле (10). Тогда вероятностная ошибка сможет быть учтена при помощи получившегося шумового гейта. Это позволяет при симуляции работы квантовой памяти использовать формализм векторов состояний вместо матриц плотности, оптимизируя таким образом расходование вычислительных ресурсов компьютера.

Шумовой гейт для амплитудного затухания может быть записан как

$$N_{AD} = \begin{pmatrix} 1 & i\varphi(t, t_0) \\ 0 & \exp[-\gamma \frac{(t-t_0)}{2}] \end{pmatrix} . \quad (17)$$

Здесь  $\gamma$  — по-прежнему вероятность перехода в состояние  $|0\rangle$ , а параметр  $\varphi$  выражается через стохастический интеграл по случайному процессу  $W$ :

$$\varphi(t, t_0) = \sqrt{\gamma} \int_{t_0}^t \exp[\frac{-\gamma}{2T}] dW_T . \quad (18)$$

Особенностью данного шумового гейта является то, что его матрица не эрмитова и не унитарна. Соответственно, данный гейт, в отличие от предыдущих, не может быть описан как поворот на сфере Блоха и нарушает нормировку вектора состояния, что будет требовать дополнительного внимания при численном моделировании.

## 3. Численное моделирование

### 3.1. Детали реализации квантового симулятора

Ввиду отсутствия доступа к функционирующим прототипам квантовых вычислителей на основе холодных атомов, оригинальная часть данной работы ограничивается численным моделированием. Для достижения поставленных целей первоочередной задачей являлось создание эффективного симулятора работы квантовых вычислительных схем<sup>1</sup>.

В качестве вычислительного ядра симулятора была выбрана библиотека QuEST – Quantum Exact Simulation Toolkit, с открытым исходным кодом, находящаяся в свободном доступе [32]. Данная библиотека реализует базовый функционал взаимодействия с абстрактным кубитным регистром, предоставляет инструменты для применения гейтов, базового учёта шумов и измерения полученных состояний.

В первую очередь, с использованием данной библиотеки был создан программный класс, симулирующий работу двумерного кубитного регистра на основе холодных атомов, учитывающий необходимые модели ошибок и использующий шумовые гейты. Удалось успешно реализовать все типы ошибок, оговоренные в разделе «Модели ошибок», за исключением амплитудного демпинга. К сожалению, базовый функционал QuEST не позволяет применять неэрмитовый шумовой гейт к квантовому состоянию, хранящемуся в формате волновой функции. Таким образом учёт амплитудного демпинга был возможен только в случае отказа от использования шумовых гейтов. По заявлению разработчиков QuEST, в ближайшем будущем возможность применения неэрмитовых гейтов к

---

<sup>1</sup> Исходные коды программ, использованных при численном моделировании, находятся в открытом доступе: [github.com/AlexeyMoiseevskiy/RandomQuantumAlgorithms](https://github.com/AlexeyMoiseevskiy/RandomQuantumAlgorithms)

волновым функциям будет добавлена, и симулятор сможет быть усовершенствован.

Вероятностная ошибка обращения фазы следует за каждым применённым гейтом и учитывается с помощью шумовых операторов так, как это объяснено в предыдущем разделе.

Для учёта дефазировки кубитов со временем каждому гейту присваивается время исполнения. В текущей реализации симулятора поддерживается задание длительности однокубитных и двухкубитных гейтов, поскольку их отличие, как правило, гораздо значительнее отличий в длительности между гейтами одного типа. Поскольку в каждом слое квантовой схемы участвуют как однокубитные, так и двухкубитные гейты, у кубитов в процессе работы схемы неизбежно появится время простоя. В симулятор встроен функционал, позволяющий, по мере возможностей, автоматически оптимизировать послойную топологию программируемой квантовой схемы — независимые гейты, действующие на отдельные кубиты по умолчанию размещаются в одном слое и выполняются в симуляции условно одновременно. При этом для каждого кубита запоминается момент последнего применения к нему какого-либо гейта. Таким образом, при каждом следующем обращении к кубиту производится расчёт времени, которое данный кубит находился в простое, и в зависимости от этой величины рассчитывается шумовой гейт, применение которого к кубиту реализует учёт дефазировки.

Поскольку все гейты, используемые в квантовом алгоритме, являются унитарными вращениями, учёт динамической ошибки требует только реализации гейтов непосредственно через вращения, а не через сторонние специализированные функции, и генерации случайной нормально распределённой поправки к углу поворота.

Учёт потери атома также не представляет сложности. Перманентный коллапс данного кубита к состоянию  $|0\rangle$  реализуется через встроенную функцию проективного измерения с фиксированным исходом.

Симуляция ошибки измерения и инициализации, как было упомянуто ранее, нетривиальна в случае получения информации о кубитном регистре через расчёт квадратов модулей амплитуд вектора состояния. Для правильного учёта ошибки в этом случае требуется возвращать квадрат модуля амплитуды с изменённым номером и с дополнительной нормировкой.

Пусть имеется кубитный регистр из  $n$  кубитов. Тогда, если он находится в чистом состоянии, он может быть охарактеризован вектором из  $N = 2^n$  амплитуд. Номер  $x$  каждой амплитуды тогда представляет собой  $n$ -битное число. Каждому его биту может быть сопоставлен один кубит. Таким образом, номер  $k$  отображает некоторое состояние кубитного регистра, а точнее некоторый результат его измерения. Вероятность наблюдения этого результата характеризуется квадратом модуля амплитуды с данным номером.

Пусть теперь имеются вероятности  $E_{\Gamma_{0 \rightarrow 1}}$  и  $E_{\Gamma_{1 \rightarrow 0}}$  ошибок при измерении нуля и единицы соответственно. Тогда номер амплитуды, модуль квадрата которой должен быть возвращён вместо искомого, определяется подменой в побитовой записи  $x$  каждого нуля на единицу с вероятностью  $E_{\Gamma_{1 \rightarrow 0}}$ , и каждой единицы на нуль с вероятностью  $E_{\Gamma_{0 \rightarrow 1}}$ . В среднем это будет эквивалентно ошибкам в измерении кубитов, сопоставленных изменяемым битам.

Значения вероятностей ошибок  $E_{\Gamma_{0 \rightarrow 1}}$  и  $E_{\Gamma_{1 \rightarrow 0}}$ , вообще говоря, не должны быть равными. Но их различие приводит к нарушению нормировки суммы квадратов модулей амплитуд  $P(x)$ . В отсутствие ошибки

$$\sum_{x=0}^n P(x) = 1 \quad . \quad (19)$$

Суммирование ведётся по номерам амплитуд. Если добавлена ошибка, то вместо искомого значения  $P(x)$  будет возвращено значение  $P(y)$ , соответствующее другому номеру:

$$P(y) = \sum_{x=0}^n P(y|x)P(x) . \quad (20)$$

Здесь  $P(y|x)$  – вероятность перехода номера  $x$  в номер  $y$  при учёте ошибки.

Введём нормировочный коэффициент  $C_y$

$$\sum_{x=0}^n \frac{P(y|x)}{C_y} = 1 . \quad (21)$$

Он может быть выражен через произведение условных вероятностей для отдельных битов:

$$C_y = \sum_x \prod_i^N P(y_i|x_i) = \sum_x \prod_i^N (P(0|x_i)P(y_i=0) + P(1|x_i)P(y_i=1)) . \quad (22)$$

Запись  $x_i$  обозначает бит номер  $i$  в числе  $x$ . Отсюда получаем, что

$$C_y = \prod_i^N b_y^1(i)(1 + Err_{0 \rightarrow 1} - Err_{1 \rightarrow 0}) + b_y^0(i)(1 - Err_{0 \rightarrow 1} + Err_{1 \rightarrow 0}) = \prod_i^N p_i . \quad (23)$$

Функция  $b_y^{val}(i) = (y_i + 1 + val) \bmod 2$  здесь равна 1, если бит числа  $y$  под номером  $i$  совпадает со значением  $val$ , и 0 иначе. Из общих соображений также понятно, что искомым коэффициентом для амплитуды  $x$  является произведением параметров  $p_i$ , характеризующих способность бита  $y_i$  принимать значения  $x_i$ . Если ошибки нет,  $p_i \equiv 1$ . Если есть — имеем  $p_i = (1 - Err_{1 \rightarrow 0} + Err_{0 \rightarrow 1})$  для бита в значении 1 и  $p_i = (1 + Err_{0 \rightarrow 1} - Err_{1 \rightarrow 0})$  для бита в значении 0. Если ошибка симметрична, то есть  $Err_{1 \rightarrow 0} = Err_{0 \rightarrow 1}$ , то в среднем опять  $p_i = 1$  и нарушения нормировки не происходит. По смыслу, параметр  $p_i$  похож на вероятность, с той лишь разницей, что в случае  $Err_{1 \rightarrow 0} \neq Err_{0 \rightarrow 1}$  способен принимать значения больше единицы.

Таким образом, при расчёте амплитуды  $A_x$  для корректного учёта ошибки измерений необходимо  $A_x$  вместо использовать значение

$$\tilde{A}_x = C_x A_y, \quad (24)$$

где  $y$  получен из  $x$  путём вероятностной подмены битов, описанным выше.

Вторым этапом разработки симулятора стало добавление класса-генератора случайных алгоритмов. Позволю себе сделать оговорку, что, несмотря на фактическую тривиальность, в техническом плане эта задача требует большого внимания и аккуратности, поскольку заключается в обеспечении точного взаимодействия большого количества элементов. Важной задачей класса-генератора является обособление понятия квантового алгоритма от понятия кубитного регистра. Это обеспечивает, например, возможность выполнения одной случайно сгенерированной квантовой схемы на двух отдельных кубитных регистрах с различными показателями ошибок для дальнейшего сравнения результатов.

### 3.2. Тестирование симулятора с различными ошибками

Маркером, показывающим корректность работы квантового симулятора случайных алгоритмов может служить, опять же, способность симулятора генерировать битовые строки с распределением Портера-Томаса. Вспомним, что гистограмма распределения вероятностей битовых строк для идеального симулятора аппроксимируется экспоненциальной зависимостью, а по мере нарастания ошибок, схлопывается к дельта-функции, соответствующей равновероятной генерации всех возможных строк. Данная зависимость для растущей вероятности ошибки обращения фазы в каждом гейте приводилась

как иллюстрация на рисунке 3. Рассмотрим также случаи других ошибок на рисунках 6, 7 и 8.

Прежде всего, из рисунков 6, 7 и 8 можно констатировать, что разработанный симулятор и генератор случайных алгоритмов функционируют корректно, при отсутствии ошибок успешно воспроизводя распределение Портера-Томаса. На всех рисунках распределение Портера-Томаса обозначено сплошной линией, а данные получены для схемы глубиной 40, с усреднением по 50 алгоритмам и 100 реализациям шума.

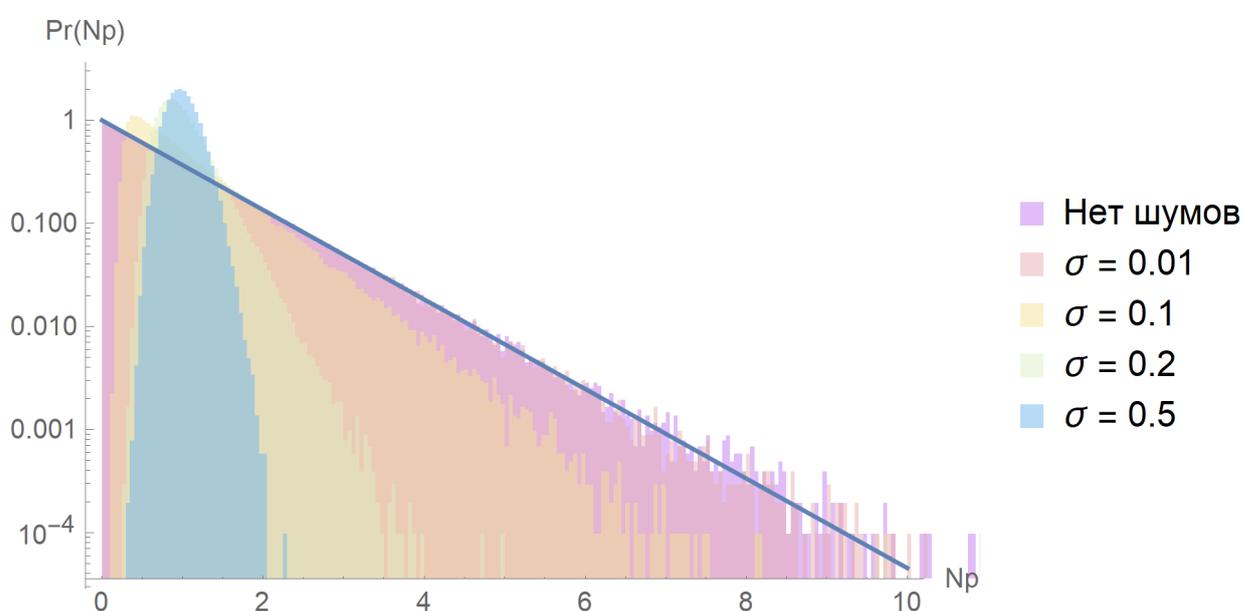


Рисунок 6: Гистограмма распределения перенормированных на общее число  $N$  вероятностей выходных битовых строк для случайного квантового алгоритма при разных значениях стандартного отклонения динамической ошибки поворота для каждого гейта в радианах. Регистр из 12 кубитов, глубина схемы 40, усреднение по 50 алгоритмам, и 100 реализациям шумов. Сплошной линией обозначена зависимость  $\exp(-Np)$ , соответствующая распределению Портера-Томаса

Рисунок 6 иллюстрирует влияние динамической ошибки на распределение. Зависимость вида гистограммы от величины стандартного отклонения ошибки унитарного поворота в целом подобна зависимости от вероятности ошибки обращения фазы.

Также на этом рисунке видно, что при росте ошибки динамика вида гистограммы замедляется и при  $\sigma = 0.5$  радиан становится близка к нормальному распределению с конечной шириной. Это обусловлено ограниченностью числа усреднений по различным алгоритмам. Согласно центральной предельной теореме, при большом количестве усреднений  $n$ , ширина распределения будет оцениваться как  $n^{-1/2}$ .

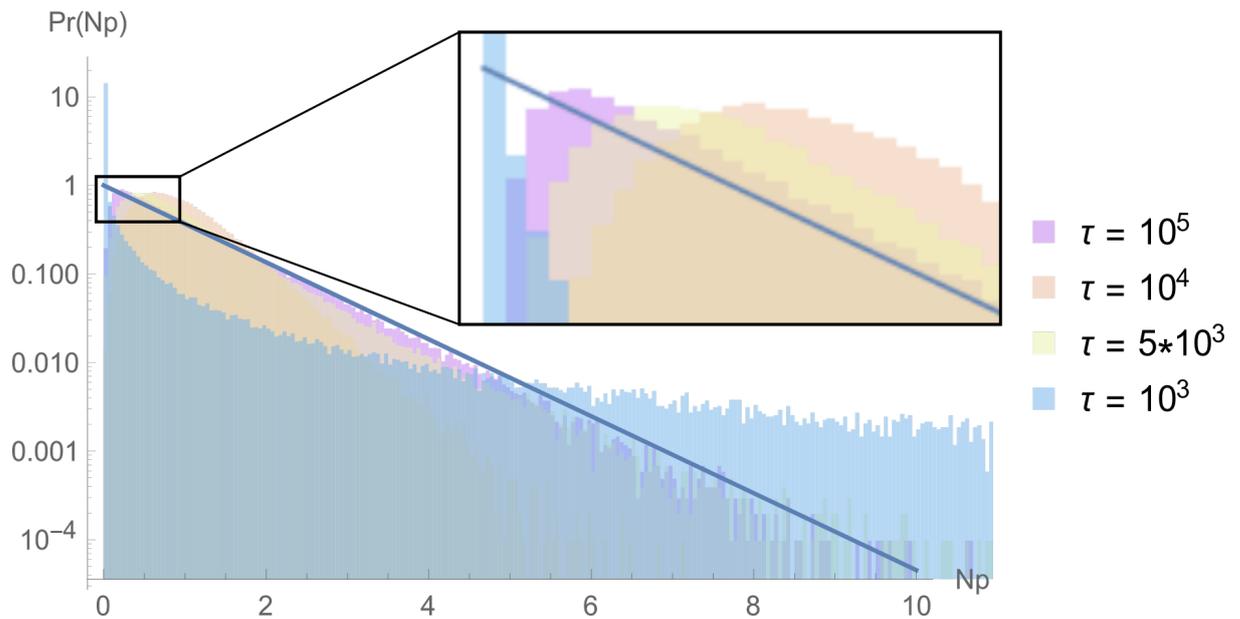


Рисунок 7: Гистограмма распределения перенормированных на общее число  $N$  вероятностей выходных битовых строк для случайного квантового алгоритма при разных значениях характерных значениях времени потери атома в единицах времени, равных продолжительности однокубитного гейта. Сплошной линией обозначена зависимость  $\exp(-Np)$ , соответствующая распределению Портера-Томаса

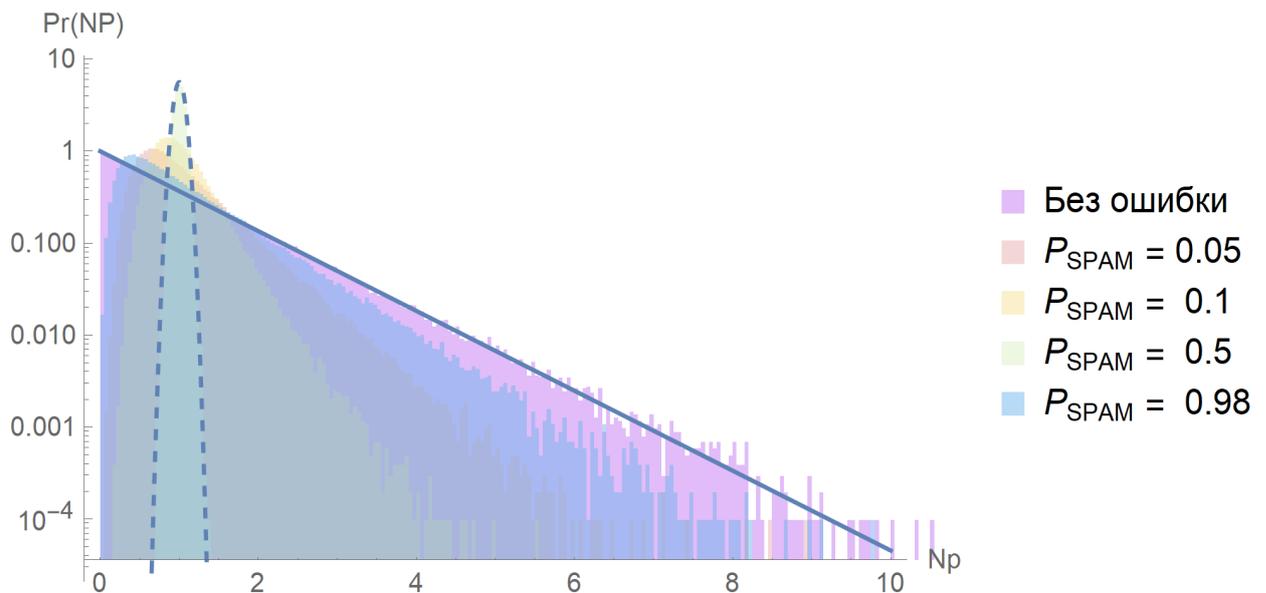


Рисунок 8: Гистограмма распределения перенормированных вероятностей выходных битовых строк для случайного квантового алгоритма при разных значениях вероятности ошибки измерения. Сплошной линией обозначена зависимость  $\exp(-Np)$ . Для значения ошибки  $P_{\text{SPAM}} = 0.5$  выполнено 200 усреднений по алгоритмам. Пунктирной линией обозначено нормальное распределение со средним 1 и отклонением  $1/\sqrt{200}$

Демонстрация влияния потерь атомов на вид распределения приведена на рисунке 7. Характерное время потери атома  $\tau$  приведено в единицах, равных продолжительности однокубитного гейта. Продолжительность двухкубитного гейта принята за 10 единиц. На масштабной врезке можно увидеть, что пик распределения при уменьшении  $\tau$  первоначально движется так, как если бы распределение стремилось к нормальному, подобно предыдущим случаям. Однако, по достижении некоторого характерного времени, пик устремляется в область нуля и зависимость приобретает характер резко спадающей. Это связано с тем, что первоначально, пока количество потерянных за время работы схемы атомов мало, вклад потерь подобен вкладу вероятностной ошибки. Однако когда характерное время потери атома по порядку становится равно времени работы схемы, умноженному на количество кубитов, вклад потерянных атомов в конечные состояния становится определяющим, и поскольку потерянные кубиты при

измерении дадут результат 0, вероятности возникновения разнообразных битовых строк резко падают, смещая пик гистограммы к нулю.

Влияние SPAM-ошибки приведено на рисунке 8. Тут вероятности ошибки при считывании нуля и единицы положены равными. В целом, картина снова похожа на влияние вероятностной ошибки, но с той разницей, что по достижении вероятности ошибки  $P_{SPAM} = 0,5$ , при дальнейшем росте распределение возвращается к исходному.  $P_{SPAM} = 0,5$  соответствует ситуации, когда измерения не могут отличить 1 от 0, и считываемые значения определяются только случайной ошибкой. Это же подтверждается наложением на гистограмму нормального распределения со средним 1 и отклонением  $1/\sqrt{N_{avr}}$ , где  $N_{avr}$  — число усреднений по алгоритмам. При значении  $P_{SPAM} = 0,98$  с высокой вероятностью вместо нулей считываются единицы, а вместо единиц — нули. Таким образом, статистика исходов совпадает со случаем  $P_{SPAM} = 0,02$  и гистограмма вновь приближается к распределению Портера-Томаса.

Большинство симуляций в данной работе проводились для квантовых схем с глубиной 40. Влияние глубины схемы на распределение вероятностей исходов отражено на рисунке 9. Видно, что схема с глубиной 16 недостаточно хорошо аппроксимируется распределение Портера-Томаса, в то время как разница между схемами с глубиной 32 и 128 практически не видна.

Подобные графики были построены также для определения оптимального количества усреднений по алгоритмам и реализациям шумов, однако здесь они не приводятся ввиду своей малой информативности. Значения, используемые в большинстве симуляций — усреднение по 30 или 50 алгоритмами и по 50 либо 100 реализациям шумов.

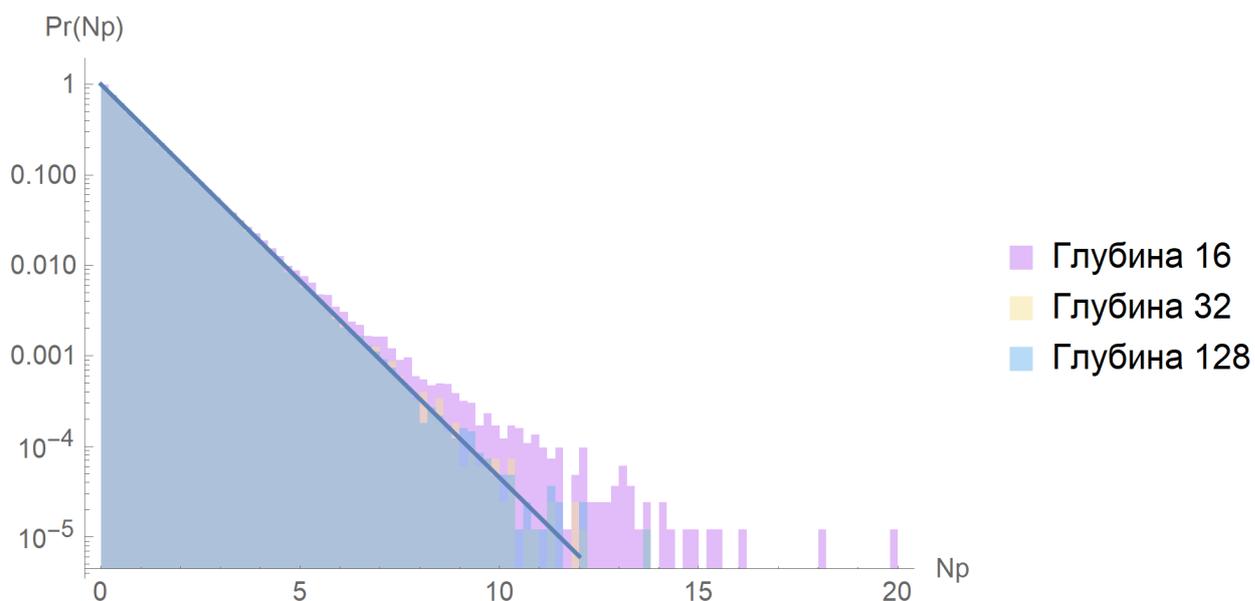


Рисунок 9: Гистограмма распределения перенормированных вероятностей выходных битовых строк для случайных квантовых алгоритмов с разными глубинами. Сплошной линией обозначена зависимость  $\exp(-N_p)$

### 3.3. Анализ распространения ошибки в кубитном регистре

Интерес представляет анализ поведения регистра при внесении в состояние отдельного кубита намеренной ошибки сверх всех учитываемых шумов. Можно ожидать, что по мере выполнения запутывающих гейтов с кубитом, в который была добавлена ошибка, влияние ошибки будет распространяться по регистру. Когда же запутывающий гейт будет применяться между кубитами, уже испытывающими влияние ошибки, могут возникнуть интересные эффекты взаимодействия, такие как взаимное усиление или наоборот, компенсация влияния ошибки.

Для изучения подобных явлений была создана программа-симулятор, генерирующая случайный алгоритм и исполняющая его в двух регистрах.

После прохождения трёх полных циклов запутывающих гейтов, что соответствует глубине в 24 слоя (смотри схему применения двухкубитных гейтов в случайном алгоритме, рисунок 4) в один из регистров, в кубит с

координатами  $(0, 0)$ , добавляется детерминированная X-ошибка. После этого, для каждого последующего слоя измеряется модуль разности проекции состояний одинаковых кубитов в разных регистрах на базисный вектор, то есть отличие вероятностей получения одних и тех же исходов при одинаковых однокубитных измерениях соответствующих кубитов. Результат такого исследования представляется как динамическая во времени картина изменения модуля разности проекций для каждого кубита из двумерного регистра. Избранные кадры этой динамики представлены на рисунке 10.

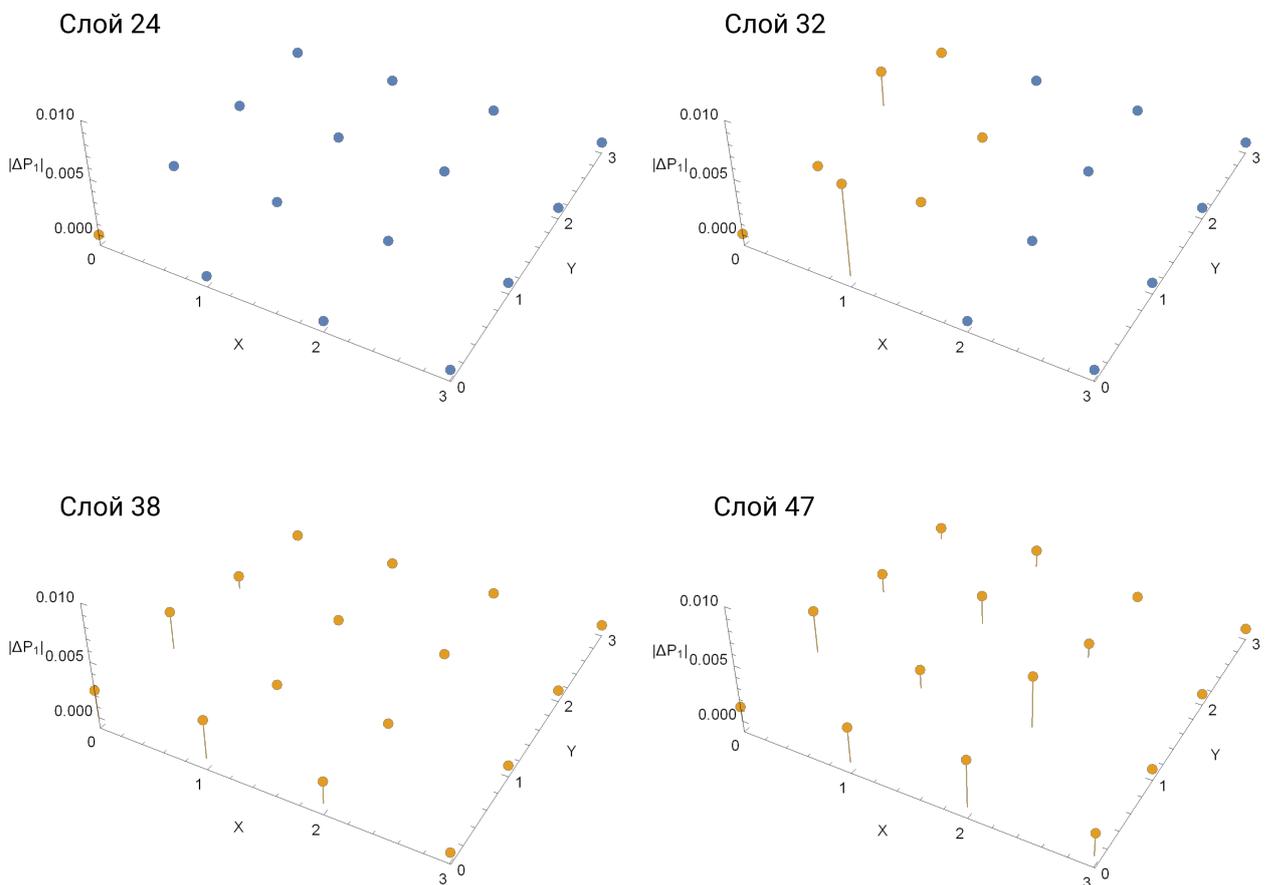


Рисунок 10: Модуль разности проекции кубитных состояний на базисный вектор для регистра с ошибкой и без. Синим отмечены кубиты, ещё не испытавшие на себе влияние ошибки.

Видно, что выраженную картину «волнового фронта» ошибки пронаблюдать не удалось. Более того, крайне интересным представляется тот факт, что отклонения модуля разности проекций от нуля начинают проявляться только в 32 слое, то есть ещё через один полный цикл запутывающих гейтов от момента внесения ошибки. Если считать, что ошибка распространяется через двухкубитные гейты, то к этому моменту под её влиянием окажутся уже семь кубитов. Единственной особенностью слоя 32, которая бросается в глаза, является то, что на этом слое впервые происходит применение запутывающего гейта к двум кубитам, которые оба уже испытали влияние ошибки. Однако тут речь идёт о кубитах с координатами  $(1, 0)$  и  $(1, 1)$ , а как можно видеть из рисунка отклонение наблюдается для кубитов  $(1, 0)$  и  $(0, 2)$ .

Некоторое подобие волнового фронта начинает формироваться только к слою номер 38, когда ошибка уже распространилась по всему регистру. Сравнив изображения для слоёв 38 и 47, можно увидеть, как видимое возмущение распространяется по кубитному регистру. Чётко видно, по направлению движения волны, что она исходит со стороны кубита  $(0, 0)$ . К сожалению, после достижения волновым фронтом противоположного края регистра, читаемость картины теряется. Подобное развитие событий наблюдалось и для других позиций внедрения ошибки, и на примере регистра линейной топологии. Возможно, данное исследование смогло бы привести к более обширным выводам при проведении симуляции для регистров значительно больших размеров, хотя бы порядка  $10 \times 10$  кубитов. Однако, симуляция квантовых вычислений на столь объёмных регистрах для классического компьютера сегодня является непосильной задачей.

### 3.4. Определение реалистичных параметров ошибок

Определение границ квантового превосходства через численное моделирование требует учёта реалистичных ошибок. Используемую модель ошибок вместе с техническими подробностями их учёта мы подробно обсудили ранее. Необходимо, однако, прежде чем делать выводы о квантовом превосходстве, на практике убедиться, что данная модель в достаточной степени реалистична.

Обеспечение работы двухкубитного гейта — одна из самых больших сложностей в реализации квантового компьютера на основе холодных атомов. Именно двухкубитный гейт обладает наибольшими временем исполнения и характерной вероятностью ошибки. Однокубитные гейты такой сложности в реализации не вызывают, поскольку не требуют взаимодействия нейтральных атомов. Опираясь на это, можно предположить, что усреднённая таблица истинности двухкубитного гейта CZ может быть хорошим маркером качества модели ошибок.

В статье [30] авторы приводят экспериментальную таблицу истинности гейта CNOT. Функционал симулятора, по сторонним причинам, предоставляет в качестве двухкубитного гейта гейт CZ. Чтобы получить таблицу истинности CNOT такого симулятора, потребуется использование двух операторов Адамара для поворота базиса. Помимо этого, в статье [30] для считывания состояния атома используется выталкивающий импульс (pushout pulse), что приводит к получению сигнала от атомов в состоянии  $|0\rangle$ , а не  $|1\rangle$ . Поскольку переобозначения состояний не происходит, таблица истинности CNOT в оригинальной статье зеркально отражена относительно её привычного вида, приведённого на рисунке 11. Таблица для симулятора на рисунке 11 получена после  $10^6$  усреднений по шумам.

Можно видеть, что в основном таблицы достаточно близки. Для симулятора наблюдаются лишь более сильные по сравнению с экспериментом занижение количества исходов, ортогональных корректному. Максимальное отличие вероятностей исходов действия CNOT для симулятора и эксперимента при любых входных значениях не превышает 1,85% от суммы элементов в строке.

		а).				б).			
Вход									
00⟩		97.11	1.93	0.97	0.02	97.3	1.7	0.8	0.2
01⟩		3.97	95.02	0.03	0.99	2.4	96.4	0.2	1.0
10⟩		0.15	2.91	3.76	93.25	2.0	1.5	3.2	93.3
11⟩		2.90	0.06	95.00	1.91	1.7	1.3	94.4	2.6
		00⟩	01⟩	10⟩	11⟩	00⟩	01⟩	10⟩	11⟩
		Выход				Выход			

Рисунок 11: Таблицы истинности гейта CNOT а). — используемого симулятора, полученная по  $10^6$  усреднений; б). — по экспериментальным данным статьи [30]. Значения указаны в процентах исходов. Условие нормировки требует суммирования вероятностей по строкам в 100%

Данные значения достигнуты при следующих параметрах ошибок:

- вероятность обращения фазы для однокубитного гейта — 0,001;
- вероятность обращения фазы для мультикубитного гейта — 0,01;
- постоянная связывания со средой — 0,001;
- длительность гейтов: 0,1 — однокубитного, 1,0 — двухкубитного;
- характерное время потери атома —  $10^4$ ;
- стандартное отклонение динамического шума — 0,04 рад.;

— вероятность ошибки считывания нуля — 0,01;

— вероятность ошибки считывания единицы — 0,03.

Величины представлены в безразмерных относительных единицах.

Эти значения будут в дальнейшем использоваться при моделировании работы квантовых схем.

### 3.5. Демонстрация квантового превосходства

В предыдущих разделах мы обсудили метод кросс-энтропийного бенчмаркинга. Напомним, что ключевую роль в нём играет величина  $F_{ХЕВ} = 2^n \langle P(x_i) \rangle_i - 1$ , характеризующая близость заданного набора битовых строк к выходным данным идеальной квантовой схемы. Интерес представляет зависимость данной величины от количества кубитов в регистре, глубины схемы и параметров ошибки. Поскольку все результаты в данной работе получаются из численного моделирования, для расчёта  $F_{ХЕВ}$  проводится симуляция работы одного и того же случайного квантового алгоритма в регистре с ошибками и без. После исполнения алгоритма для регистров производится расчёт квадратов амплитуд векторов их состояний, что сразу даёт исчерпывающую информацию о распределении вероятностей генерации битовых строк. В ходе расчётов производится усреднение по шумам. Далее, значения  $F_{ХЕВ}$ , полученные для различных случайных алгоритмов могут быть также усреднены, образуя независимую характеристику квантовой вычислительной машины с заданным числом кубитов, глубиной исполняемого алгоритма и параметрами ошибок.

На рисунке 12 приведена зависимость величины  $F_{ХЕВ}$  от количества кубитов, полученная по результатам численного моделирования для схем с

разной глубиной. Интересно отметить, что для глубины 40 значения  $F_{ХЕВ}$  для  $N = 12$ ,  $N = 16$  и  $N = 18$  несколько выпадают из общей тенденции. Это, как выяснилось, происходит из-за изменения топологии регистра. Данные регистры были реализованы как  $3 \times 4$ ,  $4 \times 4$  и  $3 \times 6$  соответственно, в то время как остальные регистры, по понятным причинам, имели вид  $2 \times N/2$ . При проведении симуляций для схем с глубиной 20 все регистры были представлены как  $2 \times N/2$ , и провал для точки  $N = 12$  на графике не наблюдается.

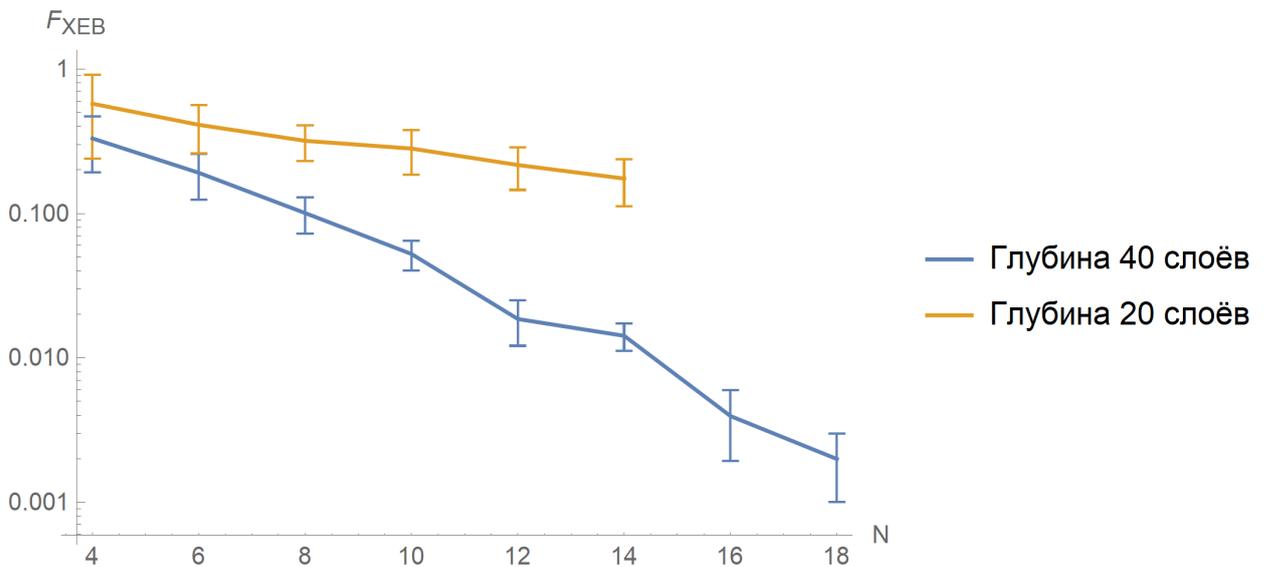


Рисунок 12: Зависимость величины  $F_{ХЕВ}$  квантовой схемы от количества кубитов  $N$

Пересчитаем теперь полученные данные так, чтобы получить график, наглядно демонстрирующий квантовое превосходство. Как отмечалось ранее, количество битовых строк, генерация которых необходима для определения величины  $F_{ХЕВ}$  с точностью до стандартного отклонения, оценивается как  $N_s = O(F_{ХЕВ}^{-2})$ . Иными словами, это минимальное количество строк, которое нужно получить, чтобы утверждать, что исследуемый алгоритм имеет с идеальной квантовой схемой хоть что-то общее. Получим из этой величины оценку времени, необходимого для генерации такого количества строк.

Для оценки времени исполнения алгоритма на квантовом компьютере возьмём следующие оценочные параметры. Время двухкубитного гейта — 1мкс, время приготовления регистра — 20мс, время измерения — 10мс. Общее время работы схемы в миллисекундах будет выражено через глубину схемы  $D$  как

$$T_{\text{quantum}}(n, D) = 0.001 D F_{\text{ХЕВ}}(n, D)^{-2} + 20 + 10 . \quad (25)$$

Здесь учтено, что гейты в одном слое выполняются параллельно и в каждом слое есть хотя бы один двухкубитный гейт, из чего следует, что время выполнения слоя равно времени выполнения двухкубитного гейта. Полученные данные и их экспоненциальная аппроксимация представлены на рисунке 13.

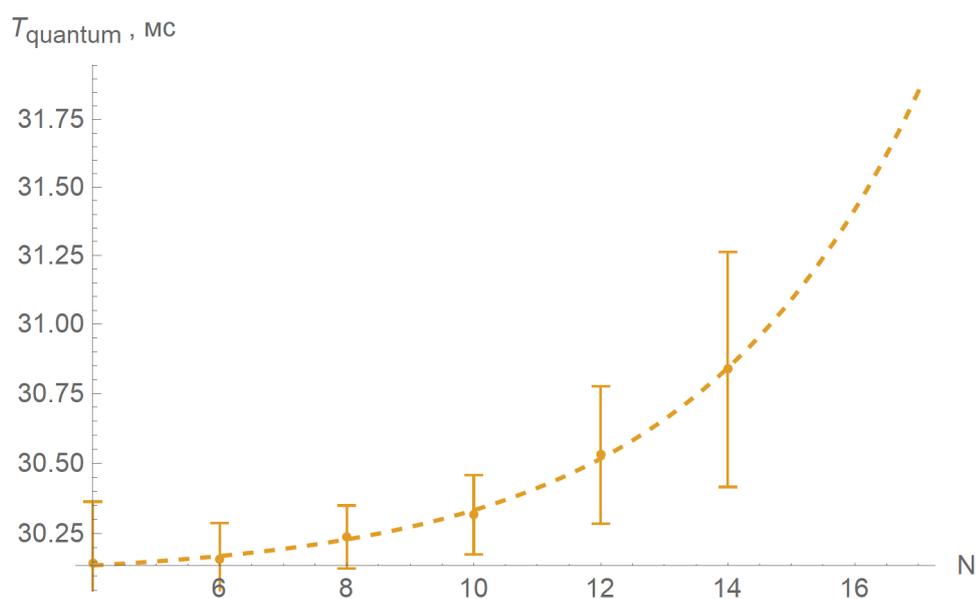


Рисунок 13: Аппроксимация зависимости времени расчётов на квантовом компьютере от числа  $N$  кубитов в схеме. Глубина схемы — 20 слоёв. Пунктирной линией обозначена зависимость  $30.0 + \exp[-4.29 + 0.29 N]$

Оценка времени классического расчёта требует более аккуратного подхода. Прежде всего, поскольку квантовая схема составлена из простых

гейтов и нет нужды перемножать полные матрицы, сложность применения гейта можно оценить как  $2^n$ , где  $n$  – число кубитов. Далее можно использовать правило, по которому происходит генерация случайного алгоритма: два однокубитных гейта не применяются к одному и тому же кубиту, если между ними на данный кубит не действовал оператор CZ. Поэтому можно утверждать, что за каждым CZ следует два однокубитных гейта, и при большой глубине схемы общее число гейтов может быть оценено как утроенное число гейтов CZ. Также необходимо помнить, что для классического компьютера, в отличие от квантового, нет нужды исполнять весь алгоритм для генерации каждой битовой строки. Поскольку разрушения сгенерированных состояний при измерении не происходит, достаточно исполнить алгоритм один раз. В завершение, полученную величину необходимо разделить на быстродействие вычислителя. Здесь и далее предлагается использовать быстродействие суперкомпьютера Ломоносов-2 по результатам теста Linpack, опубликованным на сайте мирового рейтинга суперкомпьютеров ТОП-500:  $F^{Lomonosov2} = 2,478 \text{ PFlop/s} = 2,478 \text{ TFlop/ms}$  [33].

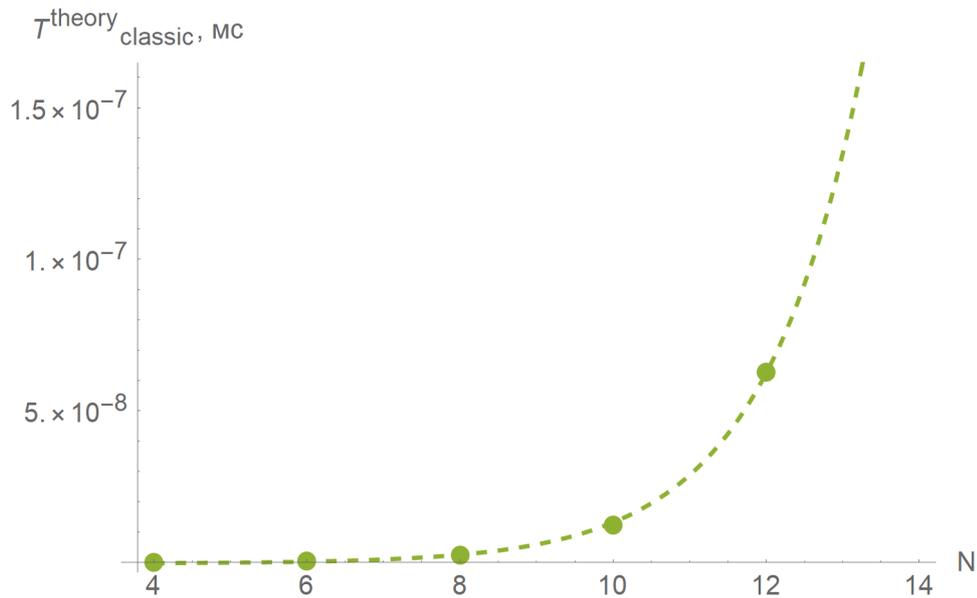


Рисунок 14: Аппроксимация зависимости времени расчётов на классическом компьютере, оцененном из общих соображений, от числа  $N$  кубитов в схеме. Глубина схемы — 20 слоёв. Пунктирной линией обозначена зависимость  $\exp[-18.93 + 0.44 N]$

Итак, формула для теоретической оценки времени классического расчёта:

$$T_{classic}^{theory}(n, D) = \frac{3 N_{CZ}(n, D) 2^n}{F^{Lomonosov 2}} . \quad (26)$$

$N_{CZ}$  – число гейтов CZ. Полученные значения и их экспоненциальная аппроксимация представлены на рисунке 14.

Однако расчёты на классическом компьютере могут иметь большое количество неучтённых в последней формуле особенностей, начиная от использования многопоточности, заканчивая затратами времени на выделение памяти. Поэтому, помимо предыдущих оценок, было произведено экспериментальное измерение времени расчётов на классическом компьютере, а также оценка его быстродействия программой Linpack, с тем чтобы пересчитать полученное время в оценку времени выполнения расчётов на суперкомпьютере. При измерении, время непосредственно симуляции квантовой схемы и время последующих расчётов битовых строк учитывалось отдельно.

Вычисления проводились на процессоре Intel Core i7 8565u. Тест Linpack в однопоточном режиме показал быстродействие  $F^{Corei7} = 114.9 \pm 15.0$  GFlop/s. Формула пересчёта

$$T_{classic}^{exp} = [\langle T_{симуляции} \rangle + F_{ХЕВ}^{-2} \langle T_{измерения} \rangle] \frac{F^{Corei7}}{F^{Lomonosov 2}} . \quad (27)$$

Полученные значения и их экспоненциальная аппроксимация представлены на рисунке 15.

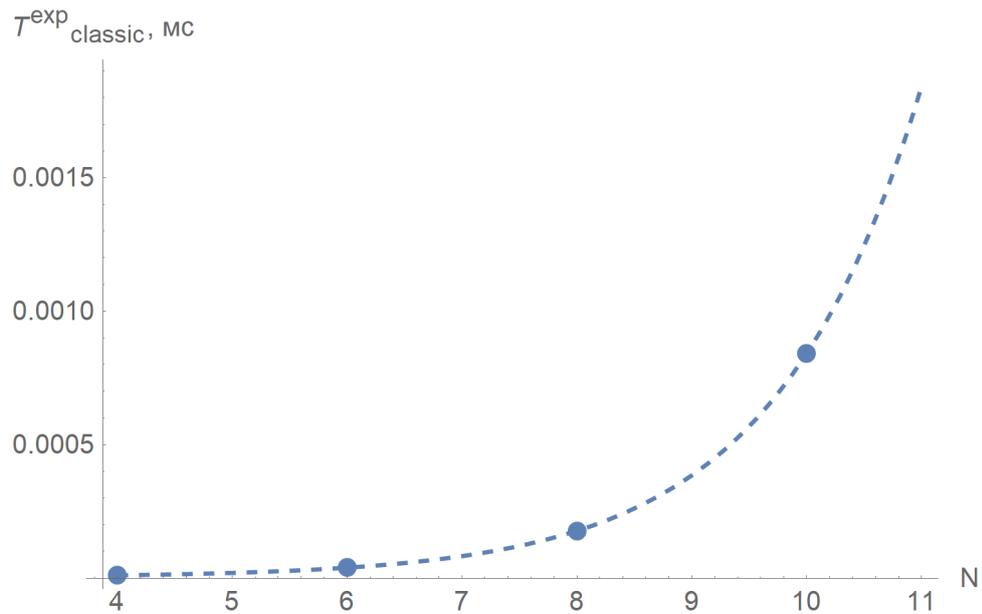


Рисунок 15: Аппроксимация зависимости времени расчётов на классическом компьютере, пересчитанном из экспериментальных данных, от числа  $N$  кубитов в схеме. Глубина схемы — 20 слоёв. Пунктирной линией обозначена зависимость  $\exp[-14.95 + 0.79 N]$

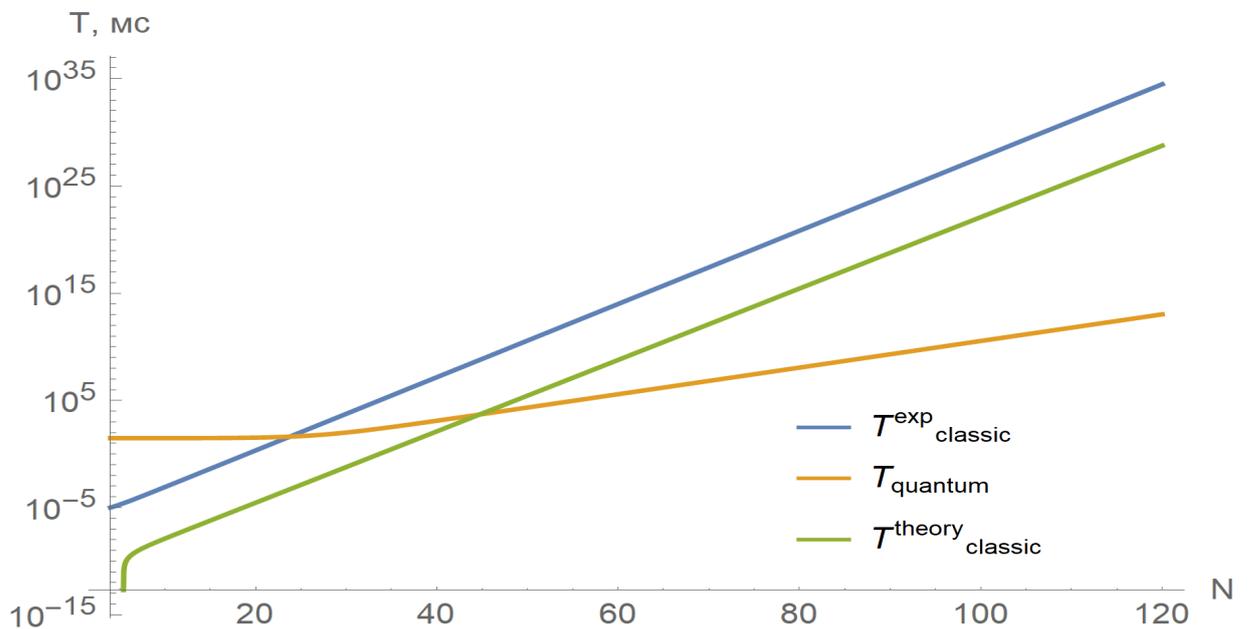


Рисунок 16: Экспоненциальная аппроксимация зависимостей времени классического и квантового расчёта от числа кубитов  $N$  в схеме. Глубина схемы — 20 слоёв. Точки пересечения представляют собой границу квантового превосходства

### 3.6. Результаты

На рисунке 16 аппроксимации всех трёх оценок времени изображены вместе. Это является основным результатом работы. Точки пересечения зависимости  $T_{\text{quantum}}(N)$  с зависимостями  $T_{\text{classic}}^{\text{exp}}(N)$  и  $T_{\text{classic}}^{\text{theory}}(N)$  есть не что иное, как границы практического квантового превосходства для соответствующих оценок быстродействия классического компьютера.

Из полученных данных следует, что квантовый компьютер на основе холодных атомов в задаче расчёта случайных квантовых схем глубиной в 20 слоёв окажется производительнее суперкомпьютера Ломоносов-2 при количестве кубитов в регистре от 21 и выше. Абстрактный классический компьютер с идентичным быстродействием начнёт уступать квантовому при 41 кубите в регистре. Данные количества кубитов являются вполне достижимыми. Таким образом можно уверенно говорить о достижимости практического квантового превосходства в реализуемой модели квантового компьютера на основе холодных атомов.

## ВЫВОДЫ

В ходе данной работы поставленные цели были достигнуты. Изучен метод определения границы квантового превосходства с помощью кросс-энтропийного бенчмаркинга (рисунки 12, 16). Разработан симулятор эволюции кубитного регистра, реализующий оптимизацию памяти посредством использования шумовых гейтов. Разработан генератор случайных квантовых схем для изучения задачи сэмплирования рандомизируемых квантовых алгоритмов. Путём анализа распределения вероятностей выходных битовых строк подтверждена корректность работы симулятора и определены оптимальные параметры для проведения расчётов (рисунки 3, 6 — 9). Проведены наблюдения особенностей динамики влияния детерминированной ошибки в кубитном регистре. Построена картина волнового фронта (рисунок 10). Определен список ошибок симулятора и значений их параметров, учёт которых обеспечивает совпадение модельной и экспериментальной таблицы истинности гейта CNOT с максимальным отклонением не более 1.85% от нормировки строк (рисунок 11). Показано, что для реалистичной модели квантового компьютера на основе холодных атомов возможно достижение квантового превосходства для схем глубиной в 20 слоёв при количестве кубитов от 21 и более (рисунок 16).

## ЗАКЛЮЧЕНИЕ

Данная работа оставляет огромное пространство для дальнейших исследований. В данный момент в симулятор уже добавлена возможность учёта амплитудного демпинга и ведётся работа над комплексным уточнением параметров ошибок.

Большой интерес представляет разработка альтернативного симулятора, использующего какой-либо метод оптимизации [17, 18, 26, 27, 28].

Есть потенциал получения из дальнейших симуляций более обширных и данных о зависимости величины  $F_{ХЕВ}$  от глубины схемы, количества кубитов, величин различных ошибок а также от совокупностей этих параметров. Однако же, кросс-энтропийный бенчмаркинг для схем с большим количеством кубитов или большой глубиной является крайне ресурсоёмкой задачей, что ограничивает возможности прямого исследования.

По той же причине остался открытым вопрос строгого описания динамики распространения влияния детерминированной ошибки в кубитном регистре. Это исследование требует объёмного и высокоточного моделирования, что невозможно без привлечения существенных вычислительных ресурсов.

Тем не менее, основной вывод данной работы оказывается крайне оптимистичным. Квантовое превосходство для архитектуры на основе сверхпроводников было показано на реальном 53-кубитном процессоре. Кубитные регистры с размерами менее 50 вполне поддаются контролю. Возможность демонстрации квантового превосходства уже на 40 кубитах в атомной архитектуре выглядит достаточно амбициозно и может открывать новые перспективы внедрения квантовых компьютеров, использующих данную архитектуру.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. М. Нильсен, И. Чанг. Квантовая информация и квантовые вычисления, перевод М. Н. Вялого и П. М. Островского, Москва «Мир» (2006)
2. P. Benioff, The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines J. Statist. Phys. 22, 563 — 591 (1980)
3. Manin Yu. I., Computable and Uncomputable, Sov.Radio. pp. 13 — 15, (1980)
4. R. Feynman, Simulating physics with computers, Internat. J. Theoret. Phys. 21, 467 — 488 (1982)
5. D. Deutsch, Quantum theory, the Church-Turing principle and the universal quantum computer, Proceedings of the Royal Society A. 400 (1818), 97 — 117 (1985)
6. S. Arora and B. Barak, Computational complexity: a modern approach, Cambridge University Press, 39 – 60, 115 – 119, 126 – 131 (2009)
7. Christos H. Papadimitriou, Computational Complexity, University of California – San Diego, Addison-Wesley, 139 –159, 181 – 219, 329 – 357 (1995)
8. D. Deutsch, R Jozsa, Rapid solution of problems by quantum computation, Proceedings of the Royal Society of London A, 439, (1907), 553-558 (1992)
9. D. R. Simon, On the power of quantum computation, SFCS '94: Proceedings of the 35th Annual Symposium on Foundations of Computer Science, 116 – 123 (1994)
10. P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Computing 26, 1484 – 1509 (1997)
11. W. Diffie, M. Hellman New directions in cryptography, IEEE Transactions on Information Theory, v. 22, Issue 6 (1976)
12. L. K. Grover, A fast quantum mechanical algorithm for database search, In Proceedings of the twenty-eighth, annual ACM symposium on Theory of computing, 212 – 219, ACM (1996)

13. S. Aaronson, Quantum computing and hidden variables, *Phys. Rev. A* 71, 032325 (2005)
14. S. Aaronson, A. Arkhipov, The Computational Complexity of Linear Optics, *Proceedings of the forty-third annual ACM symposium on Theory of computing*, 333 – 342 (2011)
15. F. Arute, K. Arya, John M. Martinis, Quantum supremacy using a programmable superconducting processor, *Nature*, 574, 505-510 (2019)
16. E. Pednault, J. Gunnels, On “Quantum Supremacy”, [ibm.com/blogs/research/2019/10/on-quantum-supremacy](https://ibm.com/blogs/research/2019/10/on-quantum-supremacy) (2019)
17. Y. Zhou, E. M. Stoudenmire, X. Waintal, What limits the simulation of quantum computers?, [arxiv.org/abs/2002.07730](https://arxiv.org/abs/2002.07730) (2020)
18. C. Huang, F. Zhang, M. Newman, Classical Simulation of Quantum Supremacy Circuits, [arxiv.org/abs/2005.06787](https://arxiv.org/abs/2005.06787) (2020)
19. A. Zlokapa, S. Boixo, D. Lidar, Boundaries of quantum supremacy via random circuit sampling, [arxiv.org/abs/2005.02464](https://arxiv.org/abs/2005.02464) (2020)
20. A. Bouland, B. Fefferman, Ch. Nirkhe, Quantum Supremacy and the Complexity of Random Circuit Sampling, [arxiv.org/abs/1803.04402](https://arxiv.org/abs/1803.04402) (2018)
21. S. Boixo, S.V. Isakov, John M. Martinis, Characterizing quantum supremacy in near-term devices, *Nature Physics*, 14, 595-600 (2018)
22. A. J. Scott, T. A. Brun, C. M. Caves, R. Schack, Hypersensitivity and chaos signatures in the quantum baker's maps, *J. Phys. A* 39, 13405–13433 (2006)
23. J. Emerson, S. Lloyd, David G Cory, Pseudo-random Unitary Operators for Quantum Information Processing, *Science* 302, 2098 (2003)
24. Y. S. Weinstein, L. Viola, Parameters of Pseudo-Random Quantum Circuits, *Phys. Rev. A* 78, 052332 (2008)
25. S. Aaronson, D. Gottesman, Improved Simulation of Stabilizer Circuits, *Phys. Rev. A* 70, 052328 (2004)
26. B. Barak, Chi-Ning Chou, X. Gao, Spoofing Linear Cross-Entropy Benchmarking in Shallow Quantum Circuits, [arxiv.org/abs/2005.02421](https://arxiv.org/abs/2005.02421) (2020)

27. A. Strikis, D. Qin, Y. Li, Learning-based quantum error mitigation, [arxiv.org/abs/2005.07601](https://arxiv.org/abs/2005.07601) (2020)
28. Y. Nakata, M. Muraio, Diagonal quantum circuits: their computational power and applications, [arxiv.org/abs/1405.6552](https://arxiv.org/abs/1405.6552) (2020)
29. T. M. Graham, M. Kwon, M. Saffman, Rydberg-Mediated Entanglement in a Two-Dimensional Neutral Atom Qubit Array, *Phys. Rev. Lett.* 123, 230501 (2019)
30. H. Levine, A. Keesling, M. D. Lukin, Parallel implementation of high-fidelity multi-qubit gates with neutral atoms, *Phys. Rev. Lett.* 123, 170503 (2019)
31. Angelo Bassi, Dirk-André Deckert, Noise gates for decoherent quantum circuits, *Phys. Rev. A* 77, 032323 (2008)
32. T. Jones, A. Brown, I. Bush, QuEST and High Performance Simulation of Quantum Computers. *Sci Rep* 9, 10736 (2019)
33. Lomonosov-2, T-Platform A-Class Cluster, TOP-500, [top500.org/system/178444](https://top500.org/system/178444)