

С.М. Гушанский, В.С. Потапов

**ОПТИМИЗАЦИЯ МОДЕЛЕЙ КВАНТОВЫХ ВЫЧИСЛИТЕЛЕЙ  
С ИСПОЛЬЗОВАНИЕМ НИЗКОУРОВНЕВЫХ КВАНТОВЫХ СХЕМ  
И ВАРИАТИВНОСТИ ЯДЕР И УЗЛОВ\***

*Определены и проанализированы основные понятия в области квантовой информации, среди которых квантовые схемы, логические гейты и кубиты. Также произведен вывод различных вариаций оптимизации, регулируемых количественными соотношениями ядер (одно ядро или несколько) и узлов (один узел или несколько), что может быть использовано для реализации симуляции мультиузловой квантовой схемы. Актуальность данных исследований заключается в нахождении всевозможных методов оптимизации моделей квантовых вычислительных устройств. Что станет еще одним шагом вперед в исследованиях элементарной теоретической базы квантового вычислительного устройства и, как следствие, практической, физической реализации этого устройства. Актуальность работ в данном направлении объясняется также большим числом как экспериментальных, так и теоретических работ, посвящённых данным вопросам. Научная новизна данного направления в первую очередь выражается в постоянном обновлении и дополнении поля квантовых исследований по ряду направлений, а компьютерная симуляция квантовых физических явлений и особенностей, достаточно слабо освещена в мире. Целью работы является компьютерное моделирование низкоуровневых случайных квантовых схем, что позволит проанализировать функционирование созданной ранее модели квантового вычислительного устройства, а также оптимизировать ее с использованием низкоуровневых квантовых схем и вариативности ядер и узлов.*

*Матрица плотности; кубит; оптимизация; квантовая схема; вектор; квантовое ядро.*

S.M. Gushanskiy, V.S. Potapov

**OPTIMIZATION OF QUANTUM COMPUTER MODELS USING LOW-LEVEL  
QUANTUM SCHEMES AND VARIABILITY OF CORES AND NODES**

*The article defines and analyzes the basic concepts in the field of quantum information, among which are quantum circuits, logic gates and qubits. Also, various optimizations, regulated by quantitative correlations of cores (one or several cores) and nodes (one node or several), are derived, which can be used to implement simulation of a multi-node quantum circuit. The relevance of these studies is to find all possible methods for optimizing the models of quantum computing devices. What will become one more step forward in the research of the elementary theoretical base of a quantum computing device and, as a consequence, the practical, physical realization of this device. The urgency of the work in this direction is also explained by the large number of both experimental and theoretical works devoted to these questions. The scientific novelty of this direction is primarily expressed in the constant updating and supplementation of the field of quantum research in a number of directions, and computer simulation of quantum physical phenomena and features is poorly illuminated in the world. The aim of the work is computer simulation of low-level random quantum circuits, which will allow to analyze the functioning of the previously created model of a quantum computing device, and also to optimize it using low-level quantum schemes and the variability of cores and nodes.*

*Density matrix; qubit; optimization; quantum circuit; vector; quantum core.*

\* Работа выполнена в рамках проектной части госзадания Минобрнауки России № 2.3928.2017/4.6 в Южном федеральном университете.

**Введение.** Квантовые вычислители способны решать задачи, которые не под силу классическим машинам. Такие не являются средством повышения производительности каких-либо приложений, тем не менее, можно выделить сферы, в которых они могли бы совершить революцию. Среди таких областей квантовая химия, материальные науки, машинное обучение и криптография.

Экспериментальные устройства, содержащие около 50 кубитов [1] в скором времени будут доступны и смогут выполнять четко определенные вычислительные задачи [2], которые требовали бы самых мощных классических суперкомпьютеров. Однако дабы не повторять ошибок прошлого в области вычислительных систем, необходимо уже на данном, подготовительном этапе определиться с оптимизационными, теоретическими и практическими вопросами, стоящими перед квантовыми вычислительными устройствами и их компонентами. А именно, квантовые схемы, оптимизация распределения вентилях и кубит, что сократит число связей в многоузловой настройке и число  $k$ -кубитных ядер с вентилями на уровне одного узла. Немаловажным достоинством также будет служить сокращение тактового цикла работы квантовой схемы.

**1. Низкоуровневые случайные квантовые схемы.** Квантовая схема – это последовательность физических преобразований из конечного набора базисных элементарных преобразований – гейтов. На вход квантовая схема получает квантовые биты. Результат ее работы вероятностный. Физически можно реализовать только линейные, сохраняющие постоянной сумму квадратов коэффициентов (унитарные) преобразования над малым количеством квантовых битов. Следовательно, любое преобразование однозначно задается значениями на базисных состояниях и преобразование над  $k$  квантовыми битами можно записать в виде матрицы  $2^k \times 2^k$ .

Хотя одна из вычислительных задач, предложенная для демонстрации вычислительных мощностей квантового компьютера – выполнение низкоуровневых случайных квантовых цепей – не является научно полезной сама по себе, схемы ее реализации по-прежнему очень полезны для калибровки, проверки и тестирования временных квантовых устройств. Низкоуровневая случайная квантовая схема, изображенная на рис. 1, демонстрирует квантовое превосходство.

Точки на рис. 1 представляют квантовые биты информации, применяемые в квантовой схеме. Ни одна квантовая схема не может функционировать без определенного набора квантовых гейтов, отображенными на рис. 1 квадратиками. Данная низкоуровневая случайная квантовая схема является частным случаем, т.е. нет принципиальной разницы в выборе конкретного такта работы квантовой схемы.

Созданы идентичные схемы с использованием следующих правил: при такте 0 к каждому кубиту применяется гейт Адамара. Как следствие, восемь разных шаблонов ( $6 \times 6$  кубитов) управляемых  $Z$  ( $CZ$  – controlled  $Z$ ) гейтов применяемых повторно до тех пор, пока не будет достигнута желаемая глубина схемы.  $CZ$ -гейты, являющиеся логическими двухкубитовыми гейтами и описываемыми матрицами Паули, представлены линией между двумя кубитами (в дальнейшем будем отождествлять эту линию с понятием узла). Физически узел представляет собой часть исполняемого кода в рамках симулятора квантового вычислителя. Линия преобразуется в узел путем добавления определенного гейта в схему. Выбор конкретного гейта ограничен лишь рамками применимого набора базисных гейтов. Этот шаблон гарантирует, что все возможные взаимодействия кубитов в этой двумерной архитектуре ближайшего соседа выполняются каждые 8 циклов. Одиночные кубитовые гейты выбираются случайным образом.

Поэтому, помимо проверки квантовых алгоритмов и проведения исследования их работы при наличии помех, модель квантовой схемы может предоставить средства для проведения калибровок и контрольных измерений, позволяющих получить наиболее производительную вычислительную структуру.

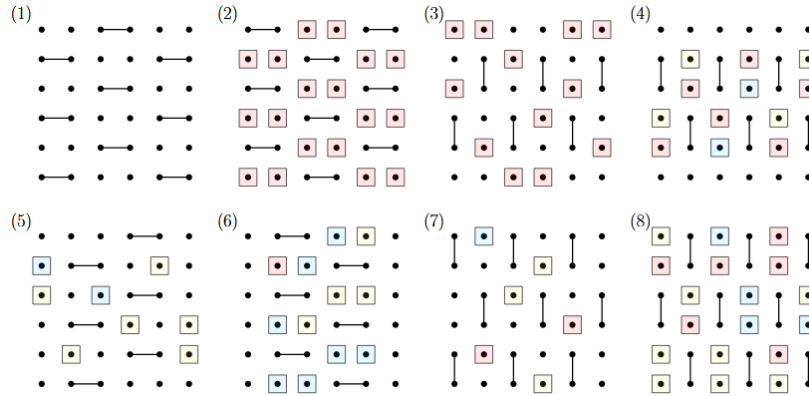


Рис. 1. Низкоуровневая случайная квантовая схема

Ограничения в низкоуровневых случайных квантовых схемах аналогичны классическим, в которых общее количество ветвей (линий) квантовой схемы и количество используемых гейтов ограничено масштабами самой схемы.

Квантовые симуляторы сравнимы с инструментами структурного моделирования, позволяющими реализовывать моделирование работы классического вычислителя. Разработанный ранее симулятор квантового вычислителя [3] был реализован и оптимизирован посредством многоуровневого подхода.

**2. Варианты оптимизаций.** Для моделирования больших систем необходимости в реальном хранении матрицы  $2^n \times 2^n$ , действующей на вектор состояния, нет. Можно использовать его регулярную структуру и применять методы, которые, учитывая его состояние, имитирует умножение на эту матрицу. Стандартная реализация имеет два вектора состояния [4] (входной и выходной).

В динамических системах реакция  $y(t)$  в момент времени  $t$  зависит не только от значения входа  $u(t)$  в момент  $t$ , но и от начального условия, т.е. начального состояния. Пусть динамическая система [5] имеет  $r$  входных переменных и  $m$  выходных, а также  $n$  внутренних переменных. Переменные  $x_i(t)$ ,  $i = \overline{1, n}$  – переменные состояния, если задать их значения  $x_i(t_0)$ ,  $i = \overline{1, n}$  в момент  $t_0$  и закон изменения входных переменных  $u_k(t)$ ,  $k = \overline{1, r}$  на интервале  $[t_0; t]$ , то можно однозначно определить значения  $x_i(t)$  для всех  $i$  в момент времени  $t$ . Переменные  $x_1(t), \dots, x_n(t)$  характеризуют начальное состояние системы. Определим вектора:

$$x(t) = \begin{bmatrix} x_1(t) \\ \dots \\ x_n(t) \end{bmatrix} = [x_1(t), \dots, x_n(t)]^T \text{ – вектор переменных состояния, вектор со-}$$

стояния.

$$u(t) = [u_1(t), \dots, u_r(t)]^T \text{ – вектор входных переменных.}$$

$$y(t) = [y_1(t), \dots, y_m(t)]^T \text{ – вектор выходных переменных.}$$

Чтобы определить одну запись выходного вектора, два комплексных произведения [6] и одно комплексное дополнение должны быть выполнены на двух записях входного вектора посредством применения общего однокубитового вентиля. В общем виде операция может быть представлена формулой

$$2 \cdot (4[\text{mul}] + 2[\text{add}]) + 2[\text{add}] = 14 \text{ FLOP}$$

Для одной сложной записи с двойной точностью требуется 16 байт памяти; входной вектор должен быть выгружен из памяти; выходной вектор - записан обратно в память. Таким образом, интенсивность работы меньше 1/2, это является примером того, что пропускная способность памяти ограничивает большинство систем.

**2.1. Одно ядро.** Чтобы уменьшить требования к памяти в 2 раза, может быть выполнено комплексное разреженное матрично-векторное умножение благодаря доступу к кэшу [7]. Чем больше кубит входит в состав вентиля, больше операций необходимо реализовать, что позволяет использовать аппаратные вычислительные мощности эффективнее. Действительно, число операций растет экспоненциально [8] с ростом числа кубит вентиля, т.к. использование  $k$ -кубитного вентиля позволяет выполнить одно скалярное произведение [9] размерности  $2^k$ .

Чтобы воздействовать  $k$ -кубитным вентиляем (размерность  $2^k \times 2^k$ ) на вектор состояния размера  $2^n$ , где  $n$  – число кубит, значения, соответствующие каждому из  $2^k$  индексов матрицы [10] вентиля, должны быть представлены в виде временного вектора размерности  $2^k$ , который, прежде чем быть загруженным в вектор состояния, умножается на матрицу вентиля. Индексы записей вектора состояния, представленные в двоичном коде, являются битовыми последовательностями вида

$$C_{n-k-1} \cdot X_{i_{k-1}} \dots C_j \dots X_{i_1} \dots C_0,$$

где  $i_0, i_1, i_{k-1}$  – индексы  $k$  кубитов, которым вентиль был применен. Извлечение и комбинирование битов  $X_{ij}$  индекса записи:  $x = X_{i_{k-1}} \dots X_{i_1} X_{i_0}$  дает индекс записи относительно временного вектора. Все  $2^k$  записи, имеющие одинаковую подстроку индекса  $C = C_{n-k-1} C_{n-k-2} \dots C_0$ , являются частью произведения матрицы на вектор. Как только все записи, умноженные на матрицу, получены и сохранены в виде вектора-состояния, можно переходить к следующей подстроке индекса  $c'=c+1$ . В общем, реализуются  $2^{n-k}$  комплексных матрично-векторных умножений размерности  $2^k$ .

Таким образом, одна матрица используется  $2^{n-k}$  раз, и элементы матрицы можно представлять заранее, чтобы индексы кубитов и, как следствие, доступ к памяти, всегда были упорядочены. Матрично-векторное произведение

$$U'_l = \sum_{i=0}^{2^k-1} m_{l,i} * U_i$$

может потребовать все записи временного вектора  $U$  (их загрузку из памяти). В таком случае производится блокировка вычисления и определение размера блока с применением автоматической кодогенерации [11] или сравнительной оценки [12]. Для индекса каждого из блоков:  $b = 0, 1, \dots, \frac{2^k}{B} - 1$  все индексы  $l$  временного выходного вектора, перед тем как перейти к следующему блоку, преобразуются в

$$\text{соответствии с } U'_l + = \sum_{j < B} m_{l,i(b,j)} * U_{i(b,j)}, \text{ где } i(b,j) = b * B + j.$$

Для параллелизации обновлений как средства упорядочения значений 1 используется явная векторизация. И, т.к. имеют место быть комплексные значения двойной точности, векторизация [13], теоретически, позволяет ускорить выполнение в 2 или даже в 4 раза.  $a_R$  и  $a_I$  – действительная и мнимая части  $a$ . Умножение одной комплексной записи  $v_I = (v_R, v_I)$  временного вектора  $v$  на одну комплексную запись матрицы вентиля  $m = (m_R, m_I)$  и сложение результата с временным выходным вектором  $v'$  может быть записано как  $(v'_R, v'_I) = (v_R m_R - v_I m_I, v_I m_R + v_R m_I)$

Тем не менее, реализация этого обновления приводит к истощению вычислительных ресурсов из-за искусственных зависимостей и дополнительных перестановок. Однако, возможна следующая реорганизация, способствующая повышению максимально достижимой производительности.

$$(v'_R, v'_I) = (v_R m_R, v_I m_R), (v'_R, v'_I) = (v_I \cdot -1 \cdot m_I, v_R m_I)$$

Стоит отметить, что  $v_I$  можно переставить один раз при загрузке (и затем в регистре), поскольку он повторно используется для  $2^k$  комплексных произведений. Так как матрица  $m$  используется в  $2^{n-k}$  матрично-векторных произведениях, предварительное вычисление для построения этих двух матриц, состоящих из  $(m_R, m_R)$  and  $(-1 \cdot m_I, m_I)$ , достаточно свободны.

**2.2. Один узел.** Оптимизация, описанная выше, не умаляет тот факт, что оперативная эффективность применения однокубитового вентиля [14] очень низкая, что затрудняет полное использование мощности мульти- и многоядерных процессоров [15]. Однако, как упоминалось ранее, чем больше кубитов содержит вентиль, тем больше операций требуется, и пока их применение остается связанным с памятью, многкубитовые вентиля (например, Тоффоли [16]) и однокубитовые вентиля (Not [17], Hadamard [18]) отрабатывают почти за равное время. Выигрыш, помимо увеличения интенсивности работы, заключается в том, что большие вентиля могут быть использованы для выполнения целой последовательности одно- и двухбитных вентиляей. В частности, множество вентиляей, действующих на  $k$  различных кубитов могут быть объединены в один многкубитовый  $k$ -кубитный вентиль.

Величина  $k$  выбирается в зависимости от максимальной производительности, пропускной способности памяти, размера кэша, и схемы. Параметры кэша особенно важны при применении вентиляей к кубитам с большим индексом. При низкоассоциативном кэшировании, это приводит к возникновению конфликтов, если ядро небольшого размера. Поскольку  $2^k$  значений должны быть загружены из вектора состояний (минимальный размер которого  $2^m$ ,  $m$  – наименьший индекс кубита) для каждого из  $2^{n-k}$  матрично-векторного произведения, строки  $2^k$ -направленного кэша должны быть соотнесены с соответствующим расположением (адресом), не имеет значения насколько велико  $m$ . Это позволяет напрямую обращаться к данным в кэше при последующих операциях матрично-векторного умножения.

**2.3. Множество узлов.** Для симуляции свойств более чем 30-ти кубитного квантового вычислителя [19] необходимо множество узлов, чтобы вектор состояния было возможно хранить в памяти. Для обеспечения связи  $2^g$  узлов каждый из которых имеет собственный вектор состояния размером  $2^l$  ( $g, l$  – число общих и локальных кубитов соответственно). Применение вентиляей к локальным кубитам, т.е. кубитам с индексом  $i < l$  не требует связи. Кубитам с индексом  $i \geq l$  – наоборот, связь необходима.

Существуют две базовые схемы, которые могут быть использованы для реализации симуляции мультиузловой квантовой схемы [20]. Первая из них содержит общие кубиты (характерный пример – соседние кубиты квантовой схемы на рис. 1) и использует общие вентили (общность вентиляей вытекает из общности кубитов) посредством использования двух парных обменов половины вектора состояния. Вторая схема меняет местами общие и локальные кубиты, применяя вентили к локальным кубитам в обычном порядке и, если необходимо, снова меняет их местами с общими кубитами. Обмен с глобальным кубитом и последующее его немедленное возвращение обратно требует такой же связи, как и первая схема. Рассмотрим схему с обменом подробнее.

Пример с 1 кубитом (рис. 2,а). В случае с двумя разрядами, обмен кубита наивысшего порядка (старший бит локального индекса) с общим кубитом (первый бит номера разряда) может быть реализован следующим образом: первый блок 0 разряда остается неизменным, т.к. замена 0 на 0 безрезультатна. Обмен 0 (общего) на 1 (локальный) для второго блока требует перевода всего блока в 1 разряд, где эти коэффициенты связаны с локальным кубитом, равным 0.

Пример с 2 кубитами (рис. 2,б). Чтобы поменять два общих кубита с двумя локальными кубитами высшего порядка в случае с 4 разрядами, i-я четверть вектор состояния каждого разряда переводится в i-й разряд.

Разряд	Базисные состояния			
0	0...			1...
1	0...			1...

Разряд	Базисные состояния			
00	00...	01...	10...	11...
01	00...	01...	10...	11...
10	00...	01...	10...	11...
11	00...	01...	10...	11...

Рис. 2. Обмен пары кубит (а) и пар кубит (б)

Под разрядами на рис. 2 подразумеваются соответствующие разряды памяти для хранения вектора состояния для симуляции [21] свойств n-кубитного квантового вычислителя с множеством узлов.

Для разрешения произвольного обмена локальных кубитов, в первую очередь, используются оптимизированные ядра. С их помощью производится обмен между кубитами с наибольшим индексом и теми, с которыми обмен должен быть осуществлен. Затем производится групповой локальный обмен по принципу «все ко всем» и, если необходимо, прочие локальные обмены (с кубитами меньших индексов) в целях повышения локальности данных в ядрах, состоящих из k-кубитных вентиляей.

**3. Оптимизации схемы: распределение вентиляей и кубит.** В дополнение к реализации оптимизации, необходима оптимизация схемы, направленная на уменьшение «шагов» связи и использование точных и более эффективно настроенных ядер. Существуют различные варианты оптимизации распределения вентиляей и кубит на основе схем «квантового превосходства». Оптимизации носят общий характер и могут применяться в любой квантовой цепи. Фактически, схемы «квантового превосходства» могут быть спроектированы таким образом, который является наименее подходящим для данных видов оптимизации производительности.

Построение таких случайных, низкоуровневых квантовых схем показано на рис. 1. Схемы созданы для использования в составе квантовой вычислительной архитектуры с двунаправленной связью. Все возможные двухкубитные вентили применяются в пределах 8 циклов, что делает систему сильно запутанной. Симулятор может пропустить начальный вентиль Адамара в 0 цикле инициализировать волновую функцию [22] непосредственно  $(2^{-\frac{n}{2}}, \dots)^T$ , вместо начала с состояния  $|0\dots 0\rangle = (1, 0, \dots, 0)^T$ . Более того, симуляция последних CZ-вентилей [23] не производится, т.к. они только изменяют фазы амплитуд вероятности [24]  $\alpha_i$ , а не вероятности  $p_i = |\alpha_i|^2$ , являющиеся объектом интереса.

**3.1. Распределение вентиляей.** Самая важная оптимизация квантовой схем – распределение вентиляей, т.к. она резко сокращает число связей в многоузловой настройке и число к-кубитных ядер с вентилями на уровне одного узла. Оптимизация может быть разделена на 3 этапа:

**Минимизация количество «шагов» связи.** На первом шаге распределение вентиляей минимизирует число обще-локальных обменов, являющееся наиболее важным параметром многоузловой настройки. Выполнение каждого тактового цикла схемы само по себе требует по меньшей мере одного «шага» связи на каждом цикле, характеризованного общим недиагональным вентилем. Однако, в соответствии с многоузловым принципом, этап оказывается полезным не для работы общих вентиляей, а скорее для обмена общих и локальных кубит и последующей локальной отработки вентиляей. Для того чтобы схема была наиболее полезной алгоритм распределения вентиляей реорганизует (если возможно) работу вентиляей. Они обрабатываются поэтапно, причем каждая группа состоит из последовательности квантовых вентиляей, действующих только на локальных кубитах (рис. 3, 4).

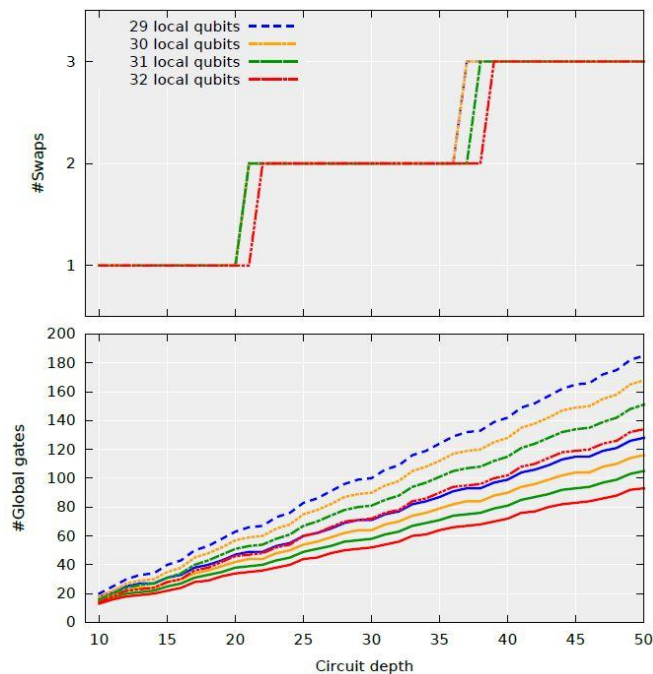


Рис. 3. Масштабирование требуемой связи для глубин квантовой схемы

После завершения этапа, некоторые локальные кубиты заменяются общими, и начинается общий этап. Эта схема значительно сокращает количество «шагов» связи.

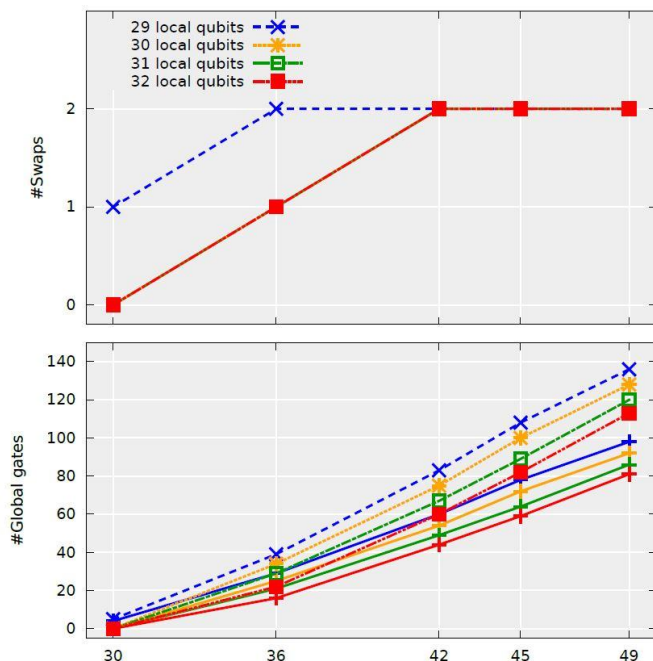


Рис. 4. Масштабирование требуемой связи для увеличения числа кубитов для схем с фиксированной глубиной 25

Для 42-кубитовой квантовой схемы глубины 25 необходимо только два обще-локальных обмена (рис. 4). Важной особенностью алгоритма распределения вентилей является то, число обще-локальных обменов меньше всего зависит от числа локальных кубит (29, 30, 31 или 32).

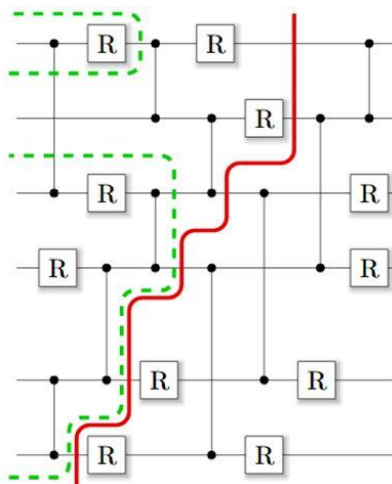


Рис. 5. Пример распределения вентилей



**Минимизация числа  $k$ -кубитных вентиляей.** На втором этапе, все вентили распределяются так, чтобы можно было объединить в цепочки последовательные 1- или 2-кубитные вентили в  $k$ -кубитные и использовать  $k$ -кубитный вентиль вместо множества одно- или двухкубитных. Необходимо увеличить число  $k$  кубит, приходящихся на один кластер, выполнив при этом условие  $k \leq k_{\max}$ , где наибольшее  $k_{\max}$  значение  $k$ , для которого производительность ядра  $k$ -кубитного вентиля будет показывать хорошую производительность в конечной системе.

**Локальные корректировки общих обменов.** Последний кластер на каждом этапе содержит меньшее число одно- и двухкубитных вентиляей.

Чтобы увеличить среднее число вентиляей в каждом кластере, уменьшив тем самым общее число кластеров в схеме, нужно удалить последний кластер на каждом этапе посредством обще-локальных обменов раньше, и, если это возможно, без увеличения их общего числа.

**3.2. Распределение кубит.** Позиция бита в каждом кубите оптимизирована в соответствии с числом кластеров [25], чему сопутствует снижение производительности ввиду ассоциативности кэша последнего уровня. Поскольку это снижение производительности происходит только в том случае, когда вентили применяются к позициям бит высокого порядка, может быть применено переназначение (перераспределение).

Следующая эвристика [26] позволяет уменьшить время решения в два раза: связать кубит и битовое местоположения 0 так, чтобы количество кластеров, имеющего  $k$  битовому местоположению 0 было максимальным. Далее игнорировать все кластеры, которые действуют на этот кубит. Произвести те же действия для положения бит 1, 2 и 3.

**Заключение.** Оптимизация и реализация квантовых вычислений в области моделей квантовых вычислителей, основанные на описанном материале, позволяют:

- ◆ Производить калибровку, проверку и тестирование квантовых вычислительных устройств.
- ◆ Уменьшить требования к памяти в  $n$  раз и тем самым ускорить вычислительный процесс при помощи выполнения комплексного разреженного матрично-векторного умножения благодаря доступу к кэшу.
- ◆ Осуществить наиболее рациональный выбор одной из существующих базовых схем, которые могут быть использованы для реализации симуляции мультиузловой квантовой схемы.
- ◆ Находить новые способы применения квантовых вычислений для моделирования каких-либо параметров в исполняемой задаче.
- ◆ В процессе написания данной работы были определены и проанализированы основные понятия в области квантовой информации такие, как квантовые схемы, логические гейты и кубиты. Это послужило основой для разработки ряда понятий (линия, узел, взаимодействие кубитов и их соседство) теоретической и практической составляющих низкоуровневых случайных квантовых схем. Также произведен вывод различных вариаций оптимизации, регулируемых количественными соотношениями ядер (одно ядро или несколько) и узлов (один узел или несколько), что может быть использовано для реализации симуляции мультиузловой квантовой схемы.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Denis D. Sukachev, Alp. Sipahigil, Mikhail D. Lukin. Silicon-Vacancy Spin Qubit in Diamond: A Quantum Memory Exceeding 10 ms with Single-Shot State Readout // Physical review letters. – 2017. – Vol. 199.
2. Mikhail D. Lukin. Probing many-body dynamics on a 51-atom quantum simulator // Nature. – November 2013. – Vol. 551.

3. *Guzik V., Gushanskiy S., Polenov M., Potapov V.* Architecture and Software Implementation of a Quantum Computer Model // Artificial Intelligence Perspectives in Intelligent Systems. Proceedings of the 5<sup>th</sup> Computer Science On-line Conference 2016 (CSOC). – Springer International Publishing AG Switzerland, 2016. – P. 59-68.
4. Квантовое состояние // Википедия. – URL: <http://ru.wikipedia.org/?oldid=90522592> (дата обращения: 26.01.2018).
5. *Koen De Raedt, Kristel Michielsen, Hans De Raedt, Binh Trieu, Guido Arnold, Marcus Richter, Th. Lippert, H. Watanabe, and N. Ito.* Massively parallel quantum computer simulator // Computer Physics Communications. – 2007. – No. 2 (176), – P. 121-136.
6. *Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, John M. Martinis, and Hartmut Neven.* Characterizing quantum supremacy in near-term devices // arXiv preprint arXiv:1608.00263. – 2016.
7. *Stierhoff G.C., Davis A.G.* A History of the IBM Systems Journal // IEEE Annals of the History of Computing. – 1998. – Vol. 20, No. 1. – P. 29-35. – DOI:10.1109/85.646206.
8. Экспоненциальный рост // Википедия. – URL: <https://ru.wikipedia.org/?oldid=79197692> (дата обращения: 26.06.2016).
9. *Lipschutz S., Lipson M.* Linear Algebra (Schaum's Outlines) // 4th McGraw Hill. – 2009. – ISBN 978-0-07-154352-1.
10. *Халмош П.* Конечномерные векторные пространства = Finite-dimensional vector spaces. – М.: Физматгиз, 1963. – 264 с.
11. *Альфред В. Ахо, Моника С. Лам, Рави Сети, Джеффри Д. Ульман.* Компиляторы: принципы, технологии и инструментарий = Compilers: Principles, Techniques, and Tools. – 2-е изд. – М.: Вильямс, 2008. – ISBN 978-5-8459-1349-4.
12. *Collier David.* The Comparative Method. In Ada W. Finifter, ed. Political Sciences: The State of the Discipline // Washington, DC: American Science Association. – 1993. – P. 105-119.
13. Vectorization // Wikipedia, The Free Encyclopedia. Retrieved 13:38, June 16, 2018, from <https://en.wikipedia.org/w/index.php?title=Vectorization&oldid=829988201>.
14. *C.P. Williams.* Explorations in Quantum Computing // Texts in Computer Science. – Springer-Verlag, 2011. – P. 51-122. – ISBN 978-1-84628-887-6, doi:10.1007/978-1-84628-887-6\_2.
15. *Kunle Olukotun.* Chip Multiprocessor Architecture // Techniques to Improve Throughput and Latency. – Morgan and Claypool Publishers, 2007. – 154 p. – ISBN 159829122X.
16. *Tzvetan Metodi, Arvin I. Faruque, Frederic T. Chong.* Quantum Computing for Computer Architects. – 2nd Ed. – Morgan & Claypool Publishers, 2011. – 15 p. – ISBN 9781608456192.
17. Inverter (logic gate) // Wikipedia, The Free Encyclopedia. Retrieved 13:41, June 16, 2018, from [https://en.wikipedia.org/w/index.php?title=Inverter\\_\(logic\\_gate\)&oldid=844691629](https://en.wikipedia.org/w/index.php?title=Inverter_(logic_gate)&oldid=844691629).
18. *Lachowicz Dr. Pawel.* Walsh–Hadamard Transform and Tests for Randomness of Financial Return-Series (April 7, 2015). – <http://www.quantatrisk.com/2015/04/07/walsh-hadamard-transform-python-tests-for-randomness-of-financial-return-series/>.
19. *Guzik V., Gushanskiy S., Polenov M., Potapov V.* The Computational Structure of the Quantum Computer Simulator and Its Performance Evaluation // CSOC2018 2018: Software Engineering and Algorithms in Intelligent Systems. – Springer International Publishing AG Switzerland, 2018. – P. 198-207.
20. *Потапов В.С., Гушанский С.М.* Разработка методики моделирования запутанных квантовых вычислений, применимых в квантовых алгоритмах // Информатизация и связь. – 2018. – № 2. – С. 38-42.
21. *Sayirci E.* Modeling and simulation as a cloud service: a survey // Simulation Conference (WSC). – 2013. – P. 389-400.
22. *Zwiebach Barton.* A First Course in String Theory. – Cambridge University Press, 2009. – ISBN 978-0-521-88032-9.
23. *Nielsen M., Chuang I.* Quantum Computation and Quantum Information. – Cambridge University Press, 2000.
24. What is the Amplitude of a Wave? // Regents of the University of California. Universe of Light. – 1996. – Retrieved 2008-08-22.
25. Data cluster // Wikipedia, The Free Encyclopedia. Retrieved 13:52, June 16, 2018, from [https://en.wikipedia.org/w/index.php?title=Data\\_cluster&oldid=784026338](https://en.wikipedia.org/w/index.php?title=Data_cluster&oldid=784026338).
26. *Латыпов Н.Н., Ёлкин С.В., Гаврилов Д.А.* Инженерная эвристика / под ред. А.А. Вассермана. – М.: Астрель, 2012. – 320 с.

## REFERENCES

1. Denis D. Sukachev, Alp. Sipahigil, Mikhail D. Lukin. Silicon-Vacancy Spin Qubit in Diamond: A Quantum Memory Exceeding 10 ms with Single-Shot State Readout, *Physical review letters*, 2017, Vol. 199.
2. Mikhail D. Lukin. Probing many-body dynamics on a 51-atom quantum simulator, *Nature*, November 2013, Vol. 551.
3. Guzik V., Gushanskiy S., Polenov M., Potapov V. Architecture and Software Implementation of a Quantum Computer Model, *Artificial Intelligence Perspectives in Intelligent Systems. Proceedings of the 5<sup>th</sup> Computer Science On-line Conference 2016 (CSOC)*. Springer International Publishing AG Switzerland, 2016, pp. 59-68.
4. Kvantovoe sostoyanie [Quantum state], *Vikipediya* [Wikipedia]. Available at: <http://ru.wikipedia.org/?oldid=90522592> (accessed 26 January 2018).
5. Koen De Raedt, Kristel Michielsen, Hans De Raedt, Binh Trieu, Guido Arnold, Marcus Richter, Th. Lippert, H. Watanabe, and N. Ito. Massively parallel quantum computer simulator, *Computer Physics Communications*, 2007, No. 2 (176), pp. 121-136.
6. Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, John M. Martinis, and Hartmut Neven. Characterizing quantum supremacy in near-term devices, *arXiv preprint arXiv:1608.00263*, 2016.
7. Stierhoff G.C., Davis A.G. A History of the IBM Systems Journal, *IEEE Annals of the History of Computing*, 1998, Vol. 20, No. 1, pp. 29-35. DOI:10.1109/85.646206.
8. Eksponentsial'nyy rost [Exponential growth], *Vikipediya* [Wikipedia]. Available at: <https://ru.wikipedia.org/?oldid=79197692> (accessed 26 June 2016).
9. Lipschutz S., Lipson M. Linear Algebra (Schaum's Outlines), *4th McGraw Hill*, 2009. ISBN 978-0-07-154352-1.
10. Halmosh P. Konechnomernyye vektornyye prostranstva = Finite-dimensional vector spaces [Finite-dimensional vector spaces = Final-dimensional vector spaces]. Moscow: Fizmatgiz, 1963, 264 p.
11. Al'fred V. Aho, Monika S. Lam, Ravi Seti, Dzheffri D. Ul'man. Kompilyatory: printsipy, tekhnologii i instrumentariy = Compilers: Principles, Techniques, and Tools [Compilers: principles, technologies and tools = Compilers: Principles, Techniques, and Tools]. 2 ed. Moscow: Vil'yams, 2008. ISBN 978-5-8459-1349-4.
12. Collier David. The Comparative Method. In Ada W. Finifter, ed. *Political Sciences: The State of the Discipline*, Washington, DC: American Science Association, 1993, pp. 105-119.
13. Vectorization // Wikipedia, The Free Encyclopedia. Retrieved 13:38, June 16, 2018, from <https://en.wikipedia.org/w/index.php?title=Vectorization&oldid=829988201>.
14. C.P. Williams. Explorations in Quantum Computing, *Texts in Computer Science*. Springer-Verlag, 2011, pp. 51-122. ISBN 978-1-84628-887-6, doi:10.1007/978-1-84628-887-6\_2.
15. Kunle Olukotun. Chip Multiprocessor Architecture, *Techniques to Improve Throughput and Latency*. Morgan and Claypool Publishers, 2007, 154 p. ISBN 159829122X.
16. Tzvetan Metodi, Arvin I. Faruque, Frederic T. Chong. Quantum Computing for Computer Architects. 2nd Ed. Morgan & Claypool Publishers, 2011. 15 p. ISBN 9781608456192.
17. Inverter (logic gate), *Wikipedia, The Free Encyclopedia*. Retrieved 13:41, June 16, 2018, from [https://en.wikipedia.org/w/index.php?title=Inverter\\_\(logic\\_gate\)&oldid=844691629](https://en.wikipedia.org/w/index.php?title=Inverter_(logic_gate)&oldid=844691629).
18. Lachowicz Dr. Pawel. Walsh-Hadamard Transform and Tests for Randomness of Financial Return-Series (April 7, 2015). Available at: <http://www.quantatrisk.com/2015/04/07/walsh-hadamard-transform-python-tests-for-randomness-of-financial-return-series/>.
19. Guzik V., Gushanskiy S., Polenov M., Potapov V. The Computational Structure of the Quantum Computer Simulator and Its Performance Evaluation, *CSOC2018 2018: Software Engineering and Algorithms in Intelligent Systems*. Springer International Publishing AG Switzerland, 2018, pp. 198-207.
20. Potapov V.S., Gushanskiy S.M. Razrabotka metodiki modelirovaniya zaputannykh kvantovykh vychisleniy, primenimyykh v kvantovykh algoritmah [Development of methods for modeling of entangled quantum computing, applicable in quantum algorithms], *Informatizatsiya i svyaz'* [Informatization and communication], 2018, No. 2, pp. 38-42.
21. Cayirci E. Modeling and simulation as a cloud service: a survey, *Simulation Conference (WSC)*, 2013, pp. 389-400.

22. *Zwiebach Barton*. A First Course in String Theory. Cambridge University Press, 2009. ISBN 978-0-521-88032-9.
23. *Nielsen M., Chuang I.* Quantum Computation and Quantum Information. Cambridge University Press, 2000.
24. What is the Amplitude of a Wave?, *Regents of the University of California. Universe of Light*, 1996. Retrieved 2008-08-22.
25. Data cluster. Wikipedia, The Free Encyclopedia. Retrieved 13:52, June 16, 2018, from [https://en.wikipedia.org/w/index.php?title=Data\\_cluster&oldid=784026338](https://en.wikipedia.org/w/index.php?title=Data_cluster&oldid=784026338).
26. *Latypov N.N., Elkin S.V., Gavrilov D.A.* Inzhenernaya evristika [Engineering heuristics], ed. by A.A. Vassermana. Moscow: Astrel', 2012, 320 p.

Статью рекомендовал к опубликованию д.т.н., профессор Г.В. Куповых.

**Гушанский Сергей Михайлович** – Южный федеральный университет»; e-mail: smgushanskiy@sfedu.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371656; кафедра вычислительной техники; к.т.н.; доцент.

**Потапов Виктор Сергеевич** – e-mail: vitya-potapov@rambler.ru; кафедра вычислительной техники; аспирант.

**Gushanskiy Sergey Mikhailovich** – Southern Federal University; e-mail: smgushanskiy@sfedu.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371656; the department of computer engineering; cand. of eng. sc.; associate professor.

**Potapov Victor Sergeevich** – e-mail: vitya-potapov@rambler.ru; the department of computer engineering; postgraduate.