УДК 530.145.3

05.00.00 Технические науки

РАЗРАБОТКА ВЫЧИСЛИТЕЛЬНОЙ СТРУКТУРЫ СИМУЛЯТОРА КВАНТОВОГО ВЫЧИСЛИТЕЛЯ И ОПРЕДЕЛЕНИЕ ЕГО ПРОИЗВОДИТЕЛЬНОСТИ

Потапов Виктор Сергеевич
аспирант 3 года обучения
SPIN: 2121-3430
e-mail: vitya-potapov@rambler.ru

Гушанский Сергей Михайлович
к.т.н., доцент
SPIN: 1556-5238, Scopus ID: 56225520500
e-mail: kron-93@bk.ru
Южный федеральный университет,
Таганрог, Россия

Данная работа описывает основы производительности, а также структурной и функциональной составляющей разработки и реализации СКВ. В соответствии с этим выведена вычислительная структура симулятора квантового вычислителя, учитывающая все имеющиеся особенности построения симулятора квантового вычислительного устройства. А также выполнена программная реализация выведенной универсальной вычислительной структуры СКВ, удовлетворяющей и работающей по принципам этой схемы

Ключевые слова: КВАНТОВЫЙ РЕГИСТР, СИМУЛЯТОР КВАНТОВОГО ВЫЧИСЛИТЕЛЯ, КОМПЛЕКСНАЯ ПЛОСКОСТЬ, КУБИТ

Doi: 10.21515/1990-4665-134-078

UDC 530.145.3

Engineering

DEVELOPMENT OF COMPUTATIONAL STRUCTURE OF THE SIMULATOR OF A QUANTUM CALCULATOR AND DETERMINING ITS PERFORMANCE

Potapov Viktor Sergeevich
3 year post-graduate training
SPIN: 2121-3430
e-mail: vitya-potapov@rambler.ru

Gushanskiy Sergei Mikhailovich
Cand.Tech.Sci., associate professor
SPIN: 1556-5238, Scopus ID: 56225520500
e-mail: kron-93@bk.ru
South Federal University,
Taganrog, Russia

This article describes the basics of performance, as well as the structural and functional component of the development and implementation of SQC. In accordance with this, a computational structure of the simulator of a quantum calculator has been derived, taking into account all the available features of constructing a simulator of a quantum computing device. Also, a software implementation of the derived universal computational structure of SQC that satisfies and operates according to the principles of this scheme is implemented

Keywords: QUANTUM REGISTER, SIMULATOR OF QUANTUM CALCULATOR, COMPLEX PLANE, QUBIT

Введение

При создании СКВ разработчики преследуют совершенно разные цели (моделирование квантовых систем и кубит, воздействия отдельных гейтов или моделирование квантовых алгоритмов) и различные подходы в реализации интерфейса (графический, консольный), видах моделирования.

На данный момент существует большое количество различных сред моделирования (как консольных, так и графических), библиотек API, а также моделей отдельных алгоритмов. Некоторые модели строятся на базе существующих математических сред моделирования, например модель

алгоритма Гровера, разработанная в Санкт-Петербургском Государственном Университете. Различаются модели и по подходам к моделированию: QuIDDPro [1] имеет математическое ядро построено на графовом подходе, который в отличии от обычного матричного подхода позволяет значительно экономить память при моделировании. Отдельного упоминания стоят библиотеки APIlibquantum [2], Cove [3] для построения программ моделирования квантовых вычислений, которые предоставляют готовый функционал для построения собственной модели.

1. Вычислительная структура симулятора квантового вычислителя

Вычислительная структура симулятора квантового вычислительного устройства [4], показанная на рис. 1, позволяется разработать симулятор квантового вычислителя, соблюдая требования, отображенные в схеме. Данная структурная и функциональная иллюстрация вычислительной модели симулятора квантового вычислителя является универсальной для его построения. $k_{1,1}$ обозначает первый кубит регистра (КР) QR1; $k_{1,2}$ – второй КР QR1; $k_{1,m}$ – текущий кубит КР QR1; $k_{1,\hat{n}}$ – последний кубит.

Вычислительная структура СКВ условно содержит два КР QR1 и QR2. Единственный реальный регистр можно условно разделить и на большее число вспомогательных частей. Такое разделение может быть сделано в целях, например, удаления квантового мусора при подготовке к процессу измерения. Каждый квантовый разряд $k_{1,m}$ регистра QR1 и каждый квантовый разряд $k_{2,m}$ регистра QR2 является кубитом. Размерность каждого кубита одинакова: $\dim \mathrm{H}_m = \hat{\nu}$.

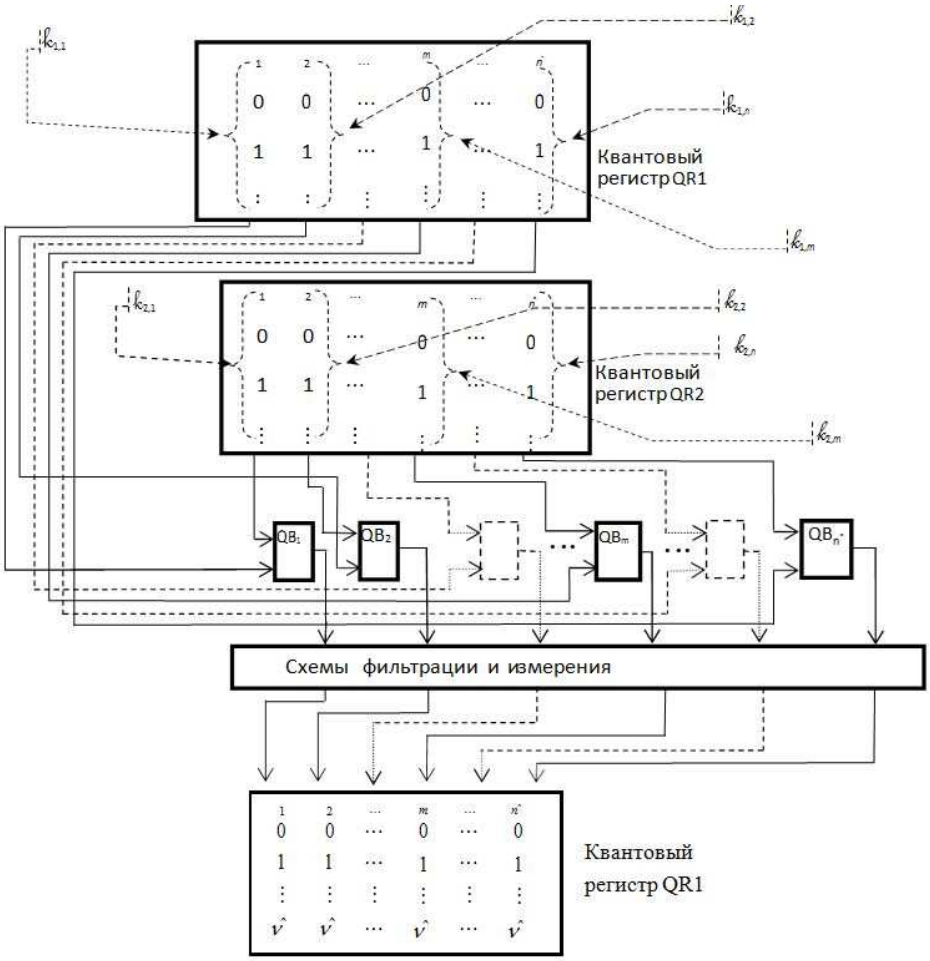

Рисунок 1. Схема вычислительной структуры СКВ

Это означает, что в одном кубите может быть записано и хранится $\hat{v}$ различных чисел. Число кубитов k_m в регистре QR1 и QR2 одинаково и равно $\hat{n}$. Таким образом, регистры QR1 и QR2 могут быть частями одного реально существующего КР QR. Например, все нечетные квантовые разряды относятся к регистру QR1, а все четные кубиты – к КР QR2. Каждый кубит k_m регистра QR1 соединен с квантовым блоком пересечений – блоком QB_m (т.е. является входом блока QB_m). То же относится и кубитам регистра QR2. Выход каждого блока QB_m передается на схемы фильтрации и измерения. Сам блок QB_m реализует интегральную операцию пересечения в форме блочно-диагональной матрицы Рота [5].

2. Программная симуляция квантовых алгоритмов и процессов

В соответствии с вышеописанной и разработанной схемой вычислительной модели СКВ, была построена программная симуляция. Данная модель может принимать унитарную функцию f и эффективно определять ее след (вектор направления) в соответствии с использованием векторной, матричной алгебры. Этот след функции f на n кубитов можно найти вдоль одной из осей X или Y.

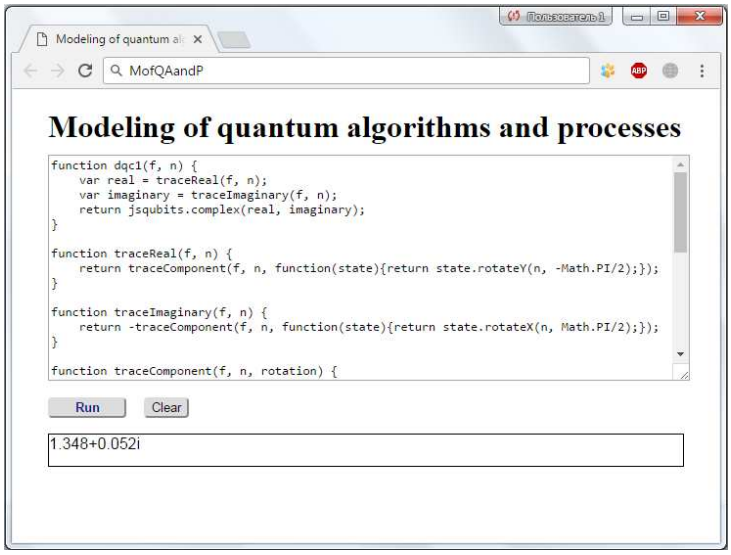

Рисунок 2. Главная форма/форма вывода результата работы модели

Результат работы модели представляется в виде раскрашенного набора пикселей. На рисунке 3 разобрана карта цветов в комплексной плоскости. Оттенок однозначно зависит от фазы ψ. Например, положительное число отображается красным пикселем, отрицательное – светло-голубым. Регистр кубит размера n определяется как 2 в степени n базисных состояний. Каждое базисное состояние представлено как единый квадрат (пиксель) амплитуда Z которого определяется его цвет в соответствии с цветовой картой. Если кубит |j> имеет реальную амплитуду a ϵ (0, 1], то j-ый квадрат в регистре красного цвета. Все кубиты, которые имеют амплитуды Z = 0 окрашены в чёрный цвет.

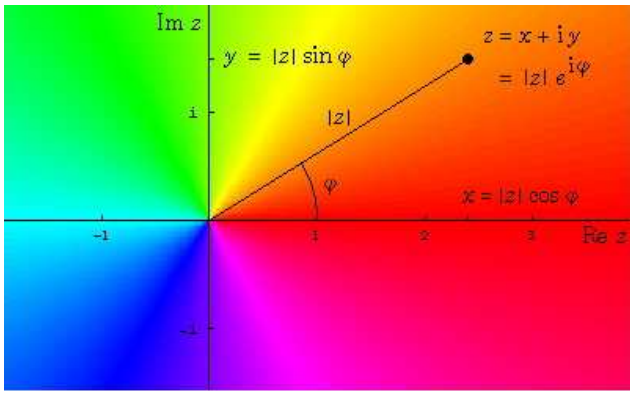

Рисунок 3. Карта цветов квантового регистра в комплексной плоскости

Данная модель может принимать унитарную функцию f и эффективно определять ее след (вектор направления) в соответствии с использованием векторной, матричной алгебры. Этот след функции f на n кубитов можно найти вдоль одной из осей X или Y.

3. Определение производительности СКВ

Что влияет на производительность симуляторов квантовых вычислительных устройств?

Квантовая схема, ее размер и глубина. Рассуждая о данном важнейшем компоненте симулятора, возможно выделить: число элементов N_{gates} (квантовый логические гейты) и ветвей $N_{branches}$ квантовой схемы, число взаимосвязанных между собой ветвей N_{deep} (глубина; некоторые гейты функционируют с привязкой нескольких ветвей, например CCNOT). Квантовой схемой над базисом B называется последовательность унитарных операторов $U_1[S_1], U_2[S_2], ..., U_l[S_l]$, где $U_l \in B$, $S_l \subseteq \{1,...,n\}$, n - число кубитов. Более формально, матричные элементы оператора U[S] имеют следующий вид. Пусть

$$U = \sum_{x,y \in \{0,1\}^d} u_{x,y} \mid x \rangle\langle y \mid, \quad S = \{j_1, j_2, ..., j_d\}, \; 1 \le j_1 < j_2 < ... < j_d \le n \qquad (1)$$

Обозначим x[S] как под-последовательность битов, стоящих на местах из множества S. Тогда оператор U[S] записывается как

$$U[S] = \sum_{x,y\in\{0,1\}^n : x[\bar{S}]=y[\bar{S}]} u_{x[S],y[S]} \mid x >< y \mid , \quad (2)$$

$$U_1[S_1], U_2[S_2], ..., U_l[S_l] = \frac{N_{gates}}{N_{branches} + N_{deep}} . \quad (3)$$

Квантовые регистры и их количество. С точки зрения увеличения производительности два КР для распараллеливания лежащих на них задачах, как например в описанной в п. 1 схеме вычислительной структуры СКВ.

Начальное базисное состояние СКВ (НБС СКВ). Квантовое состояние - это любое возможное состояние, в котором может находиться квантовая система. Пусть состояние квантовой системы задано вектором $|\psi\rangle$. При разложении в соответствии с полными наборами векторов системы $|\varphi_i\rangle$ и ее окружения $|\theta_j\rangle$: $|\psi\rangle = \sum_{i,j} c_{ij} * |\varphi_i\rangle * |\theta_j\rangle$, где c_{ij} - амплитуда вероятности нахождения системы в j-м состоянии. Очевидно, что производительность СКВ с точки зрения НБС зависит от числа ветвей $N_{branches}$ квантовой схемы и вида базисного состояния (БС). Стоит отметить, что само по себе НБС СКВ не влияет на его производительность. Однако при рассмотрении НБС в рамках жизненного цикла квантовой схемы нельзя не учесть величину манипуляций пользователя при работе с СКВ в рамках ее начального приготовления и настройки (СКВ относится к типу автоматизированных систем).

Получаем уравнение, отражающее величину производительности в рамках начального базисного состояния СКВ:

$$|\psi\rangle = N_{branches} * k , \quad (4)$$

где k – коэффициент вида базисного состояния. Так как значение $N_{branches}$ может быть бесконечно большим, не представляется возможным оценить данный параметр никак кроме процентного соотношения.

Запуск различных алгоритмов в рамках разработанного СКВ

Разработанный СКВ также предполагает реализацию набора квантовых алгоритмов, что в свою очередь влияет на производительность всего СКВ. Затраты на производительность в данном случае прямопропорциональны классу сложности конкретного квантового алгоритма, что подробно реализовано в работе [6]:

$$\sum_n S = \frac{O(n)}{N_o + N_q} * (P \mid NP \mid ZPP \mid BPP \mid BQP), \quad (5)$$

где $O(n)$ – временная сложность квантового алгоритма, или, если проще, время работы конкретного алгоритма, N_o – количество выполняемых в алгоритме операций, N_q – количество запросов к квантовому оракулу в процессе работы квантового алгоритма, $P \mid NP \mid ZPP \mid BPP \mid BQP$ – классы сложности квантовых алгоритмов (| – операция логическое ИЛИ).

В системе ниже представлены математические формализации некоторых классов сложности.

$$\begin{cases} P : C_M(n) = \max\limits_{x:|x|=n} T_M(x)(1), C_M(n) < n^c(6) \\ NP : \cup_{i=0}^{\infty} NTIME(i * n^i) = \cup_{i=0}^{\infty} \cup_{k=0}^{\infty} NTIME(i * n^k)(7), \\ NTIME(f(n)) = \{L \mid \exists m : L(m) = L, T(m, x) \le f(|x|)\}(8) \\ ZPP : 0 \mid 1 \\ BPP : P(\text{полиномиальное время выполнения}), error = (2\sqrt{p(1-p)})^n \\ BQP : P(\text{полиномиальное время выполнения}), 0 \le error \le \frac{1}{3} \end{cases}$$

Алгоритм при полиномиальном времени (P) эквивалентен детерминированной машиной Тьюринга (МТ), вычисляющей ответ по данному на входную ленту слову из входного алфавита. Сложностью

функции f называется функция C, зависящая от длины входного слова и равная максимуму времени работы машины по всем входным словам фиксированной длины (1). Если для функции f существует МТ M такая, что (6) для некоторого числа C, то говорят, что она принадлежит классу P.

NP - класс задач, ответ на которые можно получить за время P, а классом NTIME(f) называется класс задач, для которых существует недетерминированная МТ, работа которой останавливается при превышении длины входа (8). Формальное определение класса NP через класс NTIME выглядит так (7).

Суммируя все вышеизложенное, получим:

$$P = \left[\frac{\sum_{l} U_l[S_l] * |\psi\rangle}{2} \right] * \sum_{n} S \tag{9}$$

$$P = \left[\frac{\frac{N_{gates}}{N_{branches} + N_{deep}} * N_{branches} * k}{2} \right] * \frac{O(n)}{N_o + N_q} * (P \mid NP \mid ZPP \mid BPP \mid BQP)$$

$$P = \left[\frac{N_{gates} * N_{branches} * k}{2 * (N_{deep} + N_{branches})} \right] * \frac{O(n)}{N_o + N_q} * (P \mid NP \mid ZPP \mid BPP \mid BQP)$$

$$P = \left[\frac{N_{gates} * N_{branches} * k}{2 * (N_{deep} + 1) * N_{branches}} \right] * \frac{O(n)}{N_o + N_q} * (P \mid NP \mid ZPP \mid BPP \mid BQP)$$

$$P = \left[\frac{N_{gates} * k}{2N_{deep} + 2} \right] * \frac{O(n)}{N_o + N_q} * (P \mid NP \mid ZPP \mid BPP \mid BQP)$$

Заключение

Построенный на основе предложенной вычислительной структуры СКВ позволяет:

1) оптимизировать вычислительный процесс симулятора в соответствии с набором компонентов: сдвоенный квантовый регистр (QR1

и QR2), схема фильтрации и измерения, квантовые блоки пересечений QB_m и число кубитов k_m.

2) находить новые способы применения данной универсальной структурной схемы для моделирования каких-либо параметров в исполняемой задаче.

Такие исследования необходимо применять прежде, чем реализовывать квантовые алгоритмы и вычислительные устройства в реальности. Например, для того, чтобы оценить требуемый уровень запутанности для эффективной работы алгоритма или оптимальное количество элементов СКВ и их взаимодействие.

Кроме всего прочего моделирование квантовых алгоритмов, определение производительности СКВ доказывает целесообразность исполнения алгоритма на дорогостоящем квантовом вычислительном устройстве.

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

Литература

1. QuIDDPro: High-Performance Quantum Circuit Simulation // URL: http://vlsicad.eecs.umich.edu/Quantum/qp/ (Дата обращения: 11.06.2017);

2. libquantum 1.1.1 // URL: http://www.libquantum.de/api/1.1/index.html (Дата обращения: 11.06.2017);

3. Cove: A Practical Quantum Computer Programming Framework // URL: http://cove.purkeypile.com/ (Дата обращения: 11.06.2017);

4. Guzik V., Gushanskiy S., Polenov M., Potapov V. Models of a quantum computer, their characteristics and analysis // 9th International Conference on Application of Information and Communication Technologies (AICT). – Institute of Electrical and Electronics Engineers, 2015. – P. 583-587;

5. Правильщиков П.А. Квантовый параллелизм и новая модель вычислений // Труды XII Всероссийского совещания по проблемам управления ВСПУ-2014.

6. Потапов В.С., Гузик В.Ф., Гушанский С.М. О производительности и вычислительной сложности квантовых алгоритмов // Информатизация и связь. – 2017, № 3. – С.24-29.

References

1. QuIDDPro: High-Performance Quantum Circuit Simulation // URL: http://vlsicad.eecs.umich.edu/Quantum/qp/ (Data obrashcheniya: 11.06.2017);
2. libquantum 1.1.1 // URL: http://www.libquantum.de/api/1.1/index.html (Data obrashcheniya: 11.06.2017);
3. Cove: A Practical Quantum Computer Programming Framework // URL: http://cove.purkeypile.com/ (Data obrashcheniya: 11.06.2017);
4. Guzik V., Gushanskiy S., Polenov M., Potapov V. Models of a quantum computer, their characteristics and analysis // 9th International Conference on Application of Information and Communication Technologies (AICT). – Institute of Electrical and Electronics Engineers, 2015. – P. 583-587;
5. Pravil'shchikov P.A. Kvantovyj parallelizm i novaya model' vychislenij // Trudy XII Vserossijskogo soveshchaniya po problemam upravleniya VSPU-2014.
6. Potapov V.S., Guzik V.F., Gushanskij S.M. O proizvoditel'nosti i vychislitel'noj slozhnosti kvantovyh algoritmov // Informatizaciya i svyaz'. – 2017, № 3. – S.24-29.