

УДК 004.942

ПОСТРОЕНИЕ КВАНТОВЫХ СИМУЛЯТОРОВ ДЛЯ ИССЛЕДОВАНИЯ КВАНТОВЫХ АЛГОРИТМОВ*

В.Ф. Гузик¹, Е.В. Ляпунцова²,
С.М. Гушанский¹

¹ ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ,

² МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ

В работе предполагается описание основ построения квантовых симуляторов для исследования квантовых алгоритмов. Проанализированы общий подход к моделированию квантовых вычислений и структура симулятора квантового вычислителя, а также математическое ядро модели квантового вычислителя. Рассмотрены средства создания ядра вопросы по построению интерфейса пользователя.

КЛЮЧЕВЫЕ СЛОВА: моделирование, симулятор, квантовый компьютер, модель, модуль, кубит, вычислитель, модель квантового вычислителя, открытые системы.

АКТУАЛЬНОСТЬ ПРОБЛЕМЫ

Разработка симулятора квантового вычислителя (СКВ) – это сложная и актуальная задача, так как в настоящее время существует еще достаточно ограниченное количество квантовых алгоритмов, которое имеет тенденцию к росту [2]. С другой стороны, реально использовать квантовый вычислитель (КВ), посредством которого можно было бы проверить полученные при моделировании результаты, не представляется возможным, в связи с естественными технологическими и, самое главное, большими финансовыми затратами [4]. Более того, мы не знаем конкретно какие технологии будут положены в основу квантового вычислителя при построении квантовых схем или квантовых вентилях (гейтов): ионные ловушки, впадины QED, соединения Джозефсона, ЯМР-технологии или другие. Поэтому моделирование как самостоятельная часть разработки КВ, использующего новую технологию, для моделирования физической модели квантовых процессоров может оказаться необходимым.

* Работа выполнена при финансовой поддержке РФФИ, грант № НК 15-01-01270\15

CONSTRUCTION OF QUANTUM SIMULATORS FOR INVESTIGATING QUANTUM ALGORITHMS

V.F. GUZIK,
E.V. LYAPUNTSOVA, S.M. GUSHANSKY

In this work supposed to construction of the foundations of quantum description of the simulation for the study of quantum algorithms. Analyzed the general approach to the modeling of quantum computation and simulation of the quantum structure of the calculator, as well as the core of mathematical models of quantum calculator. Considered a means of creating and calculating the quantum core issues on the construction of the user interface.

KEYWORDS: modeling; simulator, quantum computing, model, module, qubit, calculator, model of quantum computer, open systems.

В противоположность обычным цифровым вычислителям, где внутренняя работа каждой базовой единицы не связана с логической работой всей машины, внутренняя квантовая динамика каждого логического элемента – ключевая составная часть КВ. Следовательно, важным элементом для соединения в моделируемый макет, является физика элементарных единиц, формирующая КВ. Теоретическая работа над квантовыми вычислениями обычно предполагает существование единиц, выполняющих высоко идеализированных единых операций. Однако на практике эти операции сложны в реализации: не учитывая декогерентизацию, физические КВ будут выполнять отдельные операции, которые сложнее, чем рассматриваемые в теоретической работе. Следовательно, для физической реализации квантовых процессоров важно иметь теоретические средства. Таким образом, СКВ должен предоставлять необходимые абстрактные конструкции, предназначенные для расчетов, как физикам-экспериментаторам, так и составителям квантовых алгоритмов.

Имеется также ряд других, не менее важных причин актуальности построения СКВ [1, 3]:

– Разработанная квантовая модель позволит наглядно увидеть сильные и слабые стороны модели, а также усовершенствовать ее в будущем.

– Пока не создан квантовый компьютер, единственная возможность практического изучения квантового компьютеринга – моделирование квантового компьютера на классическом или использование различных элементов физики и химии в не характерных для них состояниях и направлениях.

– В случае создания квантового компьютера моделирование его прототипа станет наглядным пособием для понимания основных процессов и явлений, благодаря которым стало возможным его создание.

– Разработанная квантовая модель и ее доказанные преимущества позволят легче привлечь инвестиции в физическое создание и совершенствование квантового компьютера.

ОБЩИЙ ПОДХОД К МОДЕЛИРОВАНИЮ КВАНТОВЫХ ВЫЧИСЛЕНИЙ

При моделировании вычислений квантового характера на некотором суперкомпьютере объем памяти, необходимый для успешного выполнения вычислений, растет экспоненциально от числа кубитов. Следовательно, возможности моделирования квантового компьютера на классических устройствах весьма ограничены. К примеру, на современных персональных компьютерах доступное для моделирования и вычислений количество кубитов составляет примерно 25–30, а для моделирования 35 идеальных кубитов уже потребуется более 1 терабайта оперативной памяти. Моделирование квантовых вычислений требует специализированного программного обеспечения, которое позволит легко эмулировать выбранную квантовую схему. Существует также возможность использования больших кластеров с распределенной памятью. Для этого требуется производить взаимодействия между вычислительными узлами. Протокол MPI обычно используется для этих целей. Примером реализации программного обеспечения, удовлетворяющего таким условиям, является написанный на языке Фортран с использованием MPI набор программ QCMPI [15]. Он представляет собой не что иное, как большой выбор операторов:

- оператор Паули;
- преобразование Адамара;
- контролируемое отрицание;
- фазовый сдвиг;
- квантовое преобразование Фурье и др.

Используются библиотеки линейной алгебры BLAS [15], LAPACK [7] и SCALAPACK [14]. В случае моделирования квантовых вычислений целесообразно использовать специальные алгоритмы для разреженных матриц.

В статье [9] описывается алгоритм моделирования квантовых вычислений на компьютере с общей памя-

тью. В этой работе используются специальные структуры данных для хранения разреженных матриц, при помощи которых представляется эволюция квантовых состояний в ходе вычислений. В результате такого подхода было достигнуто моделирование 32 кубитов на довольно небольших ресурсах: 16 гигабайт памяти и 64 процессора. Однако указанный алгоритм применим только для систем с общей памятью, когда нет необходимости в пересылках между различными вычислительными узлами.

В статье [8] описывается аналогичный подход к реализации модели квантового компьютера. В этой статье показано, что при моделировании 37-кубитного квантового компьютера можно добиться эффективности в 1 квантовую операцию за 10 секунд. Кроме того, возможно использование реконфигурируемых вычислителей [13] для моделирования квантовых алгоритмов, однако и в этом случае размерность моделируемой системы ограничена памятью вычислителя.

Общая схема построения квантовых симуляторов приведена на рис. 1.

СТРУКТУРА СИМУЛЯТОРА КВАНТОВОГО ВЫЧИСЛИТЕЛЯ

Для удобства разработки, модификации, портируемости СКВ, связи с языками программирования и средами разработки он имеет многослойную архитектуру:

- Ядро СКВ.
- Квантовые вентили.
- Библиотека базовых квантовых алгоритмов.
- Интерфейсы для языков программирования и сред разработки.
- Учебная интегрированная среда разработки.
- Интерфейс пользователя.

Ядро – это компонент, который содержит в себе все необходимое для моделирования кубит (кванто-

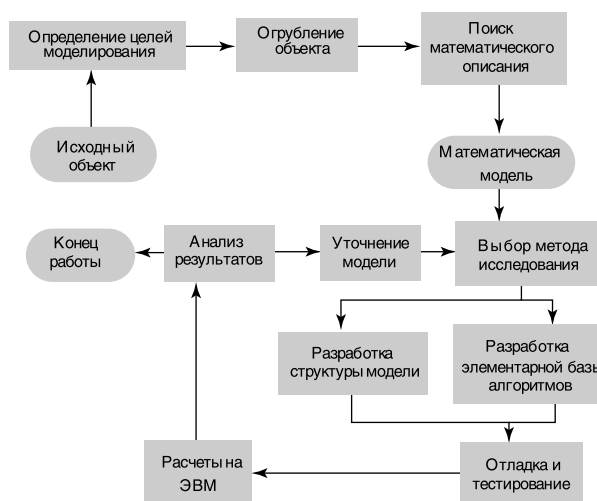


РИС. 1.

Общая схема построения квантовых симуляторов

вого регистра) и набора операций (гейтов), которые применяются к данным кубита.

Ядро эмулятора осуществляет симуляцию одного квантового регистра с помощью его вектора состояния, реализацию произвольных одно- и двухкубитовых вентилях, предоставляет функции для чтения и записи как для регистра, так и для его отдельных кубитов. В отладочных целях имеется доступ непосредственно к вектору состояния.

Для облегчения работы с ядром имеются функции, реализующие конкретные, одно- и двухкубитовые вентили.

В настоящее время реализованы все функции ядра и основные квантовые вентили. Библиотека написана на языке Си и допускает как статическую, так и динамическую компоновку с программами пользователя.

Библиотека базовых квантовых алгоритмов включает реализацию наиболее часто используемых рутинных квантовых алгоритмов, например квантового фурье-преобразования (QFT). Она будет использовать ядро и реализацию конкретных одно- и двухкубитовых вентилях. Имеется возможность динамической компоновки (DLL).

Интерфейсы для языков программирования дают возможность использования ядра эмулятора с уже существующими языками, средами разработки и библиотеками без его доработки и перекомпиляции.

Учебная интегрированная среда разработки создается на основе уже существующих программных продуктов. Она должна обеспечивать быструю и эффективную разработку и отладку квантовых алгоритмов. В случае языка высокого уровня эмулятор будет выглядеть для программиста как набор API для квантовых вычислений, управляющие конструкции «несущего» языка остаются без изменений. При применении языка низкого уровня введение квантовых вычислений целесообразно в виде специализированных команд, дополняющих возможности x86 (они могут быть реализованы как макрорасширения уже существующих ассемблеров). При этом в качестве прототипа может быть взят язык QASM для описания квантовых схем [12].

МАТЕМАТИЧЕСКОЕ ЯДРО МОДЕЛИ КВАНТОВОГО ВЫЧИСЛИТЕЛЯ

Можно ли использовать квантовые системы для решения других вычислительных задач? Какова должна быть математическая модель квантового компьютера, в той же степени не зависящая от физической реализации, что и модели классических вычислений? Эти вопросы, по-видимому, впервые были поставлены в книге Ю.И. Манина «Вычислимое и невычислимое» (1980). Они обсуждались также в работах Р. Фейнмана и других авторов. В 1985 г. Д. Дойч предложил конкретную математическую модель – квантовую машину Тьюринга [11], а в 1989 г. – эквивалентную, но более удобную модель – квантовые схемы [10].

Что такое квантовая схема? Пусть в нашем распоряжении имеется N спинов, каждый из которых находится в отдельном ящичке и идеально изолирован от окружающего мира. В каждый момент времени мы можем выбрать, по нашему усмотрению, любые два спина и действовать на них любой унитарной матрицей 4×4 . Последовательность таких операций называется квантовой схемой. Каждая операция определяется парой номеров спинов и шестнадцатью комплексными числами, поэтому квантовую схему можно записать на бумаге. Это своего рода программа для квантового компьютера.

Чтобы использовать квантовую схему для вычисления функции, нужно уметь вводить входные данные, продвигать вычисления и считывать результат. Ввести в квантовый компьютер последовательность (x_1, \dots, x_n) нулей и единиц – значит приготовить начальное состояние $|x_1, \dots, x_n, 0, \dots\rangle$. (Объем входных данных обычно меньше общего количества «ячеек памяти», т. е. спинов, N . Оставшиеся ячейки заполняются нулями.) К начальному состоянию применяется квантовая схема, зависящая от решаемой задачи, но не от конкретных входных данных. В итоге возникает квантовое состояние

$$|\Psi(x_1, \dots, x_n)\rangle = \sum_{y_1, \dots, y_n} c_{y_1, \dots, y_n}(x_1, \dots, x_n) |y_1, \dots, y_n\rangle,$$

зависящее от (x_1, \dots, x_n) . Теперь нужно считать результат. Предполагаем, что ответ должен содержаться в первых m битах строки (y_1, \dots, y_m) , т. е. мы ищем такие (y_1, \dots, y_m) , что $(y_1, \dots, y_m) = F(x_1, \dots, x_n)$. Для получения ответа производится измерение значений всех спинов. Результатом измерения может быть любая последовательность нулей и единиц (y_1, \dots, y_m) , вероятность получить ее равна $|c_{y_1, \dots, y_m}(x_1, \dots, x_n)|^2$. Таким образом, квантовый компьютер может, с некоторой вероятностью, дать любой ответ. Квантовая схема является «правильной» для данной функции F , если правильный ответ $(y_1, \dots, y_m) = F(x_1, \dots, x_n)$ получается с вероятностью, достаточно близкой к единице. Повторив все вычисление несколько раз и выбрав тот ответ, который встречается чаще, можно снизить вероятность ошибки до сколь угодно малой величины.

СРЕДСТВА СОЗДАНИЯ ЯДРА КВАНТОВОГО ВЫЧИСЛИТЕЛЯ

Средствами по созданию ядра выберем набор библиотек, способных осуществлять ряд задач, поставленных на квантовый вычислитель. В первую очередь необходимо решить для каких целей будет использоваться библиотека квантовых вычислений:

– Средство разработки квантового ПО в привычной современной среде – на языке C/C++/C# и т. д.

– Эмуляция квантового регистра, унитарных преобразований, условных операторов, операторов изменения.

– Удобный набор стандартных операторов для квантовых вычислений: элемент Адамара, операторы Паули, CNOT, Тоффоли и т.д.

В настоящее время известно не так много средств для моделирования ядра квантового вычислителя, но можно выделить ряд библиотек квантовой направленности если не для реализации самого ядра, то для осуществления его функциональности – QLib [16]. QLib представляет собой пакет Matlab, предназначенный для обеспечения доступности широкой аудитории в рамках сообщества QIT (Quantum information theory) инструментами, необходимыми для ускорения исследований по:

1. Быстроте и эффективности обрaмления и исследования вопросов.
2. Формированию интуиции с помощью визуализации.
3. Проверке гипотезы с помощью оптимизации.

QLib в настоящее время охватывает большинство, если не весь базис примитивов и дает богатый набор инструментов, с которыми становится возможным заниматься вопросами квантовых вычислителей в экспериментальной теории.

Еще одна библиотека – QSim [5].

QSim – это проект, который направлен на проблемы моделирование при участии Технологического университета Джорджии, создания цепи безопасных библиотек эмуляции многоядерных систем на основе эмулятора QEMU. QSim обеспечивает управление на уровне инструкций эмулируемой среды и подробную информацию о выполняющемся потоке команд. Применение QSim включают трассировку-сбор, разработку многопоточного программного обеспечения, и его первичной цели: микроархитектура моделирования.

ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ

Рассмотрим алгоритм работы СКВ (рис. 2) и взаимодействия различных его составляющих: каркасом, ядром, квантовыми вентилями, квантовой схемой, библиотеками базовых квантовых алгоритмов, интерфейсами для языков программирования и сред разработки, учебной интегрированной средой разработки и интерфейсом пользователя. Зависимость компонентов друг от друга очень высока.

После инициализации каркаса (формы) необходимо сначала инициализировать моделирующее ядро, в котором по умолчанию задано количество кубит в системе, на основе этих данных инициализируется квантовая схема и отображает данные о кубитах из ядра. После этого необходимо инициализировать интерфейс пользователя, в котором будет происходить большая часть работы на компьютере. Далее, последовательно или параллельно, может быть выбран один из путей для дальнейшей работы.

Интерфейс пользователя (ПИ) – разновидность интерфейсов, в котором одна сторона представле-

на человеком (пользователем), другая – машиной/ устройством. Представляет собой совокупность средств и методов, при помощи которых пользователь взаимодействует с различными, чаще всего сложными, машинами, устройствами и аппаратурой.

Очень часто пользовательский интерфейс понимают только как внешний вид программы. Однако на деле пользователь воспринимает через него всю программу в целом, а значит, такое понимание является слишком узким. В действительности ПИ объединяет в себе все элементы и компоненты программы, которые способны оказывать влияние на взаимодействие пользователя с программным обеспечением (ПО), это не только экран, который видит пользователь. Общая структурная схема интерфейса пользователя приведена на рис. 3.

Графические акселераторы (ускорители) — это специализированные графические сопроцессоры, увеличивающие эффективность видеосистемы. Их применение освобождает центральный процессор от большого объема операций с видеоданными, так как акселераторы самостоятельно вычисляют, какие пиксели отображать на экране и каковы их цвета.

ЗАКЛЮЧЕНИЕ

В заключение выделим некоторые характеристики, которыми должна обладать наиболее совершенная модель. Проанализировав возможности существующих моделей, можно отметить, что модель должна иметь следующие показатели:

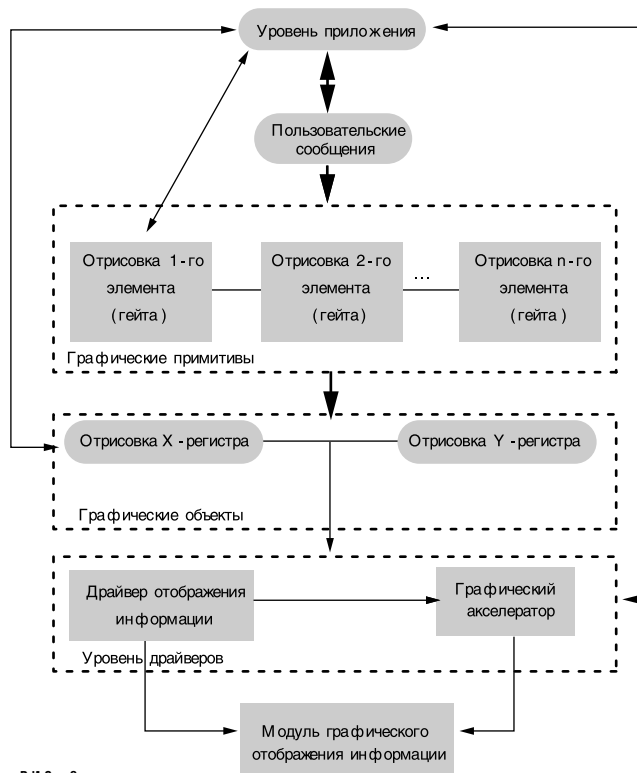


Рис. 2. Алгоритм работы СКВ

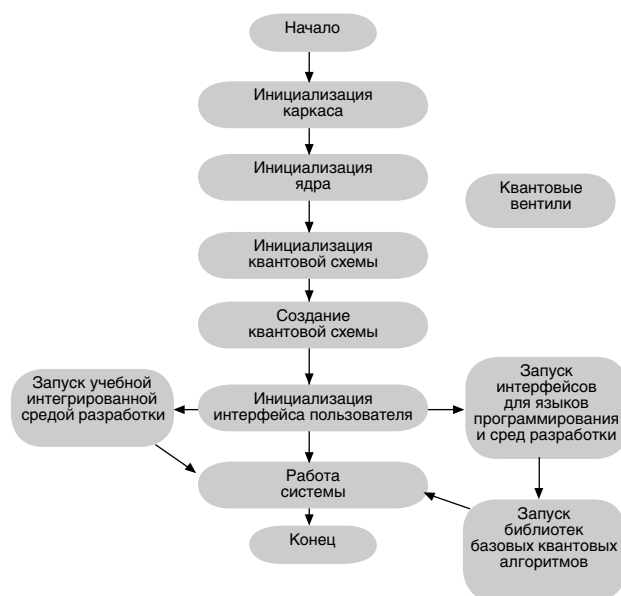


Рис. 3.

Структурная схема интерфейса пользователя

1. Архитектура типа «Блок», которая позволит добавлять в общую архитектуру новые составляющие.

2. Наличие визуализаций для всех вычислений и действий в модели при помощи цветowych схем, графиков и др.

3. Осуществление контролируемым образом логических операции над кубитами («условная динамика»).

4. Модель должна быть масштабируемой. То есть, с одной стороны, количество используемых кубит не должно быть преградой для успешного проведения вычислений, а, с другой стороны, в модели должна присутствовать возможность не только увеличения второстепенных элементов (квантовый регистр, устройство записи), но и главного – ядра модели.

ЛИТЕРАТУРА

1. Гузик В.Ф., Гушанский С.М., Потапов В.С. Проведение полного факторного эксперимента для характеристик модели квантового вычислителя // Изв. Южного фед. ун-та. Технические науки. Компьютерные и информационные технологии в науке, инженерии и управлении. 2015. № 3 (164). С. 46–54.
2. Квантовые алгоритмы. [Электронный ресурс]. https://en.wikipedia.org/wiki/Quantum_algorithm/ (Дата обращения 07.08.2015).
3. Ляпунцова Е.В., Авдеев В.А., Гузик В.Ф. Схемотехника ЭВМ и программирование в функциональных диалоговых моделях. М.: Физматлит, 2014. 636 с.
4. Стартовали продажи первого квантового компьютера. [Электронный ресурс]. <http://albatron.ru/86-startovali-prodazhi-pervogo-kvantovogo-kompyutera.html> (Дата обращения 07.08.2015).

5. About QSim // URL: <http://www.cdkersey.com/qsim-web/> (Дата обращения 08.08.2015).
6. ALTSCHUL S., GISH W., MILLER W., MYERS E., LIPMAN D. Basic local alignment search tool // J. Molecular Biol. 1990. Vol. 215, N 3. P. 403–410.
7. ANDERSON E., BAI Z., BISCHOF C., BLACKFORD S., DEMMEL J., DONGARRA J., DU CROZ J., GREENBAUM A., HAMMARLING S., MCKENNEY A., SORENSEN D. LAPACK Users' Guide (Third Ed.). Philadelphia: SIAM, 1999.
8. ARNOLD G., LIPPERT T., POMPLUN N., RICHTER M. Large scale simulation of ideal quantum computers on SMP-clusters // Proc. Conf. Parallel Computing (ParCo). Malaga, 2005. P. 447–454.
9. BURGER J.R. New approaches to quantum computer simulation in a classical supercomputer // Computing Research Repository (CoRR). 2003. Vol. Quant-ph/0308158.
10. DEUTSCH D. Quantum computation networks // Proc. Royal Soc. London. Ser. A. Mathematical and Physical Sciences. 1989. P. 73–90.
11. DEUTSCH D. Quantum theory, the Church-Turing principle and the universal quantum computer // Proc. Royal Soc. London. Ser. A. Mathematical and Physical Sciences. 1985. P. 97–117.
12. Quantum Computer Architecture Resources // URL: <http://www.media.mit.edu/quanta/qasm2circ/> (Дата обращения 02.08.2015).
13. NEGOVETIC G., PERKOWSKI M., LUKAS M., BULLER A. Evolving quantum circuits and an FPGA-based quantum computing emulator // Int. Workshop on Boolean Problems. Freiberg, 2002. P. 15–22.
14. ScaLAPACK — Scalable Linear Algebra PACKage // URL: <http://www.netlib.org/scalapack/> (Дата обращения 02.08.2015).
15. ТАВАКИН Ф., JULI 'A-D 'IAZ B. QCMPI: A parallel environment for quantum computing // Computer Physics Communications. 2009. Vol. 18. P. 948–964.
16. What is QLib? // URL: <http://www.tau.ac.il/~quantum/qlib/qlib.html> (Дата обращения 08.08.2015).

Всего по данной проблематике авторами опубликовано более 110 работ.

Гузик Вячеслав Филиппович,
д.т.н., профессор, зав. кафедрой вычислительной техники
Южного федерального университета,
☎ 344006, Ростов-на-Дону, ул. Б. Садовая, д. 105

Ляпунцова Елена Вячеславовна,
д.т.н., профессор кафедры интеллектуальные транспортные системы
Московского государственного университета путей сообщения,
советник сенатора СФ РФ,
☎ 127994, г. Москва, ул. Образцова, д. 9, стр. 9

Гушанский Сергей Михайлович,
к.т.н., доцент, доцент кафедры вычислительной техники
Южного федерального университета,
☎ г. Таганрог, П. Тольятти, 36-34,
тел.: +7 (918) 556-16-25, e-mail: selen2616@mail.ru