УДК 004.021-004.8
DOI: 10.25559/SITITO.18.202204.756-766

Оригинальная статья

Реализация векторной модели данных с использованием SIMD регистров

В. Ю. Ходченков*, В. И. Мунерман
ФГБОУ ВО «Смоленский государственный университет», г. Смоленск, Российская Федерация
Адрес: 214000, Российская Федерация, г. Смоленск, ул. Пржевальского, д. 4
* tansdf@mail.ru

Аннотация

В статье рассматривается возможность реализации векторной модели данных с использованием SIMD регистров. Необходимость решения этой проблемы вытекает из того факта, что человечество находится на пороге перехода от обычных компьютеров к их квантовым аналогам. На данный момент эмуляторы развиты недостаточно хорошо для свободной работы над квантовыми алгоритмами. Однако потребность в построении квантовых алгоритмов растёт с каждым новым шагом в разработке полноценного общедоступного квантового компьютера. В статье предлагается метод реализации векторной модели данных с использованием SIMD регистров. Наиболее близкой к параллельной векторной реализации является гипотетическая реализация квантового компьютера, поэтому именно она может послужить основой к разработке модели отвечающей всем критериям квантовой модели, но без использования реальной квантовой системы, что позволит сэкономить средства и увеличить эффективность разработки квантовых систем. Используется терминология области разработки квантовых компьютеров. На сегодняшний день, в большинстве источников, которые приводят метод формирования квантового сумматора, речь идёт о создании неких кубитовых регистров для хранения складываемых чисел. То есть, число хранится в двоичной форме в регистре кубит, каждый из которых находится в определённом квантовом состоянии, не запутанном, а измеренном. Значит, регистр такого вида является по своей сути обыкновенным битовым регистром. Такой подход относительно лёгок в реализации, но он не имеет практического смысла и применения, ведь выигрыш в скорости вычисления в этом случае крайне низок. Поэтому, можно сказать, что превращение кубита в обычный бит, не имеет смысла. Целью данной работы является оформление полученных ранее теоретических результатов в виде полноценной модели вычислений, которая позволит разработать новый тип сумматора на основе матричных операций.

Ключевые слова: квантовые вычисления, квантовый параллелизм, векторный процессор

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Ходченков В. Ю., Мунерман В. И. Реализация векторной модели данных с использованием SIMD регистров // Современные информационные технологии и ИТ-образование. 2022. Т. 18, № 4. С. 756-766. doi: https://doi.org/10.25559/SITITO.18.202204.756-766

Original article

Implementation of a Vector Data Model Using SIMD Registers

V. Yu. Khodchenkov*, V. I. Munerman
Smolensk State University, Smolensk, Russian Federation
Address: 4 Przhevalsky St., Smolensk 214000, Russian Federation
* tansdf@mail.ru

Abstract
The article considers the possibility of implementing a vector data model using SIMD registers. The need to solve this problem stems from the fact that humanity is on the verge of transition from conventional computers to their quantum counterparts. Emulators are not well-developed for free work on quantum algorithms. However, the need to build quantum algorithms is growing with each new step in the development of a full-fledged public quantum computer. The article proposes a method for implementing a vector data model using SIMD registers. The closest to a parallel vector implementation is a hypothetical implementation of a quantum computer, so it can serve as the basis for developing a model that meets all the criteria of a quantum model, but without using a real quantum system, which will save money and increase the efficiency of developing quantum systems. The terminology of the field of development of quantum computers is used. Today, in most sources that give a method for forming a quantum adder, we are talking about creating some kind of qubit registers for storing added numbers. That is, the number is stored in binary form in a register of qubits, each of which is in a certain quantum state, not entangled, but measured. This means that a register of this kind is inherently an ordinary bit register. This approach is relatively easy to implement, but it has no practical meaning and application, because the gain in calculation speed in this case is extremely low.

Keywords: quantum computing, quantum parallelism, vector processor

Conflict of interests: The authors declare no conflict of interest.

For citation: Khodchenkov V.Yu., Munerman V.I. Implementation of a Vector Data Model Using SIMD Registers. *Modern Information Technologies and IT-Education.* 2022;18(4):756-766. doi: https://doi.org/10.25559/SITITO.18.202204.756-766

Введение

Алгебра векторов [0;1] может широко использоваться для решения различного рода задач, поэтому целесообразно произвести разработку модели вычислений на основе этих векторов. Основой для построения собственного вычислителя становятся любые доступные регистры одной размерности.

Наиболее близкой к параллельной векторной реализации является гипотетическая реализация квантового компьютера, поэтому именно она может послужить основой к разработке модели отвечающей всем критериям квантовой модели, но без использования реальной квантовой системы, что позволит сэкономить средства и увеличить эффективность разработки квантовых систем. Далее будем использовать именно терминологию области разработки квантовых компьютеров [1-10].

Реализация бита в квантовых компьютерах отличается от классической реализации. Битом назовём минимальную единицу информации, которая, в общепринятом случае является током низкого напряжения 0 или током высокого напряжения 1. Квантовый бит (кубит) может находиться в суперпозиции состояний 0 и 1. Обозначения Дирака – приняты для обозначения квантовых бит. В этих обозначениях формула состояния кубита может быть представлена как:

$$|f\rangle = a|0\rangle + b|1\rangle$$

Где a и b – комплексные коэффициенты, которые обязательно соответствуют требованию нормализации:

Для формирования модели из регистров формируются пары, для получения эмулированных кубит. Следовательно, в нашем распоряжении шестнадцать кубит, так как в нашем случае используются 32 свободных регистра.

Предположим, есть два вектора a и b, такие что: $a \in Q^2$, $b \in Q^2$

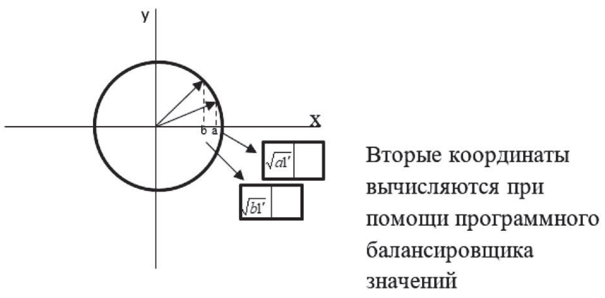
И норма каждого из этих векторов: $\|a\| = \|b\| = r$

Положим: $a' = a/r$, $b' = b/r$

Тем самым мы приводим оба вектора к единичной норме.

Затем значения a', b', поступают на вход квантового сумматора. Первая координата первого вектора записывается в первую координату первого кубита как: $\sqrt{a1'}$

И первая координата второго вектора записывается в первую координату второго кубита как: $\sqrt{b1'}$



Р и с. 1. Представление кубит на единичной окружности

F i g. 1. Representation of a qubit on the unit circle

После этого, в общем случае, последующие: третий, пятый, ..., $2n + 1$ кубиты принимают значение, эквивалентное первому кубиту, аналогично четвертый, шестой, ..., $2n + 2$ кубиты принимают значение, эквивалентное второму кубиту. С увеличением количества дублирующих векторов, возрастает точность вычисления. Далее, увеличим первые координату $2n+1$-го и $2n$-ого кубита в k^n раз. Если быть точным, первая координата каждых двух последующих кубитов будет увеличена в раз по сравнению с предыдущим кубитом. Проведём измерение полученных эмулированных квантовых битов. Результатом данной операции является набор базисных состояний кубита – обычных 0 и 1. Измерение производится компонентом системы, имеющим доступ к текущим значениям регистров, на основе алгоритма описанного в главе 2. Для дальнейших преобразований, нам выбираем номер m, начиная с которого пары эмулированных кубит находятся во втором базисном состоянии (что указывает на то, что вероятность получения этого состояния выше, нежели первого состояния). Следовательно, мы можем определить произведение первых координат пар кубитов как:

$$\frac{1}{a1' \times b1'} = k^{2m} \times \left(1 + \frac{1}{k^2}\right) \quad (1)$$

Поскольку число m означает число, после которого знак неравенства квадратов координат изменился, учитывая выбор шага k и количество дополнительных кубитов, мы получим пределы определения a1'*b1', затем выберем значение как среднее значение между пределами.

Назовем правую часть (1) X.

Точно так же, зная число p, начиная с которого последовательность будет содержать единицы в каждой позиции, произведение вторых координат определим, как:

$$\frac{1}{1 - (a1' + b1') + a1' \times b1'} = k^{2p} \times \left(1 + k^2\right) \quad (2)$$

Назовем правую часть (2) как Y.

Учитывая вышеизложенное, мы можем получить:

$$a1' + b1' = \frac{1}{X} - \frac{1}{Y} + 1 \quad (3)$$

Аналогично для вторых координат векторов.

Таким образом, мы получаем сумму первой и второй координат векторов, что означает, что мы можем получить сумму векторов, заменив:

$$a + b = (a' + b') \times r \quad (4)$$

Таким образом, операция сложения 2-мерных векторов одной нормы реализуется на основе способности компьютера производить операции над произвольным числом SIMD-регистров. Эта реализация является основой для создания процессора на основе алгебры многомерных матриц, поскольку, как упоминалось ранее, любой 2-мерный вектор, выступающий в качестве элемента этой алгебры, можно рассматривать как линейную комбинацию базовых нормализованных векторов.

Построение векторной модели с использованием SIMD открывает возможность выполнять операции в рамках этой модели параллельно. Сама основа SIMD регистров покрывает возможности параллелизма, а значит модель, построенная на их основе, будет включать параллельные вычисления как основу, что в свою очередь говорит о возможном увеличении скорости обработки данных.

Сейчас идеи квантовых сумматоров лежат в использовании кубита как обычного бита.

На сегодняшний день, в большинстве источников, которые приводят метод формирования квантового сумматора, речь идёт о создании неких кубитовых регистров для хранения складываемых чисел. То есть, число хранится в двоичной форме в регистре кубит, каждый из которых находится в определённом квантовом состоянии, не запутанном, а измеренном. Значит, регистр такого вида является по своей сути обыкновенным битовым регистром. Такой подход относительно лёгок в реализации, но он не имеет практического смысла и применения, ведь выигрыш в скорости вычисления в этом случае крайне низок. Поэтому, можно сказать, что превращение кубита в обычный бит, не имеет смысла.

Рассмотрим один из примеров такого подхода к созданию квантового сумматора.

Сложение обычных чисел в столбик предполагает два действия над каждым разрядом числа, а именно само сложение чисел и сложение полученной суммы со значением переноса с предыдущего разряда.

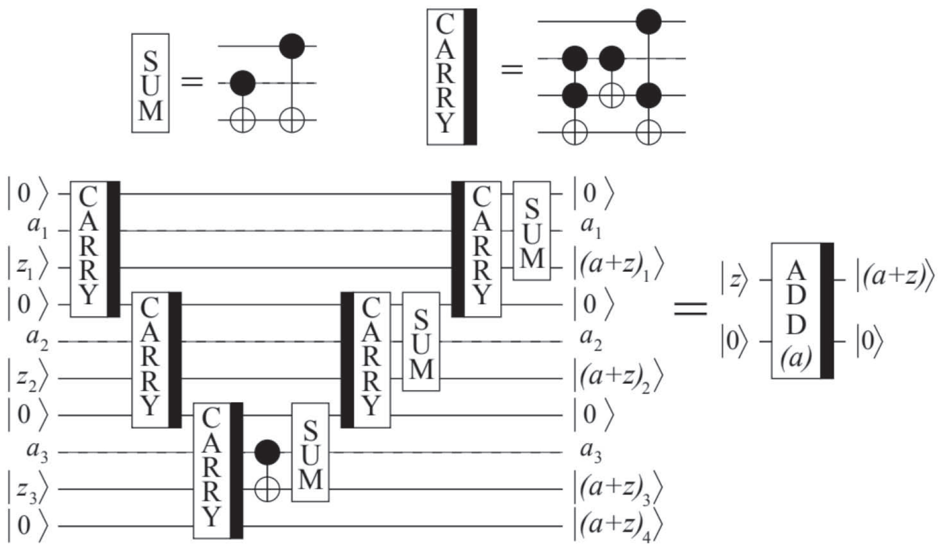
Попробуем составить подобный алгоритм для квантового вычислителя. В этом алгоритме будут использованы семь кубит, два для занесения чисел для суммирования, два для учёта перенесения с перенесения с предыдущих разрядов и три для хранения полученных результатов [5]:

1) На начальном шаге зададим два первых кубита состоянием ноль.
2) Используя логический вентиль CNOT, определим первый перенос, а именно, перенос есть, если конъюнкция первого кубита и единицы равна единице, и переноса нет, если само значение первого кубита равно единице.
3) Далее переведём кубит под номером четыре в состояние $|1\rangle$ а затем заносим в него же результат применения вентиля XOR применённого к первому кубиту.
4) Следом, используем сложный вентиль CCNOT, который является более сложным аналогом вентиля CNOT, так как этот вентиль инвертирует значения контролируемого кубита лишь в случае, когда два управляющих кубита находятся в состоянии $|1\rangle$. Комбинация из этих вентилей позволяет совершить перенос.
5) Последним этапом суммируем кубиты полученных значений и кубиты переносов.

Этот алгоритм на примитивном уровне реализует логический квантовый сумматор для двух бит [5].

Теперь рассмотрим другой алгоритм. Суммирование целых чисел.

Сумматор целых числе это простой процессор, складывающий классические числа с квантовым регистром. Мы рассмотрим два способа сделать это. Первым является адаптация сумматора с ведущими суммами Ведрала, Баренко и Экерта.



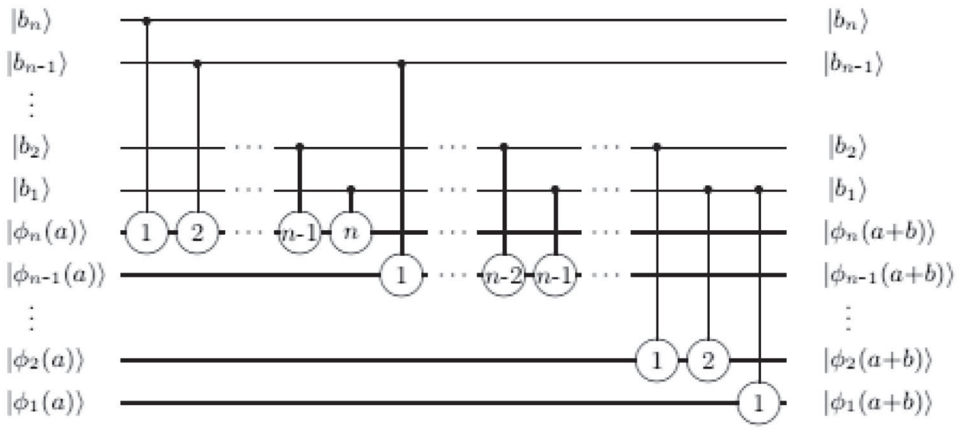
Р и с. 2. Схема адаптации сумматора Ведрала, Баренко и Экерта

F i g. 2. Adaptation scheme of Vedral, Barenko and Ekert adder

Эта схема требует *2n* кубитов, чтобы добавить значение без переполнения, потому что первый кубит на рисунке не нужен. Он использует *O(n)* элементарных вентилей с линейной сложностью. Второй метод использует сумматор Драпера, который мы назовем -сумматор. Ф-сумматор применяет квантовое преобразование Фурье кубитового регистра $|z\rangle$ к квантовому преобразованию Фурье суммы $z+a$, где является классическим значением, встроенным в φ-сумматор.

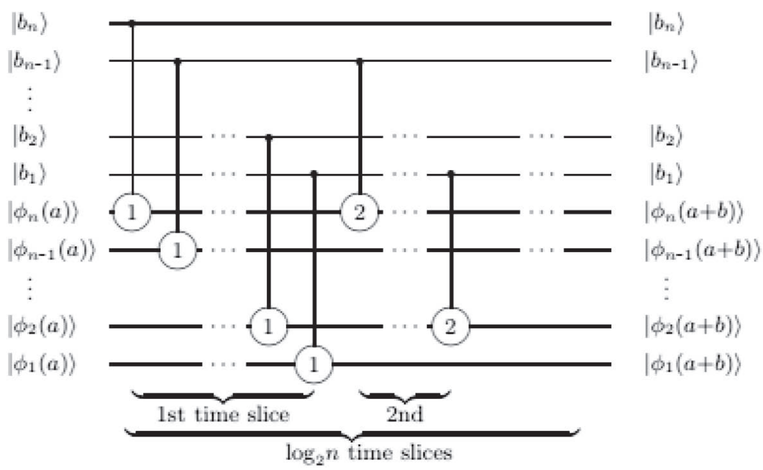
Преимущество этого метода в том, что ему не требуются дополнительные кубиты для переносов. Кроме того, тот факт, что мы используем только добавление классического значения, помогает упростить φ-сумматор. Тем не менее, квантовое преобразование Фурье должно быть применено к квантовому регистру до и после φ-сумматора, поэтому мы в конечном итоге используем более элементарные вентили, чем этот сумматор.

Рассмотрим более общую картину:
Квантовое сложение выполняется с использованием последовательности условных поворотов, напоминающих квантовое преобразование Фурье (такие повороты взаимно коммутативны, но обусловлены внешними битами). С практической точки зрения, так как все контрольные биты известны заранее, добавление квантовых данных к стандартным – это просто вопрос реализации результирующих поворотов. На рисунке 5 представлена схема квантового сложения [11].



Р и с. 3. Схема сложения в квантовом компьютере
F i g. 3. Scheme of addition in a quantum computer

Поскольку реализация квантового сложения очень похоже на КПФ, неудивительно, что существует более эффективная приблизительная реализация квантового сложения, использующая ту же технику, что и АКПФ. Используя аналогичные рассуждения, совершенно ясно, что приближенное квантовое сложение, может быть рассчитано с эффективностью $n * \log_2 n$.
Основное различие между квантовым сложением и квантовым преобразованием Фурье заключается в том, что все операций коммутируют друг с другом в квантовом сложении, тогда как Преобразования Адамара, необходимые для выполнения преобразования Фурье, требуют определенного порядка. Отсюда следует, что если квантовый компьютер может реализовать множество независимых вентилей одновременно, время выполнения будет уменьшаться пропорционально количеству этих операций: «Квантовый компьютер способен вычислять $\frac{n}{2}$ независимых 2-кубитных операций одновременно, и может выполнять квантовое сложение за время эквивалентное $n+1$. Если мы используем метод АКПФ для устранения поворотов ниже определенного порога, квантовое сложение может быть выполнено за $\log_2 n$ операций. Одним из возможных параллельных методов будет выполнение всех поворотов глубины 1 одновременно, а затем всех поворотов глубины 2 и так далее. Понятно, что каждая из этих операций работает на независимых кубитах» [11].



Р и с. 4. Полная схема сумматора
F i g. 4. Complete adder circuit

Цель исследования

Основной целью данной работы является оформление полученных ранее теоретических результатов в виде полноценной модели вычислений, которая позволит разработать новый тип сумматора на основе матричных операций.

Модель предназначена для точной эмуляции операций квантового компьютера, средствами аппаратного обеспечения, более доступного и стабильного, чем квантовый компьютер.

Актуальность данной проблемы подтверждается тем фактом, что в данный момент появилось множество задач, требующих больших объёмов вычислений. А именно обработки реляционных баз данных, которые очень распространены на данный момент времени. Эффективность их обработки можно существенно повысить за счет использования параллелизма, основанного на изоморфизме реляционной алгебры баз данных алгебре многомерных матриц. Для чего и нужно представить операции, производимые квантовым компьютером, в виде, отвечающем алгебре многомерных матриц. Такое представление может послужить отправной точкой для применения алгоритмов параллельной обработки данных с использованием доступных возможностей квантового компьютера для ускорения обработки во множество раз [12-18].

Помимо этого, активная разработка квантовых компьютеров представителями передовых корпораций, говорит о его скорой массовой реализации. Тем не менее, даже имея реализованный массовый квантовый компьютер, логичным кажется использование эмуляторов для создания программ, ведь тестирование неотлаженного процесса на настоящем квантовом процессоре может обойтись компании или индивидуальному разработчику весьма недешёво, в то время как достоверный эмулятор позволит проводить необходимые действия много дешевле и безопаснее. Ведь на данный момент полноценного квантового компьютера нет, и его доступность в обозримом будущем находится под вопросом.

Именно поэтому реализация достоверного эмулятора квантовых процессов является первоочередной задачей разработки квантовых систем.

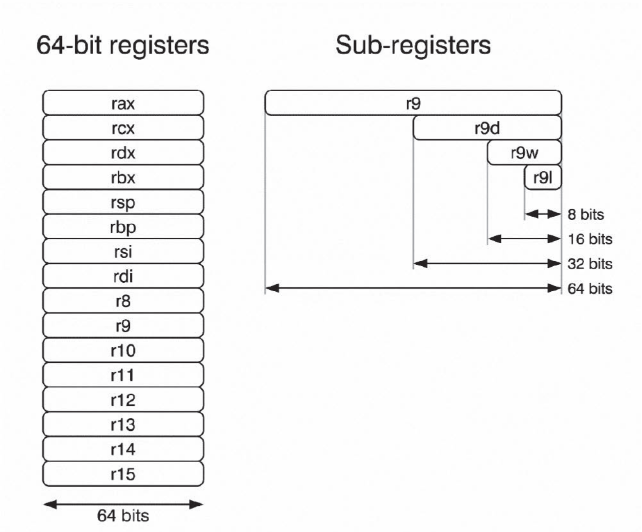
Идеология эмуляции

Рассмотрим основные принципы предложенной схемы эмуляции поведения кубита при помощи регистров процессора.

Регистр процессора — особые ячейки памяти процессора, которые предназначены для хранения промежуточных вычислений или данных, которые используются для работы процессора. Отличительная черта регистров – скорость их работы, которая превышает даже скорость работы кэш-памяти.

Основной идеей построения эмулятора является использование SIMD (single instruction, multiple data) регистров. Из названия следует, что одна и та же процессорная команда может быть применена одновременно к нескольким регистрам, то есть обработать одной командой несколько разных ячеек данных.

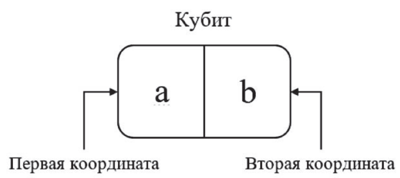
Для решения задачи эмуляции квантовых процессов целесообразно использовать, так называемые, свободные регистры данных. В случае 64-битного процессора Intel(R) Core(TM) i5 шестого поколения – регистры – . Для работы удобнее использовать 128-битные регистры – , так как взаимодействие с ними, сложение или умножение оставит значение в рамках 256-битного регистра, нам не понадобится использовать разбиение значения на два 128-битных регистра, так как запись будет осуществляется в регистр старшей битности.



Р и с. 5. Схема реализации 64-битных регистров
F i g. 5. Implementation scheme for 64-bit registers

На рисунке выше представлен список 64-битных регистров, и схема работы регистров меньшей разрядности. Схема наглядно демонстрирует, что регистры 32-битной, 16-битной и 8-битной разрядностей – являются частью регистра 64-битной разрядности, взаимодействие с ними, это точно такая же запись в регистр, за исключением того, что зона данных не входящая в разряды регистра остаётся неизменной в случае записи данных в регистры меньшей разрядности. Аналогичная схема применяется к 512-битным, 256-битным и 128-битным регистрам.

Сам по себе регистр представляет собой ячейку для хранения двоичных данных, так что его использование накладывает необходимость конвертации вводимых значений в двоичную или шестнадцатеричную систему счисления, так как прямая запись десятичного числа невозможна. Преимущества регистров состоит в возможности параллельного сложения находящихся в них данных, что наталкивает мысль о сходстве регистра и кубита в плане параллельности вычислений.



Р и с. 6. Информационная схема кубита
F i g. 6. Information scheme of a qubit

Кубит содержит в себе два числа, координаты вектора единичной окружности, связанные между собой по правилу $\sqrt{x^2 + y^2} = 1$. Такая связь означает что при изменении одной из координат, вторая изменится пропорционально в силу физических особенностей кубита [19].

Р и с. 7. Подробная схема реализации эмулированного кубита
F i g. 7. Detailed implementation scheme for emulated qubit

Аналогично кубиту, регистр может содержать в себе записанное в него число, в нашем случае одну из координат вектора единичной окружности. Но настоящий физический кубит содержит в себе две координаты вектора, значит нам понадобится и второй регистр, для хранения второй координаты. Смысл связи, заключается в аналогичной формуле и будет соблюдаться путём применения балансировки значений при любой операции с любым из двух связанных регистров. Пара связанных регистров и балансировщик будет называться эмулированным кубитом. Однако, есть ещё одно допущение, которое необходимо сделать. Записываемы в регистр числа, должны быть дробной частью координаты вектора единичной окружности, так как запись в регистр дробных чисел в данной работе не рассматривается. Работа ведётся с масштабированными целыми числами, и число знаков после запятой может храниться в отдельной переменной в высокоуровневой части модели.
Такой подход делает возможным применение описанного выше алгоритма к эмулированным кубитам, и построения на этой основе квантовой векторной модели вычислений.

Компоненты модели

Рассмотрим компоненты предлагаемой модели вычислений.
Основным компонентом является эмулированный кубит структура его определена в пункте 1.
Далее приведены операции над эмулированным кубитом определённые в модели [20-22].

1. Операция записи исходных данных в SIMD регистры. Запись осуществляется путём помещения в компонент памяти регистра числа в интервале (0; 1). Эта операция применяется синхронно к каждому регистру, участвующему в расчёте. Запись в регистры осуществляется путём вызова ассемблерной команды и записи в регистр преобразованного в шестнадцатиричную или двоичную систему счисления координаты вектора единичной окружности. Так как взаимодействие с регистрами возможно лишь на уровне команд ассемблера, для доступа к нему используется библиотека на языке C++. Последовательно трансляцию данных можно описать следующим образом: получение данных с устройства ввода, проверка их соответствия условию , конвертация десятичных чисел в шестнадцатеричный формат, резервирование пары регистров для записи данных, трансляция данных в библиотеку на C++, ввод данных в ассемблерную вставку, и наконец запись в регистр.
2. Операция дублирования регистра. Так как изначальная задача модели эмулировать операции квантового компьютера, определяем операцию дублирования, которая принимает на вход два регистра, оригинальный и регистр дубликат, и присваивает дубликату соответствующее число из оригинала. Дублирование необходимо для досконального исполнения квантового алгоритма, дублирующие кубиты позволят проводить анализ последовательности элементарных состояний после измерения эмулированных кубитов, благодаря этому будет получен номер, с которого последовательность приходит к одному, фиксированному состоянию. Дублирование проводится аналогично, с использованием ассемблерной команды , на вход опирации подаётся пара эмулированных кубит, родитель и наследник, далее имена конкретных регистров трансдируются в библиотеку на C++, преобразуются в фрагменты ассемблерных команд, и исполняются, перенося значения из родительских регистров в наследные.
3. Операция умножения на число. Согласно описанному выше алгоритму, записанные в SIMD регистры данные должны быть пропорционально изменены относительно предыдущего регистра на коэффициент , определяющий шаг приближения. Шагом приближения называется коэффициент, на который отличаются координаты в двух соседних парах эмулированных кубит, он используется для увеличения вероятности получения одного из базисных состояний. Чем выше вероятность, тем чаще это состояние будет получено при измерении, а значит появится возможность получить цепочку базисных состояний необходимого вида. Умножение более сложная операция, и умножение в регистрах реализовано специфическим образом, в частности, оператор является унарным, то есть принимает как аргумент тот регистр, который надо умножить, а вот число, на которое будет произведено умножение хранится в другом, специально отведённом для этого регистре. Так для умножения значения в 32-битном регистре на число, будет использоваться регистр . Таким образом умножение на число определим как занесение в регистр необходимого числа, произвдение умножения одного из двух регистров эмулированного кубита, затем балансировка значения во втором регистре эмулированного кубита. Так же нельзя не отметить, что умножение на число имеет определённое ограничение, произведение обязано оставаться в промежутке (0; 1). Это условие достигается сдвигом вправо полученного значения, до попадания в оригинальные границы регистра, так как по изначальному предположению, мы храним в регистре дробную часть координаты вектора единичной окружности.
4. Операция измерения. Квантовые алгоритмы предполагают полноценное взаимодействие с кубитом, поэтому модель должна содержать операцию, соответствующую измерению кубита. Измерением кубита называют приведение его в одно из базисных состояний 0 или 1. В квантовой механике измерение кубита это коллапс состояния

фотона в одно из состояний 0 или 1, то есть точное определение его спина. В случае эмуляции, данная операция выделяет значение из SIMD регистра и проводит сравнение случайного числа из интервала (0; 1) и числа содержащегося в SIMD регистре. Если случайное число больше содержимого измеряемого регистра, операция возвращает 0, иначе 1.

5. Операция анализа последовательности. При работе квантового алгоритма происходит анализ последовательности элементарных состояний для нахождения номера , с которого система переходит к одному конкретному состоянию. Операция принимает на вход всю доступную последовательность и возвращает номер согласно описанному выше алгоритму (такая-то глава).
6. Операция сложения пары SIMD регистров. Применяется к паре регистров, с указанием коэффициента . Возвращает сумму, вычисленную с применением квантового алгоритма. Эта операция допускает несколько вариантов реализации, в частности, допустимо расширить область ответственности эмулированного кубита, и записать в него, например, два числа меньшей битности, и оперировать с двумя числами в одном эмулированном кубите. Такой подход позволяет увеличить объём обрабатываемых данных при меньшем числе доступных эмулированных кубитов. Но такой подход не позволит построить достоверный эмулятор, ведь в таком случае придётся предположить, что кубит может содержать две пары чисел, что на данный момент невозможно в физическом и механическом плане.
7. Операция произведения пары SIMD регистров. Принимает на вход данные аналогичные операции сложения. Возвращает произведение, вычисленное с применением квантового алгоритма. Произведение пары кубит описывается в главе 1 и производится аналогичным образом на уровне эмулятора. Применить альтернативный подход заполнения регистров данными, то есть поместить, например, две пары чисел в один эмулированный кубит, было бы не совсем эффективным, так как при произведении, в определённых случаях, число может превысить битность доступного регистра, и быть записанным в регистр старшей битности, чем может повредить результат вычисления другой пары чисел [23-25].

Более сложные операции, такие как сложение, умножение и операции (λ, μ)-свернутого произведения многомерных матриц, состоят из описанных операций. Алгоритмически их реализация мало отличается от ранее изложенных.

Заключение

Результатом данной работы может служить заключение, что векторные модели с использованием SIMD регистров способны решать множество ресурсоёмких задач, таких как выполнение операций над многомерными матрицами. Существующие и разрабатываемые квантовые компьютеры отвечают принципиальным требованиям для оптимальной обработки многомерных матриц, а значит использование квантовых принципов для построения модели целесообразно. Применение только базисных состояний кубита – означает полную потерю всего многообразия возможностей использования суперпозиции базисных состояний, недостатки используемых в литературе методов привели к разработке в прошлых работах, и применении в данной, нового подхода к рассмотрению и эмуляции кубита. Предложена модель вычислений, опирающаяся на предыдущие результаты автора в области построения квантовых алгоритмов, открывших возможность, используя примитивные операции, проводить сложение и умножение многомерных матриц, а также реализовать операцию -свернутого произведения.

Для дальнейшего развития построенной модели и реализованной системы достаточно ввести операции алгебры многомерных матриц, что увеличит эффективность вычислений многомерных матриц на несколько порядков.

Главная цель работы, оформление полученных ранее теоретических результатов в виде полноценной модели вычислений, применимой для разработки программных систем и создания матричного сумматора на её основе, достигнута, предложенная модель вычислений предназначена для точной эмуляции операций квантового компьютера, средствами аппаратного обеспечения, более доступного и стабильного, чем квантовый компьютер. Поскольку каждый двумерный вектор, действующий согласно алгебре многомерных матриц, может быть представлен в виде комбинации базисных нормализованных векторов, описанный выше подход может стать ядром для создания процессора на основе алгебры многомерных матриц.

На данный момент полноценного квантового компьютера нет, и его доступность в обозримом будущем находится под вопросом. Тем не менее, даже имея реализованный массовый квантовый компьютер, использование эмуляторов для создания программ позволит проводить необходимые действия много дешевле и безопаснее.

Именно поэтому реализация достоверного эмулятора квантовых процессов является серьёзным достижением на пути к разработке общедоступных квантовых систем.

Расширением модели с единичной окружности на нечто более весомое может быть получено путём построения гомоморфного отображения единичной окружности в единичный круг произвольного радиуса. Такое отображение позволит перенести работу из строго ограниченной области к более открытым множествам чисел, что в свою очередь даст обоснование для применения модели к реальным данным без необходимости их приведения к специфическому, подходящему для модели виду.

Список использованных источников

[1] Feynman R. P. Simulating physics with computers // International Journal of Theoretical Physics. 1982. Vol. 21, issue 6. P. 467-488. doi: https://doi.org/10.1007/BF02650179

[2] Мунерман В. И., Самойлова Т. А. Алгебраический подход к алгоритмизации задач маршрутизации // Системы высокой доступности. 2018. Т. 14, № 5. С. 50-56. doi: https://doi.org/10.18127/j20729472-201805-08

[3] Quantum Algorithms for String Processing / F. Ablayev [и др.] // Mesh Methods for Boundary-Value Problems and Applications. Lecture Notes in Computational Science and Engineering ; ed. by I. B. Badriev, V. Banderov, S. A. Lapin. Vol. 141. Cham : Springer, 2022. doi: https://doi.org/10.1007/978-3-030-87809-2_1

[4] Валиев К. А. Квантовые компьютеры и квантовые вычисления // Успехи физических наук. 2005. Т. 175, № 1. С. 3-39. URL: https://www.elibrary.ru/item.asp?id=8896345 (дата обращения: 14.09.2022).

[5] Draper T. G. Addition on a Quantum Computer // arXiv:quant-ph/0008033. 2000. doi: https://doi.org/10.48550/arXiv.quant-ph/0008033

[6] Бочаров Л. Ю., Мальцев П. П. Основные направления и перспективы развития квантовых информационных технологий за рубежом // Нано- и микросистемная техника. 2007. № 5. С. 2-10. URL: https://www.elibrary.ru/item.asp?id=9594914 (дата обращения: 14.09.2022).

[7] Мунерман В. И. Алгебраический подход к подготовке данных для вывода ассоциативных правил // Системы высокой доступности. 2017. Т.13, № 3. С. 34-37. URL: https://www.elibrary.ru/item.asp?id=30554619 (дата обращения: 14.09.2022).

[8] Алгоритм сложения векторов на квантовом компьютере / Г. М. Григорьева, А. И. Миронов, В. И. Мунерман, В. Ю. Ходченков // Системы компьютерной математики и их приложения. 2019. № 20-1. С. 124-128. URL: https://www.elibrary.ru/item.asp?id=39103168 (дата обращения: 14.09.2022).

[9] Creating a Vector Processor Based on Quantum Computing / G. Grigoryeva, V. Khodchenkov, A. Mironov, V. Munerman // 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). Saint Petersburg and Moscow, Russia : IEEE Computer Society, 2019. P. 1745-1748. doi: https://doi.org/10.1109/EIConRus.2019.8657294

[10] Grigoryeva G., Khodchenkov V. Implementation of Operations of a Quantum Computer in the Language of the Algebra of Multidimensional Matrices // 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus). St. Petersburg, Moscow, Russia : IEEE Computer Society, 2021. P. 2181-2184. doi: https://doi.org/10.1109/ElConRus51938.2021.9396119

[11] Мунерман В. И. Объектно-ориентированная модель массовой обработки данных // Системы высокой доступности. 2011. Т. 7, № 4. С. 72-74. URL: https://www.elibrary.ru/item.asp?id=18049084 (дата обращения: 14.09.2022).

[12] Ambainis A. Understanding quantum algorithms via query complexity // Proceedings of the International Congress of Mathematicians ; ed. by B. Sirakov, P. N. de Souza, M. Viana. Vol. 4. Rio de Janeiro : World Scientific, 2018. P. 3283-3304. URL: https://eta.impa.br/dl/ICM-2018-vol4-ver1-eb.pdf (дата обращения: 14.09.2022).

[13] Захаров В. Н., Мунерман В. И., Самойлова Т. А. Параллельные методы вывода ассоциативных правил в технологиях in-database и in-memory // CEUR Workshop Proceedings. 2017. Т. 2064. С. 219-225. URL: https://ceur-ws.org/Vol-2064/paper26.pdf (дата обращения: 14.09.2022).

[14] Sokolov N. P. Functions of multidimensional matrices and their applications for the solutions of linear systems of partial differential equations // Ukrainian Mathematical Journal. 1970. Vol. 22, issue 6. P. 657-674. doi: https://doi.org/10.1007/BF01086271

[15] Ильин П. Л., Мунерман В. И. Рекурсивное вычисление детерминанта многомерной матрицы // Системы компьютерной математики и их приложения. 2019. № 20-1. С. 162-167. URL: https://elibrary.ru/item.asp?id=39103176 (дата обращения: 14.09.2022).

[16] Fujiki D., Mahlke S., Das R. In-Memory Data Parallel Processor // ACM SIGPLAN Notices. 2018. Vol. 53, no. 2. P. 1-14. doi: https://doi.org/10.1145/3296957.3173171

[17] Liu J.-Sh., Lin J.-Yu., Chung Y.-C. Efficient parallel algorithms for multi-dimensional matrix operations // Proceedings International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN 2000). Dallas, TX, USA : IEEE Computer Society, 2000. P. 224-229. doi: https://doi.org/10.1109/ISPAN.2000.900289

[18] Pan V. Y., Yu Y., Stewart C. Algebraic and Numerical Techniques for the Computation of Matrix Determinants // Computers & Mathematics with Applications. 1997. Vol. 34, issue 1. P. 43-70. doi: https://doi.org/10.1016/S0898-1221(97)00097-7

[19] Григорьева Г. М., Ходченков В. Ю. О возможности построения эмулятора квантового компьютера с использованием XMM регистров // Системы компьютерной математики и их приложения. 2021. № 22. С. 113-116. URL: https://www.elibrary.ru/item.asp?id=46649884 (дата обращения: 14.09.2022).

[20] Мунерман В. И., Мунерман Д. В. О соответствии моделей данных и моделей вычислений // Системы компьютерной математики и их приложения. 2021. № 22. С. 146-152. URL: https://elibrary.ru/item.asp?id=46649891 (дата обращения: 14.09.2022).

[21] Объектно-ориентированный подход к разработке моделей данных / Е. П. Емельченков, В. И. Мунерман, Д. В. Мунерман, Т. А. Самойлова // Современные информационные технологии и ИТ-образование. 2020. Т. 16, № 3. С. 564-574. doi: https://doi.org/10.25559/SITITO.16.202003.564-574

[22] Мунерман В. И. Архитектура программно-аппаратного комплекса для массовой обработки данных на базе многомерно-матричной модели // Системы высокой доступности. 2015. Т. 11, № 2. С. 13-18. URL: https://www.elibrary.ru/item.asp?id=23819273 (дата обращения: 14.09.2022).

[23] Furtak T., Amaral J. N., Niewiadomski R. Using SIMD registers and instructions to enable instruction-level parallelism in sorting algorithms // Proceedings of the nineteenth annual ACM symposium on Parallel algorithms and architectures (SPAA '07). New York, NY, USA : Association for Computing Machinery, 2007. P. 348-357. doi: https://doi.org/10.1145/1248377.1248436

[24] Nuzman D., Rosen I., Zaks A. Auto-vectorization of interleaved data for SIMD // ACM SIGPLAN Notices. 2006. Vol. 41, no. 6. P. 132-143. doi: https://doi.org/10.1145/1133255.1133997

[25] SIMD-scan: ultra fast in-memory table scan using on-chip vector processing units / T. Willhalm [и др.] // Proceedings of the VLDB Endowment. 2009. Vol. 2, no. 1. P. 385-394. doi: https://doi.org/10.14778/1687627.1687671

Поступила 14.09.2022; одобрена после рецензирования 18.11.2022; принята к публикации 29.11.2022.

Об авторах:

Ходченков Валерий Юрьевич, аспирант физико-математического факультета, ФГБОУ ВО «Смоленский государственный университет» (214000, Российская Федерация, г. Смоленск, ул. Пржевальского, д. 4), **ORCID: https://orcid.org/0000-0001-5792-7059**, tansdf@mail.ru

Мунерман Виктор Иосифович, доцент кафедры информатики физико-математического факультета, ФГБОУ ВО «Смоленский государственный университет» (214000, Российская Федерация, г. Смоленск, ул. Пржевальского, д. 4), кандидат технических наук, доцент, **ORCID: https://orcid.org/0000-0002-9628-4049,** vimoon@gmail.com

Все авторы прочитали и одобрили окончательный вариант рукописи.

References

[1] Feynman R.P. Simulating physics with computers. *International Journal of Theoretical Physics*. 1982;21(6):467-488. doi: https://doi.org/10.1007/BF02650179

[2] Munerman V.I., Samoylova T.A. Algebraic approach to algorithmization of routing problems. *Highly available systems*. 2018;14(5):50-56. (In Russ., abstract in Eng.) doi: https://doi.org/10.18127/j20729472-201805-08

[3] Ablayev F., Ablayev M., Khadiev K., Salihova N., Vasiliev A. Quantum Algorithms for String Processing. In: Badriev I.B., Banderov V., Lapin S.A. (eds.) Mesh Methods for Boundary-Value Problems and Applications. Lecture Notes in Computational Science and Engineering. Vol. 141. Cham: Springer; 2022. doi: https://doi.org/10.1007/978-3-030-87809-2_1

[4] Valiev K.A. Quantum computers and quantum computing. *Physics-Uspekhi*. 2005;48(1):1-36. doi: https://doi.org/10.1070/PU2005v048n01ABEH002024

[5] Draper T.G. Addition on a Quantum Computer. arXiv:quant-ph/0008033. 2000. doi: https://doi.org/10.48550/arXiv.quant-ph/0008033

[6] Bocharov L.Yu., Malcev P.P. The main directions development and future of quantum information technologies. *NANO-and MICROSYSTEMS TECHOLOGY*. 2007;(5):2-10. Available at: https://www.elibrary.ru/item.asp?id=9594914 (accessed 14.09.2022). (In Russ., abstract in Eng.)

[7] Munerman V.I. Algebraic approach the data preparation for the associative rules derivation. *Highly available systems*. 2017;13(3):34-37. Available at: https://www.elibrary.ru/item.asp?id=30554619 (accessed 14.09.2022). (In Russ., abstract in Eng.)

[8] Grigorieva G.M., Mironov M.A., Munerman V.I., Khodchenkov V.Yu. Vector addition algorithm in a quantum computer. *Computer Mathematics Systems and Their Applications.* 2019;(20-1):124-128. Available at: https://www.elibrary.ru/item.asp?id=39103168 (accessed 14.09.2022). (In Russ., abstract in Eng.)

[9] Grigoryeva G., Khodchenkov V., Mironov A., Munerman V. Creating a Vector Processor Based on Quantum Computing. In: 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). Saint Petersburg and Moscow, Russia: IEEE Computer Society; 2019. p. 1745-1748. doi: https://doi.org/10.1109/EIConRus.2019.8657294

[10] Grigoryeva G., Khodchenkov V. Implementation of Operations of a Quantum Computer in the Language of the Algebra of Multidimensional Matrices. In: 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus). St. Petersburg, Moscow, Russia: IEEE Computer Society; 2021. p. 2181-2184. doi: https://doi.org/10.1109/ElConRus51938.2021.9396119

[11] Munerman V.I. The object-oriented model of mass data processing. *Highly available systems*. 2011;7(4):72-74. Available at: https://www.elibrary.ru/item.asp?id=18049084 (accessed 14.09.2022). (In Russ., abstract in Eng.)

[12] Ambainis A. Understanding quantum algorithms via query complexity. In: Sirakov B., de Souza P.N., Viana M. (eds.) Proceedings of the International Congress of Mathematicians. Vol. 4. Rio de Janeiro: World Scientific; 2018. p. 3283-3304. Available at: https://eta.impa.br/dl/ICM-2018-vol4-ver1-eb.pdf (accessed 14.09.2022).

[13] Zakharov V.N., Munerman V.I., Samoilova T.A. Parallel methods for deriving associative rules with the usage in-database and in-memory technologies. *CEUR Workshop Proceedings*. 2017;2064:219-225. Available at: https://ceur-ws.org/Vol-2064/paper26.pdf (accessed 14.09.2022). (In Russ., abstract in Eng.)

[14] Sokolov N.P. Functions of multidimensional matrices and their applications for the solutions of linear systems of partial differential equations. *Ukrainian Mathematical Journal*. 1970;22(6):657-674. doi: https://doi.org/10.1007/BF01086271

[15] Iljin P.L., Munerman V.I. Recursive Computation of the Multidimensional Matrix Determinant. *Computer Mathematics Systems and Their Applications.* 2019;(20-1):162-167. Available at: https://elibrary.ru/item.asp?id=39103176 (accessed 14.09.2022). (In Russ., abstract in Eng.)

[16] Fujiki D., Mahlke S., Das R. In-Memory Data Parallel Processor. *ACM SIGPLAN Notices*. 2018;53(2):1-14. doi: https://doi.org/10.1145/3296957.3173171

[17] Liu J.-Sh., Lin J.-Yu., Chung Y.-C. Efficient parallel algorithms for multi-dimensional matrix operations. In: Proceedings International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN 2000). Dallas, TX, USA: IEEE Computer Society; 2000. p. 224-229. doi: https://doi.org/10.1109/ISPAN.2000.900289

[18] Pan V.Y., Yu Y., Stewart C. Algebraic and Numerical Techniques for the Computation of Matrix Determinants. *Computers & Mathematics with Applications*. 1997;34(1):43-70. doi: https://doi.org/10.1016/S0898-1221(97)00097-7

[19] Grigorieva G.M., Khodchenkov V.Yu. On the possibility of building a quantum computer emulator using XMM registers. *Computer Mathematics Systems and Their Applications.* 2021;(22):113-116. Available at: https://www.elibrary.ru/item.asp?id=46649884 (accessed 14.09.2022). (In Russ., abstract in Eng.)

[20] Munerman V.I., Munerman D.V. About the correspondence of data models and calculation models. *Computer Mathematics Systems and Their Applications.* 2021;(22):146-152. Available at: https://elibrary.ru/item.asp?id=46649891 (accessed 14.09.2022). (In Russ., abstract in Eng.)

[21] Emelchenkov Ye.P., Munerman V.I., Munerman D.V., Samoilova T.A. The Object Oriented Approach to Designing Data Models. *Modern Information Technologies and IT-Education*. 2020;16(3):564-574. (In Russ., abstract in Eng.) doi: https://doi.org/10.25559/SITITO.16.202003.564-574

[22] Munerman V.I. Construction of hardware-software complexes architecture to improve massively data processing. *Highly Available Systems.* 2015;11(2):13-18. Available at: https://www.elibrary.ru/item.asp?id=23819273 (accessed 14.09.2022). (In Russ., abstract in Eng.)

[23] Furtak T., Amaral J.N., Niewiadomski R. Using SIMD registers and instructions to enable instruction-level parallelism in sorting algorithms. In: Proceedings of the nineteenth annual ACM symposium on Parallel algorithms and architectures (SPAA '07). New York, NY, USA: Association for Computing Machinery; 2007. p. 348-357. doi: https://doi.org/10.1145/1248377.1248436

[24] Nuzman D., Rosen I., Zaks A. Auto-vectorization of interleaved data for SIMD. *ACM SIGPLAN Notices*. 2006;41(6):132-143. doi: https://doi.org/10.1145/1133255.1133997

[25] Willhalm T., Popovici N., Boshmaf Y., Plattner H., Zeier A., Schaffner J. SIMD-scan: ultra fast in-memory table scan using on-chip vector processing units. *Proceedings of the VLDB Endowment.* 2009;2(1):385-394. doi: https://doi.org/10.14778/1687627.1687671

Submitted 14.09.2022; approved after reviewing 18.11.2022; accepted for publication 29.11.2022.

About the authors:

Valerii Yu. Khodchenkov, Postgraduate Student of the Faculty of Physics and Mathematics, Smolensk State University (4 Przhevalsky St., Smolensk 214000, Russian Federation), **ORCID: https://orcid.org/0000-0001-5792-7059**, tansdf@mail.ru

Victor I. Munerman, Associate Professor of the Chair of Computer Science, Faculty of Physics and Mathematics, Smolensk State University (4 Przhevalsky St., Smolensk 214000, Russian Federation), Cand. Sci. (Eng.), Associate Professor, **ORCID: https://orcid.org/0000-0002-9628-4049,** vimoon@gmail.com

All authors have read and approved the final manuscript.