ЖЭТФ, 2019, том 155, вып. 6, стр. 999–1008

О РЕАЛИЗАЦИИ ПОДСТАНОВОК В ВИДЕ КВАНТОВЫХ СХЕМ БЕЗ ИСПОЛЬЗОВАНИЯ ДОПОЛНИТЕЛЬНЫХ КУБИТОВ

Д. В. Денисенко*

Московский государственный технический университет им. Н. Э. Баумана
105005, Москва, Россия

Поступила в редакцию 23 января 2019 г.,
после переработки 27 января 2019 г.
Принята к публикации 29 января 2019 г.

Представлен способ реализации подстановок (S-боксов) в виде квантовых схем без использования дополнительных кубитов. Представлены квантовые схемы, реализующие подстановки блочного шифра ГОСТ Р 34.12-2015 «Магма» на четырех логических кубитах, а также описания квантовых схем, реализующих S-боксы блочных шифров AES и ГОСТ Р 34.12-2015 «Кузнечик» на восьми логических кубитах — минимальные по количеству логических кубитов квантовые схемы.

DOI: 10.1134/S004445101906004X

1. ВВЕДЕНИЕ

Теория квантовых вычислений стремительно развивается с конца прошлого века. Построен ряд формальных моделей квантовых вычислений, согласно которым квантовая природа объектов, с использованием которых проводятся вычисления, теоретически позволяет более эффективно решать некоторые вычислительные задачи [1–3].

В настоящее время фундаментальные исследования направлены на создание квантовых симуляторов и квантовых процессоров: в 2017 г. группа физиков заявила о создании программируемого 51-кубитного квантового симулятора [4], разработан 53-кубитный симулятор, основанный на ионах в оптических ловушках [5]. В компании IBM успешно испытан прототип 50-кубитного квантового процессора [6], а в декабре 2017 г. опубликована статья [7], в которой представлен проект масштабируемого кремниевого квантового процессора, представляющего собой массив из $24 \times 20 = 480$ кубитов. В январе 2018 г. компания Intel сообщила о создании 49-кубитного сверхпроводящего квантового чипа «Tangle Lake». В марте 2018 г. компания Google объявила о создании 72-кубитного квантового процессора «Bristlecone». В компании надеются, что «Bristlecone» позволит продемонстрировать «квантовое превосходство» [8, 9].

В конце 2018 г. опубликован отчет [10], в котором авторы резюмируют текущее состояние уровня развития квантовых технологий в области квантовых вычислений. В частности, сделан вывод о том, что уровень прогресса в гейтовой модели квантовых вычислений можно отслеживать по ключевым параметрам, определяющим качество квантового процессора: уровню ошибок при выполнении базовых операций с одним и двумя кубитами, глубине взаимосвязи кубитов в одном аппаратном модуле. В отчете отмечено, что «квантовое превосходство» (решение задачи, которое трудно получить на классическом компьютере, независимо от того, имеет ли эта задача практическую полезность) еще не продемонстрировано, при этом опубликованы оценки необходимого количества ресурсов квантового вычислителя для решения некоторых задач, в том числе связанных с реализацией криптографических алгоритмов в виде квантовых схем.

Данная работа является развитием работ [11,12], в которых рассмотрена задача поиска секретного ключа алгоритмов шифрования с помощью квантового алгоритма Гровера. В работе [11] представлена минимальная по количеству логических кубитов квантовая схема, реализующая поиск ключа SDES по одной паре открытого и шифрованного текстов: для реализации функции зашифрования SDES требуется 18 кубитов, так как она представляет собой 8 булевых функций f_i: $V_{10} \times V_8 \to V_1$, $i \in \overline{1,8}$, зави-

* E-mail: DenisenkoDV@bmstu.ru

сящих от 18 булевых переменных. Для поиска ключа SDES алгоритмом Гровера потребуется еще один флаговый кубит, в работе [11] представлена соответствующая квантовая схема на 19 логических кубитах, т. е. показано, что минимальная оценка количества кубитов для поиска ключа SDES квантовым алгоритмом Гровера ($18 + 1 = 19$ кубитов) достижима. Результаты данной работы показывают, что заявленное в [10,13,14] количество логических кубитов, требуемое для реализации криптографических алгоритмов в виде квантовых схем, завышено.

В данной работе представлен алгоритм построения квантовых схем, реализующих подстановки (S-боксы — базовые структурные элементы криптографических алгоритмов шифрования и хэш-функций) без использования дополнительных логических кубитов. Представлены квантовые схемы S-боксов блочного шифра ГОСТ Р 34.12-2015 «Магма», а также распределение квантовых вентилей в квантовых схемах, реализующих S-боксы ГОСТ Р 34.12-2015 «Кузнечик» и AES без использования дополнительных кубитов. Для реализации S-боксов блочного шифра ГОСТ Р 34.12-2015 «Магма» достаточно 4 логических кубита, а для реализации S-боксов ГОСТ Р 34.12-2015 «Кузнечик» и AES достаточно 8 логических кубитов.

Полученные результаты позволяют сделать новый вывод о минимальных по количеству логических кубитов квантовых схемах, реализующих криптографические алгоритмы AES [15], ГОСТ Р 34.12-2015 [16], SHA-2 [17], SHA-3 [18] и ГОСТ Р 34.11-2012 [19] (см. табл. 3, 4 ниже). Представленные на рис. 1–8 квантовые схемы могут быть полезны при экспериментальном тестировании и верификации работы физических реализаций квантовых вычислительных устройств.

2. ПОСТРОЕНИЕ КВАНТОВОЙ СХЕМЫ, РЕАЛИЗУЮЩЕЙ ПРОИЗВОЛЬНУЮ ПОДСТАНОВКУ $s \in S(V_n)$ БЕЗ ИСПОЛЬЗОВАНИЯ ДОПОЛНИТЕЛЬНЫХ КУБИТОВ

Алгоритм построения квантовой схемы, соответствующей произвольному унитарному оператору, описан в работе [20], разд. 4.5.

Определение 1. Пусть $N = 2^n$, $n \in \mathbb{N}$, и e_1, $e_2, \ldots, e_N$ — базис векторного пространства $L_{\mathbb{C}^N}$ над полем комплексных чисел $\mathbb{C}$. Унитарные матрицы $U \in \mathbb{C}_{2^n,2^n}$, нетривиально действующие не более чем на два базисных вектора $e_1, e_2, \ldots, e_N$, называются двухуровневыми унитарными матрицами.

Требуется для унитарного оператора $U \in \mathbb{C}_{2^n,2^n}$, соответствующего выбранной подстановке $s \in S(V_n)$, найти представление

$$U_{N-1}U_{N-2}\ldots U_1U = I.$$

Чтобы построить квантовую схему, реализующую заданную подстановку $s \in S(V_n)$ без использования дополнительных кубитов, достаточно выполнить следующие действия.

1. Найти унитарную матрицу $U \in \mathbb{C}_{2^n,2^n}$, соответствующую выбранной подстановке $s \in S(V_n)$. Матрица U — подстановочная матрица в пространстве V_{2^n}, т. е. состоит только из нулей и единиц.

2. Представить матрицу $U \in \mathbb{C}_{2^n,2^n}$ в виде произведения двухуровневых матриц

$$U = V_1 \ldots V_k,$$

где $V_i \in \mathbb{C}_{2^n,2^n}$ — двухуровневые унитарные матрицы (см. [20], разд. 4.5.1), $i \in \overline{1,k}$, $k \leq (2^n - 1) + (2^n - 2) + \ldots + 1 = \frac{2^n(2^n-1)}{2}$ (сумма арифметической прогрессии). Любая унитарная матрица, действующая на n кубитов, может быть разложена в произведение не более чем $2^{n-1}(2^n - 1)$ двухуровневых унитарных матриц.

3. В разд. 4.5.2 работы [20] показано, что любую двухуровневую унитарную матрицу можно легко реализовать с помощью однокубитовых операторов и CNOT. Таким образом, построив квантовые схемы, реализующие двухуровневые унитарные матрицы $V_1 \ldots V_k$, получим квантовую схему, реализующую оператор U, соответствующий выбранной подстановке $s \in S(V_n)$.

Заметим, что двухуровневые унитарные матрицы представляются в виде квантовых схем неоднозначно. При реализации матриц V_i и V_{i+1}, $i \in \overline{1,k}$, возможно последовательное применение взаимно обратных квантовых вентилей. Перебрав все возможные реализации двухуровневых унитарных матриц $V_1 \ldots V_k$, можно найти минимальную квантовую схему, реализующую оператор $U = V_1 \ldots V_k$, который соответствует подстановке $s \in S(V_n)$.

Построение квантовой схемы по произвольному унитарному оператору $U \in \mathbb{C}_{2^n,2^n}$ реализовано в квантовом симуляторе Quipper (см. [21–23]), причем в реализации не учитывается возможность оптимизации квантовых схем путем удаления последовательных взаимно обратных квантовых вентилей.

В разд. 3 представлены оптимизированные квантовые схемы, реализующие S-боксы ГОСТ Р 34.12-2015 «Магма», распределения квантовых вентилей в оптимизированных квантовых схемах,

Таблица 1. Распределение количества квантовых вентилей в квантовых схемах на рис. 1–8

Подстановка	Распределение количества квантовых вентилей	Подстановка	Распределение количества квантовых вентилей
π_0	1: «X, arity 1» controls 0+3 4: «X, arity 1» controls 1+2 4: «X, arity 1» controls 2+1 4: «X, arity 1» controls 3 20: «not, arity 1» controls 1 Total gates: 33	π_1	1: «X, arity 1» controls 0+3 2: «X, arity 1» controls 1+2 4: «X, arity 1» controls 2+1 2: «X, arity 1» controls 3 20: «not, arity 1» controls 1 Total gates: 29
π_2	1: «X, arity 1» controls 0+3 4: «X, arity 1» controls 1+2 5: «X, arity 1» controls 2+1 3: «X, arity 1» controls 3 24: «not, arity 1» controls 1 Total gates: 37	π_3	1: «X, arity 1» controls 0+3 3: «X, arity 1» controls 1+2 5: «X, arity 1» controls 2+1 4: «X, arity 1» controls 3 16: «not, arity 1» controls 1 Total gates: 29
π_4	1: «X, arity 1» controls 0+3 4: «X, arity 1» controls 1+2 3: «X, arity 1» controls 2+1 3: «X, arity 1» controls 3 20: «not, arity 1» controls 1 Total gates: 31	π_5	1: «X, arity 1» controls 0+3 4: «X, arity 1» controls 1+2 6: «X, arity 1» controls 2+1 2: «X, arity 1» controls 3 22: «not, arity 1» controls 1 Total gates: 35
π_6	1: «X, arity 1» controls 0+3 3: «X, arity 1» controls 1+2 6: «X, arity 1» controls 2+1 3: «X, arity 1» controls 3 18: «not, arity 1» controls 1 Total gates: 31	π_7	1: «X, arity 1» controls 0+3 4: «X, arity 1» controls 1+2 4: «X, arity 1» controls 2+1 4: «X, arity 1» controls 3 18: «not, arity 1» controls 1 Total gates: 31

Таблица 2. Распределение количества квантовых вентилей в квантовых схемах, реализующих S-боксы ГОСТ Р 34.12-2015 «Кузнечик» и AES на 8 логических кубитах

$\pi_{Kuznechik}$	1: «X, arity 1» controls 0+7 8: «X, arity 1» controls 1+6 28: «X, arity 1» controls 2+5 56: «X, arity 1» controls 3+4 70: «X, arity 1» controls 4+3 54: «X, arity 1» controls 5+2 27: «X, arity 1» controls 6+1 6: «X, arity 1», controls 7 1020: «not, arity 1», controls 1 Total gates: 1270	π_{AES}	1: «X, arity 1» controls 0+7 8: «X, arity 1» controls 1+6 28: «X, arity 1» controls 2+5 56: «X, arity 1» controls 3+4 70: «X, arity 1» controls 4+3 55: «X, arity 1» controls 5+2 26: «X, arity 1» controls 6+1 7: «X, arity 1», controls 7 958: «not, arity 1», controls 1 Total gates: 1209

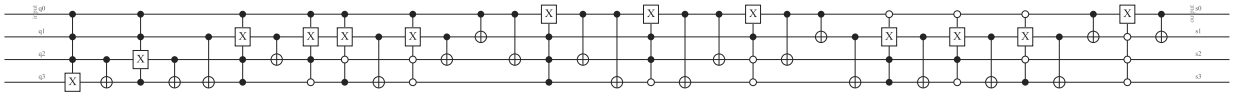


Рис. 1. Квантовая схема, реализующая подстановку $\pi_0 = (12, 4, 6, 2, 10, 5, 11, 9, 14, 8, 13, 7, 0, 3, 15, 1)$

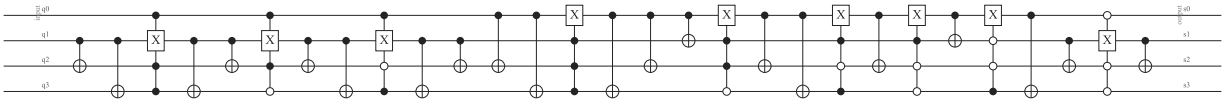


Рис. 2. Квантовая схема, реализующая подстановку $\pi_1 = (6, 8, 2, 3, 9, 10, 5, 12, 1, 14, 4, 7, 11, 13, 0, 15)$

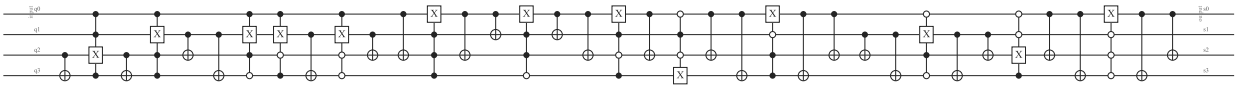


Рис. 3. Квантовая схема, реализующая подстановку $\pi_2 = (11, 3, 5, 8, 2, 15, 10, 13, 14, 1, 7, 4, 12, 9, 6, 0)$

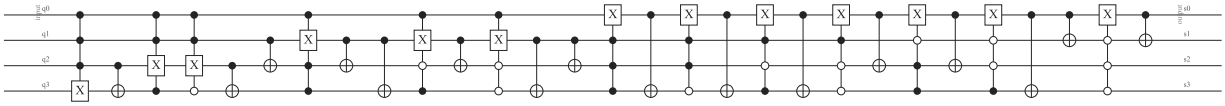


Рис. 4. Квантовая схема, реализующая подстановку $\pi_3 = (12, 8, 2, 1, 13, 4, 15, 6, 7, 0, 10, 5, 3, 14, 9, 11)$

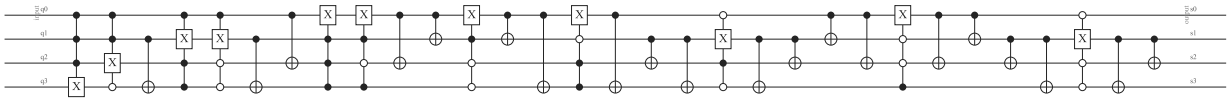


Рис. 5. Квантовая схема, реализующая подстановку $\pi_4 = (7, 15, 5, 10, 8, 1, 6, 13, 0, 9, 3, 14, 11, 4, 2, 12)$

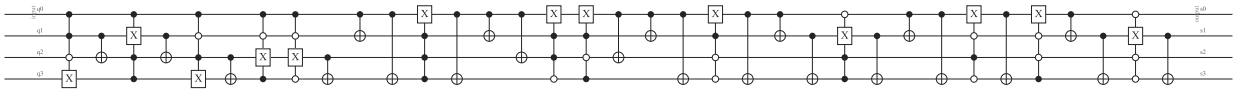


Рис. 6. Квантовая схема, реализующая подстановку $\pi_5 = (5, 13, 15, 6, 9, 2, 12, 10, 11, 7, 8, 1, 4, 3, 14, 0)$

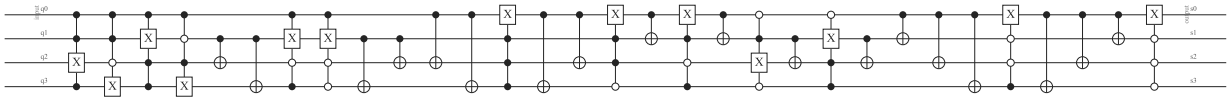


Рис. 7. Квантовая схема, реализующая подстановку $\pi_6 = (8, 14, 2, 5, 6, 9, 1, 12, 15, 4, 11, 0, 13, 10, 3, 7)$

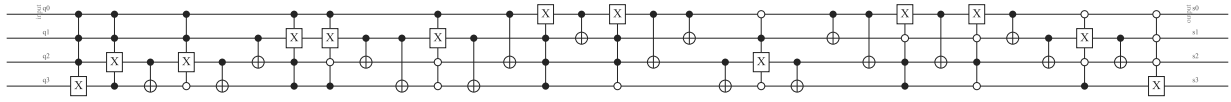


Рис. 8. Квантовая схема, реализующая подстановку $\pi_7 = (1, 7, 14, 13, 0, 5, 8, 3, 4, 15, 10, 6, 9, 12, 11, 2)$

реализующих S-боксы ГОСТ Р 34.12-2015 и AES. В Приложении А представлены соответствующие программные реализации для квантового симулятора Quipper. В Приложении В подробно разобран пример построения квантовой схемы, реализующей подстановку $s \in S(V_4)$.

3. КВАНТОВЫЕ СХЕМЫ S-БОКСОВ ГОСТ Р 34.12-2015 И AES

На рис. 1–8 представлены квантовые схемы, реализующие S-боксы ГОСТ Р 34.12-2015 «Магма». Распределение количества квантовых вентилей, необходимых для реализации S-боксов ГОСТ Р 34.12-2015 «Магма», представлено в табл. 1. Распределение количества квантовых вентилей, необходимых для реализации S-боксов ГОСТ Р 34.12-2015 «Кузнечик» и AES, представлено в табл. 2.

Отметим, что в работах [13, 14] для реализации S-бокса AES требуется 40 логических кубитов, 3584 T-вентилей и 4569 вентилей Клиффорда (множество вентилей {T, H, S, CNOT}, см. [20, 24]). В работе [13] описан способ построения квантовой схемы, реализующей S-бокс AES на 9 логических кубитах, но при этом потребуется 9695 T-вентилей и 12631 вентилей Клиффорда. В табл. 2 показано, что для реализации S-бокса AES на 8 логических кубитах требуется 958 вентилей CNOT и 251 обобщенный вентиль $\mathrm{CNOT}(C|t)$ (см. [20, 25], управляемый кубит с номером t контролируется множеством кубитов C). Обобщенные вентили $\mathrm{CNOT}(C|t)$ можно реализовать без использования дополнительных кубитов (см. [20], с. 184), но при этом для их реализации может потребоваться значительное количество T-вентилей и вентилей Клиффорда.

4. ЗАКЛЮЧЕНИЕ

Полученные результаты позволяют сделать вывод о том, что для реализации блочных шифров $E : V_n \times V_m \to V_m$ с ключом $K \in V_n$ и блоками открытого и шифрованного текстов $P, C \in V_m$, в структуре которых отсутствует операция сложения по модулю 2^t, $t > 1$, достаточно $n + m$ логических кубитов.

При наличии возможности применения квантового преобразования Фурье операцию модульного сложения можно реализовать без использования вспомогательных кубитов [26].

Если же в структуре блочного шифра применяется модульное сложение, а возможность применения квантового преобразования Фурье отсутствует, то для реализации операции $P \boxplus K \mod 2^n$ может потребоваться один дополнительный кубит и $\frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{25}{6}n + 8$ квантовых вентилей (см. [27]), а для реализации всего блочного шифра E: $V_n \times V_m \to V_m$ потребуется $n + m + 1$ логических кубитов.

Таблица 3. Минимальное количество логических кубитов, требуемое для реализации алгоритмов ГОСТ Р 34.12-2015 и AES

Алгоритм	Минимальное количество кубитов для реализации в виде квантовой схемы
ГОСТ Р 34.12-2015 «Магма»	$256 + 64 = 320$
ГОСТ Р 34.12-2015 «Кузнечик»	$256 + 128 = 384$
AES-128	$128 + 128 = 256$
AES-192	$192 + 128 = 320$
AES-256	$256 + 128 = 384$

Таблица 4. Минимальное количество логических кубитов, требуемое для реализации алгоритмов SHA-2, SHA-3 и ГОСТ Р 34.11-2012

Алгоритм	Минимальное количество кубитов для реализации в виде квантовой схемы
SHA-2 (224, 256)	512
SHA-2 (384, 512)	1024
SHA-3	1600
ГОСТ Р 34.11-2012	1024

Минимальное количество логических кубитов, требуемое для реализации алгоритмов шифрования ГОСТ Р 34.12-2015 и AES, представлено в табл. 3, и оно значительно меньше, чем значения, приведенные в работах [10, 13, 14].

Кроме того, можно сделать вывод о том, что минимальное количество логических кубитов, необходимое для реализации хэш-функций в виде квантовых схем, определяется максимальной длиной внутреннего состояния соответствующей хэш-функции. В табл. 4 представлены оценки минимального количества кубитов, достаточного для реализации хэш-функций SHA-2, SHA-3 и ГОСТ Р 34.11-2012 в виде квантовых схем.

ПРИЛОЖЕНИЕ А

Построение квантовых схем S-боксов с помощью квантового симулятора Quipper

Приведем программные реализации, с помощью которых можно построить неоптимальные квантовые схемы, реализующие S-боксы ГОСТ Р 34.12-2015 без использования дополнительных кубитов. Для того чтобы затем получить оптимальные квантовые схемы, достаточно будет убрать пары квантовых вентилей, реализующие тождественные преобразования.

1. MAGMA.hs — строим квантовые схемы S-боксов ГОСТ Р 34.12-2015 «Магма».

2. KUZNECHIK.hs — строим квантовую схему S-бокса ГОСТ Р 34.12-2015 «Кузнечик».

Приведем содержимое указанных файлов.

MAGMA.HS

```
import Quipper
import QuipperLib.Synthesis
import Quipper.Printing
import QuipperLib.Simulation
import Quipper.QData
import Quantum.Synthesis.Ring
import Quantum.Synthesis.Matrix
import Quantum.Synthesis.MultiQubitSynthesis
import System.Random
-------------------------------------------------------------------------
--s = [12,4,6,2,10,5,11,9,14,8,13,7,0,3,15,1]
--s(x) = y;  x.U = y;
opertor :: [Qubit] -> Circ [Qubit]
opertor (input)  = (exact_synthesis op) (input)
  where
    op :: Matrix  (Ten_and Six) (Ten_and Six) DRComplex--DOmega
    op = matrix
     [[0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1],
     [0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0],
     [0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0],
     [0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0],
     [0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0],
     [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0],
     [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0],
     [0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0]]

sub :: ([Qubit]) -> Circ ([Qubit])
sub (input) = do
  let [q0, q1, q2, q3] = input
  comment_with_label "input"
    (q0, q1, q2, q3)
    ("s0", "s1", "s2", "s3")
  [q0, q1, q2, q3] <- opertor ([q0, q1, q2, q3])
  return ([q0, q1, q2, q3])

test1_circuit :: IO ()
test1_circuit = do
  putStrLn "Substitution functionality test:"
  --print_generic GateCount sub  (replicate 4 qubit)
```

```
39   --print_generic Preview sub (replicate 4 qubit)
40   --print_generic ASCII sub (replicate 4 qubit)
41   print_generic PDF sub (replicate 4 qubit)
42
43 test1_exec :: IO ()
44 test1_exec = do
45   putStrLn "Substitution functionality test:"
46   print_generic GateCount sub  (replicate 4 qubit)
47   g <- newStdGen
48   print $ run_generic g (0.0 :: Double) sub ([False,False,False,False])
49   print $ run_generic g (0.0 :: Double) sub ([False,False,False,True])
50   print $ run_generic g (0.0 :: Double) sub ([False,False,True,False])
51   print $ run_generic g (0.0 :: Double) sub ([False,False,True,True])
52   print $ run_generic g (0.0 :: Double) sub ([False,True,False,False])
53   print $ run_generic g (0.0 :: Double) sub ([False,True,False,True])
54   print $ run_generic g (0.0 :: Double) sub ([False,True,True,False])
55   print $ run_generic g (0.0 :: Double) sub ([False,True,True,True])
56   print $ run_generic g (0.0 :: Double) sub ([True,False,False,False])
57   print $ run_generic g (0.0 :: Double) sub ([True,False,False,True])
58   print $ run_generic g (0.0 :: Double) sub ([True,False,True,False])
59   print $ run_generic g (0.0 :: Double) sub ([True,False,True,True])
60   print $ run_generic g (0.0 :: Double) sub ([True,True,False,False])
61   print $ run_generic g (0.0 :: Double) sub ([True,True,False,True])
62   print $ run_generic g (0.0 :: Double) sub ([True,True,True,False])
63   print $ run_generic g (0.0 :: Double) sub ([True,True,True,True])
64 -- | Run any of the above main functions:
65 main = do
66   test1_circuit
67   test1_exec
```

KUZNECHIK.HS

```
 1 import Quipper
 2 import QuipperLib.Synthesis
 3 import Quipper.Printing
 4 import QuipperLib.Simulation
 5 import Quipper.QData
 6 import Quantum.Synthesis.Ring
 7 import Quantum.Synthesis.Matrix
 8 import Quantum.Synthesis.MultiQubitSynthesis
 9 import Data.Time
10 import System.Random
11 ------------------------------------------------------
12 main = do
13   g <- newStdGen
14   startTime <- getCurrentTime
15   let scheme = (exact_synthesis op)
16   --output of the quantum scheme to a text file for further optimization
17   print_generic ASCII scheme [qubit, qubit, qubit, qubit, qubit, qubit, qubit, qubit]
18   stopTime <- getCurrentTime
```

```
19  let deltaTime = show $ diffUTCTime stopTime startTime
20  putStrLn deltaTime
21  print_generic GateCount scheme [qubit, qubit, qubit, qubit, qubit, qubit, qubit, qubit]
22  print $ run_generic g (0.0 :: Double) scheme ([False,False,False,False,False,False,False,
     False])
23  print $ run_generic g (0.0 :: Double) scheme ([False,False,False,False,False,False,False,
     True])
24  print $ run_generic g (0.0 :: Double) scheme ([True,True,True,True,True,True,True,True])
25  where
26    op :: Matrix  (Times (Times Four Four) (Times Four Four)) (Times (Times Four Four)
      (Times Four Four))  DRComplex --QRComplex--DOmega  --DRComplex
27    op = matrix [[ here insert the appropriate unitary matrix of size 256x256 ]]
```

ПРИЛОЖЕНИЕ В

Построение квантовой схемы, реализующей подстановку $\pi_1 = (6, 8, 2, 3, 9, 10, 5, 12, 1, 14, 4, 7, 11, 13, 0, 15)$

Построим квантовую схему, реализующую подстановку $\pi_1 = (6, 8, 2, 3, 9, 10, 5, 12, 1, 14, 4, 7, 11, 13, 0, 15)$. Подстановка $\pi_1 \in S(V_4)$. Обозначим $y = \pi_1(x)$, $x, y \in V_4$. Состояния $|x\rangle$, $|y\rangle$ представляют собой векторы-столбцы из $L_{\mathbb{C}^{2^4}}$, действие оператора $U|x\rangle = |y\rangle$ представляет собой умножение вектора-столбца $|x\rangle$ на матрицу $U \in \mathbb{C}_{2^4,2^4}$.

1. Подстановке π_1 соответствует унитарная матрица

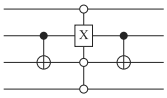
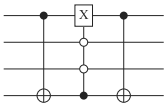
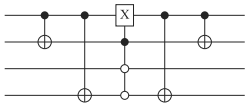
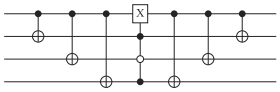
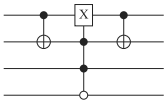
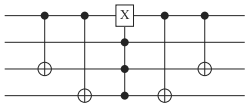
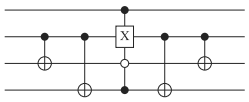
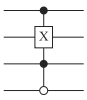
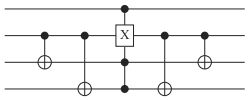
$$U_{\pi_1} = \begin{pmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{pmatrix}.$$

2. Матрицу U_{π_1} можно представить в виде произведения двухуровневых унитарных матриц:

$$U_{\pi_1} = V_1 \cdot V_2 \cdot V_3 \cdot V_4 \cdot V_5 \cdot V_6 \cdot V_7 \cdot V_8 \cdot V_9.$$

В табл. 5 приведены двухуровневые матрицы $V_1, \ldots, V_9$, участвующие в разложении U_{π_1}, состояния s и t, на которые двухуровневые матрицы действуют нетривиально, и квантовые схемы, реализующие двухуровневые матрицы $V_1, \ldots, V_9$. Матрицы записаны в виде списка строк, каждая строка пред-

Таблица 5. Представление матрицы U_{π_1} в виде произведения двухуровневых матриц

$V_1 = \{200, 4000, 2000, 1000, 800, 400, 8000, 100, 80, 40, 20, 10, 8, 4, 2, 1\}$	$s = \|0110\rangle$ $t = \|0000\rangle$	
$V_2 = \{8000, 80, 2000, 1000, 800, 400, 200, 100, 4000, 40, 20, 10, 8, 4, 2, 1\}$	$s = \|0001\rangle$ $t = \|1000\rangle$	
$V_3 = \{8000, 4000, 2000, 1000, 40, 400, 200, 100, 80, 800, 20, 10, 8, 4, 2, 1\}$	$s = \|0100\rangle$ $t = \|1001\rangle$	
$V_4 = \{8000, 4000, 2000, 1000, 800, 20, 200, 100, 80, 40, 400, 10, 8, 4, 2, 1\}$	$s = \|0101\rangle$ $t = \|1010\rangle$	
$V_5 = \{8000, 4000, 2000, 1000, 800, 400, 20, 100, 80, 40, 200, 10, 8, 4, 2, 1\}$	$s = \|0110\rangle$ $t = \|1010\rangle$	
$V_6 = \{8000, 4000, 2000, 1000, 800, 400, 200, 8, 80, 40, 20, 10, 100, 4, 2, 1\}$	$s = \|0111\rangle$ $t = \|1100\rangle$	
$V_7 = \{8000, 4000, 2000, 1000, 800, 400, 200, 100, 80, 2, 20, 10, 8, 4, 40, 1\}$	$s = \|1001\rangle$ $t = \|1110\rangle$	
$V_8 = \{8000, 4000, 2000, 1000, 800, 400, 200, 100, 80, 40, 2, 10, 8, 4, 20, 1\}$	$s = \|1010\rangle$ $t = \|1110\rangle$	
$V_9 = \{8000, 4000, 2000, 1000, 800, 400, 200, 100, 80, 40, 20, 8, 10, 4, 2, 1\}$	$s = \|1011\rangle$ $t = \|1100\rangle$	

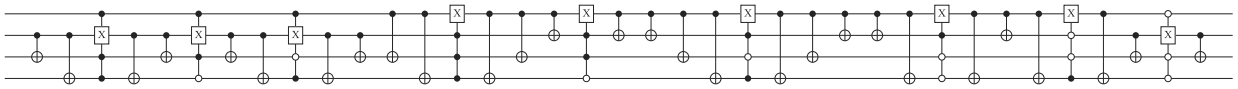


Рис. 9. Квантовая схема, реализующая подстановку $\pi_1 = (6, 8, 2, 3, 9, 10, 5, 12, 1, 14, 4, 7, 11, 13, 0, 15)$

ставляет собой вектор $v_i \in V_{16}$, $||v_i|| = 1$, $i \in \overline{1,16}$ и записана в шестнадцатеричной системе счисления.

Поскольку $|y\rangle = U|x\rangle$, $|y\rangle = V_1 \cdot \ldots \cdot (V_8 \cdot (V_9 \cdot |x\rangle))$, квантовая схема, соответствующая U_{π_1}, имеет вид приведенной на рис. 9.

После оптимизации квантовой схемы на рис. 9 получим квантовую схему на рис. 2.

ЛИТЕРАТУРА

1. P. W. Shor, SIAM J. Comput. **26**, 1484 (1997).

2. L. K. Grover, in *Proc. of STOC 1996*, ed. by G. L. Miller, ACM (1996), pp. 212–219.

3. D. R. Simon, SIAM J. Comput. **26**, 1474 (1997).

4. H. Bernien, S. Schwartz, A. Keesling, H. Levine, and A. Omran, Nature **551**, 579 (2017), DOI:10.1038/nature24622; arXiv:1707.04344.

5. J. Zhang, G. Pagano, P. W. Hess, A. Kyprianidis, and P. Becker, Nature **551**, 601 (2017), DOI:10.1038/nature24654; arXiv:1708.01044.

6. https://www.ibm.com/blogs/research/2017/11/the-future-is-quantum/.

7. M. Veldhorst, H. G. J. Eenink, C. H. Yang, and A. S. Dzurak, Nature Comm. **8**, 1766 (2017), DOI:10.1038/s41467-017-01905-6; https://doi.org/10.1038/s41467-017-01905-6.

8. C. S. Calude and E. Calude, arXiv:1712.01356v1.

9. J. Kelly, *A Preview of Bristlecone, Google's New Quantum Processor*, Quantum AI Lab, 05.03.2018, https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html.

10. *Quantum Computing*: *Progress and Prospects*, National Academies of Sciences, Engineering, and Medicine (2018), National Acad. Press, Washington, DC, https://doi.org/10.17226/25196.

11. Д. В. Денисенко, М. В. Никитенкова, ЖЭТФ **155**, 32 (2019).

12. Д. В. Денисенко, Г. Б. Маршалко, М. В. Никитенкова, В. И. Рудской, В. А. Шишкин, ЖЭТФ **155**, 645 (2019).

13. M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt, arXiv:1512.04965v1.

14. P. Kim, D. Han, and K. C. Jeong, Quant. Inf. Process. **17**, 339 (2018), https://doi.org/10.1007/s11128-018-2107-3.

15. *NIST, Specification for the Advanced Encryption Standart* (*AES*), Federal Inf. Process. Stand. Publ. 197 (2001).

16. ГОСТ Р 34.12-2015, *Информационные технологии. Защита информации. Блочные шифры.*

17. *NIST*: *Secure Hash Standard* (*SHS*), FIPS PUB 180-4 (2015); http://dx.doi.org/10.6028/NIST.FIPS.180-4.

18. *SHA-3 Standard*: *Permutation-Based Hash and Extendable-Output Functions*, Federal Inf. Process. Stand. (NIST FIPS) 202 (2015); https://doi.org/10.6028/NIST.FIPS.202.

19. ГОСТ Р 34.11–2012, *Информационная технология. Криптографическая защита информации. Функция хэширования.*

20. M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge Univ. Press, Cambridge (2010).

21. *Квантовый симулятор Quipper*, http://www.mathstat.dal.ca/-selinger/quipper/.

22. A. S. Green, P. L. Lumsdaine, N. J. Ross, P. Selinger, and B. Valiron, arXiv:1304.5485v1.

23. S. Siddiqui, M. J. Islam, and O. Shehab, arXiv:1406.4481v2 [quant-ph].

24. G. H. Low, V. Kliuchnikov, and L. Schaeffer, https://arxiv.org/abs/1812.00954v1.

25. A. Younes and J. Miller, arXiv:quant-ph/0304099v1.

26. T. G. Draper, https://arxiv.org/abs/quant-ph/0008033.

27. P. Kaye, arxiv.org/abs/quant-ph/0408173.