УДК 519.714.5, 003.26
DOI: 10.25559/SITITO.15.201903.541—552

Обратимые вычисления: обзор проблемы и новые результаты (отказоустойчивость и криптография)

С. И. Гуров[1*], А. Е. Жуков[2], Д. В. Закаблуков[3], Г. В. Кормаков[1]
[1] Московский государственный университет имени М.В. Ломоносова, г. Москва, Россия
119991, Россия, г. Москва, ГСП-1, Ленинские горы, д. 1
[*] sgur@cs.msu.ru
[2] Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет), г. Москва, Россия
105005, Россия, г. Москва, 2-я Бауманская ул., д. 5, стр. 1
[3] ООО «Алгоритмы и данные», г. Москва, Россия
117218, Россия, г. Москва, ул. Дмитрия Ульянова, д. 42, стр. 1

Аннотация

В работе рассмотрены основные положения обратимости как новой парадигмы развития вычислительной техники. Первые разделы носят обзорный характер. Показана неизбежность т. н. «теплового проклятия» при сохранении традиционной парадигмы создания средств вычислительной техники (ВТ). Изложены основы обратимой логики, рассмотрены основные обратимые логические элементы и модели обратимых вычислений, в том числе обратимые клеточные автоматы. Кратко рассмотрены обратимые языки программирования. Во второй части затронуты основные вопросы логического синтеза схем из обратимых элементов и физическая реализация обратимой схемотехники. Кратко описана проблематика синтеза отказоустойчивых схем в парадигме обратимой схемотехники. Предлагается техника синтеза сбоеустойчивых обратимых элементов в хэммингово пространстве и описываются некоторые такие схемы. Далее рассматривается проблематика применения схем из обратимых логических элементов в криптографии. Описывается предлагаемая общая схема создания обратимых схем с «уборкой мусора», предназначенных для криптографических применений.

Ключевые слова: обратимая логика, обратимые логические элементы, отказоустойчивые схемы, хэммингово пространство, защита информации, обратимые схемы с «уборкой мусора».

Для цитирования: *Гуров С. И., Жуков А. Е., Закаблуков Д. В., Кормаков Г. В.* Обратимые вычисления: обзор проблемы и новые результаты (отказоустойчивость и криптография) // Современные информационные технологии и ИТ-образование. 2019. Т. 15, № 3. С. 541-552. DOI: 10.25559/SITITO.15.201903.541-552

Reversible Computing: Review of the Problem and New Results (Fault Tolerance and Cryptography)

S. I. Gurov[1*], A. E. Zhukov[2], D. V. Zakablukov[2], G. V. Kormakov[1]
[1] Lomonosov Moscow State University, Moscow, Russia
1, Leninskie gory, Moscow 119991, Russia
* sgur@cs.msu.ru
[2] Bauman Moscow State Technical University, Moscow, Russia
5/1 2-nd Baumanskaya Str., Moscow 105005, Russia
[3] "Algorithms and Data" LLC, Moscow, Russia
42, bld. 1, Dmitriya Ul'yanova Str., Moscow 117218, Russia

Abstract

The paper considers the main provisions of reversibility as a new paradigm for the development of computer technology. The first sections are of an overview nature. The inevitability of the so-called "heat curse" while maintaining the traditional paradigm of creating means of computer engineering. The fundamentals of reversible logic are presented, the main reversible logic elements and models of reversible computations, including reversible cellular automata, are considered. Reversible programming languages are briefly reviewed. The second part addresses the basic issues of the logical synthesis of circuits from reversible elements and the physical implementation of reversible circuitry. The synthesis of fault-tolerant circuits in the paradigm of reversible circuitry is briefly described. A technique for synthesizing fault-tolerant reversible elements in a hamming space is proposed and some such schemes are described. Next, the problems of using circuits of reversible logic elements in cryptography are considered. The proposed general scheme for creating reversible schemes with "garbage collection" intended for cryptographic applications is described.

Keywords: reversible logic, reversible logic elements, fault-tolerant circuits, hamming space, information protection, reversible circuits with "garbage collection".

For citation: Gurov S.I., Zhukov A.E., Zakablukov D.V., Kormakov, G.V. Reversible Computing: Review of the Problem and New Results (Fault Tolerance and Cryptography). *Sovremennye informacionnye tehnologii i IT-obrazovanie* = Modern Information Technologies and IT-Education. 2019; 15(3): 541-552. DOI: 10.25559/SITITO.15.201903.541-552

Введение

В начале 1960-х г. был открыт казавшийся сначала парадоксальным эффект Неймана-Ландауэра, и Ландауэр сформулировал принцип: *в любой вычислительной системе, независимо от её физической реализации, при потере одного бита информации выделяется теплота в количестве не менее* $\varepsilon_0 = k \cdot T \cdot ln2$ [J], *k – постоянная Больцмана, T – абсолютная температура* [1]. В 2012 г. эффект удалось обнаружить на практике [2]. Ландауэр даже придумал слоган *информация осязаема*, она не существует абстрактно и воплощена материально.

При комнатной температуре $\varepsilon_0 \approx 3 \cdot 10^{-21}$ [J], что сравнимо со средней энергией одной молекулы воздуха при комнатной температуре. Однако в пересчёте на процессор рассеиваемая мощность вырастает уже до величин порядка 1 [W]. Если в начале XXI в. компьютеры при обработке одного бита рассеивали примерно в миллион раз больше тепла, чем ε_0, то на начало 2010-х это отношение снизилось до нескольких тысяч, и исследователи предсказывают приближение этого соотношения к 1 в течение ближайших десятилетий.

По результатам анализа, проведённого в рамках проекта ITRS-2001, использовании 22-нм технологии ИМС, выделение тепла составит 5-10 [MW] на кв. см поверхности процессора. Отметим, что к 2020 г. Intel планирует освоение 7-нм техпроцесса, а к 2022 г. – 4 нм. Ясно, что в данных условиях игнорировать эффект Ландауэра уже нельзя: исследователи пришли к выводу, что, без учёта тепловых шумов и требований надёжности, физический предел традиционных технологий вычислителя с плавающей точкой – 10^{22} операций в секунду [3]. Данную критическую ситуацию можно назвать *тепловым проклятием* развития вычислительной техники в рамках существующих технологий.

Имеется, очевидно, только два способа, уменьшить рассеивание энергии на бит ниже ε_0. Первый очевидной способ – пытаться снижать рабочую температуру вычислителя. При этом общая рассеиваемая энергия не уменьшается, а просто переносится на холодильное оборудование.

Другой подход состоит в использовании обратимых вычислительных устройств. Обратимые вычисления блокируют потерю информация, что теоретически обеспечивает нулевые потери энергии на её обработку. Если в традиционных компьютерах ненужная для дальнейших вычислений информация не сохраняется, исчезает, то обратимые вычисления организуют так, чтобы этого не произошло.

Здесь важно указать: обратимость необходимо поддерживать на всех уровнях вычислений: алгоритмических языков, реализации прикладных программ, схемотехники и физической реализации обратимых элементов, поскольку необратимость хотя бы на одном уровне полностью разрушает положительные эффекты остальных [4].

Обратимые комбинационные элементы

Вычисления обратимы, если по выходным величинам полностью восстанавливаются входные.

На логическом уровне такие вычисления реализуют на обратимых элементах. Обратимые комбинационные элементы осуществляет биективные преобразования входного булевого *n*-вектора в выходной. Как следствие, такой элемент должен иметь *n* выходов. В общем случае элементы с *n* входами и *m* выходами будем называть n×m-элементами.

На выходе обратимого комбинационного элемента кроме значений вычисляемой функции (*информационных* битов) обычно формируются ещё и дополнительные *мусорные* (garbage или *стоковые*, sink) биты. Они и дают возможность по выходному вектору однозначно восстановить входной. Их нельзя потерять, поскольку тогда произойдёт рассеяние энергии, для борьбы с которым и разрабатывают обратимые элементы.

Далее входы элементов (вентилей, гейтов, gate) будем обычно обозначать $A, B, \ldots$ или $A_1, A_2, \ldots$, а выходы – *P, Q, ...* Иногда появятся и другие обозначения, но они будут понятны. Входы обратимых элементов разделяются на *управляющие (адресные)* и *управляемые (целевые)*. Значения адресных входов передаются на выход без изменений, в то время как значения целевых изменяются в зависимости от значений адресных. Символом ($\bullet$) обозначаем операцию конъюнкции, символом ($\oplus$) – операцию *сумма по mod* 2.

К настоящему времени разработано много обратимых комбинационных элементов. Простейшими из них являются NOT ($P = \bar{A}$) и SWAP ($P = B, Q = A$), CNOT (Controlled NOT или элемент Фейнмана FG: $P = A, Q = A \oplus B$), элементы Тоффоли ($P = A, Q = B, R = C \oplus A \cdot B$) и Фредкина ($P = A, Q = \bar{A} \cdot B \ ? \ A \cdot C, \ R = A \cdot B \ ? \ \bar{A} \cdot C$).

Рассмотрим ещё некоторые обратимые вентили.

Обобщённый гейт Тоффоли (C^{n}NOT, MCT, Multiple-Control Toffoli Gate, GT, generalized n-bit Toffoli) реализует преобразование $P_1 = A_1, \ldots, P_{n-1} = A_{n-1}, P_n = A_1 \cdot \ldots \cdot A_{n-1} \oplus A_n$, $n > 2$.

Обобщённый (множественно управляемый) гейт Фредкина (обобщённый CSWAP, CnSWAP, MCF, Multiple-Control Fredkin Gate) – n×n-гейт, n>2, всегда передающий на выход входы $A_1, \ldots A_{n-2}$ без изменения, при этом если все они равны 1, то входы A_{n-1} и A_n также передаются на выход без изменений (т.е. $P_1 = A_1, \ldots, P_n = A_n$), иначе A_{n-1} и A_n меняются на выходе местами ($P_{n-1} = A_n, P_n = A_{n-1}$) [6].

Вентиль Переса (PG) реализует обратимый полусумматор и не является универсальным. Он реализует преобразование $P = A, Q = A \oplus B, R = C \oplus A \cdot B$ (здесь A и B – суммируемые разряды, Q – их сумма, R – перенос в следующий разряд и P – мусорный бит). Вентиль Переса может быть реализован на гейтах TG и CNOT.

Описание других обратимых вентилей находится в следующих работах [5, 6, 7].

Математика обратимой логики

Схемы из функциональных элементов

Такие схемы классически определяются как ориентированный графы без циклов с помеченными рёбрами и вершинами. Обратимая схема из функциональных элементов есть ациклическая комбинационная логическая схема, в которой все элементы обратимы и соединены друг с другом последовательно без ветвлений.

В самой простой математической модели обратимых схем все элементы имеют одинаковое количество входов и выходов *n*. В ориентированном графе, описывающем такую обратимую схему, все вершины нумеруются от 1 до *l* и имеют ровно *n* занумерованных входов и выходов. При этом *i*-й выход *m*-й вершины, *m* < *l*, соединяется только с *i*-м входом (*m*+1)-й вер-

шины. Входами обратимой схемы являются входы первой вершины, а выходами – выходы *l*-й вершины. Величина *l* называется *сложностью схемы*.

Обратимый элемент по определению задаёт биективное отображение на некотором множестве, к примеру, на множестве двоичных векторов длины *n*. Любое такое преобразование можно описать подстановкой на данном множестве. Следовательно, последовательное соединение обратимых элементов задаёт подстановку, равную произведению соответствующих подстановок. Отсюда следует очевидная связь между обратимыми схемами и подстановками на множестве.

В случае, когда элемент самообратим, задаваемая им подстановка является обратной к самой себе. Если обратимая схема состоит только из самообратимых элементов, то схема, состоящая из тех же самых элементов, но соединенных в обратном порядке, задает биективное отображение, обратное к отображению, задаваемому оригинальной схемой.

Вычисления с ограниченной памятью

Модель обратимых схем можно свести к модели функциональных схем с ограниченной памятью, на операции в которой наложены дополнительные условия. В случае обратимой схемы с *n* входами каждой линии ставится в соответствие один из *n* регистров памяти (ячеек памяти), хранящих результат вычислений на каждом шаге работы схемы. Входы и выходы элементов, подключенные к линиям схемы, считывают и записывают значения в соответствующий линии регистр памяти.

Для реализации биективного отображения на множестве двоичных векторов длины n необходимо как минимум n регистров памяти. Если требуемое отображение невозможно реализовать при помощи заданного семейства обратимых элементов на n регистрах памяти, но можно реализовать на $n+q$ регистрах памяти, то говорят, что схема реализует отображение с q дополнительными входами. В большинстве случаев перед началом работы в регистры памяти, соответствующие дополнительным входам, записывают нулевые значения, однако могут быть записаны и единичные значения.

Функциональная полнота семейств обратимых элементов

Семейство логических элементов, из которых строиться схема, называют *библиотекой*. Схему, построенную из элементов библиотеки **B**, называют **B**-*схемой* и если такая схема реализует некоторую подстановку, то последнюю назовём **B**-*конструируемой*. Множество всех подстановок, реализуемых **B**-схемами с *n* входами и *n* выходами, является группой, обозначаемой B_n. Эта группа, в свою очередь, является подгруппой (может быть несобственной) симметрической группы S_{2^n}. Подстановки, соответствующие элементам библиотеки $\hat{B}$, являются образующими элементами группы B_n. В связи с этим задачи построения схем из обратимых элементов, реализующих элементы группы подстановок, и получения оценок для их сложности сводятся к задачам нахождения длин элементов группы подстановок в заданной системе образующих, длины самой группы и мощностей её слоев.

При этом естественно возникает вопрос функциональной полноты заданного семейства обратимых элементов: какие биективные отображения (какой класс отображений) могут быть реализованы и при каком количестве дополнительных входов.

Библиотека, образованную элементами NOT, CNOT, TG называют *NCT-библиотекой* [18]. Будем называть *N-конструируемой* (*C-конструируемой, T-конструируемой*) подстановку, построенную исключительно с помощью элементов NOT (соответственно CNOT, TG). В работе [19] показано:

- группа S_{2^n} содержит 2^n N-конструируемых подстановок, $\prod_{i=0}^{n-1}\left(2^n-2^i\right)$ C-конструируемых подстановок, $\frac{1}{2}\left(2^n-n-1\right)!$ T-конструируемых подстановок. Все три указанных множества являются подгруппами в S_{2^n}.
- в схеме с $n>3$ входами каждый из элементов NCT-библиотеки задаёт чётную подстановку, как следствие и сама обратимая схема также реализует чётную подстановку.
- для любой заданной подстановки на множестве двоичных векторов длины n " 3 существует реализующая её NCT-схема с n входами.
- для любой заданной чётной подстановки на множестве двоичных векторов длины $n\geq 4$ существует реализующая её NCT-схема с n входами.
- для любой заданной нечётной подстановки на множестве двоичных векторов длины $n\geq 4$ не существует реализующей её NCT-схемы с n входами, однако существует реализующая ее NCT-схема с $n+1$ входами (один дополнительный).
- зафиксируем набор библиотек $\mathbf{B}_1$, ..., $\mathbf{B}_k$ и построим $(\mathbf{B}_1|...|\mathbf{B}_k)$ -схему, присоединяя выходы $\mathbf{B}_i$-схемы к входам $\mathbf{B}_{i+1}$-схемы, $i=1,\ldots,k-1$. Входом построенной схемы есть вход $\mathbf{B}_1$-схемы, а выходом $\mathbf{B}_k$-схемы. Каждая подстановка из S_{2^n} при $n=1,2,3$ и каждая четная подстановка при $n\geq 4$ может быть реализована некоторой (T|C|T|N)-схемой и, следовательно, является NCT-конструируемой.

Аналогично, для других семейств обратимых элементов можно установить их функциональную полноту. Пусть **B** – библиотека из обратимых логических элементов. Тогда **B** универсальна, если для любого n и любой подстановки $\pi\in S_{2^n}$ существует такое *q*, что некоторая **B**-схема вычисляет $\neq$, используя *q* линий дополнительной (вспомогательной) памяти. Доказано, что для любой универсальной библиотеки **B** и достаточно большого *n* подстановки из знакопеременной группы A_{2^n} являются **B**-конструируемыми, а подстановки из S_{2^n} реализуются с помощью не более, чем одной дополнительной линии.

Схемы с дополнительными входами

Считаем, что схема из обратимых элементов с *n* значимыми и *q* дополнительными входами реализует функцию $f:\mathbb{Z}_2^n\to\mathbb{Z}_2^m$, если подавая на управляемые (*значимые*) входы значение аргумента функции **x**, а на дополнительные входы – 0, на *m* значимых (*информационных*) выходах мы получаем значение вектора **y** (значения, полученные на остальных выходах, игнорируются). Если в конце работы схемы значение на незначимом выходе не равно входной константе, линия называется *мусорной*, а полученное значение – *вычислительным мусором*. Достаточно часто мусорными считаются все неинформационные выходы.

В общем, число основных (значимых) входов плюс число дополнительных входов должно равняться числу основных

(значимых) выходов плюс число незначимых выходов, включая мусорные, как показано на рисунке 1.

Р и с. 1. Входы и выходы обратимой схемы
F i g. 1. The inputs and outputs of the reversible circuit

Чтобы реализовать необратимое отображение с помощью обратимых логических элементов необходимо сделать его обратимым, добавив дополнительные входы, на которые подаются константы 0. Для реализации необратимой функции, в которой совпадающие выходные наборы встречаются до M раз, требуется как минимум $\log_2 M$ дополнительных линий [20]. В частности, для любого заданного отображения на множестве двоичных векторов длины n существует реализующая её обратимая схема, состоящая из элементов данного семейства и имеющая не более $2n$ входов (n дополнительных).

Поскольку произвольная схема с $n+q$ входами, построенная из обратимых элементов реализует некоторую подстановку на множестве двоичных векторов длины $n+q$, реализация этой схемой функции $f:\mathbb{Z}_2^n \to \mathbb{Z}_2^m$ может быть математически описана при помощи расширяющего $\varphi_{n,n+q}:\mathbb{Z}_2^n \to \mathbb{Z}_2^{n+q}$ вида

$$\varphi_{n,n+q}(x_1,\ldots,x_n) = x_1,\ldots,x_n,0,\ldots,0$$

и редуцирующего отображения $\psi^{\pi}_{n+q,m}:\mathbb{Z}_2^{n+q} \to \mathbb{Z}_2^n$ вида

$$\psi^{\pi}_{n+q,m}\left(\langle x_1,\ldots,x_{n+q}\rangle\right) = \langle x_{\pi(1)},\ldots,x_{\pi(m)}\rangle$$

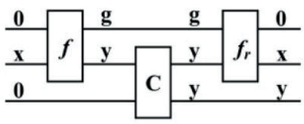
где π – некоторая подстановка на множестве $\mathbb{Z}_{n+q}$.

Тогда обратимая схема с $n+q \geq m$ входами, задающая подстановку $g:\mathbb{Z}_2^{n+q} \to \mathbb{Z}_2^{n+q}$, реализует отображение $f:\mathbb{Z}_2^n \to \mathbb{Z}_2^m$ с использованием $q \geq 0$ дополнительных входов (дополнительной памяти), если существует такая подстановка $\pi \in \mathbb{Z}_{n+q}$, что

$$\psi^{\pi}_{n+q,m}\left(g\left(\varphi_{n,n+q}(x)\right)\right) = f(x),$$

где $\mathbf{x} \in \mathbb{Z}_2^n, f(\mathbf{x}) \in \mathbb{Z}_2^m$.

В основополагающей работе Ч. Беннетта была предложена общая конструкция для обратимого вычисления произвольной (обратимой или необратимой) функции [4]. Конструкция Беннетта схематично представлена на рис. 2. Здесь ***f*** – схема из обратимых элементов, реализующая функцию $f:\mathbb{Z}_2^n \to \mathbb{Z}_2^m$, $\mathbf{x} \in \mathbb{Z}_2^n$ – вход функции, $\mathbf{y} \in \mathbb{Z}_2^m$ – ее выход, ***g*** – вычислительный мусор, $\boldsymbol{f_r}$ – зеркальная схема. В конструкции используется подсхема копирования **C**, которая может быть построена из элементов CNOT.



Р и с. 2. Конструкция Беннета
F i g. 2. Bennett's Design

С помощью любого семейства, содержащего универсальный элемент, можно реализовать в обратимой схеме любое заданное отображение на множестве двоичных векторов длины n, однако количество необходимых для этого дополнительных входов схемы может кардинально отличаться для различных семейств: оно может быть либо константным, либо зависящим от количества элементов в схеме.

От количества дополнительных входов обратимой схемы зависит и класс отображений на множестве двоичных векторов, реализуемых при помощи заданного семейства обратимых элементов, даже если оно содержит универсальный элемент. К примеру, вентиль Фредкина является универсальным, однако при помощи обратимой схемы без дополнительных входов, состоящей только из элементов данного типа, невозможно реализовать отображение, изменяющее вес Хемминга двоичного вектора (количество единичных координат в нем).

Помимо этого, от количества дополнительных входов обратимой схемы существенно зависит и ее сложность.

Обратимые функции и языки программирования

Обратимые надстройки над функциями

Для обеспечения обратимости при вычислениях используют функции, инъективные по первому аргументу, которые получают из обычных функций, применяя процедуру обратимой надстройки [11].

Определение. Бинарную функцию $A \times B \xrightarrow{\odot} C$ называют *инъективной по первому аргументу*, если

$a_1 \odot b = a_2 \odot b \Rightarrow a_1 = a_2$.

Пример: $n+3 = m+3 \Rightarrow n = m$; антипример: $n \times 0 = m \times 0 \not\Rightarrow n = m$.

Утверждение. *Пусть* $\odot$ *– бинарная функция, инъективная по первому аргументу. Тогда существует функция* $C \times B \xrightarrow{\oslash} A$ *такая, что* $(a \odot b) \oslash b = a$.

Неформально говорят, что операция $\oslash$ «обратна» к операции $\odot$. Примеры пар $(\odot, \oslash)$: для чисел – $(+,-)$, при работе с ненулевыми числами – $(\times,\div)$, для булевых векторов – $(\oplus,\oplus)$.

Обратимая надстройка над унарной функцией $f(y)$ есть бинарная функция $g(x,y) = (x \odot f(y), y)$, такая, что операция $\odot$ обратима по первому аргументу:

$$x \to x \odot f(y) \qquad (x \odot f(y)) \oslash f(y) = x \leftarrow x \odot f(y)$$
$$y \to y \qquad\qquad y \leftarrow y$$

Пример: если $g(x,y) = (x + f(y), y)$, то $g^{-1}(x,y) = (x - f(y), y)$. Пусть $f(y) = \sin y$, x – произвольное, тогда

$$g(x,y) = (x + \sin y, y), g^{-1}(x,y) = (x - \sin y, y),$$
$$g^{-1}(x + \sin y, y) = (x + \sin y - \sin y, y) = (x, y).$$

Обратимые надстройки существуют у всех функций. Обратная детерминированность программы позволяет проводить вычисления программы в обратном порядке.

Двунаправленные программы называют линзами (*lens*) [11].

Janus и другие

Язык Janus – первый обратимый язык программирования, был предложен Т. Йокоямой и Р. Глюком в 1982 году [12]. Преобразователь и интерпретатор Janus свободно доступен[1]. Janus также реализован на языке Prolog.

В языке Janus все функции заменены обратимыми надстройками, в циклах и условиях возвращается информация о пройденном пути («история» Беннета), параметры процедур передаются только по ссылке, глобальных переменных нет, при инициализации процедур все переменные и элементы массива обнуляются, а стеки опустошаются.

Janus – императивный язык программирования: исполнение программы состоит в последовательном выполнении команд. Программа на языке Janus $prog$ состоит из основной процедуры p_{main}, за которой следует последовательность d_proc^* определений процедур. Основная процедура p_{main} не имеет параметров и состоит из указания типов переменных и оператора. Имеются следующие типы переменных данных: скаляр (32-разрядное целое), одномерный массив целых и стек целых. Логические значения суть ‘целое ненулевое’ = *true*, ‘целое нулевое’ = *false*. Массивы индексируются целыми числами, начиная с нуля.

Управляющие структуры языка Janus традиционны. Это операторы присваивания, обмена двух значений ($\langle = \rangle$), условный (if e_1 then s_1 else s_2 fi e_2) и цикла (from e_1 do s_1 until e_2). В Janus’e также имеются операторы работы со стеком – поместить данное в стек извлечь его из стека (*push, pop*), прямого (call) и обратного (uncall) вызовов процедуры.

Пример вычисления чисел Фибоначчи на языке Janus смотрите, например, [11, 13].

Известны и другие обратимые языки: CRL, PsiLisp, R, In [14, 15].

Обратимое программирование тесно связано с алгебраическим (функциональным). Некоторые результаты по алгебрам программ в связи с обратимыми вычислениями можно найти в работах Н. Н. Непейводы [16, 17].

Логический синтез и физическая реализации

Классические методы логического синтеза не могут быть напрямую применены в схемах, построенных на обратимых элементах, и такие схемы проектируются для каждого конкретного устройства с заданной булевой функцией. При этом не существует единых правил, с помощью которых можно было бы спроектировать произвольное устройство.

Проблема стоков

Получаемые при вычислениях мусорные биты просто отбросить нельзя, их нужно неким особым образом утилизировать. Если на каждый триггер процессора будет приходиться свой сток, то какого-либо выигрыша по энерговыделению не получится. Утилизация мусора – самая энергозатратная операция, она присуща и обычной схемотехнике. Полностью избежать появления мусора, как мы отмечали, не удаётся.

Простейший подход, заключающийся в сохранении каждого мусорного значения на выходе, ясно, быстро «съест» всю доступную память. Есть предположение, что если программа выполняет N команд, то для хранения стоков потребуется порядка $\log N$ бит памяти.

Для решения проблемы Фредкин, Тоффоли и их студенты предложили [18]:

1. Произвести вычисления, получив большое количество мусора;
2. Записать результат, получив ещё немного мусора;
3. Выполнить вычисления в обратном направлении, уничтожив мусор, полученный на шаге 1).

При этом придется выделить дополнительную память под информацию, связанную с принципиально командами. При качественной разработке кода программы, однако, количество такого мусора можно ограничивать.

Методы синтеза обратимых схем

Под обратимой схемотехникой мы понимаем методы синтеза обратимых вычислителей, в т. ч. со специальными свойствами (например, сбоеустойчивость).

Обратимые схемы можно рассматривать в качестве модели реального вычислительного устройства. В зависимости от технологических ограничений при производстве устройств и предъявляемым к ним требованиям меняются цели и задачи методов синтеза. Среди них можно выделить следующие:

- минимизация количества элементов схемы (её сложности);
- минимизация количества дополнительных входов схемы в случаях, когда добавить дополнительный вход в схему технологически гораздо сложнее, чем увеличить её сложность;
- минимизация количества элементов определённого типа (к примеру, если элемент Тоффоли технологически реализуется в квантовом устройстве гораздо сложнее, чем NOT или CNOT, то лучше синтезировать обратимую схему большей сложности, но с меньшим количеством элементов Тоффоли.

Наиболее известные из библиотек синтеза обратимых элементов можно найти в [21, 22].

Для синтеза обратимых схем в основном применяют традиционные методы, адаптированные под обратимую схемотехнику [23].

Особо отметим парадоксальный, казалось бы, факт: при реализации классическая необратимая логика требует в разы больше вентилей, чем обратимая [24]. Например, однобитный обратимый сумматор требует четырёх вентилей и срабатывает за 2τ (τ – время задержки вентиля).

Cycle-based алгоритм

Особняком стоит метод на базе cycle-based алгоритма [7]. Он основан на представлении заданного биективного отображения в виде произведения независимых циклов (подстановок). Данное представление заменяется на эквивалентное и,

[1] Janus Extended Playground. URL: http://topps.diku.dk/pirc/janus-playground (дата обращения: 16.07.2019).

возможно, избыточное, но удобное для синтеза по частям. Алгоритмы синтеза такого типа позволяют получать обратимые схемы с асимптотически оптимальной сложностью.

Cycle-based алгоритм позволяет получить обратимую схему без дополнительных входов со сложностью $L(n,0) \lesssim \frac{48n2^n}{\log_2 n}$.

Доказано, что почти все чётные подстановки на множестве двоичных векторов длины n реализуются со сложностью $L \gtrsim n2^n / \log_2(n+q)$. Таким образом, алгоритм синтеза является асимптотически оптимальным.

Для синтеза произвольного отображения на множестве двоичных векторов длины n был разработан алгоритм синтеза, являющийся модификацией стандартного метода Лупанова и оптимизированный под различное количество дополнительных входов q. Данный алгоритм позволяет получить обратимую схему со сложностью $L(n,q) \lesssim \frac{8n2^n}{\log_2 q}$ при $n^2 \lesssim q \lesssim 2^{n-o(n)}$. Доказано, что почти все отображения на множестве двоичных векторов длины n реализуются со сложностью $L \gtrsim n2^n / \log_2(n+q)$. Таким образом, алгоритм синтеза является асимптотически оптимальным для указанного диапазона значений q.

Стоит отметить, что сложность синтезированной обратимой схемы, как это видно из оценок выше, существенно зависит от количества дополнительных входов: чем их больше, тем меньше сложность схемы в худшем случае. Данная зависимость является характерной для обратимых схем и выражена не столь явно для классических, необратимых схем. Таким образом, во время синтеза обратимой схемы приходится искать компромисс и выбирать между большей сложностью схемы либо большим количеством дополнительных входов.

Самокорректируемые элементы

Повышение надёжности функционирования интегральных микросхем (ИМС) остаётся актуальной проблемой синтеза. Важной стороной задачи остаётся проблема устойчивости схем к отказам, как к кратковременным самоустраняемым (сбоям, SEU, single event upsets), так и к неустранимым. Причинами таких отказов являются воздействие на схему различных видов помех: радиационных, скачков напряжения питания, а также естественная деградация сигналов во времени. Устойчивость работы ИМС является важнейшим требованием к аппаратуре, работающей в тяжёлых условиях космоса. В настоящее время в мире активно развивается направление радиационно-стойкого проектирования (RHBD, Radiation Hard by Design), основанного на использовании схемотехнических, топологических и алгоритмических методов повышения сбоеустойчивости.

Отметим, что неисправность того или иного логического элемента ИМС может привести выходе всей схемы как к кратной ошибке на (*конус ошибок*), так и к её непроявлению (*маскирование* ошибки).

С целью обнаружения и исправления ошибок вычислений, осуществляемых *основной* аппаратурой, её дополнят корректирующей *дополнительной*. В этом случае говорят о *схемной избыточности*. Создание самокорректируемых схем является задачей синтеза вычислительных устройств с дополнительными требованиями.

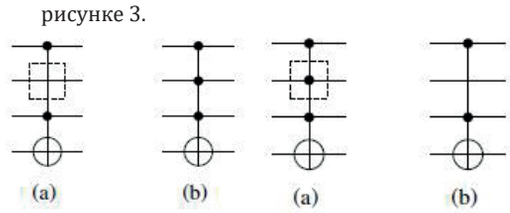
Под *самокорректируемостью* понимают свойство обнаруживать и исправлять ошибки, возникающие как в основной, так и в дополнительной аппаратуре. Последнее свойство позволяет избежать проблемы «сторожа над сторожем». Создание самокорректируемых схем является задачей синтеза вычислительных устройств с дополнительными требованиями.

Для анализа ошибок схем и разработке методов их парирования рассматривают различные модели возникающих ошибок. Укажем некоторые модели логических неисправностей, используемых при анализе сбоеустойчивости обратимых схем [23-26].

- *Константная ошибка* (stuck-at fault model) – данный бит выхода всегда принимает константное значение 0 или 1.
- *Битовая ошибка* (bit fault model) – инвертирование выходного бита.

Эти модели неисправности используется при исследовании и обычных, необратимых схем.

- *Узловая ошибка* (crosspoint fault model) – специфическая ошибка обратимых элементов Тоффоли описанных типов, описывает ошибки возникновения (Appearance Fault) или исчезновения (Disappearance Fault) в нём контрольных разрядов, как показано на рисунке 3.



Р и с. 3. Схемы появления узловых ошибок возникновения (1) и исчезновения (2):
(a) – до появления сбоя, (b) – после

F i g. 3. The appearance of nodal errors of occurrence (1) and disappearance (2):
(a) - before the failure, (b) - after

В простейшем случае сбоеустойчивая схема обеспечивает лишь обнаружение возникшей ошибки. Исправить обнаруженную ошибку далее возможно методом многократного пересчёта выхода, используя, таким образом, *временную* избыточность схемы (понятно, что использование любого вида избыточности, временной или информационной, в конечном счёте реализуется дополнительными схемами). На сегодняшний день исследования методов синтеза сбоеустойчивых обратимых схем ограничиваются практически исключительно указанным простейшим вариантом, не приводящий к значительному увеличению сложности схемы.

В [5] дан обзор известных подходов к синтезу обратимых схем. Эти подходы можно разделить на два класса. Первый заключается в построении элементов, обеспечивающих контроль чётности своих выходов и использованием далее общих методов синтеза. Второй связан с приданием свойств контроль чётности уже синтезированных схем.

Большинство методов находятся в рамках первого класса. Разработан универсальный способ преобразования произвольных обратимых элементов в гейты, сохраняющие чётность.

Методы второго класса часто требуют значительного перепроектирования уже имеющихся схем. При этом увеличивается количество ключей, добавляются новые проверяющие элементы и увеличивается мусор.

В последнее время предложены следующие схемы сбоеустойчивых вычислительных устройств, использующих контроль чётности [23]: полный сумматор из сохраняющих чётность блоков (2010), АЛУ (2013), компрессор (устройство для сжатия динамического диапазона звукового сигнала), полный сумматор (2015).

Отметим, что существует простейший метод *троирования* или *тройного модульного резервирования* (TMR, Triple Modular Redundancy) построения сбоеустойчивых, при котором результат определяется *мажорированием* «голосования 2 из 3» выходов трёх экземпляров основной схемы. Данную схему мажорирования называют *воутером*.

Помехоустойчивое кодирование в хэмминговом пространстве

Предложим новый метод построения сбоеустойчивых обратимых элементов на основе помехоустойчивого кодирования в пространстве Хэмминга [26]. Метод заключается в замене обратимых гейтов на их сбоеустойчивые аналоги, обеспечивающих гарантированное автоматическое исправление любой одиночной ошибки, т. е. основанный на принципе селективной защиты на уровне отдельных элементов.

В простейшем случае помехоустойчивого кодирования будем кодировать булевы значения 0 и 1 тремя битами 000 и 111, используя 3 проводника вместо одного и называя данные 3-битовые значения Полюсом_0 и Полюсом_1 соответственно. Пространство *n*-мерного единичного куба 2^n называют пространством Хэмминга; мы будем работать в 3-мерном таком пространстве. Булевы операции будем производить над сигналами как над соответствующими полюсами. При возникновении не более чем одиночной ошибки хэммингово расстояние (число несовпадающих бит) не превосходит 1. Коррекция ошибки происходит автоматически в неявном виде.

Для обратимой схемотехники это существенно: происходит исправление ошибки, а не просто фиксация факта, что ошибка произошла.

Нетрудно заметить, что предлагаемый подход имеет сходство с методом TMR на уровне элементов. Различие заключается в том, что при троировании элемента имеется воутер, который не защищен от ошибок; т. е. один элемент защищается элементами, каждый из которых также подвержен сбоям. При использовании метода кодирования в пространствах Хэмминга эта проблема принципиально отсутствует. Дело в том, что воутером в этом подходе выступают последующие элементы в схеме, воутером которых в свою очередь, являются следующие за ними и т.д. Это обеспечивает исправление любой однократной битовой ошибки элемента. Более того, предлагаемый метод имеет существенные преимущества относительно многократных ошибок. Строго говоря, любое число кратных ошибок гарантированно исправляется, при условии, что на один расширенный элемент приходится не больше одного сбоя.

Очевидной платой за столь высокий уровень сбоеустойчивости являются значительные аппаратные затраты. Кратное увеличение числа используемых ключей делают этот метод вряд ли применимым для всей схемы, оставляя его целесообразным для наиболее уязвимых элементов и подсхем, сбоеустойчивость которых критически важна для функционирования всей вычислительной системы.

В [26] представлены разработанные авторами данной статьи самокорректирующиеся обратимые элементы в пространстве Хэмминга: обратимый воутер «2 из 3», гейты HNOT, (инвертирование), HCNOT (управляемое инвертирование), Тоффоли HTG, Фредкина HFRG, полный одноразрядный сумматор HADD.

В цитируемой работе также указаны примеры элементов, реализованных в поляризованном Хэмминговом пространстве, когда качестве полюсов выбраны векторы 010 и 101. Такой выбор реализует более сбалансированное представления сигналов 0 и 1, и, соответственно, нагрузку на транзисторные вентили при реализации схем на кристалле. Переход к такому поляризованному хеммингову пространству геометрически эквивалентен вращению единичного куба вокруг своего центра с переносом нулевого вектора в вершину, представляемую вектором поляризации.

Сравнение сложности и моделирование сбоеустойчивости обычных необратимых схем с их аналогами, построенных на элементах в хэмминговом пространстве позволило утверждать, что, во-первых, число вентилей в методе тройного резервирования возрастает примерно в четыре раза, в то время как при кодировании в пространстве Хэмминга оно возрастает на порядок и, во-вторых, сбоеустойчивость предлагаемого метода во всех случаях превосходит метод тройного резервирования, что также продемонстрировано при соответствующем сравнении.

Физическая реализация

Адекватной физической реализации обратимых вычислений создать до сих пор не удалось. Здесь применяются следующие подходы [25]: КМОП технология с пониженным потреблением энергии; ПЗС-структуры (приборы с зарядовой связью, CCD, Charge-Coupled Device); оптические вычислительные устройства (оптические солитоны); квантовые вычислительные устройства.

Перспективными в плане обратимости эксперты считают стремительно развивающиеся микро/нано-электромеханические технологии, системы клеточных автоматов. Возможно, такие автоматы удастся построить из молекул.

Использование схем из обратимых логических элементов в криптографии

Идея использование в криптографии обратимой логики и схем из обратимых элементов не нова. Криптографы видят в них возможности дополнительной защиты информации от утечек по побочным каналам.

Одним из действенных методов современного криптоанализа является так называемая *разностная атака по мощности*, которая использует информацию об энергопотреблении криптографического устройства. Поскольку энергия, потребляемая устройством, меняется в зависимости от обрабатываемых данных и выполняемых на разных этапах алгоритма команд, её измерение дает возможность определить характеристики криптографического устройства и даже используемый ключ криптоалгоритма. При этом идеальная реализация схемы из обратимых элементов, теоретически, вообще лишена недостатков, связанных с возможностью утечки информации по соответствующему побочному каналу. Это объясняет большое внимание специалистов к схемам из обратимых элементов, реализующим арифметические операции,

поскольку такие операции используются в большинстве ассиметричных криптосистем. Например, в современных реализациях системы шифрования с открытым ключом RSA используются модульные операции сложения и умножения чисел длиной до 3072 бит.

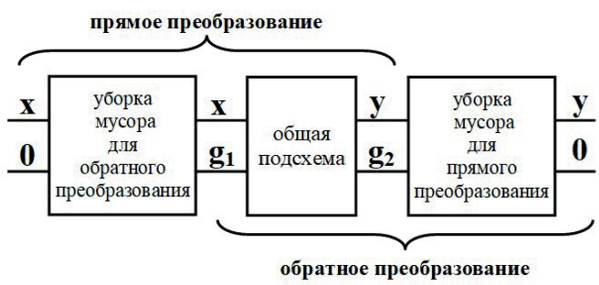
Исследования в этой области за последнее время ведутся со все возрастающей интенсивностью и различными авторами были предложены схемы обратимых полусумматора, и полного сумматора, триггера, умножителя и др. Среди данных схем умножитель имеет особое значение. Ещё в 1994 г. разрабатывалась схема обратимого умножителя, потребляющая на 99% меньше энергии, чем её классическая КМОП-реализация. В работе [27] предлагается обратимая реализация схем сложения и умножения в поле, как наиболее затратных по энергопотреблению операций АЛУ криптопроцессора. Также реализованы полный сумматор и умножитель Монтгомери. Далее, в работе [28] авторы предлагают реализацию схем обратимых мультиплексоров, регистров и сдвиговых регистров. Это позволило им реализовать схему умножителя Монтгомери, более выгодную в вопросах элементной сложности, количества мусорных выходов и квантового веса. Вскоре были предложены другие обратимые схемы для умножения, имеющие меньшую сложность и количество мусорных бит [29]. Это, однако, было достигнуто с помощью введения новых обратимых элементов.

Операцией, обратной к операции возведения в степень элемента циклической группы, является операция дискретного логарифмирования, играющая ключевую роль в современной криптографии. В работах [30, 7] рассматривается алгоритм дискретного логарифмирования по основанию примитивного элемента в конечном поле характеристики 2 и его реализация обратимыми схемами, как с дополнительной памятью, так и без неё. Доказывается верхняя асимптотическая оценка сложности обратимой схемы, реализующей алгоритм дискретного логарифмирования и показывается, что уже при использовании n дополнительных входов (n -степень расширения поля), сложность таких схем существенно снижается. Данная оценка асимптотически ниже, чем для произвольного булева преобразования, и достигается при асимптотически меньшем количестве дополнительных входов.

Одним из важнейших понятий для современной криптографии является однонаправленная функция (one-way function), т. е. функция, вычислить значение которой можно за время, полиномиальное от числа битов входа, в то время как вычислить за полиномиальное время функцию, обратную к ней не представляется возможным. Это определение, на первый взгляд, идёт в разрез с принципами обратимой логики, согласно которым «зеркальная» схема для обратного преобразования должна иметь ту же сложность. Из этого противоречия можно сделать вывод, что-либо однонаправленные функции на деле не существуют, либо что их невозможно реализовать при помощи схем из обратимых элементов.

В работе [31] исследуется этот вопрос и доказывается, что такой вывод является ложным, обратимая логика позволяет реализовывать однонаправленные функции, а её использование не исключает возможность их существования. Авторы показывают, что различие в сложности реализации схем для прямой и обратной функций кроется в следующем факте: для построения схемы из обратимых элементов, реализующей однонаправленную функцию, необходимо использовать дополнительные (мусорные) линии, значения на которых после выполнения алгоритма схемой не входят в число битов ответа и могут быть проигнорированы, однако необходимы для осуществления обратного преобразования. Именно незнание значений этих битов (garbage outputs) обусловливает сложность обращения прямого преобразования.

В [32, 33] предлагается развитие данного взгляда на вопрос однонаправленности. Схема из обратимых элементов, реализующая однонаправленную функцию, должна иметь некоторое количество мусорных линий с неопределёнными значениями на выходе и, в следствие этого, не является обратимой. Автор (также один из авторов данной статьи) рассматривает подход, заключающийся в модификации таких схем с целью получения на мусорных линиях константных значений, не зависящих от значений, поданных на вход. Такая процедура называется *уборкой мусора* и требует введения в схему дополнительных элементов. Количество таких элементов в схемах, реализующих прямое и обратное преобразование, различается, что и обусловливает различия в сложности вычисления однонаправленной функции и обратного ей преобразования, как представлено на рисунке 4.



Р и с. 4. Обратимая схема с уборкой мусора
F i g. 4. Reversible scheme with garbage collection

Выводы

Обратимые вычисления – новая активно развивающаяся парадигма вычислений. Она обеспечивает принципиальную возможность выхода из ситуации «теплового проклятия». Важным является то, что обратимость необходимо поддерживать на всех уровнях вычислительных технологий, от схемотехники до физической реализации вычислений.

В области логического синтеза на сегодняшний день настоятельной необходимостью является создание таких обратимых сбоеустойчивых стандартных элементов, как мультиплексор, демультиплексор, шифратор, дешифратор, сумматор и триггеры и др. Разные исследователи предлагают те или иные подходы к данной проблеме, но общих методов такого синтеза сбоеустойчивых схем ещё не разработано.

Список использованных источников

[1] *Landauer R.* Irreversibility and Heat Generation in the Computing Process // IBM Journal of Research and Development. 1961. Vol. 5, No. 3. Pp. 183-191. DOI: 10.1147/rd.53.0183

[2] *Bérut A., Arakelyan A., Petrosyan A.* et al. Experimental verification of Landauer's principle linking information and thermodynamics // Nature. 2012. Vol. 483. Pp. 187-189.

DOI: 10.1038/nature10872

[3] *DeBenedictis E. P.* Will Moore's Law Be Sufficient? // SC '04: Proceedings of the 2004 ACM/IEEE Conference on Supercomputing, Pittsburgh, PA, USA, 2004. Pp. 45-45. DOI: 10.1109/SC.2004.68

[4] *Bennett C. H.* Logical Reversibility of Computation // IBM Journal of Research and Development. 1973. Vol. 17, No. 6. Pp. 525-532. DOI: 10.1147/rd.176.0525

[5] *Saeedi M., Markov I. L.* Synthesis and Optimization of Reversible Circuits – A Survey // ACM Computing Surveys (CSUR). 2013. Vol. 45, Issue 2, Article 21. DOI: 10.1145/2431211.2431220

[6] *Merkle R. C.* Reversible electronic logic using switches // Nanotechnology. 1993. Vol. 4, No. 1. Pp. 21-40. DOI: 10.1088/0957-4484/4/1/002

[7] *Закаблуков Д. В.* Методы синтеза обратимых схем из функциональных элементов NOT, CNOT и 2-CNOT: дисс. … канд. физ.-мат. наук. Москва, 2018. 151 с.

[8] *Toffoli T.* Computation and construction universality of reversible cellular automata // Journal of Computer and System Sciences. 1977. Vol. 15, Issue 2. Pp. 213-231. DOI: 10.1016/S0022-0000(77)80007-X

[9] *Feynman R. P.* Quantum Mechanical Computers // Optics News. 1985. Vol. 11, Issue 2. Pp. 11-20. DOI: 10.1364/ON.11.2.000011

[10] *Zhirnov V. V., Cavin R. K., Hutchby J. A., Bourianoff G.* Limits to Binary Logic Switch Scaling - A Gedanken Model // Proceedings of the IEEE. 2003. Vol. 91, No. 11. Pp. 1934-1939. DOI: 10.1109/JPROC.2003.818324

[11] *Foster J. N.* Bidirectional Programming Languages // Technical Report MS-CIS-10-08. Department of Computer & Information Science University of Pennsylvania. March 13, 2010.

[12] *Yokoyama T., Glück R.* A reversible programming language and its invertible self-interpreter // Proceedings of the 2007 ACM SIGPLAN symposium on Partial evaluation and semantics-based program manipulation (PEPM '07). Association for Computing Machinery, New York, NY, USA, 2007. Pp. 144-153. DOI: 10.1145/1244381.1244404

[13] *Yokoyama T.* Reversible Computation and Reversible Programming Languages // Electronic Notes in Theoretical Computer Science. 2010. Vol. 253, Issue 6. Pp. 71-81. DOI: 10.1016/j.entcs.2010.02.007

[14] *Fredkin E. F., Toffoli T.* Conservative logic // International Journal of Theoretical Physics. 1982. Vol. 21, Issue 3-4. Pp. 219-253. DOI: 10.1007/BF01857727

[15] *Perumalla K. S.* Introduction to Reversible Computing. CRC Press, 2014.

[16] *Непейвода Н. Н.* Алгебры как альтернатива численному параллелизму // Первый Национальный Суперкомпьютерный Форум (НСКФ-2012). Переславль-Залесский: ИПС имени А.К. Айламазяна РАН, 2012.

[17] *Непейвода Н. Н.* От численного моделирования к алгебраическому // Параллельные вычисления и задачи управления (PACO'2012). Шестая международная конференция. Том 1. Москва, 2012. С. 93-103. URL: https://elibrary.ru/item.asp?id=22032867 (дата обращения: 16.07.2019).

[18] *Toffoli T.* Reversible computing // Automata, Languages and Programming. ICALP 1980. Lecture Notes in Computer Science / de Bakker J., van Leeuwen J. (eds). Springer, Berlin, Heidelberg, 1980. Vol. 85. Pp. 632-644. DOI: 10.1007/3-540-10003-2_104

[19] *Shende V.V., Prasad A.K., Markov I.L., Hayes, J.P.* Synthesis of reversible logic circuits // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2003. Vol. 22, No. 6. Pp. 710-722. DOI: 10.1109/TCAD.2003.811448

[20] *Маслов Д.* Роль обратимости в компьютерных технологиях будущего // Компьютерра. 2004. № 14. URL: http://www.kinnet.ru/cterra/538/33163.html (дата обращения: 16.07.2019).

[21] *Wille R., Große D., Teuber L., Dueck G. W., Drechsler R.* RevLib: An Online Resource for Reversible Functions and Reversible Circuits // 38th International Symposium on Multiple Valued Logic (ismvl 2008). Dallas, TX, 2008. Pp. 220-225. DOI: 10.1109/ISMVL.2008.43

[22] *Maslov D., Dueck G.W., Scott N.* Reversible logic synthesis benchmarks page. Technical report, 2003. URL: http://www.cs.unb.ca/profs/gdueck/quantum (дата обращения: 16.07.2019).

[23] *Jain A.* Fault Tolerant Synthesis of Reversible Circuits // arXiv: 1310.5231, 2013.

[24] *Bruce J. W., Thornton M. A., Shivakumaraiah L., Kokate P. S., Li X.* Efficient adder circuits based on a conservative reversible logic gate // Proceedings IEEE Computer Society Annual Symposium on VLSI. New Paradigms for VLSI Systems Design. ISVLSI 2002, Pittsburgh, PA, USA, 2002. Pp. 83-88. DOI: 10.1109/ISVLSI.2002.1016879

[25] *Бобков С. Г.* Высокопроизводительные вычислительные системы / под ред. В.Б. Бетелина. М.: НИИСИ РАН, 2014. 296 с.

[26] *Кормаков Г. В., Гуров С. И.* Сбоеустойчивые обратимые схемы и метод их синтеза в пространстве Хэмминга // Прикладная математика и информатика: Труды факультета Вычислительной математики и кибернетики. М.: МАКС Пресс, 2018. № 57. С. 21-35.

[27] *Thapliyal H., Zwolinski M.* Reversible Logic to Cryptographic Hardware: A New Paradigm // 2006 49th IEEE International Midwest Symposium on Circuits and Systems, San Juan, 2006. Pp. 342-346. DOI: 10.1109/MWSCAS.2006.382067

[28] *Noor Muhammed Nayeem, Lafifa Jamal, Hafiz Md. Hasan Babu.* Efficient Reversible Montgomery Multiplier and Its Application to Hardware Cryptography // *Journal of Computer Science. 2009.* Vol. 5, Issue 1. Pp. 49-56. DOI: 10.3844/jcssp.2009.49.56

[29] *Banerjee A., Pathak A.* An analysis of reversible multiplier circuits // arXiv: 0907.3357v1 [quant-ph] 20 Jul 2009.

[30] *Жуков А. Е., Закаблуков Д. В., Засорина Ю. В., Чикин А. А.* Вычислительно асимметричные преобразования и схемы из обратимых элементов // Вопросы кибербезопасности. 2015. № 2(10). С. 49-55. URL: https://elibrary.ru/item.asp?id=23293952 (дата обращения: 16.07.2019).

[31] *Chau H. F., Lo H.-K.* One-way Functions in Reversible Computations // Cryptologia. 1997. Vol. 21, Issue 2. Pp. 139-148. DOI: 10.1080/0161-1197918858

[32] *Жуков А. Е.* Схемы из обратимых логических элементов: Один подход к изучению однонаправленности // Труды III Международной конференции «Информационные системы и технологии» (IST'2006). Минск, 2006. с. 85.

[33] *Жуков А. Е.* Один подход к изучению однонаправленности // Информационная безопасность. 2018. № 1. С. 40-43. URL: http://lib.itsec.ru/articles2/crypto/odin-podhodk-izucheniyu-odnonapravlennosti (дата обращения: 16.07.2019).

Поступила 16.07.2019; принята к публикации 25.08.2019; опубликована онлайн 30.09.2019.

Об авторах:

Гуров Сергей Исаевич, старший научный сотрудник, доцент кафедры математических методов прогнозирования, факультет вычислительной математики и кибернетики, Московский государственный университет имени М.В. Ломоносова (119991, Россия, г. Москва, ГСП-1, Ленинские горы, д. 1), кандидат физико-математических наук, доцент, ORCID: http://orcid.org/0000-0001-5486-1357, sgur@cs.msu.ru

Жуков Алексей Евгеньевич, доцент кафедры ИУ8 «Информационная безопасность», факультет информатики и систем управления, Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет) (105005, Россия, г. Москва, 2-я Бауманская ул., д. 5, стр. 1), кандидат физико-математических наук, доцент, ORCID: http://orcid.org/0000-0001-5459-3831, aez_iu8@rambler.ru

Закаблуков Дмитрий Владимирович, программист, ООО «Алгоритмы и данные» (117218, Россия, г. Москва, ул. Дмитрия Ульянова, д. 42 стр. 1), кандидат физико-математических наук, ORCID: http://orcid.org/0000-0001-6584-6443, dmitriy.zakablukov@gmail.com

Кормаков Георгий Владимирович, студент, факультет вычислительной математики и кибернетики, Московский государственный университет имени М.В. Ломоносова (119991, Россия, г. Москва, ГСП-1, Ленинские горы, д. 1), ORCID: http://orcid.org/0000-0002-7728-0392, egor2898@mail.ru

Все авторы прочитали и одобрили окончательный вариант рукописи.

References

[1] Landauer R. Irreversibility and Heat Generation in the Computing Proces. *IBM Journal of Research and Development.* 1961; 5(3):183-191. (In Eng.) DOI: 10.1147/rd.53.0183

[2] Bérut A., Arakelyan A., Petrosyan A. et al. Experimental verification of Landauer's principle linking information and thermodynamics. *Nature.* 2012; 483:187-189. (In Eng.) DOI: 10.1038/nature10872

[3] DeBenedictis E.P. Will Moore's Law Be Sufficient? In: *SC '04: Proceedings of the 2004 ACM/IEEE Conference on Supercomputing*, Pittsburgh, PA, USA, 2004, pp. 45-45. (In Eng.) DOI: 10.1109/SC.2004.68

[4] Bennett C.H. Logical Reversibility of Computation. *IBM Journal of Research and Development.* 1973; 17(6):525-532. (In Eng.) DOI: 10.1147/rd.176.0525

[5] Saeedi M., Markov I.L. Synthesis and Optimization of Reversible Circuits – A Survey. *ACM Computing Surveys (CSUR).* 2013; 45(2):21. (In Eng.) DOI: 10.1145/2431211.2431220

[6] Merkle R.C. Reversible electronic logic using switches. *Nanotechnology.* 1993; 4(1):21-40. (In Eng.) DOI: 10.1088/0957-4484/4/1/002

[7] Zakablukov D.V. Synthesis methods for reversible circuits from NOT, CNOT, and 2-CNOT functional elements: dis. ... Ph.D. (Phys.-Math.). Moscow, 2018. (In Russ.)

[8] Toffoli T. Computation and construction universality of reversible cellular automata. *Journal of Computer and System Sciences*. 1977; 15(2):213-231. (In Eng.) DOI: 10.1016/S0022-0000(77)80007-X

[9] Feynman R.P. Quantum Mechanical Computers. *Optics News.* 1985; 11(2):11-20. (In Eng.) DOI: 10.1364/ON.11.2.000011

[10] Zhirnov V.V., Cavin R.K., Hutchby J.A., Bourianoff G. Limits to Binary Logic Switch Scaling - A Gedanken Model. *Proceedings of the IEEE.* 2003; 91(11):1934-1939. (In Eng.) DOI: 10.1109/JPROC.2003.818324

[11] Foster J.N. *Bidirectional Programming Languages*. Technical Report MS-CIS-10-08. Department of Computer & Information Science University of Pennsylvania. March 13, 2010. (In Eng.)

[12] Yokoyama T., Glück R. A reversible programming language and its invertible self-interpreter. In: *Proceedings of the 2007 ACM SIGPLAN symposium on Partial evaluation and semantics-based program manipulation (PEPM '07).* Association for Computing Machinery, New York, NY, USA, 2007, pp. 144-153. (In Eng.) DOI: 10.1145/1244381.1244404

[13] Yokoyama T. Reversible Computation and Reversible Programming Languages. *Electronic Notes in Theoretical Computer Science.* 2010; 253(6):71-81. (In Eng.) DOI: 10.1016/j.entcs.2010.02.007

[14] Fredkin E.F., Toffoli T. Conservative logic. *International Journal of Theoretical Physics.* 1982; 21(3-4):219-253. (In Eng.) DOI: 10.1007/BF01857727

[15] Perumalla K.S. Introduction to Reversible Computing. CRC Press, 2014. (In Eng.)

[16] Nepejvoda N.N. Algebras as an alternative to numerical parallelism. In: *NSCF-2012, First National Supercomputing Forum*. Russia, Pereslavl-Zalessky, Program Systems Institute of RAS, November 29-30, 2012.

[17] Nepejvoda N.N. From Numerical Modeling to Algebraic. In: *The parallel calculations and the tasks of the control. The works of the International Conference PACO-2012.* Moscow, vol. 1, 2012, pp. 93-103. Available at: https://elibrary.ru/item.asp?id=22032867 (accessed 16.07.2019). (In Russ., abstract in Eng.)

[18] Toffoli T. Reversible computing. In: de Bakker J., van Leeuwen J. (eds). *Automata, Languages and Programming. ICALP 1980. Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, 1980; 85:632-644. (In Eng.) DOI: 10.1007/3-540-10003-2_104

[19] Shende V.V., Prasad A.K., Markov I.L., Hayes, J.P. Synthesis of reversible logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.* 2003; 22(6):710-722. (In Eng.) DOI: 10.1109/TCAD.2003.811448

[20] Maslov D. The role of reversibility in computer technology of the future. *Computerra.* 2004; 14. Available at: http://www.kinnet.ru/cterra/538/33163.html (accessed 16.07.2019). (In Russ.)

[21] Wille R., Große D., Teuber L., Dueck G.W., Drechsler R. RevLib: An Online Resource for Reversible Functions and Reversible Circuits. In: *38th International Symposium on*

Multiple Valued Logic (ismvl 2008). Dallas, TX, 2008, pp. 220-225. (In Eng.) DOI: 10.1109/ISMVL.2008.43

[22] Maslov D., Dueck G.W., Scott N. *Reversible logic synthesis benchmarks page.* Technical report, 2003. Available at: http://www.cs.unb.ca/profs/gdueck/quantum (accessed 16.07.2019). (In Eng.)

[23] Jain A. Fault Tolerant Synthesis of Reversible Circuits. *arXiv:* 1310.5231, 2013. (In Eng.)

[24] Bruce J.W., Thornton M.A., Shivakumaraiah L., Kokate P.S., Li X. Efficient adder circuits based on a conservative reversible logic gate. In: *Proceedings IEEE Computer Society Annual Symposium on VLSI. New Paradigms for VLSI Systems Design. ISVLSI 2002.* Pittsburgh, PA, USA, 2002, pp. 83-88. (In Eng.) DOI: 10.1109/ISVLSI.2002.1016879

[25] Bobkov S.G. *Vysokoproizvoditelnye vychislitelnye systemy* [High-Performance Computer Systems]. Moscow, NIISI RAN Publ., 2014. (In Russ.)

[26] Kormakov G. V., Gurov S.I. Failure-proof reversible circuits and a method for their synthesis in Hamming space. In: *Applied Mathematics and Informatics. Proceedings of the Faculty of Computational Mathematics and Cybernetics Lomonosov Moscow State University.* Moscow, MAKS Press, 2018, no. 57, pp. 21-35. (In Russ.)

[27] Thapliyal H., Zwolinski M. Reversible Logic to Cryptographic Hardware: A New Paradigm. In: *2006 49th IEEE International Midwest Symposium on Circuits and Systems*. San Juan, 2006, pp. 342-346. (In Eng.) DOI: 10.1109/MWSCAS.2006.382067

[28] Noor Muhammed Nayeem, Lafifa Jamal, Hafiz Md. Hasan Babu. Efficient Reversible Montgomery Multiplier and Its Application to Hardware Cryptography. *Journal of Computer Science. 2009;* 5(1):49-56. (In Eng.) DOI: 10.3844/jcssp.2009.49.56

[29] Banerjee A., Pathak A. An analysis of reversible multiplier circuits. *arXiv:* 0907.3357v1 [quant-ph] 20 Jul 2009. (In Eng.)

[30] Zhukov A., Zakablukov D., Zasorina Yu., Chikin A. Computationally Asymmetric transformations And Reversible Logic Circuits. *Voprosy kiberbezopasnosti*. 2015; 2(10):49-55. Available at: https://elibrary.ru/item.asp?id=23293952 (accessed 16.07.2019). (In Russ., abstract in Eng.)

[31] Chau H.F., Lo H.-K. One-way Functions in Reversible Computations. *Cryptologia.* 1997; 21(2): 139-148. (In Eng.) DOI: 10.1080/0161-1197918858

[32] Zhukov A. Circuits from reversible logic elements: One approach to the study of unidirectionality. In: *Proceeding of The Third International Conference on Information Systems and Technologies (IST'2006).* Minsk, 2006, p. 85. (In Russ.)

[33] Zhukov A. One approach to the study of unidirectionality. *Information Security*. 2018; 1:40-43. Available at: http://lib.itsec.ru/articles2/crypto/odin-podhod-k-izucheniyu-odnonapravlennosti (accessed 16.07.2019). (In Russ.)

Submitted 16.07.2019; revised 25.08.2019; published online 30.09.2019.

About the authors:

Sergey I. Gurov, Senior Researcher, Associate Professor of the Department of Mathematical Methods of Forecasting, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University (1, Leninskie gory, Moscow 119991, Russia), Ph.D. (Phys.-Math.), Associate Professor, ORCID: http://orcid.org/0000-0001-5486-1357, sgur@cs.msu.ru

Aleksey E. Zhukov, Associate Professor of the Department of Information Security (IU-8), Faculty of Informatics and Control Systems, Bauman Moscow State Technical University (5/1 2-nd Baumanskaya Str., Moscow 105005, Russia), Ph.D. (Phys.-Math.), Associate Professor, ORCID: http://orcid.org/0000-0001-5459-3831, aez_iu8@rambler.ru

Dmitry V. Zakablukov, programmer, "Algorithms and Data" LLC (42, bld. 1, Dmitriya Ul'yanova Str., Moscow 117218, Russia), Ph.D. (Phys.-Math.), ORCID: http://orcid.org/0000-0001-6584-6443, dmitriy.zakablukov@gmail.com

Georgy V. Kormakov, student, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University (1, Leninskie gory, Moscow 119991, Russia), ORCID: http://orcid.org/0000-0002-7728-0392, egor2898@mail.ru

All authors have read and approved the final manuscript.