

## РЕВЕРСИВНЫЕ ВЫЧИСЛЕНИЯ: ОБЗОР МИРОВОГО ОПЫТА

Реверсивные вычисления — направление современной информатики, тесно связанное с идеей сверхпроводящих и энергосберегающих высокопроизводительных вычислений. В статье кратко рассматриваются основные наработки мирового опыта в области реверсивных вычислений, главные достижения и неудачи рабочих групп в этом направлении.

**Reversible computing: an overview** / Nepeivoda A.N. (Program System Institute of RAS, Pereslavl-Zalesky, Piotr I st., 4-A, E-mail: nowater@rambler.ru). Reversible computing is a domain of modern computer science that is strongly connected with ideas of low-power high-performance computations. In the paper main results of the investigations in RC are gathered.

### 1. Введение

Неудобство распараллеливания и сложность распределения задач между процессорами в традиционных архитектурах привели к идеям поиска альтернативных архитектур для исполнения массивно параллельных задач. Одновременно поиск решений для уменьшения энергетических затрат на вычисления привел к задаче построения реверсивной машины, предназначенной для исполнения обратимых задач. Современные результаты по моделированию реверсивности показывают, что архитектурные решения для нее естественным образом порождают массивно параллельные системы.

Еще в первой половине XX века Дж. фон Нейман заметил, что потери информации и энергии в системе связаны [59]. Чуть более, чем через десять лет Ландауэр установил, что при потере одного бита информации в окружающую среду выделяется  $E_{diss} = k_B \times T \times \ln 2$  Дж тепла [30]. Эта формула показывает предел производительности традиционных процессоров. То есть при использовании 2-мегаваттной мощности фундаментальный предел производительности идеального классического вычислителя достигает  $2,5 \times 10^{26}$  операций в секунду [14]. А вычислителя, использующего операции с плавающей точкой (принципиально неточные) —  $10^{22}$  операций в секунду (там же). Соображения надежности вычислений указывают на еще более низкие верхние границы (чтобы избежать помех, связанных с тепловым шумом, надо избавиться от тепла) [26, 14, 21].

На общую потерю энергии от нереверсивности не влияет даже использование сверхнизких температур: значение  $k_B \times T \times \ln 2$  Дж формально снижается, но на о

требуется тот же порядок энергии ([24]). Результаты последних экспериментов по проверке принципа Ландауэра приведены в [11]. Одиночная частица помещалась в резервуар, состоящий из двух частей, и из неизвестного положения насильно перемещалась в заранее заданную половину. Выделение тепла при этом в точности соответствовало предсказанному Ландауэром.

В настоящее время реверсивные вычисления — критическая область, вызывающая интерес ученых из США, Японии и европейских стран, однако, несмотря на то, что все больше экспериментов подтверждают существование предела Ландауэра (а значит, и необходимости обратиться к реверсивности), исследования ведутся очень разрозненно и почти не ссылаются друг на друга. Есть ряд классических работ, результаты которых используются всеми, в остальном же типична ситуация, когда исследователи даже не знают о трудах друг друга. Поэтому, хотя утверждается, что удалось экспериментально подтвердить возможность выхода за предел Ландауэра в реверсивном случае [45], неясно, можно ли хотя бы в принципе добиться достаточно эффективной реализации реверсивного компьютера. Чтобы получить возможность приблизительно судить об этом, необходимо хотя бы собрать воедино самые свежие результаты по реверсивности в самых разных областях и оценить, как они могли бы повлиять друг на друга.

Единственный проект, в котором делается попытка как-то интегрировать знания в различных областях реверсивности — это MicroPower [5] — совместный бельгийско-датский проект по разработке реверсивной среды, объявленный на срок 2009–2013 годов. Разработчиками проекта выделено шесть уровней реализации, на которых должна выполняться реверсивность:

- 1) Физическая модель вычислителя
- 2) Архитектура машины
- 3) Языки высокого уровня и алгоритмы

Нарушение реверсивности хотя бы на одном этаже нивелирует эффект преодоления тепловых барьеров. Добавим, что над всеми этими уровнями реализации есть теоретический уровень, также требующий рассмотрения.

За годы, прошедшие после основополагающих работ Беннетта, было сделано немало попыток создать вычислительные модели на всех этих уровнях. Однако исследования велись разрозненно, а рабочие группы зачастую просто не знали друг о друге. Нетрадиционные решения во всех работах по реверсивности, как правило, не уходили дальше физического уровня, а верхние уровни базировались на модификациях классических идей. Кратко рассмотрим, что же было сделано на каждом из них к настоящему моменту.

## 2. Физический уровень

Первым на существование в природе машин, близких к реверсивным, (а значит, и возможность их построения) обратил внимание Беннет [8]. А именно: процессы, происходящие при синтезе белков, обратимы и управление ими осуществляется за счет катализа ферментами. При этом в таком сложном процессе, как синтез ДНК, затраты энергии составляют от 5 до 50  $k_b \times T$  Дж на нуклеотид, тогда как вероятность ошибки при этом у всех организмов составляет  $10^{-9}$ – $10^{-11}$ [16] (заметим, что один нуклеотид содержит информацию, эквивалентную двум битам). Хотя предел Ландауэра в биологических системах не достигается, но пересчет показывает, что подобная точность вычислений (для преодоления теплового шума) на современных MOSFET-технологиях

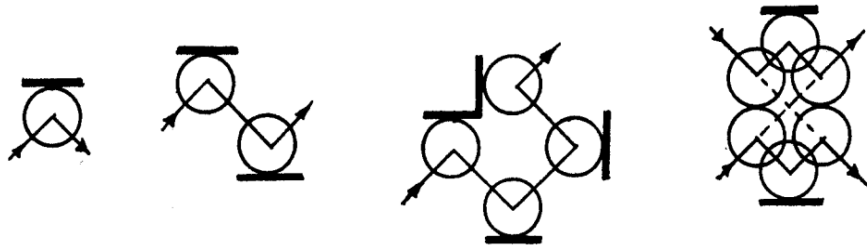


Рис. 1. Биллиардная модель: отражение, сдвиг, сохранение пути, прохождение сквозь

требовала бы минимум  $50 k_b \times T$  Дж на пару битов — следовательно, часть энергии в биологической системе должна восстанавливаться [21, 18].

Также Беннетт обратил внимание, что в так называемых «маятниковых системах» (без управления извне) расходы энергии можно сделать сколь угодно малыми, если растянуть управляющее воздействие по времени [9]. Тоффоли заметил, что если заряд в ячейках будет сохраняться постоянным (количество нулей и единиц будет оставаться неизменным, а меняться лишь их расположение), можно будет достигнуть такого же эффекта [22]. В частности, на этой идее построены модели адиабатных транзисторов традиционного типа, где один бит моделируется двумя с взаимно обратными последовательностями 01 и 10. Однако, как отмечает в [38] Меркле, данная структура страдает тем же недостатком, что и «маятниковые системы»: чтобы потеря энергии действительно почти не происходила, надо производить переход между состояниями как можно более длительное время. Моментальное переключение адиабатной системы затрачивает столько же энергии, сколько и в случае традиционной архитектуры.

Работы по оптимизации адиабатных схем и сопутствующие расчеты показывают, что использование данной технологии целесообразно при низких температурах [20, 19], в задачах, где допустима низкая скорость вычислений. Проводятся конференции, где обсуждается, как данная технология может быть применима к сверхпроводящим машинам [64]. Однако высказываются и скептические мнения: по оценкам некоторых исследователей, адиабатное переключение проигрывает магнитным моделям более, чем в тысячу раз, по затратам энергии [50].

Модели реверсивных процессоров на базе электронных схем используют ряд идей, исторически первой из которых оказалась модель биллиардных шаров Тоффоли–Фредкина [22]. Эту модель тщательно исследовал Меркле [38]: в его интерпретации роль биллиардных шаров играли электроны, а стола — алмазный контейнер, обработанный определенным образом (модель исследовалась при температуре 1 К). Меркле предусмотрел использование управляющих электронов и возможные потери энергии. Большинство современных исследований по реверсивной вычислимости в качестве физической модели используется именно модель биллиардных шаров Тоффоли–Фредкина (например, в [42]).

Блестящую физическую модель реализации реверсивных элементов, не базирующуюся на «биллиардной схеме», предложил также Меркле в [39]: в его «спиральной логике» роль нулей и единиц выполняет наличие или отсутствие электронов на определенных местах винтовой траектории. При сближении электронов настолько, чтобы их поля начали взаимодействовать, посредством этого взаимодействия вычисляется логическая функция. Меркле доказал, что в рамках такой логики можно вычислить гейт Фредкина. По оценкам потерь энергии, спиральная модель считается одной из лучших

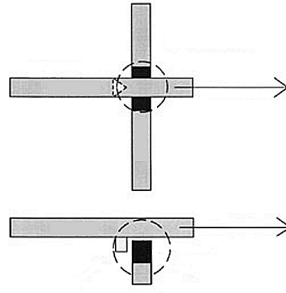


Рис. 2. Модель стержней, вид сверху и сбоку. Черным обозначен шлюз, белым — зонд, пунктирной окружностью — область сцепления.

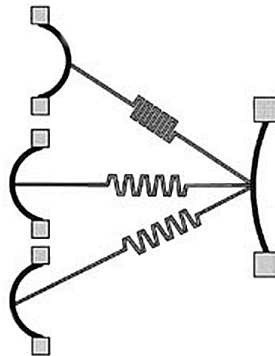


Рис. 3. Модель мембран. Гейт голосования: отклонение мембраны-выхода будет в ту сторону, в которую отклоняется большинство входов.

предложенных [18].

Многие молекулярные процессы также обладают естественной обратимостью (например, равновесные реакции). И, хотя молекулярные модели намного более медленные, чем электронные, но на данном этапе развития реверсивной технологии эта разница не может быть решающей при выборе вычислителя: как уже указывалось, легко реализуемые электронные модели также должны быть медленными, чтобы оставаться реверсивными. Почти исчерпывающий анализ молекулярных моделей дает также Меркле [37]. Он рассматривает два типа механических систем: вида «стержень-стержень» [17] и вида «закрепленные с двух сторон мембраны» [31]. С помощью первых моделируются гейты «и»–«или»; вторая модель позволяет строить гейт голосования с тремя входами и примыкает по своим свойствам к клеточным автоматам. Предлагалась практическая реализация «логики стержней» с помощью нанотрубок [23]; практическая реализация мембранной логики может связана как с магнитными элементами [?], так и с собственно мембранными технологиями [2].

Современные исследования в области молекулярных реверсивных моделей используют ДНК-технологии в комбинации с полимерными химическими реакциями [46]. Была неоднократно доказана тьюринг-полнота вычислителей, работающих на этих принципах, однако смешанная ДНК-полимерная модель (а также чистые ДНК-модели) существенно опираются на свойства стохастических вычислений (см., например, [46, 49]). Вторая проблема ДНК-направления (с точки зрения реверсивности) состоит в том, что оно почти не соприкасается с основным потоком по исследованию реверсивности и зачастую моделирует классическую необратимую логику (классический метод резолю-

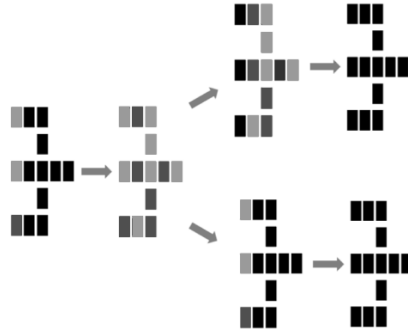


Рис. 4. Магнитный гейт голосования. Светло-серые блоки — положительно заряженные. Темно-серые — отрицательно. Сверху показана нереверсивная схема размагничивания, снизу — реверсивная.

ций в стохастическом процессе [49]; двоичные элементы [34]; исчисление силлогизмов [48]). Есть единичные работы, посвященные моделированию гейта Фредкина на ДНК-машинах, но в них не используется четырехзначность системы нуклеотидов — логика лишь двоичная [61, 54].

Модели магнитных реверсивных вычислителей во многом сходны с идеями, описанными Меркле в [37] (например, моделирование реверсивных гейтов голосования в [29]). Пока что в вышеуказанной работе признается, что для корректной работы системы гейтов необходимо размагничивание с начала, а не с конца, в силу чего реверсивность теряется. Однако подход представляется перспективным при переходе к вычислительной модели клеточного автомата [12], лишенной данного недостатка. Помимо прочего, магнитные частицы обладают естественной памятью, позволяющей использовать соответствующие логические элементы (см. [44], хотя здесь эта идея применяется к мембранным системам).

### 3. Уровень архитектуры

Направление разработки реверсивных архитектур, близких к традиционным, основывается на классической тьюринговой модели вычислений. В широких кругах принято считать, что первым поднял вопрос о симуляции обычных машин Тьюринга с помощью реверсивных Беннетт[8]. На самом деле первая публикация об этом была на десять лет раньше (Лесерф, [32], показал, что проблема останова для машин Тьюринга с обратным проходом неразрешима, поскольку те способны смоделировать все традиционные машины Тьюринга). А задача построения по (обычной) машине Тьюринга обратной рассматривалась в классической работе [36].

В теории реверсивных машин Тьюринга принято считать, что нельзя одновременно в одном состоянии на одном символе записать символ на ленту и переместиться. Четверки состояний бывают двух следующих типов:

- 1)  $\langle p, a, b, q \rangle$  — находясь в состоянии  $p$  и считав символ  $a$ , пишем символ  $b$  и переходим в состояние  $q$
- 2)  $\langle p, *, s, q \rangle$  — находясь в состоянии  $p$ , двигаемся по ленте согласно  $s$  ( $-1$  — влево,  $+1$  — вправо) и переходим в состояние  $q$ .

Обычные машины Тьюринга тоже легко представляются через такие четверки — достаточно добавить им состояний.

Реверсивная машина Тьюринга обладает тем свойством, что если она пришла в одно и то же состояние из двух разных, то гарантируется, что эти состояния принадлежали к записываемому типу и записали на ленту разные символы (заметим: это сразу же делает возможным просчет сколь угодно длинной цепочки шагов назад). Многоленточные РМТ определяются аналогично многоленточным обычным МТ с тем же ограничением.

В ранних работах Беннетта [8, 9] описывался процесс преобразования всякого неинъективного вычисления в инъективное. Приведем классические результаты и современные оценки эффективности процедур «реверсизации» [7].

*Теорема 1.* 1) («Вложение по Ландауэру» — на самом деле впервые совершенно Лесерфом) По одноленточной машине Тьюринга  $T$  можно построить двухленточную реверсивную машину Тьюринга  $L(T)$ , такую, что  $|L(T)|(x) = \lambda x. (|T|(x), \text{trace}(T, x))$ .

2) («Трюк Беннетта») По одноленточной машине Тьюринга  $T$  можно построить трехленточную реверсивную машину Тьюринга  $B(T)$ , такую, что  $|B(T)|(x) = \lambda x. (x, |T|(x))$ .

3) Пусть  $T$  —  $k$ -ленточная реверсивная машина Тьюринга с алфавитом размера  $m$ . Тогда существует  $T'$  — одноленточная реверсивная машина Тьюринга с алфавитом из трех символов, такая, что

$$|T|(x_1, \dots, x_k) = (y_1, \dots, y_k) \iff |T'|(e(\langle x_1, \dots, x_k \rangle)) = e(\langle y_1, \dots, y_k \rangle).$$

4) Пусть  $T$  —  $k$ -ленточная реверсивная машина Тьюринга, требующая  $T$  времени. Тогда эквивалентная ей РМТ с двумя лентами потребует  $O(T \log T)$  времени; с одной лентой —  $O(T^2)$  времени [4].

Также в вышеуказанных работах было доказано, что всякая инъективная функция моделируется на реверсивных машинах Тьюринга (в смысле данного определения).

Современные исследователи отмечают, что ни вложение по Ландауэру, ни трюк Беннетта или игра в камни не сохраняют семантику программы и, следовательно, не обеспечивают полной реверсивности [7].

С построением универсальной РМТ  $U$ , по коду другой РМТ  $T$  и входным данным  $x$  находящей  $|T|(x)$ , возникло то затруднение, что процедура  $\beta$ -редукции (подстановки) не является инъекцией. Поэтому явное построение универсальной РМТ оказалось достаточно трудоемким. Самая малая из них — с 17 состояниями и 5 рабочими символами (Morita, [41]). При доказательстве была использована эмуляция тьюринг-полных систем с циклическими тегами, но результат работы этой машины представляет собой не все содержимое ленты после останова, поэтому, как отмечают другие исследователи [7], нельзя считать, что на этой машине достигается полная реверсивность. Другой подход к универсальным РМТ осуществили Axelsen и Glück [7]: РМТ в их исполнении работает на трех лентах, одна из которых промежуточная, одна для вывода результата и одна с неизменными входными данными (по принципу Беннетта). Явная небиективность машины Морита здесь маскируется, но также присутствует в неявном виде.

К сожалению, все архитектурные решения в этом направлении базируются на идеях трюка Беннетта и переносе традиционных идей на реверсивный случай, из-за чего возникает ряд неустраняемых проблем. Данные проекты объединяют следующие черты:

1) За основу машинного языка берутся стандартные команды нереверсивных ассемблеров (сложение/вычитание, умножение/деление, чтение/запись), модифицируемые таким образом, чтобы информация не терялась. Чаще всего это достигается за счет сохранения одного из операндов и, в случае, если нужная функция все равно не инъективна, доопределения ее до биекции. Так, в [55] используется двоичный сдвиг вместо умножения и деления на 2.

2) За основу реализации берутся либо стандартные CMOS-элементы, модифицированные под адиабатные принципы [58], либо двоичные реверсивные гейты [60].

3) Во всех случаях, главной проблемой таких архитектур остается минимизация количества мусора. В Pendulum [58] вводится «мусорный стек» для сохранения операндов неинъективных функций: его содержимое может быть сброшено или переработано обратным проходом, в зависимости от целей программиста. В других архитектурах обратный проход по Беннетту вводится принудительно, тем самым набор инструкций дублируется (чтобы утилизировать мусорные биты в исходные константные данные) [55]. Но это приводит в точности к той же ситуации, что и при в магнитных моделях — блоки становится невозможно соединять между собой!

Не всегда соответствующие ассемблеры бывают описаны аккуратно. Почти все авторы забывают о том, что вычисления производятся в кольце  $\mathbb{Z}_{2^n}$ , и декларируют реверсивность, в частности, умножения, за исключением случаев нулевых операндов (что было бы верно, если бы соответствующая алгебраическая структура была полем). Таким образом, либо требуется вводить дорогостоящую нереверсивную проверку корректности, либо реверсивность грубо нарушается. Исключения составляют работы, где нереверсивные операции сохраняют оба операнда [60].

Более смелое направление разработки реверсивных процессоров — вычислители на базе клеточных автоматов. Теоретическая модель реверсивного клеточного автомата обладает рядом очень естественных свойств, что выгодно отличает ее от модели реверсивной машины Тьюринга.

*Теорема 2.* 1) *Всякий нереверсивный клеточный автомат размерности  $n$  можно промоделировать реверсивным клеточным автоматом размерности  $n+1$  [56].*

2) *Можно построить универсальный реверсивный клеточный автомат размерности 1 [40].*

3) *Задача определения реверсивности клеточного автомата размерности 1 разрешима и полиномиальна [3]. Задача определения реверсивности клеточного автомата размерности 2 и более неразрешима [28].*

4) *Клеточный автомат является реверсивным тогда и только тогда, когда в его конфигурации не встречаются «сады Эдема» (таким образом, на множестве клеточных автоматов из инъективности автоматически следует биективность, и «трюк Беннетта» невозможен, даже в обобщенном виде) [43].*

5) *Реверсивные клеточные автоматы размерности 1 могут достигать сколь угодно высоких степеней неразрешимости задачи достижимости [53].*

Возможность реализации клеточных автоматов на подходящей материальной базе (магнитные носители) рассматривалась в [57] (модель Изинга). Биллиардная модель Тоффоли–Фредкина была описана на основе блочных клеточных автоматов с блоками  $2 \times 2$  со следующими правилами [35]:

- 1) Если в блоке лишь одна клетка, блок поворачивается на 180 градусов.
- 2) Если в блоке ровно две клетки по диагонали, он поворачивается на 90 градусов.
- 3) Иначе блок неизменен.

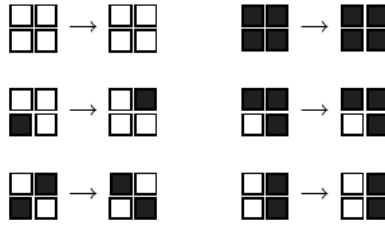


Рис. 5. Схема блочного клеточного автомата Марголуса.

Эта модель свободна от недостатков, связанных с копированием данных, однако возникает проблема эффективного программирования на клеточных автоматах. Пока что неизвестны способы ее разрешить иначе, чем переборными методами.

Совсем кратко скажем о реверсивных аналоговых машинах. То, что аналоговые машины часто обладают естественной реверсивностью, было замечено, в частности, исследователями MicroPower [15]. Однако дальше этого наблюдения направление не разрабатывалось, из-за неуниверсальности и принципиальной неточности аналоговых машин. С учетом того, что действия над действительными числами с плавающей точкой также принципиально неточны, разработка аналогового реверсивного вычислителя могла бы стать им достойной альтернативой.

#### 4. Языки программирования

Исторически первый полностью (локально и глобально) реверсивный язык программирования Janus был предложен в [33]. Помимо использования строго инъективных встроенных функций и запрета на повторение переменной в левой и правой частях присваивания Janus содержал условия-охраны на начало цикла и конец условного оператора (что позволяло осуществлять проход по программе в две стороны в любом ее месте) (сходный способ обращения программ предлагался, в частности, в [65]). Благодаря проекту MicroPower, Janus стал наиболее известным реверсивным языком программирования: в частности, на нем было написано обратимое быстрое преобразование Фурье и его же интерпретатор [63].

Для того, чтобы сделать инъективными возможно большее количество стандартных функций (например, умножение, нарушающее условие обратимости в нуле), использовалась конструкция «обратимого расширения», подробно описанная (и обобщенная) в [62] (один из частных случаев которой — вышеупомянутый «трюк Беннетта»).

*Теорема 3.* Пусть даны произвольная функция  $f$  и операция  $\circ$ , допускающая правое сокращение, то есть

$$\forall a_1, a_2, b (a_1 \circ b = a_2 \circ b \Rightarrow a_1 = a_2).$$

Тогда функция  $g(x, y) = (x \circ f(y), y)$  является инъекцией.  $g(x, y)$  — и есть «обратимое расширение»  $f$ .

Заметим, однако, что конструкция «обратимого расширения», представленная в [62] как стандарт де-факто для всех реверсивных языков программирования и используемая в Janus и семействе подобных языков (например, обратимом Forth), порождает не группу, а полугруппу с правыми сокращениями. Еще одна важная проблема, возникающая из-за использования обратимого расширения — существенная неразрешимость



задачи обратного исполнения, делающая использование реверсивного Janus и всех родственных языков крайне непрактичным. Например, можно возвести число в квадрат, но произвести обратную операцию напрямую не получится, потому что умножение превращается в обратимое расширение, и для обратной операции придется вводить один из операндов (то есть собственно сам искомый корень!). В этом смысле гораздо более практичны более традиционные техники обратного исполнения программ, написанных на традиционных языках программирования, такие, как URA (universal resolving algorithm) [1].

Попытка построения реверсивного функционального языка, лишенная недостатка обратимого расширения, имеется, например, в [13] (свежие улучшения см. в [27]). Используется свойство двусторонности секвенционального вывода в интуиционистской логике и на ее базе строится модификация классического  $\lambda$ -исчисления с типами (прямая сумма и прямое произведение). Соответствующий теоретический аппарат — симметрические моноидальные категории [51] — объявлен средством описания реверсивной логики, однако он описывает морфизмы коммутативных полугрупп, а не групп [25], и подходит для описания линейной логики Жирара, на базе которой уже с 80-х гг. прошлого века было построено типизированное  $\lambda$ -исчисление [10].

## 5. Интеграция уровней друг с другом

Приведем краткую таблицу совместимости разных физических моделей с архитектурными идеями. В тех случаях, когда исследования по совмещению уровней не проводились, ставится «неизвестно». Если существуют очень серьезные ограничения на применимость модели, ставится «условно»: таковы, например, все случаи, когда вычислитель вынужденно откатывается к входным данным для сохранения реверсивности, и тем теряет возможность использоваться в более сложных схемах как примитив.

Физическая модель	Традиционная архитектура	Клеточная архитектура
Адиабатные CMOS	условно	нет
Биллиардная	условно	да
Спиральная	условно	неизвестно
ДНК-полимерная	нет	условно
Стержневая	да	неизвестно
Мембранная	условно	неизвестно
Магнитная	условно	да

Что наблюдается сейчас в «мэйнстримовом» направлении исследований по реверсивности? Почти все разработки основаны на идее, явной или скрытой, использования трюка Беннета о копировании входа. Проблема здесь не столько в самой этой операции — эксперименты по преодолению предела Ландауэра рассматривали обратимость с копированием, и там достигался тот же эффект [45], — сколько в побочных эффектах, возникающих при использовании обратимых расширений. Обратные исполнения становятся бессмысленными почти во всех случаях, размеры схем неизбежно удваиваются, совместимость ухудшается, а перенос классических решений без исправления теории служит неиссякаемым источником ошибок. Чтобы преодолеть этот тупик, просто необходимо переходить к нетрадиционным экспериментальным моделям, основывающимся на идеях аналоговых или клеточных машин.

Второй важный момент в моделировании реверсивности следующий. Самые свежие оценки энергетических и ресурсных затрат на полную реверсивность показывают, что

даже для построения цифрового сумматора эффективнее смешанная модель [52], и использование нереверсивных операций там, где возникает естественная необходимость, снижает общую стоимость вычислений. Таким образом, нет необходимости дополнять полугруппу преобразований до моноида с правым сокращением, пытаясь добиться полной обратимости — выделение истинно групповых операций на реверсивный процессор более целесообразно с точки зрения физики.

Эти моменты уже осознаны некоторыми группами исследователей, однако из-за очень плохой интеграции высказанные ими предупреждения не были услышаны остальными. Таким образом, в настоящее время еще слишком рано говорить о возможности появления работающей реверсивной машины.

## СПИСОК ЛИТЕРАТУРЫ

1. *S. M. Abramov, R. Gluck*. "Principles of inverse computation and the universal resolving algorithm". // *The Essence of Computation: Complexity, Analysis, Transformation*, LNCS, Vol. 2566, p. 269–295. Springer-Verlag, 2002.
2. *A. Alhazov, K. Morita* "On Reversibility and Determinism in P Systems" // *Membrane Computing, 10th International Workshop, WMC 2009, Lecture Notes in Computer Science* vol. 5957, 2010, p. 158–168.
3. *Amoroso, S.; Patt, Y. N.* «Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures» // *Journal of Computer and System Sciences* 6: p. 448–464, 1972.
4. *Axelsen, H. B.* (2012) "Time Complexity of Tape Reduction for Reversible Turing Machines" // *Lecture Notes in Computer Science*, 2012, Vol. 7165(2012), p. 1–13.
5. *Axelsen, H. B., Glück, R. A., De Vos, M. K. Thomsen* "MicroPower: Towards Low-Power Microprocessors with Reversible Computing". // *ERCIM News* 01/2009; 2009.
6. *Axelsen, H.B. Glück, R.* (2011) "What Do Reversible Programs Compute?". // *Foundations of Software Science and Computation Structure*, p. 42–56.
7. *Axelsen, H. B., Glück, R. A.* "A simple and efficient universal reversible Turing machine" // *LATA'11 Proceedings of the 5th international conference on Language and automata theory and applications* P. 117–128. 2011.
8. *Bennett, C.H.* (1973). "Logical reversibility of computation". // *IBM J. Res. Develop.* 17 (6): p. 525–532.
9. *Bennett, C. H.* "The thermodynamics of computation — a review". // *Int. J. Theoret. Phys.*, 21(1982), p. 905-940.
10. *N. Benton, G. Bierman, V. Paiva, M. Hyland* "Linear  $\lambda$ -calculus and categorical models revisited" // *Computer Science Logic*, Springer-Verlag, 1992.
11. *A. Berut, A. Arakelyan, A. Petrosyan, S. Ciliberto, R. Dillenschneider, E. Lutz* "Experimental verification of Landauer's principle linking information and thermodynamics" // *Nature* 483, p. 187–189. 2012.

12. *Bhanja, S.* "A review of magnetic cellular automata systems". // Circuits and Systems (ISCAS), 2011 IEEE International Symposium, P. 2373–2376
13. *Bowman, W.J., James, R.P., Sabry, A.* "Dagger Traced Symmetric Monoidal Categories and Reversible Programming". // Reversible Computation (2011)
14. *E. DeBenedictis,* "Will Moore's Law be Sufficient,"// Proceedings of the 2004 ACM/IEEE Conference on Supercomputing, 2004.
15. *A. De Vos* "Reversible Computing: fundamentals, quantum computing and applications". Wiley-VCH, Weinheim. 2010. 261 p.
16. *Drake J.W., Charlesworth B., Charlesworth D., Crow J.F.,* 1998. Rates of spontaneous mutation // Genetics. V. 148. Vol. 4. P. 1667–1686.
17. *Drexler, K. E.* "Rod Logic and Thermal Noise in the Mechanical Nanocomputer"// Molecular Electronic Devices, North- Holland 1988.
18. *Frank, M. P.* "The Reversible Computing Question: A Crucial Challenge for Computing". // Workshop on Frontiers of Extreme Computing, Santa Cruz Oct. 24–27, 2005. Invited talk.
19. *Frank, M. P.* "Towards a More General Model of Reversible Logic Hardware". // Superconducting Electronics Approaching the Landauer Limit and Reversibility (SEALeR) workshop, Annapolis, MD, Mar. 15–16, 2012. Invited talk.
20. *Frank, M. P.* "Common mistakes in adiabatic logic design and how to avoid them"// Embedded Systems and Applications, p. 216–222, 2003.
21. *Frank, M. P.* "Introduction to Reversible Computing: Motivation, Progress, and Challenges"// Proceedings of the 2nd conference on Computing frontiers, p. 385–390.
22. *Fredkin, E., and Toffoli, T.* "Conservative logic"// Int. J. Theoret. Phys., 21, p. 219–253 (1982).
23. *J. Storrs Hall* "Nanocomputers and Reversible Logic"//Nanotechnology, 1994, Vol. 5, p. 157–167.
24. *Halliday, D. and Resnick, R.* (1988) The Fundamentals of Physics, Extended Third Edition (Wiley).
25. *M. Hasegawa* "On Traced Monoidal Closed Categories"// J. Mathematical Structures in Computer Science Vol. 19 Iss. 2, 2009, P. 217–244.
26. *J. Izydorczyk* "Three steps to the thermal noise death of Moore's law"// J. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 18 Iss. 1, January 2010. P. 161–165.
27. *James, R.P., Sabry, A.* "Isomorphic Interpreters from Logically Reversible Abstract Machines". // Reversible Computation (2011)
28. *Kari, J.* "Reversibility of 2D cellular automata is undecidable"// Cellular automata: theory and experiment (Los Alamos, NM, 1989), Physica D: Nonlinear Phenomena, 45, p. 379–385. 1990.

29. *B. Lambson, D. Carlton, J. Bokor.* "Exploring the Thermodynamic Limits of Computation in Integrated Systems: Magnetic Memory, Nanomagnetic Logic, and the Landauer Limit" // Physical Review Letters, vol. 107, pp. 010604(1)–01604(4), 2011.
30. *Landauer, R.* (1961). "Irreversibility and Heat Generation in the Computing Process". // IBM J. Res. Develop. 5 (3) p. 183–191.
31. *Landauer, R.* (1986) "Reversible Computation". //Der Informationsbegriff in Technik und Wissenschaft, (R. Oldenbourg, Munchen, 1986. p. 139–158.
32. *Lecerf, Y.* «Machines de Turing reversibles. Recursive insolubilité en  $n \in \mathbb{N}$  de l'équation  $u = \theta^n$ , où  $u$  est un "isomorphisme de codes"» // Comptes Rendus, 257(1963), p. 2597–2600.
33. *Lutz, C.* "Janus: a time-reversible language. Letter to R. Landauer". <http://www.cise.ufl.edu/~mpf/rc/janus.html>, 1986.
34. *Margolin, A. A., Stojanovic, M. N.* "Boolean calculations made easy (for ribozymes)" // Nature Biotechnology 23, p. 1374–1376 (2005)
35. *Margolus, N.* "Physics-like models of computation" // Physica D: Nonlinear Phenomena 10: p. 81–95, 1984.
36. *McCarthy, J.* "The inversion of functions defined by Turing machines". // Automata Studies , 177–181. Princeton University Press, 1956.
37. *Merkle, R. C.* "Two Types of Mechanical Reversible Logic" // Nanotechnology, Volume 4, 1993, p. 114–131.
38. *Merkle, R. C.* "Reversible Electronic Logic Using Switches" // Nanotechnology, Volume 4, 1993, p. 21–40.
39. *Merkle, R. C., Drexler, K. E.* "Helical logic" // (1996) Nanotechnology Vol. 7, p. 325–339.
40. *K. Morita.* "Computation universality of one-dimensional reversible cellular automata". // Information Processing Letters, 42. p. 325–329, 1992
41. *Morita, K., Yamaguchi, Y.* "A universal reversible Turing machine" // Proc. 5th Int. Conf. on Machines, Computations, and Universality, Springer-Verlag, p. 90–98 (2007)
42. *Morita, K.,* "Universality Issues in reversible Computing Systems and Cellular Automata" // Electronic Notes in Theoretical Computer Science, Vol. 253 Iss. 6, March, 2010, P. 23–31.
43. *Myhill, J.* "The converse of Moore's Garden-of-Eden theorem" // Proceedings of the American Mathematical Society 14: 685–686. (1963)
44. *Ts. Ogiro, A. Alhazov, Ts. Tanizawa, K. Morita* "Universality of 2-State 3-Symbol Reversible Logic Elements — A Direct Simulation Method of a Rotary Element". // Natural Computing, PICT 2, Springer Japan, 2010, p. 252–259.

45. *A.O. Orlov, C.S. Lent, C.C. Thorpe, G.P. Boechler, and G.L. Snider* (2012). Experimental Test of Landauer's Principle at the Sub-kBT Level // *Jap. J. Appl. Phys.* 51(2012)
46. *Qian, L., Soloveichik, D., Winfree, E.* "Efficient Turing-Universal Computation with DNA Polymers"// *DNA computing and molecular programming. Lecture Notes in Computer Science* (6518). Springer , Berlin, p. 123–140.
47. *Raiter, K. R., Cockburn, B. F.* "An investigation into three-level ferroelectric memory"// *Proceedings of the 2005 IEEE International Workshop on Memory Technology, Design, and Testing (MTDT'05)*, p. 38–43.
48. *Ran, T., Kaplan, S., Shapiro, E.* "Molecular implementation of simple logic programs"// *Nature Nanotechnology* 4, p. 642–648 (2009)
49. *Rodriguez-Paton A., de Murieta, I. S., Sosik, P.* "Autonomous Resolution Based on DNA Strand Displacement"// *DNA Computing and Molecular Programming. Lecture Notes in Computer Science*, 2011, Vol. 6937/2011, p. 190–203,
50. *K. Roy, S. Bandyopadhyay, J. Atulasimha,* «Hybrid spintronics and straintronics: A magnetic technology for ultra-low-energy computing signal processing» // *Applied Physics Letters*, 2011.
51. *Selinger, P.* "A survey of graphical languages for monoidal categories". // *New Structures for Physics, Lecture Notes in Physics*, vol. 813, p. 289–355. Springer Berlin / Heidelberg (2011)
52. *S.Shauck, N. Grumman,* "Reversible Logic: The cost of full reversibility"// *SEALeR Workshop*. 2012.
53. *K. Sutner* "The Complexity of Reversible Cellular Automata"// *Theoretical Computer Science* 325 (2004) p. 317–328.
54. *Thapliyal H., Srinivas, M. B.* "An Extension to DNA Based Fredkin Gate Circuits: Design of Reversible Sequential Circuits using Fredkin Gates"// *Proceedings of SPIE* Vol. 6050, p. 196–202, 2005.
55. *Thomsen, M. K.; Gluck, R., Axelsen, H. B.* "Towards Designing a Reversible Processor Architecture". // *Preliminary Proceedings of the Workshop on Reversible Computing*, pages 46–50, 2009.
56. *Toffoli, T.,* "Computation and Construction Universality of Reversible Cellular Automata"// *J. Comput. Syst. Sci.* 15 (1977), p. 213–231.
57. *Toffoli, T.; Margolus, N.* "Cellular Automata Machines: A New Environment for Modeling". MIT Press. 1987.
58. *Vieri, C. J.* "Pendulum — a reversible computer architecture". Thesis (M.S.)–Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 1995.

59. *von Neumann, J.*, 1949, The Role of High and of Extremely High Complication. Fourth University of Illinois lecture, in *Theory of Self-Reproducing Automata*, A.W. Burks (ed.), Champaign, Illinois: University of Illinois Press, 1966, pp. 64–73.
60. *R. Wille, M. Soeken, D. Grobe, E. Schonborn, R. Drechsler* "Designing a RISC CPU in Reversible Logic" // 41st International Symposium on Multiple-Valued Logic (ISMVL), p. 170–175, Tuusula, 2011.
61. *Wood, H., Junghuei Chen, D.*, "Fredkin gate circuits via recombination enzymes". // *Proceedings of Evolutionary Computation, 2004(CEC2004)*, 2004.
62. *Yokoyama, T., Axelsen, H.B. Glück, R.* "Principles of a reversible programming language". // *Proceedings of the 5th Conference on Computing Frontiers*, 2008.
63. *T. Yokoyama, R. Gluck.* "A reversible programming language and its invertible self-interpreter". // *Partial Evaluation and Program Manipulation. Proceedings*, p. 144–153. ACM Press, 2007.
64. Superconducting Electronics Approaching the Landauer Limit and Reversibility (SEALeR) Workshop. March 2012, Annapolis, MD. [http://terpconnect.umd.edu/~kosborn/index\\_files/Page699.html](http://terpconnect.umd.edu/~kosborn/index_files/Page699.html)
65. *Грис Д.* "Наука программирования". М. , 1984. 416 с.