

УДК 004.942

## ПРОЕКТИРОВАНИЕ ИНСТРУМЕНТАЛЬНОЙ ПРОГРАММНОЙ СИСТЕМЫ ПОДДЕРЖКИ МЕТОДОЛОГИИ ПРОЕКТИРОВАНИЯ СЛОЖНЫХ СИСТЕМ НА ОСНОВЕ АВТОМАТНЫХ МОДЕЛЕЙ С ИСПОЛЬЗОВАНИЕМ АЛГЕБРЫ ГИПЕРРАЗМЕРНЫХ ВЕКТОРОВ

<sup>1</sup>Мартышкин А.И., <sup>1</sup>Трокоз Д.А., <sup>1</sup>Пашенко Д.В., <sup>1</sup>Калашников В.А., <sup>2</sup>Синев М.П.

<sup>1</sup>ФГБОУ ВО «Пензенский государственный технологический университет»,

Пенза, e-mail: Alexey314@yandex.ru, dmitriy.trokoz@gmail.com,

dmitry.pashchenko@yandex.ru, mymailfordev@gmail.com;

<sup>2</sup>ФГБОУ ВО «Пензенский государственный университет», Пенза, e-mail: mix.sinev@gmail.com

В данной статье рассматривается проблема развития микроэлектроники из-за физического и экономического предела по закону Мура. В качестве решения данной проблемы приводятся алгоритмы на базе алгебры гиперразмерных векторов, так как алгоритмы на базе алгебры гиперразмерных векторов устойчивы к появлению в разрядах регистров мусорных значений, которые критичны при вычислениях классическими алгоритмами. Использование алгоритмов на базе гиперразмерных векторов позволяет увеличить частоты вычислительного устройства, не изменяя токопотребления, и способно давать верный результат несмотря на появление мусорных значений в регистрах вычислительного устройства. В статье поднимается вопрос малой изученности методов кодирования сложных систем в формате автоматов на базе алгебры гиперразмерных векторов. В данной работе представлена инструментальная программная система поддержки методологии проектирования сложных систем на основе автоматных моделей с использованием алгебры гиперразмерных векторов. Предложены методы формализованного описания систем в виде систем канонических уравнений для кодирования в гиперразмерный вектор и описание алгоритмов трансформации формализованного описания с использованием алгебры гиперразмерных векторов, а также алгоритмы вычисления текущего состояния графа, представленного как гиперразмерный вектор. Продемонстрирована работа программной инструментальной среды путем кодирования и симуляции целевой системы.

**Ключевые слова:** гиперразмерные векторы, алгебра гиперразмерных векторов, моделирование систем, программная среда моделирования, закон Мура

## DESIGNING THE TOOLING SOFTWARE SYSTEM FOR SUPPORTING THE METHODOLOGY OF DESIGNING COMPLEX SYSTEMS BASED ON AUTOMATIC MODELS USING THE ALGEBRA OF HYPERDIMENSIONAL VECTORS

<sup>1</sup>Martyshkin A.I., <sup>1</sup>Trokoz D.A., <sup>1</sup>Paschenko D.V., <sup>1</sup>Kalashnikov V.A., <sup>2</sup>Sinev M.P.

<sup>1</sup>Penza State Technological University, Penza, e-mail: Alexey314@yandex.ru, novoselich93@mail.ru,

dmitry.pashchenko@yandex.ru, mymailfordev@gmail.com;

<sup>2</sup>Penza State University, Penza, e-mail: mix.sinev@gmail.com

This article discusses the problem of the development of microelectronics due to the physical and economic side-chapel according to Moore's law. As a solution to this problem, algorithms based on the algebra of hyperdimensional vectors are presented, since algorithms based on the algebra of hyperdimensional vectors are resistant to the appearance of garbage values in the register bits, which are critical in calculations by classical algorithms. The use of algorithms based on hyperdimensional vectors allows you to increase the frequencies of the computing device without changing the current consumption and are able to give the correct result despite the appearance of garbage values in the registers of the computing device. The article raises the question of the low level of knowledge of methods for coding complex systems in the format of automata based on the algebra of hyperdimensional vectors. This paper presents an instrumental software system for supporting the methodology of designing complex systems based on automata models using the algebra of hyperdimensional vectors. Methods of formalized description of systems in the form of systems of canonical equations for coding into a hyperdimensional vector and a description of algorithms for transforming a formalized description using the algebra of hyperdimensional vectors, as well as algorithms for calculating the current state of a graph represented as a hyperdimensional vector are proposed. Demonstrated the work of the software tool environment by coding and simulation of the target system.

**Keywords:** hyperdimensional vectors, algebra of hyperdimensional vectors, system modeling, simulation software, Moore's law

В настоящее время четко прослеживается вектор развития индустрии вычислительной техники. С каждым годом вычислительные машины становятся всё более производительными, а их габариты уменьшаются. Вместе с последними экспоненциально растут вычислительные мощности, позволяющие обрабатывать все большие объемы информации. Однако сегодня про-

изводители высокопроизводительных вычислительных систем подошли к пределу по закону Мура. Дальнейшее уменьшение размеров транзисторов будет невозможно физически и нецелесообразно экономически. Уже сейчас производители центральных процессорных устройств (ЦПУ) увеличивают производительность за счет увеличения потребляемой энергии и опти-

мизации в архитектуре. Попытки снизить токопотребление при сохранении высокой производительности приводят к росту числа ошибок в значимых битах регистров ЦПУ и последующему критическому сбою всей системы. Однако алгоритмы, построенные на алгебре гиперразмерных векторов, способны справиться с такими ошибками, так как потеря значимости нескольких бит не влияет на валидность результата вычислений [1]. Данное обстоятельство позволяет создать вычислительное устройство с аппаратной поддержкой алгебры гиперразмерных векторов, которое будет работать на высоких частотах, но с низким токопотреблением, что, в свою очередь, увеличит показатель полезных вычислений на ватт потребленной энергии.

Исходя из вышеописанного, в данной работе представлена разработанная программная инструментальная среда, которая производит симуляцию моделей при помощи алгебры гиперразмерных векторов. Объектом исследования являются алгоритмы кодирования и вычисления состояний автомата при помощи алгебры гиперразмерных векторов. Разработанные алгоритмы позволяют успешно закодировать в вектор любой объект для дальнейших вычислений. Это позволит проектировать эвристические системы на базе нейросети, которая сможет принимать любые данные для вычислений, закодированные в гиперразмерный вектор, что решает проблему позиционирования данных [2].

### Материалы и методы исследования

В качестве способа кодирования целевых систем используется автоматная модель. Абстрактные автоматы – вычислительные машины, представленные в виде математических моделей, осуществляющие обработку информации по заданному алгоритму [3–6]. Самым распространённым способом представления абстрактных автоматов является графический способ. Автомат представляется как граф, где вершинами представляются все возможные состояния автомата, а стрелками – переходы между состояниями [7, 8]. В качестве примера графического представления автомата на рис. 1 продемонстрирован граф автомата D-триггера.

Разработанная инструментальная программная система позволяет моделировать системы путем кодирования в виде абстрактного автомата. В отличие от большинства существующих решений, инструментальная система моделирования представляет целевую систему в векторно-символьной архитектуре.

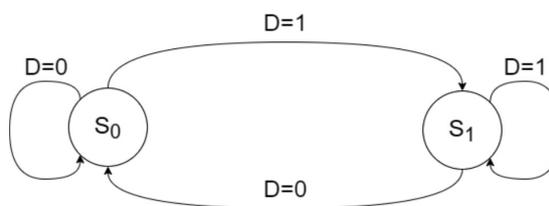


Рис. 1. Графическое представление работы D-триггера

Гиперразмерный вектор представляет собой двоичный вектор высокой размерности, который содержит случайную последовательность нулей и единиц, полученную с использованием равномерного закона распределения. Основной метрикой, которая используется при сравнении двух гиперразмерных векторов, является расстояние Хэмминга. Считается, что два гиперразмерных вектора не пересекаются, если расстояние Хэмминга  $\geq 0,5$ . Если расстояние Хэмминга  $< 0,5$ , то это говорит о том, что сравниваемый HD-вектор входит в результирующий.

Для симуляции поведения целевой системы в программе используются операции алгебры гиперразмерных векторов, такие как сложение векторов, умножение векторов, очищение результирующего вектора от шума. Одним из преимуществ использования векторно-символьной архитектуры является оптимальная алгоритмическая сложность вычисления активных состояний при представлении целевой системы в формате недетерминированного автомата.

Возможность представлять все активные состояния автомата одним гиперразмерным вектором описывается свойствами алгебры гиперразмерных векторов.

### Описание элементов интерфейса

Интерфейс программы состоит из окна ввода блока управления, окна ввода системы канонических уравнений и окна симуляции, где отображается граф автомата целевой системы [9, 10]. Окно программы продемонстрировано на рис. 2.

Блок управления программой состоит из следующих кнопок:

- 1) кнопка сохранения закодированной целевой системы в файл;
- 2) кнопка загрузки целевой системы из файла;
- 3) кнопка запуска кодирования системы уравнений в граф автомата;
- 4) таблица сигналов для установки их значений.

На рис. 3 продемонстрирован блок кнопок управления программой.



Рис. 2. Окно системы моделирования систем



Рис. 3. Кнопки управления программой

Блок управления симуляцией состоит из следующих кнопок:

- 1) кнопка запуска симуляции;
- 2) кнопка пошаговой симуляции;
- 3) кнопка остановки симуляции, если симуляция была запущена по кнопке 1;
- 4) кнопка сброса системы в исходное состояние.

На рис. 4 продемонстрирован блок кнопок управления симуляцией



Рис. 4. Кнопки управления симуляцией

#### Описание и демонстрация работы среды моделирования

Были разработаны алгоритмы кодирования целевой системы путём парсинга систе-

мы канонических уравнений (СКУ) [11, 12]. СКУ задаются как набор формул в нормальной дизъюнктивной форме (ДНФ). В качестве операндов самого уравнения выступают конъюнкты, состоящие из предыдущего состояния и выходного сигнала предыдущего состояния, само же уравнение приравнивается к новому состоянию. Количество конъюнктов определяет количество вершин графа автомата, из которых осуществляется переход.

После ввода СКУ пользователем происходит кодирование в векторно-символьную архитектуру. Для каждого состояния  $S_i$  и сигнала  $X_i$  генерируются случайные гиперразмерные вектора, далее происходит расчет вектора всего автомата  $A$  по формуле (1). При этом операции конъюнкции и дизъюнкции заменяются на операции умножения и сложения соответственно. Для кодирования однонаправленного перехода из первоначального состояния в следующее в представлении векторно-символьной архитектуры, используется операция циклического битового сдвига вправо, обозначенного в формулах как функция  $sr(V, n)$ , где  $V$  – входной  $n$  – количество бит, на которое сдвигается входной вектор  $V$ .

$$A = \sum_{i=1}^S S_i \times x_i \times sr(S_{i-1}, 1). \quad (1)$$

После завершения кодирования пользователь может начать пошаговую симуляцию. Причем вычисления происходят исключительно с HD-векторами сигналов и состояний автомата. Вычисление следующего состояния системы производится по формуле

$$S_n = sr(S_{n-1} \times x_{in} \times A, -1), \quad (2)$$

$S_n$  – следующее состояние автомата;  
 $S_{n-1}$  – предыдущее состояние автомата;  
 $x_{in}$  – входной сигнал, по которому автомат переходит в состояние  $S_n$ .

Для вычисления следующих активных состояний на шаге симуляции при представлении целевой системы в виде недетерминированного автомата, используется формула (3). Здесь и в дальнейших формулах используется знак приближения, поскольку в действительности будет получена сумма, содержащая помимо векторов частных состояний автомата, образующих вектор очередного обобщенного состояния недетерминированного конечного автомата, также ряд векторов, представляющих собой шум, т.е. векторы, неподобные ни к одному

из векторов частных состояний или входных сигналов автомата.

$$Q_n \approx \sum_{i=1}^{Q_{n-1}} sr(S_i \in Q_{n-1} \times x_i \in \sum_{in} \times A, -1), \quad (3)$$

$Q_n$  – результирующий вектор с активными состояниями;

$Q_{n-1}$  – множество предыдущих состояний автомата;

$S_i$  – предыдущее состояние;

$x_i$  – входной сигнал, по которому автомат переходит в состояние  $S_i \in Q_{n-1}$ .

$\sum_{in}$  – множество входных сигналов, при которых недетерминированный автомат входит в активные состояния  $Q_n$ .

Для демонстрации взаимодействия пользователя с системой на рис. 5 представлена UML-диаграмма последовательности.

Для демонстрации вычислений в векторно-символьной архитектуре, в качестве примера опишем абстрактную информационную систему в виде графа, продемонстрированного на рис. 6.

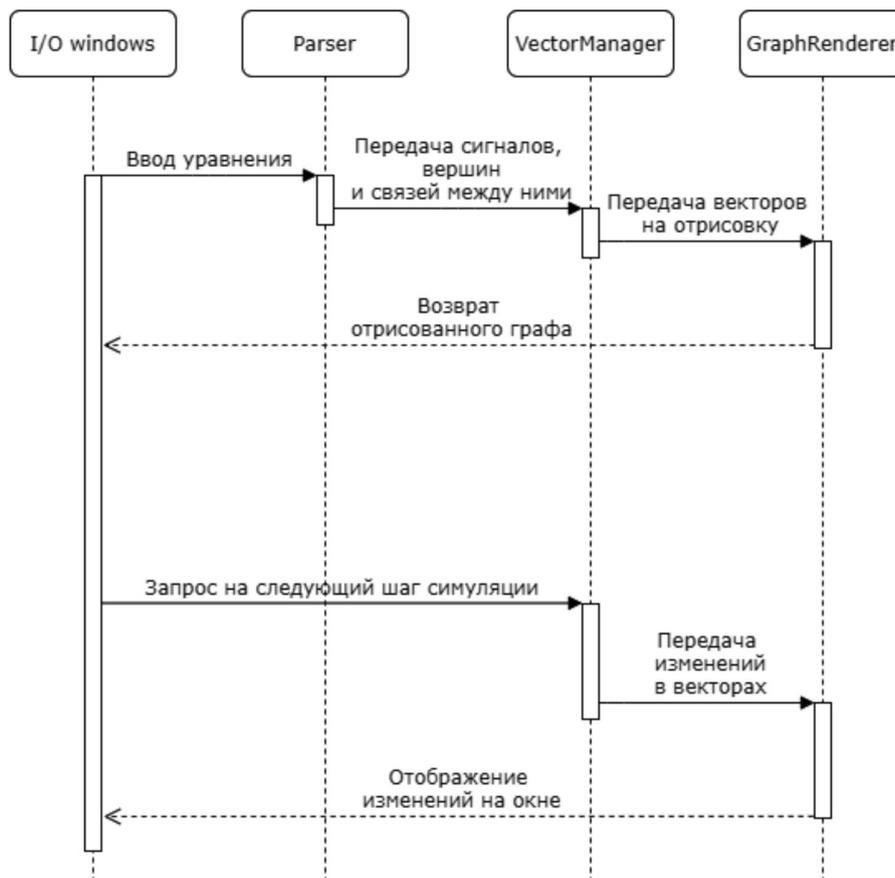


Рис. 5. Диаграмма последовательности работы системы

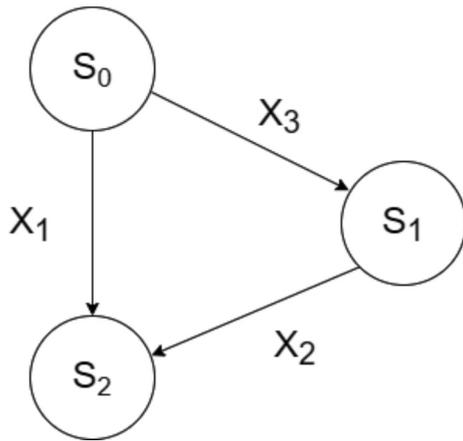


Рис. 6. Граф автомата системы

Граф целевой информационной системы можно представить в виде СКУ, описанной формулами

$$S_1 = S_0 \wedge x_3, \quad (4)$$

$$S_2 = (S_1 \wedge x_2) \vee (S_0 \wedge x_1). \quad (5)$$

После трансформации в векторно-символьную архитектуру у нас выходит следующая СКУ, представленная формулами

$$S_1 \approx sr(S_0 \times x_3 \times A, -1), \quad (6)$$

$$S_2 \approx sr(S_1 \times x_2 \times A, -1) + sr(S_0 \times x_1 \times A, -1). \quad (7)$$

Но для симуляции необходимо найти вектор всего автомата  $A$ , вычисляемый по формуле

$$A = S_0 \times x_3 \times sr(S_1, 1) + S_1 \times x_2 \times sr(S_2, 1) + S_0 \times x_1 \times sr(S_2, 1). \quad (8)$$

После ввода уравнения в систему и кодирования в векторно-символьную архитектуру, программа вывела граф, продемонстрированный на рис. 7.

После установки для всех сигналов значения 1, запускается пошаговая симуляция. После перехода из активного состояния  $S_0$ , автомат перешел в сразу несколько активных состояний  $S_1$  и  $S_2$ . Развернутый расчет следующих активных состояний автомата представлен по формуле

$$sr((S_0 + S_1 + S_2) \times (S_1 \times x_3 \times sr(S_0, -1)) \times x_3, 1) + sr((S_0 + S_1 + S_2) \times (S_2 \times x_1 \times sr(S_0, -1)) \times x_1, 1) = S_1 + S_2 + noise. \quad (9)$$

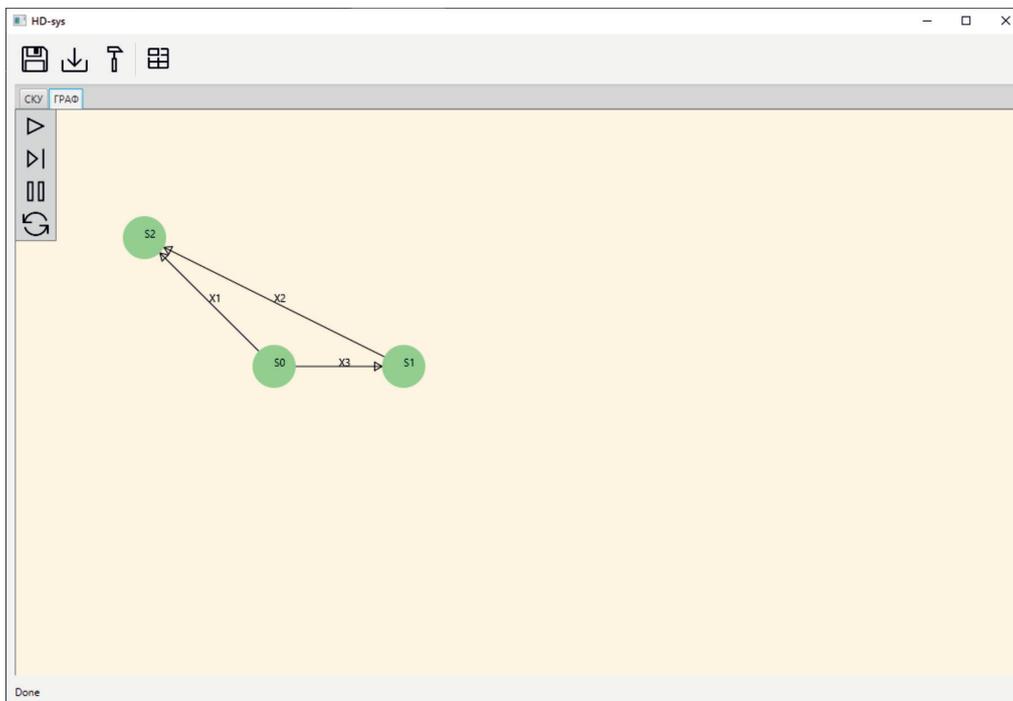


Рис. 7. Граф автомата системы в программе

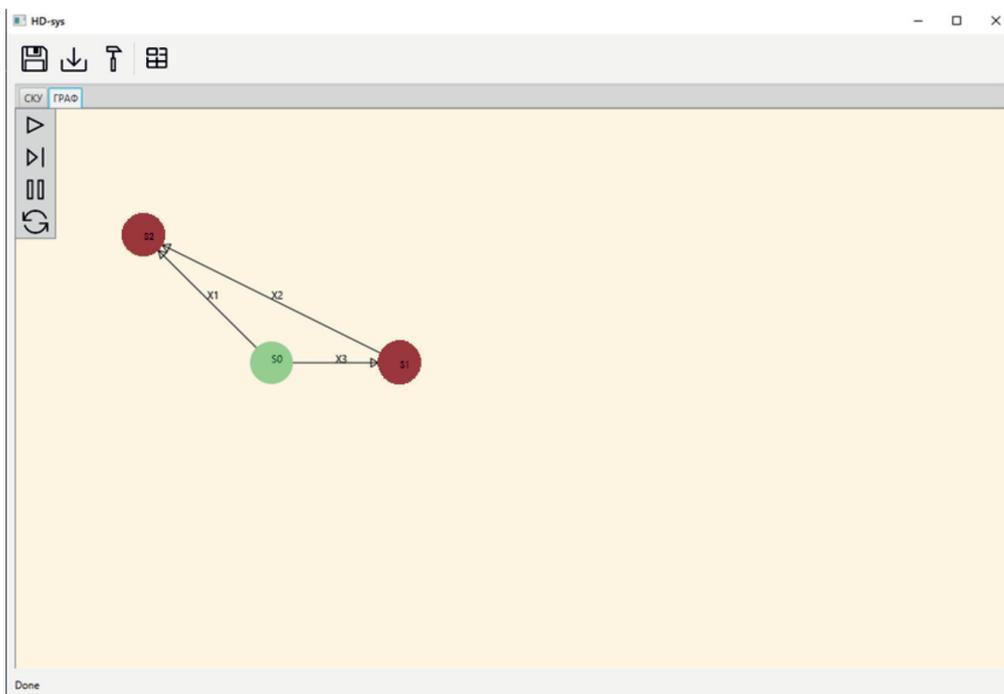


Рис. 8. Симуляция системы на шаге 2

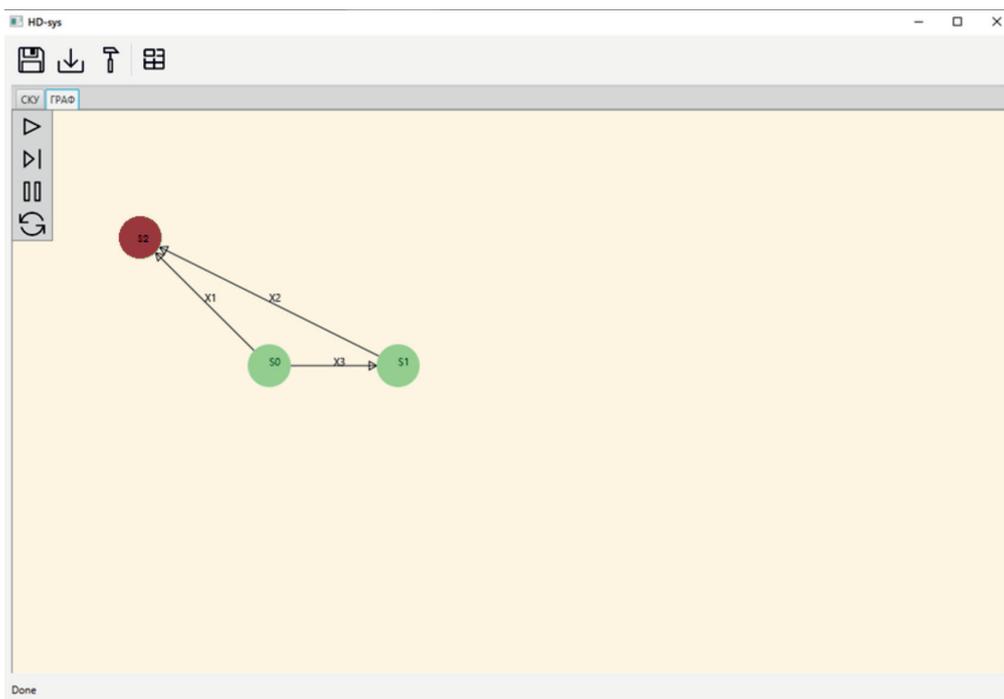


Рис. 9. Симуляция системы на последнем шаге

Результат перехода продемонстрирован на рис. 8.

Следующий шаг симуляции перевел автомат в состояние  $S_2$ , после чего симуляция завершилась. Последний шаг симуляции продемонстрирован на рис. 9.

## Выводы

Разработанная программная инструментальная среда моделирования позволяют решить задачу моделирования комплексных систем в виде недетерминированного ко-

нечного автомата с множеством состояний. В рассмотренном примере представлена простейшая автоматная модель, позволяющая проследить процесс гиперразмерных вычислений в ходе формирования результирующего вектора. Для реальных практических задач, в которых соответствующие недетерминированные автоматные модели имеют тысячи и десятки тысяч состояний, переходы между которыми идут параллельно, возникает проблема высоких вычислительных затрат.

Предложенный в работе метод моделирования процессов переходов для недетерминированного конечного автомата за счет избыточности представления позволит в будущем применять аппаратные решения, обладающие высоким уровнем параллелизма и быстродействия, но при этом некоторой долей ошибок вычислений без внедрения дополнительных методов их коррекции.

Таким образом, развитие теории гиперразмерных вычислений со временем обеспечит возможность перехода к принципиально иным уровням быстродействия вычислительных устройств.

*Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-07-00516 А.*

#### Список литературы

1. Kanerva P. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation* 1 (2). 2009. P. 139–159.
2. Пашенко Д.В., Трокоз Д.А., Мартышкин А.И., Синев М.П. Использование алгебры гиперразмерных векторов для эвристического представления данных при обучении широких нейронных сетей // Проблемы управления и моделирования в сложных системах: материалы XXI Международной конференции (Самара, 3–6 сентября 2019 г.). Самара: Издательство ООО «Офорт», 2019. С. 301–305.
3. Трокоз Д.А., Исхаков Н.В., Синев М.П., Митрохин М.А., Сивишкина Н.О. Темпоральный анализ киберфизических систем с использованием теории автоматов // XXI век: итоги прошлого и проблемы настоящего плюс. 2019. Т. 8. № 3. С. 113–117.
4. Вашкевич Н.П., Бикташев Р.А. Недетерминированные автоматы и их использование для реализации систем параллельной обработки информации: монография. Пенза: ПГУ, 2016. 391 с.
5. Захаров Н.Г., Рогов В.Н. Синтез цифровых автоматов. Ульяновск: Ульяновский государственный технический университет, 2003. 135 с.
6. Скиена С. Алгоритмы. Руководство по разработке. СПб.: БХВ-Петербург, 2017. 720 с.
7. Синев М.П., Мартышкин А.И., Трокоз Д.А. Исследование и анализ подходов к организации параллельных вычислений в распределенных системах // XXI век: итоги прошлого и проблемы настоящего плюс. 2020. Т. 9. № 2. С. 95–104.
8. Новосельцев В.Б., Соколова В.В. Обработка рекурсивных данных конечными автоматами // Известия Томского политехнического университета. 2006. Т. 309. № 7. С. 130–133.
9. Прохоренко Н. JavaFX. СПб.: БХВ-Петербург, 2020. 768 с.
10. Блох Д. Java: эффективное программирование, 3-е изд. СПб.: ООО «Диалектика», 2019. 464 с.
11. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ, 2-е изд. М.: Издательский дом «Вильямс», 2013. 1296 с.
12. Макконнелл С. Совершенный код: Практическое руководство по разработке программного обеспечения. М.: Русская редакция БХВ, 2017. 896 с.