

УДК 519.6

В.В. Кожевников

МЕТОД АНАЛИЗА ДОСТИЖИМОСТИ УСТОЙЧИВЫХ СОСТОЯНИЙ ЛОГИЧЕСКИХ СХЕМ ЦИФРОВЫХ АВТОМАТОВ ¹

Кожевников Валерий Владимирович, кандидат технических наук, окончил Пушкинское высшее командное училище радиотехники, доцент кафедры «Телекоммуникационные технологии и сети» Ульяновского государственного университета. Имеет публикации в области теории проектирования микросистем. [e-mail: vvk2861955@mail.ru].

Аннотация

Метод строится на базе представления логических схем цифровых автоматов (ЦА) в виде уравнений состояний сетей Петри (СП) из класса уравнений Мурата. Для моделирования логики ЦА используются ингибиторные СП. Ингибиторные СП обеспечивают наиболее точное моделирование логических схем автоматов, но при достаточно высокой мощности моделирования утрачивают базовые свойства СП и имеют более низкую мощность разрешения по сравнению с классическими СП. Решение проблемы достигается в результате представления ингибиторных СП в виде матричных уравнений с неявно заданными ингибиторными дугами в матрице инцидентности. Графическая форма представления СП используется в качестве инструмента перехода от исходного описания автомата к его представлению в виде уравнения состояний СП или системы линейных алгебраических уравнений. Построение сетевой модели ЦА осуществляется исходя из принципа сохранения потока однородной информации. Свойство однородности сетевой модели обеспечивает сохранение свойств ингибиторных СП в сетевой модели для моделирования логики и одновременно служит в качестве критерия достижимости устойчивых состояний. Анализ достижимости устойчивых состояний логических схем сводится к решению уравнения состояний СП при заданном критерии достижимости. Метод может быть использован для решения задач синтеза логики базовой структурной схемы автомата, генерации тестов, имитационного моделирования, моделирования и вычисления неисправностей, анализа отказоустойчивости и надежности логических схем автоматов.

Ключевые слова: метод, анализ, достижимость, цифровые автоматы, логические схемы, сети Петри, уравнение состояний, устойчивые состояния.

REACHABILITY ANALYSIS METHOD OF DIGITAL AUTOMATA LOGIC CIRCUITS STABLE STATES

Valeriy Vladimirovich Kozhevnikov, Candidate of Engineering; graduated from Pushkin Higher Command School of Radioelectronics; Associate Professor at the Department of Telecommunication Technologies and Networks of Ulyanovsk State University; an author of articles in the field of microelectronic system design theory. e-mail: vvk2861955@mail.ru.

Abstract

The method is based on representation of digital automata logic circuits in the form of Petri nets state equations of Murata equations class. The inhibitory Petri nets are used for the digital automata logic simulation. Inhibitory Petri nets provide the most accurate simulation of logic circuits but lose their basic properties at a rather high modeling capacities and have the lower capacity of resolution compared to classical Petri nets. The problem solving is achieved by the inhibitory Petri nets presentation as matrix equations with implicit specified inhibitory arcs in the incidence matrix. The Petri nets graphics is used as a transition tool from the initial description of the automata to its representation as an equation of Petri nets states or as a set of linear algebraic equations. The digital automata network model-building is performed on the base of a homogeneous data flow conservation. Homogeneity property of the network model provides preservation of the inhibitory Petri nets properties in the network model to simulate the logic and simultaneously serves as a criterion of stable states reachability. Reachability analysis of logic circuits stable states comes down to solving the equation of Petri nets states at the given reachability criteria. The method can be used for solving the basic structural circuit automata logic synthesis tasks, test generation, simulation, faults modeling and computation, fault-tolerance and reliability analysis of automata logic circuits.

Key words: method, analysis, reachability, digital automata, logic circuits, Petri nets, equation of states, stable states.

¹ Исследование выполнено при финансовой поддержке РФФИ – грант 12-07-00140 а.

ВВЕДЕНИЕ

Прикладная теория ЦА [1, 2] представляет собой одно из направлений развития общей теории автоматов, объектом исследования которой являются цифровые устройства и системы передачи, хранения и обработки цифровой информации. Для изучения процессов, протекающих в ЦА, и общих закономерностей, которым они подчинены, широко применяется алгебраический аппарат, математическая логика, комбинаторный анализ, теория графов и другие разделы дискретной математики [3]. Тем не менее, проблема математического определения понятия ЦА и понятия устойчивого состояния автомата в частности остается актуальной. Представление логических схем ЦА в виде систем логических уравнений [4] или их графических интерпретаций [5, 6] не обеспечивает возможности адекватного представления последовательностных схем автоматов, моделирования и анализа динамики параллельных и асинхронных процессов передачи, хранения и обработки цифровой информации.

Теория СП [7] во многих своих аспектах теснейшим образом связана с теорией автоматов и по существу является развитием последней [8, 9]. СП изначально создавались для моделирования и анализа динамики параллельных и асинхронных процессов взаимодействия дискретных автоматов. При этом задача анализа достижимости занимает центральное место в теории СП [3]. Необходимость адекватного представления и, что особенно важно, возможность анализа (верификации) реальных систем, в данном случае, логических схем ЦА, определяет проблему повышения эффективности соответствующего математического аппарата СП. Как правило, попытки повышения мощности моделирования сети приводят к снижению мощности разрешения (ингибиторные СП, E–сети, раскрашенные СП и т. п.) и наоборот (маркированные графы и автоматные СП).

Для моделирования логических схем ЦА обычно используются ингибиторные СП [10] или СП со сдерживающими дугами [7] и их расширения [11]. Ингибиторные СП отличаются от СП правилами запуска переходов и, соответственно, логикой функционирования сети. При этом все другие предложенные расширения СП либо на самом деле не являются расширениями, либо эквивалентны СП со сдерживающими дугами [7]. Ингибиторные СП обеспечивают наиболее точное моделирование логических схем автоматов, но при достаточно высокой мощности моделирования утрачивают базовые свойства СП и имеют более низкую мощность разрешения по сравнению с классическими СП. Решение проблемы может быть достигнуто в случае представления ингибиторных СП в виде матричных уравнений с неявно заданными ингибиторными дугами в матрице инцидентности [12].

Предлагаемый метод строится на базе представления логических схем ЦА в виде уравнений состояний СП (матричных уравнений) из класса уравнений Мурата [13]. В качестве математического средства аналитическое представление СП позволяет составлять уравнения состояний, алгебраические уравнения и другие математические соот-

ношения, описывающие динамику систем [14–16]. Графическая форма представления СП используется в качестве инструмента перехода от исходного описания автомата к его представлению в виде уравнения состояний СП или системы линейных алгебраических уравнений. Построение сетевой модели ЦА осуществляется исходя из принципа сохранения потока однородной информации [17]. Свойство однородности сетевой модели обеспечивает сохранение свойств ингибиторных СП в сетевой модели для моделирования логики и одновременно служит в качестве критерия достижимости устойчивых состояний ЦА [18].

Задача анализа достижимости устойчивых состояний ЦА сводится к решению уравнения состояний СП при заданном критерии достижимости. Фундаментальное уравнение СП представляет собой систему линейных диофантовых уравнений [19]. Решения этой системы интерпретируются как векторы счета допустимых последовательностей срабатываний переходов и поэтому должны быть неотрицательными целыми числами, что обуславливает специфику задачи. Известные методы решения линейных систем уравнений в целых неотрицательных числах [20–22] имеют асимптотически экспоненциальную вычислительную сложность, что затрудняет их применение для анализа реальных систем. Поэтому для решения уравнения состояний ЦА применяются специальные методы генерации решений с учетом специфики сетевой модели ЦА [23]. При этом проблема матричного метода анализа достижимости СП [7, 14], обусловленная наличием в сети ассиметричных обратных связей и, как следствие, возможностью получения так называемых недействительных решений (но все-таки решений), в случае анализа достижимости устойчивых состояний последовательностных схем ЦА оборачивается одним из основных свойств СП, наилучшим образом обеспечивающим механизм моделирования элементов памяти ЦА. Анализ достижимости устойчивых состояний выполняется независимо от длительности входных сигналов и задержек компонентов схемы автомата.

1 ОПРЕДЕЛЕНИЕ МАТЕМАТИЧЕСКОГО АППАРАТА СП ДЛЯ МОДЕЛИРОВАНИЯ ЛОГИЧЕСКИХ СХЕМ ЦА

СП представляют собой двудольный ориентированный граф, содержащий два типа вершин – позиции (места) и переходы, соединенные между собой ориентированными дугами. При этом граф позволяет задать структуру сети статически. Динамику в структуру сети вносит движение фишек (меток), регулируемое правилами запуска переходов и смены разметки (маркировки) сети. Собственно, правила запуска (разрешения) переходов и смены разметки сети, предложенные К. Петри (правила срабатываний переходов сети), и определяют двудольный ориентированный граф как СП или сеть с заданной (определенной) логикой функционирования.

В общем случае СП определяется четверкой:

$$N = (P, T, A, M),$$

где P – конечное множество позиций сети,

T – конечное множество переходов сети,

A – матрица инцидентности множества позиций и множества переходов сети,

M – множество достижимых разметок (состояний) сети на множестве позиций P . Все множество позиций сети разбивается на множество входных, внутренних и выходных позиций $P = \{P^x, P^0, P^y\}$. Входными называются позиции P^x , которые имеют только выходные и не имеют входных дуг. Выходными называются позиции P^y , которые имеют только входные и не имеют выходных дуг. Внутренние позиции P^0 имеют как входные, так и выходные дуги. Для сетей с обратными связями множество позиций, входящих в состав обратных связей и определяющих множество точек условного разрыва обратных связей, обозначается как $P^q \in P^0$. Матрица инцидентности $A = A^+ - A^-$ где $a_{ij} = \{0, -1, 1\}$ для ординарных сетей (рассматривается класс ординарных сетей, хотя все утверждения справедливы и в общем случае) задает множество отношений между позициями и переходами или логику функционирования сети. Каждое состояние сети задается в виде вектора разметки сети $\mu(P) \in M$. При этом множество $M(P^x)$ составляет множество входных разметок сети, множество $M(P^y)$ составляет множество выходных разметок сети, и множество $M(P^0)$ составляет множество внутренних разметок сети.

Единство статического и динамического аспектов достигается в результате представления сети в виде фундаментального уравнения состояний СП из класса уравнений Мурата:

$$\Delta\mu = A \cdot \tau, \tag{1}$$

где $\Delta\mu = \mu - \mu_0$, μ_0 – вектор начальной разметки сети,

μ – вектор конечной разметки сети,

τ – вектор покрытия переходов сети (вектор счета последовательности срабатываний переходов) или отображение Париха [7] для последовательности срабатываний переходов, который определяет только состав и не определяет последовательность срабатываний переходов.

Множество векторов покрытия переходов τ образует покрытие сети – S , где $\tau \in S$.

Правила срабатываний переходов СП могут быть заданы следующими отношениями:

$$\mu_k = \mu_{k-1} + A \cdot u_k \tag{2}$$

$$\text{и } \mu_{k-1} + A \cdot u_k \geq 0, \tag{3}$$

где u_k – вектор запуска переходов сети,

$u_k \in \tau$, μ_k – вектор текущей разметки сети.

Уравнение (2) определяет правило смены разметки СП, условие (3) – правило запуска переходов СП соответственно. Все переходы, каждый и все одновременно, удовлетворяющие условию (3), составляют вектор u_k . Для каждой последовательности переходов

$$\Delta\mu = \sum_{k=1}^n \Delta\mu_k, \tau = \sum_{k=1}^n u_k,$$

где $\Delta\mu_k = \mu_k - \mu_{k-1}$.

Множество достижимых разметок сети определяет пространство состояний или мощность моделирования сети, которая для СП ограничена собственно правилами Петри. Мощность разрешения сети зависит от возможностей ее анализа. Повышение моделирующей мощности сети может быть достигнуто путем определения дополнительных правил срабатываний переходов сети, отличных от правил Петри, применение которых приводит к неопределенной логике срабатывания переходов и соответственно к неопределенной логике функционирования сети, что выражается в нарушении уравнения (1).

Соответственно, все возможные расширения СП могут быть представлены следующими отношениями:

$$\mu_0 + A \cdot \tau - \mu \leq 0 \tag{4}$$

$$\text{и } \mu_0 + A \cdot \tau - \mu \geq 0. \tag{5}$$

При этом неопределенная логика запуска переходов (входная логика переходов) выражается отношением (4). Неопределенная логика смены разметки сети (выходная логика переходов) – отношением (5). Совмещение расширений СП (4) и (5) в рамках одной сети допускается только в случае введения дополнительных ограничений при построении исходной структуры сети.

Отношения (4) и (5) могут быть использованы для определения ингибиторных СП и их расширений. Введение специальных обозначений для ингибиторных дуг в матрице инцидентности A уравнения (1):

$$a_{ij} = -1 \rightarrow a_{ij} = -\alpha$$

$$\text{и } a_{ij} = 1 \rightarrow a_{ij} = \alpha,$$

где $\alpha \neq 0$, эквивалентно отношениям (4) и (5).

В случае неявного определения ингибиторных дуг (равными нулю) в матрице инцидентности, то есть при $\alpha = 0$, ингибиторная СП может быть представлена в виде фундаментального уравнения СП (1). Для идентификации ингибиторных дуг в матрице инцидентности выполняется программирование логики срабатывания переходов путем построения (генерации) минимального порождающего множества решений уравнения состояний (1). В общем случае программирование логики срабатывания каждого перехода заключается в определении активности смежных переходов путем частичного определения вектора τ . Вектор покрытия сети τ в данном случае определяет не только состав, но и логику запуска переходов. Покрытие сети S определяет логику функционирования сети [12].

Анализ достижимости ингибиторных СП сводится к решению уравнения (1). Процедура решения уравнения (1) строится на базе соответствующего минимального порождающего множества решений уравнения состояний (1). Вычисление вектора начальной разметки сети $\mu_0(P)$ и вектора конечной разметки сети $\mu(P)$ выполняется в результате разложения вектора $\Delta\mu(P)$. Для сетей с обратными связями вычисление вектора начальной разметки сети $\mu_0(P^q)$ и вектора конечной разметки сети $\mu(P^q)$ выполняется в результате решения уравнений: $\mu_0(P^q) = A^-(P^q, T) \cdot \tau$ и $\mu(P^q) = A^+(P^q, T) \cdot \tau$.

Возможность вычисления вектора начальной разметки сети $\mu_0(P^q)$ и вектора конечной разметки сети $\mu(P^q)$ на множестве позиций, входящих в состав обратных связей, обеспечивает решение проблемы недействительных решений матричного анализа СП.

Для построения протоколов достижимости ингибиторных СП могут быть использованы стандартные методы построения протоколов достижимости обычных СП. В общем случае процедура построения протоколов достижимости сводится к вычислению последовательности векторов запуска переходов и текущих разметок сети для каждого вектора τ , начиная с вектора начальной разметки μ_0 и до тех пор, пока не будет достигнута разметка μ . Последовательность векторов запуска переходов u_k и векторов текущей разметки μ_k может быть получена путем итеративного решения уравнения (2). На каждом шаге итерации $k=1, n$ – все переходы, входящие в состав вектора x , проверяются на выполнение условия (3). Переходы, удовлетворяющие условию (3) каждый в отдельности и все одновременно, составляют вектор u_k . На каждом шаге итерации $\tau = \tau - u_k$. В отличие от процедуры построения протоколов достижимости обычных СП, наличие исходного вектора τ является обязательным.

2 МОДЕЛЬНОЕ ПРЕДСТАВЛЕНИЕ ЛОГИЧЕСКИХ СХЕМ ЦА

Цифровой или дискретный автомат представляет собой последовательностное устройство или устройство с элементами памяти, которое может находиться как минимум в двух устойчивых состояниях и самостоятельно переходить из одного устойчивого состояния в другое в соответствии с заданным алгоритмом (диаграммой переходов и состояний). В общем случае структурная схема ЦА может быть представлена в виде набора трех узлов: последовательностной схемы элементов памяти, комбинационной схемы формирования сигналов управления элементами памяти и комбинационной схемы формирования выходных сигналов. Под устойчивым понимается состояние ЦА, в которое переходит автомат под воздействием входных сигналов и внутренних сигналов с выходов элементов памяти после завершения всех переходных процессов и которое может быть сохранено в течение заданного периода времени [4]. Для вывода автомата из устойчивого состояния необходимо сформировать соответствующую входную комбинацию сигналов. Устойчивое состояние автомата, из которого он не может быть выведен, называется тупиковым.

Исходной информацией для построения сетевой модели служит описание структурной схемы автомата. Степень декомпозиции компонентов схемы должна обеспечивать возможность представления этих компонентов в виде таблицы истинности. Точность моделирования и возможности анализа зависят от степени декомпозиции компонентов схемы. При этом точность моделирования находится в обратно пропорциональной зависимости от возможности анализа.

Структурная схема представляется в виде маркированного графа путем интерпретации входов и выходов схемы

и структурных компонентов позициями маркированного графа, а самих компонентов и линий соединений – составными и простыми переходами соответственно. Таблицы истинности компонентов представляются в виде сетевых моделей компонентов путем интерпретации наборов из таблицы истинности простыми переходами, а соответствующей логики входными и выходными дугами этих переходов. Сетевые модели компонентов структурной схемы представляют собой расширенную ингибиторную СП, в которой допускается наличие как входных, так и выходных ингибиторных дуг для моделирования входной и выходной логики переходов. Маркированный граф структурной схемы в результате подстановки сетевых моделей компонентов преобразуется в ингибиторную СП. При этом свойства маркированного графа сохраняются, так как в сетевой модели компонента одновременно может быть активизирован только один переход.

Множество входов и выходов структурной схемы интерпретируется как множество входных и выходных позиций сети. Множество входов и выходов компонентов схемы интерпретируется как множество внутренних позиций сети. Логическая единица интерпретируется как фишка в позиции сети, а ее отсутствие – как логический ноль. Перемещение сигналов в логических схемах интерпретируется как движение фишек в сети.

Графическая форма представления логических схем позволяет перейти от описания логической схемы к ее математическому представлению в виде матрицы инцидентности A . Ингибиторные дуги в матрице инцидентности задаются неявно. Каждой ингибиторной дуге в матрице инцидентности соответствует ноль. Практически таблица истинности компонента, где единичные значения входных переменных берутся со знаком минус, а выходные – со знаком плюс, представляет собой матрицу инцидентности компонента схемы с неявно заданными ингибиторными дугами.

Представление ЦА в виде двудольного ориентированного графа или матрицы инцидентности позволяет задать логические схемы статически. Динамику в модель вносит движение фишек, регулируемое правилами запуска переходов и смены разметки сети. Комплексная модель ЦА представляется в виде уравнения состояний СП (1) [18].

Цикл срабатывания (переключения) структурного автомата или дискретный интервал времени, в течение которого входные сигналы с учетом текущего внутреннего состояния автомата достигают выходов и переводят автомат в новое состояние, интерпретируется как вектор покрытия переходов сети τ . Для маркированных графов структурной схемы автомата $\tau = 1$. Для производной ингибиторной СП логической схемы автомата $\tau \leq 1$. Ограничение вектора покрытия переходов сети τ обеспечивает безопасность (ограниченность) сетевой модели.

Входная комбинация сигналов на входах автомата интерпретируется как вектор начальной разметки входных позиций сети $\mu_0(P^x)$. Начальная и конечная комбинации сигналов на выходах элементов памяти интерпретируются как вектор начальной разметки $\mu_0(P^q)$ и вектор конечной

разметки $\mu(P^q)$ на множестве внутренних позиций сети, входящих в состав обратных связей. Выходная комбинация сигналов на выходах автомата интерпретируется как вектор конечной разметки выходных позиций сети $\mu(P^y)$. Текущее и устойчивое состояния автомата формируются в процессе построения протоколов достижимости. Для переключения автомата из одного достижимого устойчивого состояния в другое может потребоваться два и более цикла переключения автомата.

Формирование критерия достижимости выполняется путем частичного определения вектора $\Delta\mu$. Известно, что для маркированных графов вектор $\Delta\mu(P^0) = 0$ [14]. Данное свойство маркированных графов сохраняется в сетевой модели и может быть использовано для определения вектора $\Delta\mu$ равным нулю на множестве внутренних позиций сетевой модели P^0 при решении уравнения (1). Более того, именно свойство однородности обеспечивает возможность матричного представления сетевой модели логических схем автоматов и генерации минимального порождающего множества решений уравнения (1). Моделирование условного разрыва обратных связей для последовательностных схем осуществляется путем исключения соответствующих позиций P^q из состава внутренних позиций сети P^0 . Состояние сети определяется как достижимое, если для него существует решение, удовлетворяющее условию: $\Delta\mu(P^q) \neq 0$, и как устойчивое, если для него существует решение, удовлетворяющее условию: $\Delta\mu(P^q) = 0$.

В качестве примера рассматривается асинхронный RS-триггер, который представляет собой простейший автомат, способный находиться в двух устойчивых состояниях. На рисунке 1 приведен маркированный граф схемы асинхронного RS-триггера и сетевая модель компонента (в данном случае логического элемента «И-НЕ»). Пунктиром на рисунке 1 обозначены ингибиторные дуги, соответствующие нулю в матрице инцидентности компонентов.

Матрица инцидентности сетевой модели RS-триггера с неявно заданными ингибиторными дугами может быть представлена следующим образом:

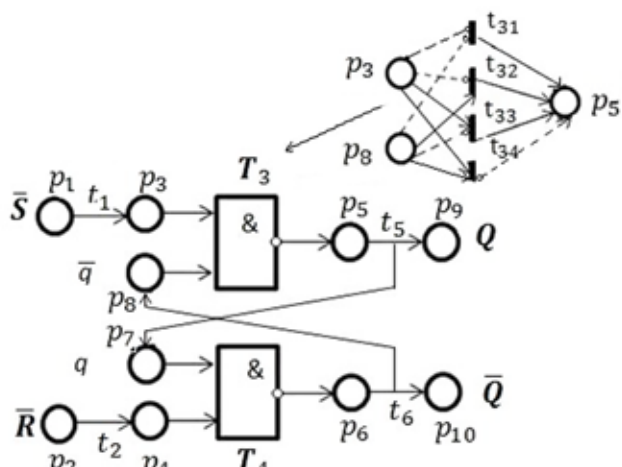


Рис.1. Графическая форма представления RS-триггера

		t_1	t_2	t_{31}	t_{32}	t_{33}	t_{34}	t_{41}	t_{42}	t_{43}	t_{44}	t_5	t_6
\bar{S}	p_1	-1	0	0	0	0	0	0	0	0	0	0	0
\bar{R}	p_2	0	-1	0	0	0	0	0	0	0	0	0	0
	p_3	1	0	0	0	-1	-1	0	0	0	0	0	0
	p_4	0	1	0	0	0	0	0	-1	0	-1	0	0
	p_5	0	0	1	1	1	0	0	0	0	0	-1	0
	p_6	0	0	0	0	0	0	1	1	1	0	0	-1
q	p_7	0	0	0	0	0	0	0	0	-1	-1	1	0
\bar{q}	p_8	0	0	0	-1	0	-1	0	0	0	0	0	1
Q	p_9	0	0	0	0	0	0	0	0	0	0	1	0
\bar{Q}	p_{10}	0	0	0	0	0	0	0	0	0	0	0	1

Подставляя матрицу инцидентности и частично определенный вектор $\Delta\mu$ в уравнение (1), получим комплексную модель RS-триггера. Исходное уравнение (1) для сетевой модели RS-триггера может быть записано следующим образом:

$$\begin{bmatrix} \Delta\mu_1 \\ \Delta\mu_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \Delta\mu_7 \\ \Delta\mu_8 \\ \Delta\mu_9 \\ \Delta\mu_{10} \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_{31} \\ t_{32} \\ t_{33} \\ t_{34} \\ t_{41} \\ t_{42} \\ t_{43} \\ t_{44} \\ t_5 \\ t_6 \end{bmatrix}$$

3 Анализ достижимости устойчивых состояний логических схем ЦА

Задача анализа достижимости устойчивых состояний ЦА является обратной по отношению к задаче синтеза структурной схемы ЦА. Если для задачи синтеза структурной схемы автомата в качестве исходной информации служит таблица или диаграмма переходов и состояний автомата (диаграмма Мура), то для задачи анализа структурная схема автомата служит в качестве исходной информации. При этом таблица переходов (переключений) может быть задана в качестве исходной информации или получена в результате анализа структурной схемы автомата. В первом случае выполняется анализ достижимости устойчивых состояний автомата (верификация структурной схемы автомата), во втором случае – генерация множества достижимых устойчивых состояний структурной схемы автомата и анализ достижимости устойчивых состояний автомата.

Процедура анализа достижимости устойчивых состояний ЦА сводится к решению уравнения (1). Множество решений, в свою очередь, ограничено и зависит от степени определенности вектора $\Delta\mu$. В случае, если вектор $\Delta\mu$ полностью определен, уравнение (1) может иметь только одно решение для каждого вектора $\Delta\mu$. Доопределение каждого вектора $\Delta\mu$ осуществляется на основе заданных значений таблицы переходов и состояний автомата. В

случае, если вектор $\Delta\mu$ определен только на множестве внутренних позиций сети (значения таблицы переходов и состояний автомата не заданы), задача анализа достижимости устойчивых состояний сводится к решению однородного уравнения $A(P^0, T) \cdot \tau = 0$. Доопределение вектора $\Delta\mu$ для каждого полученного вектора τ выполняется путем простого умножения вектора τ на матрицу инцидентности $A(P, T)$. Количество решений в данном случае должно быть равно количеству возможных переключений автомата.

Задача анализа достижимости устойчивых состояний RS-триггера (в случае, если вектор $\Delta\mu$ определен только на множестве внутренних позиций сети) сводится к решению однородного уравнения:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_{31} \\ t_{32} \\ t_{33} \\ t_{34} \\ t_{41} \\ t_{42} \\ t_{43} \\ t_{44} \\ t_5 \\ t_6 \end{bmatrix}$$

Генерация минимального порождающего множества решений или комбинаций столбцов (переходов) сетевых моделей компонентов выполняется исходя из принципа активности всех составных переходов сетевой модели для каждого состояния автомата. В составе сетевой модели каждого компонента одновременно может быть активизирован только один простой переход. Соответственно количество единиц (активных переходов) в комбинации постоянно и равно количеству компонентов схемы. Данное ограничение определяется спецификой срабатывания ЦА и необходимо для минимизации перебора комбинаций

переходов, а также для исключения возможных недействительных решений. Количество возможных комбинаций переходов определяется выражением: $N = \prod_{i=1}^m 2^{m_i}$, где m – количество компонентов схемы, m_i – количество входов i -го компонента. Процедура генерации минимального порождающего множества решений приводится в работе [23].

Минимальное порождающее множество решений для уравнения RS-триггера может быть задано путем частичного определения вектора покрытия переходов τ в виде матрицы:

$$\begin{matrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 & \tau_5 & \tau_6 & \tau_7 & \tau_8 & \tau_9 & \tau_{10} & \tau_{11} & \tau_{12} & \tau_{13} & \tau_{14} & \tau_{15} & \tau_{16} \\ \begin{bmatrix} t_1 & t_1 & t_1 & t_1 & t_1 & t_1 & t_1 & t_1 & t_1 & t_1 & t_1 & t_1 & t_1 & t_1 & t_1 & t_1 \\ t_2 & t_2 & t_2 & t_2 & t_2 & t_2 & t_2 & t_2 & t_2 & t_2 & t_2 & t_2 & t_2 & t_2 & t_2 & t_2 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ t_5 & t_5 & t_5 & t_5 & t_5 & t_5 & t_5 & t_5 & t_5 & t_5 & t_5 & t_5 & t_5 & t_5 & t_5 & t_5 \\ t_6 & t_6 & t_6 & t_6 & t_6 & t_6 & t_6 & t_6 & t_6 & t_6 & t_6 & t_6 & t_6 & t_6 & t_6 & t_6 \end{bmatrix} \end{matrix}$$

Полученные комбинации последовательно подставляются в уравнение состояний RS-триггера. Доопределение каждого вектора покрытия переходов сети τ осуществляется в результате решения уравнения любым из существующих методов решения систем линейных алгебраических уравнений. Может быть использован метод Гаусса с учетом целочисленности системы или любой другой метод решения целочисленных систем (алгоритм Диксона [24], алгоритм Сергеева [25]).

Множество полученных решений уравнения состояний RS-триггера приведено ниже:

$$\begin{aligned} \tau_1 &= [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1], \\ \tau_2 &= [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1], \\ \tau_3 &= [1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1], \\ \tau_4 &= [1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1], \\ \tau_5 &= [0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1], \\ \tau_6 &= [0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1], \\ \tau_7 &= [1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1], \\ \tau_8 &= [1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1], \\ \tau_9 &= [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1], \\ \tau_{10} &= [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1], \\ \tau_{11} &= [1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1], \\ \tau_{12} &= [1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1], \\ \tau_{13} &= [0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0], \\ \tau_{14} &= [0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0], \\ \tau_{15} &= [1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0], \\ \tau_{16} &= [1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0]. \end{aligned}$$

$$\begin{aligned} \Delta\mu_1 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1], \\ \Delta\mu_2 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1], \\ \Delta\mu_3 &= [-1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1], \\ \Delta\mu_4 &= [-1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1], \\ \Delta\mu_5 &= [0 \ -1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1], \\ \Delta\mu_6 &= [0 \ -1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1], \\ \Delta\mu_7 &= [-1 \ -1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1], \\ \Delta\mu_8 &= [-1 \ -1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1], \\ \Delta\mu_9 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1], \\ \Delta\mu_{10} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1], \\ \Delta\mu_{11} &= [-1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1], \\ \Delta\mu_{12} &= [-1 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1 \ 0 \ 0 \ 1], \\ \Delta\mu_{13} &= [0 \ -1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0], \\ \Delta\mu_{14} &= [0 \ -1 \ 0 \ 0 \ 0 \ 0 \ -1 \ 1 \ 0 \ 0], \\ \Delta\mu_{15} &= [-1 \ -1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0], \\ \Delta\mu_{16} &= [-1 \ -1 \ 0 \ 0 \ 0 \ 0 \ -1 \ -1 \ 0 \ 0]. \end{aligned}$$

В результате разложения каждого вектора $\Delta\mu$ имеем множество начальных и конечных разметок для каждого достижимого состояния RS-триггера:

$$\begin{aligned} \mu_{10} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0], & \mu_1 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1], \\ \mu_{20} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0], & \mu_2 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1], \\ \mu_{30} &= [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0], & \mu_3 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1], \\ \mu_{40} &= [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0], & \mu_4 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1], \\ \mu_{50} &= [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0], & \mu_5 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1], \\ \mu_{60} &= [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0], & \mu_6 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1], \\ \mu_{70} &= [1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0], & \mu_7 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1], \\ \mu_{80} &= [1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0], & \mu_8 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1], \\ \mu_{90} &= [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0], & \mu_9 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1], \\ \mu_{100} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0], & \mu_{10} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1], \\ \mu_{110} &= [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0], & \mu_{11} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1], \\ \mu_{120} &= [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0], & \mu_{12} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1], \\ \mu_{130} &= [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0], & \mu_{13} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0], \\ \mu_{140} &= [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0], & \mu_{14} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0], \\ \mu_{150} &= [1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0], & \mu_{15} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0], \\ \mu_{160} &= [1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0], & \mu_{16} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]. \end{aligned}$$

Приведенное множество решений определяет множество достижимых в течение цикла переключения автомата состояний. Из них устойчивые состояния определяют решения, для которых дополнительно выполняется условие: $\Delta\mu(P^q) = 0$.

Множество полученных решений может быть представлено в виде диаграммы Мура. Соответствующая диаграмма RS-триггера приведена на рисунке 2. Начальной и конечной разметке достижимых состояний на множестве позиций P^q соответствуют вершины графа. Начальная и конечная разметка для устойчивых состояний отображается в виде вершин графа, которые имеют петли обратной связи.

Для построения протоколов достижимости устойчивых состояний структурного автомата используются стандартные процедуры обычных СП. Соответствующие протоколы достижимости для устойчивых состояний RS-триггера приведены ниже:

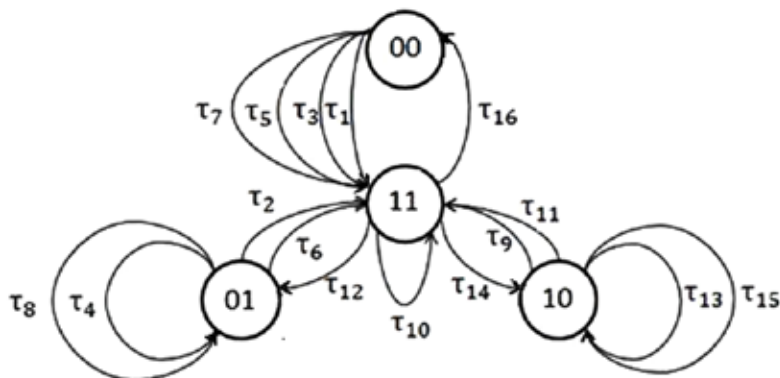


Рис. 2. Диаграмма Мура RS-триггера

$$\begin{aligned} \tau_4 &= [1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1], & \mu_{40} &= [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0], \\ u_{41} &= [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0], & \mu_{41} &= [0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0], \\ u_{42} &= [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1], & \mu_{42} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1], \\ \\ \tau_8 &= [1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1], & \mu_{80} &= [1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0], \\ u_{81} &= [1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0], & \mu_{81} &= [0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0], \\ u_{82} &= [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1], & \mu_{82} &= [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0], \\ u_{83} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1], & \mu_{83} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1], \\ \\ \tau_{10} &= [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1], & \mu_{100} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0], \\ u_{101} &= [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0], & \mu_{101} &= [0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0], \\ u_{102} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1], & \mu_{102} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1], \end{aligned}$$

$$\begin{aligned} \tau_{13} &= [0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0], & \mu_{13\ 0} &= [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0], \\ u_{13\ 1} &= [0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0], & \mu_{13\ 1} &= [0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0], \\ u_{13\ 2} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0], & \mu_{13\ 2} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0], \\ \\ \tau_{15} &= [1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0], & \mu_{15\ 0} &= [1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0], \\ u_{15\ 1} &= [1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0], & \mu_{15\ 1} &= [0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0], \\ u_{15\ 2} &= [0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0], & \mu_{15\ 2} &= [0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0], \\ u_{15\ 3} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0], & \mu_{15\ 3} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0]. \end{aligned}$$

Полученные протоколы отражают динамику параллельных и асинхронных процессов достижимости устойчивых состояний RS-триггера. Время срабатывания переходов принимается равным одному дискретному интервалу времени. Соответственно, время достижимости интерпретируется как количество итераций. Непосредственно сами устойчивые состояния автомата могут быть получены в соответствии с выражением: $S = \bigvee_{k=1}^n \mu_k$.

Анализ корректности устойчивых состояний ЦА сводится к проверке безопасности и потенциальной безопасности полученных протоколов. Каждый вектор текущей разметки сети проверяется на условие безопасности: $\mu_k \leq 1$. Если для некоторой позиции условие безопасности не выполняется, то данная позиция определяет место некорректности исходного описания структурной схемы автомата. В качестве критерия потенциальной безопасности для каждого протокола используется условие: $\sum_{k=1}^n \mu_k \leq 1$. Если для некоторой позиции условие потенциальной безопасности не выполняется (фиска в позиции сохраняется в позиции в течение двух и более итераций), то данная позиция определяет место возможного появления неустойчивого сигнала, способного привести к сбою в процессе функционирования логических схем при недостаточной длительности входного сигнала.

ЗАКЛЮЧЕНИЕ

На основании проведенных исследований можно сделать заключение, что проблема анализа достижимости устойчивых состояний ЦА на базе их представления в виде уравнения состояний СП разрешима. Представление ЦА в виде однородных уравнений состояний СП обеспечивает возможность не только анализа, но и генерации множества достижимых устойчивых состояний в случае, если они изначально не заданы.

В процессе анализа автоматически учитываются (воспроизводятся) асинхронные и параллельные процессы функционирования, что значительно упрощает подготовку исходного описания и обеспечивает более высокую точность моделирования ЦА. При разработке процедур анализа могут быть использованы стандартные методы решения систем линейных алгебраических уравнений, что, в свою очередь, значительно сокращает время разработки и выполнения процедур анализа.

Практическое применение стандартных методов решения уравнений ограничено размерами и сложностью структурной схемы ЦА. Критическим, с точки зрения

эффективности метода, является время генерации порождающего множества решений уравнения состояний ЦА, которое находится в экспоненциальной зависимости от количества компонентов схемы и количества входов каждого компонента. Приемлемое время генерации всегда может быть достигнуто в результате иерархической композиции структурной схемы автомата или разработки специальных методов решения уравнения состояний СП с учетом особенностей структурной организации автоматов. Практически речь идет о разработке методов решения уравнения состояний ЦА. Кроме того, реальные системы могут быть представлены в виде систем уравнений состояний ЦА, что обеспечивает достаточно высокую эффективность метода в случае его реализации на базе высокопроизводительных суперкомпьютеров.

Метод вписывается в общую концепцию математического моделирования логических схем ЦА на различных уровнях представления [13] и может быть использован для решения задач синтеза логики базовой структурной схемы автомата, генерации тестов, моделирования и вычисления неисправностей [26], имитационного моделирования [18], анализа отказоустойчивости и надежности логических схем автоматов. Применение ограниченных и временных СП [14] обеспечивает возможность воспроизведения реальных процессов функционирования (временных диаграмм), протекающих в ЦА. При этом может быть выполнен расчет минимальной длительности входных сигналов, необходимой для достижимости устойчивых состояний, в зависимости от задержек компонентов схем ЦА.

СПИСОК ЛИТЕРАТУРЫ

1. Савельев А.Я. Прикладная теория цифровых автоматов. – М. : Высш. Шк., 1987.
2. Глушков В.М. Синтез цифровых автоматов. – М. : Физматгиз, 1962.
3. Горбатов В.А. Основы дискретной математики: учеб. пособие для студентов вузов. – М. : АСТ, 2006.
4. Пухальский Г.И., Новосельцева Т.Я. Цифровые устройства: учебное пособие для вузов. – СПб. : Политехника, 1996.
5. Сигорский В.П. Математический аппарат инженера. – Киев : Техника, 1975.
6. Автоматизированная система контроля и диагностики логических узлов и блоков на этапе производства / В.В. Кожевников [и др.] // Автоматика и вычислительная техника. – 1983. – № 3. – С. 57–63.

7. Питерсон Д. Теория сетей Петри и моделирование систем. – М. : Мир, 1984.
8. Автоматное управление асинхронными процессами в ЭВМ и дискретных системах / В.И. Варшавский [и др.]. – М. : Наука, 1986.
9. Лазарев В.Г., Пийль Е.И. Синтез управляющих автоматов. – М. : Энергоатомиздат, 1989.
10. Котов В.Е. Сети Петри. – М. : Наука. Главная редакция физ-мат. литературы, 1984.
11. Гуревич Д.С. Поглощающие сети Петри и их использование при разработке цифровых вычислительных систем с распределенной структурой // Автоматика и вычислительная техника. – 1990. – № 2. – С. 80–87.
12. Кожевников В.В. Метод анализа достижимости ингибиторных сетей Петри // Автоматизация процессов управления. – 2013. – № 3 (33). – С. 32–37.
13. Кожевников В. В. Концепция математического моделирования микропрограммируемых устройств // Изв. РАН. Техническая кибернетика. – 1992. – № 4. – С. 175–179.
14. Мурата Т. Сети Петри. Свойства, анализ, приложения // ТИИЭР. – 1989. – Т. 77, № 44. – С. 41–85.
15. Girault C., Volk R. Petri nets for systems engineering – A guide to modelling, verification and applications. – Springer-Verlag, 2003.
16. Cortadella J., Kishinevsky M., Kondratyev A., Lavagno L., Yakovlev A. Logic synthesis of asynchronous controllers and interfaces. – Springer-Verlag, 2002.
17. Филлипс Д., Гарсиа-Диас А. Методы анализа сетей. – М. : Мир, 1984.
18. Кожевников В.В. Метод математического моделирования логических схем цифровых автоматов // Автоматизация процессов управления. – 2012. – № 4 (30). – С. 97–101.
19. Зайцев Д.А. Решение фундаментального уравнения сетей Петри в процессе композиции функциональных подсетей // Искусственный интеллект. – 2005. – № 1. – С. 59–68.
20. О нахождении решения уравнения состояний сетей Петри из класса уравнений Мурата / А.А. Хан [и др.] // ТИИЭР. – 1981. – Т. 69, № 4. – С. 71–72.
21. Кривый С.Л. О некоторых методах решения и критериях совместности систем линейных диофантовых уравнений в области натуральных чисел // Кибернетика и системный анализ. – 1999. – № 4. – С. 12–36.
22. Зайцев Д.А. К вопросу о вычислительной сложности метода Тудика // Искусственный интеллект. – 2004. – № 1. – С. 29–37.
23. Смагин А.А., Кожевников В.В., Круглова И.В. Решение уравнения состояний систем управления // Известия самарского научного центра РАН. – 2013. – Т. 15, № 4(3). – С. 705–708.
24. Dixon J. D. Exact solution of linear equations using P-adic expansions // Numer. Math. 40. 1982. pp. 137–141.
25. Сергеев М.Б. Гибридный разрядный метод решения систем уравнений в целочисленной арифметике // Информационно-управляющие системы. – 2003. – № 2–3. – С. 16–18.
26. Кожевников В.В. Методы построения и анализа сетевой диагностической модели микропрограммируемых структур // Автоматика и вычислительная техника. – 1988. – № 5. – С. 90–95.

REFERENCES

1. Savel'yev A.Ya. *Prikladnaya teoriya tsifrovyykh avtomatov* [The Applied Digital Automata Theory]. Moscow, Vysshaya Shkola Publ., 1987.
2. Glushkov V.M. *Sintez tsifrovyykh avtomatov* [Digital Automata Synthesis]. Moscow, Fizmatgiz Publ., 1962.
3. Gorbatov V.A. *Osnovy diskretnoy matematiki: uchebnoye posobiye dlya studentov vuzov* [Rudiments of Discret Mathematics: Textbook for Students]. Moscow, AST Publ., 2006.
4. Pukhalskiy G.I., Novoseltseva T.Ya. *Tsifrovyye ustroystva: uchebnoye posobiye dlya vtuzov* [Digital Devices: Textbook for Students]. St. Petersburg, Politika Publ., 1996.
5. Sigorskiy V.P. *Matematicheskii apparat inzhenera* [The Mathematical Tools of an Engineer]. Kiyev, Tekhnika Publ., 1975.
6. Kozhevnikov V.V. and Others. *Avtomatizirovannaya sistema kontrolya i diagnostiki logicheskikh uzlov i blokov na etape proizvodstva* [Automatic Test Equipment for Logical Sections and Units on the Production Phase]. *Avtomatika i vychislitel'naya tekhnika* [Automatics and Computers], 1983, no. 3, pp. 57–63.
7. Piterson D. *Teoriya setey Petri i modelirovaniye system* [The Petri Nets Theory and System Simulation]. Moscow, Mir Publ., 1984.
8. Varshavskiy V.I. and Others. *Avtomatnoye upravleniye asinkhronnymi protsessami v EVM i diskretnykh sistemakh* [Automatic Control of Asynchronous Processes in Computers and Discrete Systems]. Moscow, Nauka Publ., 1986.
9. Lazarev V.G., Piy I. Ye.I. *Sintez upravlyayushchikh avtomatov* [The Control Automata Synthesis]. Moscow, Energoatomizdat Publ., 1989.
10. Kotov V.E. *Seti Petri* [Petri Nets]. Moscow, Nauka Publ., 1984.
11. Gurevich D.S. *Pogloshchayushchiye seti Petri i ikh ispolzovaniye pri razrabotke tsifrovyykh vychislitelnykh sistem s raspredelennoy strukturoy* [Absorbing Petri Nets and their Use in Developing Digital Computer Systems with a Distributed Structure]. *Avtomatika i vychislitel'naya tekhnika* [Automatics and Computers], 1990, no. 2. pp. 80–87.
12. Kozhevnikov V.V. *Metod analiza dostizhimosti ingibitornykh setey Petri* [The Accessibility-Analysis Method of Ingibitory Petri Nets]. *Avtomatizatsiya protsessov upravleniya* [Automation of Control Processes], 2013, no. 3 (33), pp. 32–37.
13. Kozhevnikov V.V. *Kontseptsiya matematicheskogo modelirovaniya mikroprogrammuyemykh ustroystv* [The Mathematical Modeling of Programmable Units]. *Izvestiya RAN. Tekhnicheskaya kibernetika* [Proceedings of the Russian Academy of Sciences. Engineering Cybernetics], 1992, no. 4, pp. 175–179.
14. Murata T. *Seti Petri. Svoystva, analiz, prilozheniya* [Petri Nets. Features, Analysis, Applications]. *ТИИЭР*

[Proceedings of the IEEE], 1989, vol. 77, no. 44, pp. 41–85.

15. Girault C., Volk R. Petri nets for systems engineering – A guide to modelling, verification and applications. Springer-Verlag, 2003.

16. Cortadella J., Kishinevsky M., Kondratyev A., Lavagno L., Yakovlev A. Logic synthesis of asynchronous controllers and interfaces. – Springer-Verlag, 2002.

17. Phillips D., Garcia-Diaz A. *Metody analiza setey* [Network Analysis Methods]. Moscow, Mir Publ., 1984.

18. Kozhevnikov V.V. Metod matematicheskogo modelirovaniya logicheskikh skhem tsifrovyykh avtomatov [Method of Mathematical Modeling of Logic Circuits of Digital Automata]. *Avtomatizatsiya protsessov upravleniya* [Automation of Control Processes], 2012, no. 4 (30), pp. 97–101.

19. Zaytsev D.A. Resheniye fundamentalnogo uravneniya setey Petri v protsesse kompozitsii funktsionalnykh podsetey [Solution of Fundamental Equation of Petri Nets during the Functional Subnetwork Composition]. *Iskusstvennyy intellekt* [Artificial Intelligence], 2005, no.1, pp. 59–68.

20. Khan A.A. and Others. O nakhozhenii resheniya uravneniya sostoyaniy setey Petri iz klassa uravneniy Murata [On Solution of Equation of Petri Nets States from Murat Equations]. *TMIЭP* [Proceedings of the IEEE], 1981, vol. 69, no. 4, pp. 71–72.

21. Kryvy S.L. O nekotorykh metodakh resheniya i kriteriyakh sovmestimosti sistem lineynykh diofantovykh uravneniy v oblasti naturalnykh chisel [On some Solution

Methods and System-Compatibility Criteria of Linear Diophantine Equations in Natural Number Domain]. *Kibernetika i sistemnyy analiz* [Cybernetics and System Analysis], 1999, no. 4, pp. 12–36.

22. Zaytsev D.A. K voprosu o vychislitelnoy slozhnosti metoda Tudika [To the Problem on Computational Complexity of the Toudic Method]. *Iskusstvennyy intellekt* [Artificial Intelligence], 2004, no. 1, pp. 29–37.

23. Smagin A.A., Kozhevnikov V.V., Kruglova I.V. Resheniye uravneniya sostoyaniy sistem upravleniya [Solutions of Equations of the Control System States]. *Izvestiya samarskogo nauchnogo tsentra RAN* [Proceedings of Samara Research Center of the Russian Academy of Sciences], 2013, vol. 15, no. 4(3), pp. 705–708.

24. Dixon J. D. Exact solution of linear equations using P-adic expansions. *Numer. Math.*, vol.40, 1982, pp. 137–141.

25. Sergeev M.B. Gibridnyy razryadnyy metod resheniya sistem uravneniy v tselochislennoy arifmetike [The Hybrid Discharge Solution Method for systems of Equations in the Integer Arithmetic]. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2003, no. 2–3, pp. 16–18.

26. Kozhevnikov V.V. Metody postroyeniya i analiza setevoy diagnosticheskoy modeli mikroprogrammiruyemykh struktur [Methods for Building and Analysis of Network Diagnosis Model of Micro-programmed Structures]. *Avtomatika i vychislitel'naya tekhnika* [Automatics and Computers], 1988, no. 5, pp. 90–95.