

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

LOGICAL DESIGN



УДК 519.711
<https://doi.org/10.37661/1816-0301-2024-21-3-7-22>

Оригинальная статья
Original Article

Декомпозиция параллельного автомата в сеть последовательных автоматов и энергосберегающее кодирование их состояний при асинхронной реализации

Ю. В. Поттосин

*Объединенный институт проблем информатики
Национальной академии наук Беларуси,
ул. Сурганова, 6, Минск, 220012, Беларусь
E-mail: pott@newman.bas-net.by*

Аннотация

Цели. Рассматриваются задача декомпозиции параллельного автомата в сеть последовательных автоматов при асинхронной реализации и противогоночное энергосберегающее кодирование их состояний. Целью работы является исследование возможности применения приема декомпозиции при кодировании частичных состояний для снижения размерности задачи с учетом особенностей асинхронной реализации.

Методы. Заданный параллельный автомат разлагается в сеть последовательных асинхронных автоматов, состояния которых кодируются затем троичными векторами. Снижение энергопотребления проектируемого устройства достигается за счет уменьшения интенсивности переключений элементов памяти, которая оценивается с помощью вероятностей переходов между состояниями автомата. Кодирование состояний сводится к задаче минимального взвешенного покрытия. Вероятности переходов между состояниями вычисляются путем решения системы линейных уравнений согласно методу Чэпмена – Колмогорова.

Результаты. Описан способ построения сети из последовательных асинхронных автоматов, реализующей заданный параллельный автомат. Затронута задача минимизации межкомпонентных связей в сети.

Заключение. Использование декомпозиции параллельного автомата позволяет снизить размерность трудоемкой задачи кодирования состояний. Предлагаемый метод предназначен для применения в системах автоматизированного проектирования дискретных устройств.

Ключевые слова: параллельный автомат, частичное состояние, декомпозиция автоматов, кодирование состояний автомата, вероятность перехода между состояниями, взвешенное покрытие

Для цитирования. Поттосин, Ю. В. Декомпозиция параллельного автомата в сеть последовательных автоматов и энергосберегающее кодирование их состояний при асинхронной реализации / Ю. В. Поттосин // Информатика. – 2024. – Т. 21, № 3. – С. 7–22. <https://doi.org/10.37661/1816-0301-2024-21-3-7-22>

Конфликт интересов. Автор заявляет об отсутствии конфликта интересов.

Поступила в редакцию | Received 10.06.2024
Подписана в печать | Accepted 27.06.2024
Опубликована | Published 30.09.2024

Decomposition of a parallel automaton into a net of sequential automata and low power state assignment of them at asynchronous implementation

Yuri V. Pottosin

*The United Institute of Informatics Problems
of the National Academy of Sciences of Belarus,
st. Surganova, 6, Minsk, 220012, Belarus
E-mail: pott@newman.bas-net.by*

Abstract

Objectives. The problems of decomposition of a parallel automaton into a net of sequential automata at synchronous realization and low power race free state assignment of them are considered. The objective of the paper is to investigate the possibilities of applying decomposition in state assignment of partial states in order to decrease the problem dimension taking into account the peculiar properties of the asynchronous realization.

Methods. The given parallel automaton is decomposed into a net of sequential asynchronous automata whose states are assigned then with ternary vectors. The power consumption lowering of the designed device is achieved by lowering the intensity of its memory elements switching that is appreciated by probabilities of transitions between the states of the automaton. The state assignment is reduced to the problem of minimal weighted cover. The probabilities of transitions between sets are calculated by means of solving a system of linear equations according to the Chapman – Kolmogorov method.

Results. A method to construct a net of sequential asynchronous automata that realizes the given parallel automaton is described. The paper touches upon the problem of minimization of interconnections in the net.

Conclusion. Applying parallel automaton decomposition allows decreasing the dimension of the laborious problem of state assignment. The proposed method is intended for application in computer aided systems for design of discrete devices.

Keywords: parallel automaton, partial state, decomposition of automata, state assignment of automata, probability of transition between states, weighted cover

For citation. Pottosin Yu. V. *Decomposition of a parallel automaton into a net of sequential automata and low power state assignment of them at asynchronous implementation*. Informatika [Informatics], 2024, vol. 21, no. 3, pp. 7–22 (In Russ.). <https://doi.org/10.37661/1816-0301-2024-21-3-7-22>

Conflict of interest. The author declares of no conflict of interest.

Введение. Одним из способов снижения размерности задач при синтезе дискретных устройств является разложение исходного устройства на блоки. Этой цели служит решение задачи декомпозиции автоматов. Параллельный автомат является функциональной моделью дискретного устройства управления с достаточной степенью удобства, представляющей параллелизм управляемых взаимодействующих процессов [1]. Такая модель близка к широко известной сети Петри [2]. В отличие от последовательного автомата, который в какой-то момент времени находится только в одном состоянии, параллельный автомат может одновременно находиться в нескольких частичных состояниях. Любая совокупность частичных состояний, в которых параллельный автомат может находиться одновременно, не находясь ни в каком другом состоянии, называется глобальным состоянием. Синтез логической структуры устройства, поведение которого представлено параллельным автоматом, лежит через кодирование его частичных состояний. Решение этой задачи можно заменить нахождением всех его глобальных состояний и последующим их кодированием каким-либо из известных методов, разработанных применительно к последовательным автоматам. Однако при сравнительно небольшом количестве частичных состояний параллельного автомата число его глобальных состояний, как показано в работе [3], может быть настолько велико, что задачу их кодирования невозможно решить за приемлемое время. Кроме того, при асинхронной реализации параллельный автомат не всегда моделируется последовательным автоматом [3], и указанный подход вообще не может быть

использован. Поэтому были разработаны методы специально для кодирования частичных состояний параллельного автомата [4–6]. Они используют приемы, отличающиеся от тех, которые применяются при кодировании состояний последовательных автоматов. В настоящей работе предлагается сводить кодирование частичных состояний параллельного автомата к совместному кодированию состояний компонентных автоматов сети, реализующей заданный параллельный автомат. Каждый из компонентных автоматов данной сети является последовательным асинхронным автоматом, и поэтому для кодирования его состояний можно использовать какой-либо из известных методов, предназначенных для таких автоматов.

Одним из важных критериев оптимизации при проектировании дискретных устройств является величина энергии, потребляемой проектируемым устройством. Это обусловлено, с одной стороны, стремлением увеличить время действия источника энергии в портативных приборах, а с другой – стремлением снизить остроту проблемы отвода тепла при проектировании сверхбольших интегральных схем.

Как отмечено в работах [7, 8], потребляемая мощность схемы, построенной на основе КМОП-технологии, пропорциональна интенсивности переключений логических элементов и элементов памяти. Это дает возможность частично решать данную проблему на уровне логического проектирования. В частности, снижения энергопотребления можно добиваться на этапе кодирования состояний автомата, т. е. когда абстрактным символам состояний приписываются булевы векторы, что необходимо для получения системы булевых функций, представляющей структурную модель проектируемого устройства. Для последовательных автоматов при синхронной и асинхронной реализации эта задача решалась в работах [9–14].

Метод энергосберегающего кодирования частичных состояний параллельного автомата при синхронной реализации описан в статье [15], где используется подход из работы [16], сводящий данную задачу к поиску покрытия графа его полными двудольными подграфами. Декомпозиционный подход к энергосберегающему кодированию состояний при синхронной реализации представлен в статье [17], в которой показано, как применять для совместного кодирования состояний компонентных автоматов сети, реализующей параллельный автомат, итеративный метод, описанный в работе [18].

Использование приема декомпозиции параллельного автомата, т. е. разложения его в сеть последовательных автоматов, позволяет снизить, как было сказано, размерность задачи. Кроме того, разложение проектируемого устройства на блоки дает возможность снизить энергопотребление путем блокировки работы некоторых блоков [19].

Описание модели. Параллельный автомат состоит из следующих объектов: $Q = \{q_1, q_2, \dots, q_l\}$ – множества *частичных состояний*, $X = \{x_1, x_2, \dots, x_n\}$ – множества входных булевых переменных, $Y = \{y_1, y_2, \dots, y_m\}$ – множества выходных булевых переменных и последовательности строк-переходов $T = \{\tau_1, \tau_2, \dots, \tau_l\}$, имеющих вид [3]

$$\tau_i = S_i : -K_i \rightarrow K_i' \rightarrow S_i', \quad (1)$$

где S_i и S_i' – подмножества множества Q ; K_i – элементарная конъюнкция переменных из множества X и K_i' – элементарная конъюнкция переменных из множества Y .

Все множество частичных состояний, в которых рассматриваемый параллельный автомат находится в некоторый момент времени, называется *глобальным состоянием*. Смысл выражения (1) заключается в следующем. Если автомат находится одновременно в состояниях, составляющих множество S_i , и булевы переменные приняли значения, обращающие конъюнкцию K_i в единицу, то конъюнкция K_i' приобретает значение 1 и автомат переходит из частичных состояний, составляющих множество S_i , в частичные состояния, составляющие множество S_i' . Другими словами, пусть $P = \{P_1, P_2, \dots, P_p\}$ – множество всех достижимых глобальных состояний заданного параллельного автомата. Тогда если $S_i \subseteq P_g$, где P_g – текущее глобальное состояние автомата, и моделируемое устройство принимает на входе двоичные сигналы, обращающие конъюнкцию K_i в единицу, то глобальным состоянием в следующий момент времени будет $P_h = (P_g \setminus S_i) \cup S_i'$, а на выходе устройство выдаст двоичные сигналы, обращающие конъюнкцию K_i' в единицу. Любая из конъюнкций K_i или K_i' может отсутствовать в приведенной строке.

Отсутствие K_i означает ее тождественное равенство единице. Отсутствие K_i' означает в зависимости от интерпретации данной модели либо то, что все переменные из множества Y обращаются в нуль, либо то, что значения сигналов на выходе не меняются. Так же, как для последовательного автомата, возможны синхронная и асинхронная реализации параллельного автомата. Далее будет рассматриваться асинхронная реализация. При такой реализации время такта функционирования автомата не фиксировано и определяется изменениями внешнего входного сигнала. При поступлении сигнала на вход автомат приходит в устойчивые частичные состояния, из которых он не уходит до очередного изменения входного сигнала. Это изменение не должно произойти раньше, чем установятся частичные состояния. В настоящей работе не рассматриваются выходные сигналы и конъюнкция K_i' будет опускаться. Проблема устранения «дребезжания» выходных сигналов, т. е. многократного изменения значений, во время переходов между состояниями решается на уровне схемы, как, например, в работе [20].

В описываемую модель внесены следующие ограничения и допущения:

1. Вводится начальное глобальное состояние в виде одноэлементного множества. Для определенности можно взять $\{q_1\}$.

2. Для двух различных строк, i -й и j -й, если $S_i \cap S_j \neq \emptyset$, то $S_i = S_j$.

3. При переходе из одного устойчивого частичного состояния в другое автомат может проходить некоторую последовательность неустойчивых состояний. Время пребывания в любом из них пренебрежимо мало.

Не будем, как сказано выше, обращать внимание на выходные сигналы, ограничимся рассмотрением переключений элементов памяти.

Примером параллельного автомата может служить следующая последовательность строк:

$$\begin{aligned}\tau_1 &= 1: -x_2 \rightarrow 10; \\ \tau_2 &= 10: -\bar{x}_2 \rightarrow 2.3.4; \\ \tau_3 &= 2: -x_1 \rightarrow 5.6; \\ \tau_4 &= 3.5: -x_2 \rightarrow 8; \\ \tau_5 &= 4: -x_2 \rightarrow 7; \\ \tau_6 &= 4: -\bar{x}_1 \rightarrow 9; \\ \tau_7 &= 7: -\bar{x}_2 \rightarrow 9; \\ \tau_8 &= 6.8.9: -x_1 \rightarrow 1.\end{aligned}$$

Данный параллельный автомат допускает асинхронную реализацию и моделирование асинхронным последовательным автоматом. Здесь $Q = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ и $X = \{x_1, x_2\}$. За начальное глобальное состояние примем одноэлементное множество $\{1\}$. Первая строка (переход τ_1) означает, что, если $x_2 = 1$, то независимо от того, какое значение имеет x_1 , автомат уходит из состояния $\{1\}$ и входит в состояние $\{10\}$, которое также является глобальным. При $x_2 = 0$ автомат остается в состоянии $\{1\}$. Из состояния $\{10\}$ при $x_2 = 0$ автомат переходит в частичные состояния 2, 3 и 4, но при $x_1 = 0$ состояние 4 оказывается неустойчивым и меняется на частичное состояние 9 (переход τ_6). Устойчивым глобальным состоянием тогда является $\{2, 3, 9\}$. При $x_1 = 1$ состояние 2 оказывается неустойчивым, из него автомат переходит в частичные состояния 5 и 6 (переход τ_3) и оказывается в устойчивом глобальном состоянии $\{3, 4, 5, 6\}$. Проследив таким образом функционирование данного параллельного автомата, получим еще устойчивые глобальные состояния $\{3, 5, 6, 9\}$, $\{6, 7, 8\}$ и $\{6, 8, 9\}$. Далее будем рассматривать только устойчивые частичные и глобальные состояния.

Построение сети последовательных автоматов, реализующей заданный параллельный автомат. Решение данной задачи подробно представлено в статье [17]. Пусть задан некоторый параллельный автомат B , описанием которого является последовательность строк вида (1). Рассмотрим некоторую сеть $N = (X, A_1, A_2, \dots, A_n)$, где X – множество входных булевых переменных, тех же, что присутствуют в задании автомата B , и A_1, A_2, \dots, A_n – компонентные последовательные автоматы с множествами внутренних состояний Q_1, Q_2, \dots, Q_n . Каждый из автома-

тов A_j является автоматом без выходов, т. е. для него заданы только переходы, которые представим в форме, аналогичной выражению (1):

$$q^j : -\alpha_j(x_1, x_2, \dots, x_n, q^1, q^2, \dots, q^{j-1}, q^{j+1}, \dots, q^n) \rightarrow q^{j'}, \quad (2)$$

где α_j – предикат над переменными из множества X и состояниями компонентных автоматов $A_1, A_2, \dots, A_{j-1}, A_{j+1}, \dots, A_n$, принимающий значение 1 при определенной комбинации значений некоторых входных переменных и определенной совокупности состояний некоторых компонентных автоматов. Если $\alpha_j = 1$, то автомат A_j переходит из состояния q^j в состояние $q^{j'}$, в противном случае остается в состоянии q^j .

Сеть N реализует автомат B , если существует такое отображение φ множества $D \subseteq Q_1 \times Q_2 \times \dots \times Q_n$ в некоторую совокупность подмножеств множества Q , что для любого перехода (1) автомата B при $S_i \subseteq \varphi(q^1, q^2, \dots, q^n)$ имеем $S_i' \subseteq \varphi(q^{1'}, q^{2'}, \dots, q^{n'})$, если $\alpha_j = 1$, и $S_i' \subseteq \varphi(q^1, q^2, \dots, q^n)$, если $\alpha_j = 0$, согласно (2). Если в случае асинхронной реализации параллельный автомат B допускает моделирование последовательным автоматом, то можно построить последовательный автомат A , эквивалентный автомату B , с состояниями, соответствующими глобальным состояниям автомата B . Описанная сеть в этом случае реализует автомат A в смысле работ [21, 22].

Определим множество состояний Q_j компонентного автомата A_j ($j = 1, 2, \dots, n$) сети N следующим образом. Каждому состоянию $q^j \in Q_j$ поставим во взаимно однозначное соответствие некоторое частичное состояние q автомата B , причем среди всех частичных состояний, соответствующих состояниям из Q_j , не должно быть ни одной пары *параллельных* состояний, т. е. таких, в которых автомат B может находиться одновременно. Указанное соответствие обозначим $f_j(q^j) = q$. Пусть множество компонентных автоматов $\{A_1, A_2, \dots, A_n\}$ сети N достаточно для того, чтобы для каждого частичного состояния q автомата B существовала по крайней мере одна компонента A_j с таким состоянием q^j , что $f_j(q^j) = q$.

Переходы в компонентных автоматах определим согласно выражению (2) следующим образом. Каждой строке τ_i вида (1) из описания автомата B поставим в соответствие совокупность переходов в тех компонентных автоматах A_j , каждый из которых имеет такое состояние q^j , что $f_j(q^j) \in S_i \cup S_i'$. Для компонентного автомата A_j в качестве левой части выражения (2) возьмем то состояние q^j , для которого $f_j(q^j) \in S_i$. Таких состояний в Q_j ровно одно, поскольку по определению множества Q_j оно не содержит ни одной пары состояний, соответствующих параллельным частичным состояниям автомата B . Состояние $q^{j'}$ в выражении (2) определяем так, чтобы имело место $f_j(q^{j'}) \in S_i'$. При этом полагаем, что $\alpha_j = 1$ тогда и только тогда, когда $K_i = 1$ и $\{f_1(q^1), f_2(q^2), \dots, f_{j-1}(q^{j-1}), f_{j+1}(q^{j+1}), \dots, f_n(q^n)\} \supseteq S_i'$.

Множество последовательных автоматов $\{A_1, A_2, \dots, A_n\}$, образующих сеть N , которая реализует заданный параллельный автомат B , строим следующим образом. Получим все максимальные множества взаимно непараллельных частичных состояний автомата B . Максимальных в том смысле, что любое частичное состояние, не принадлежащее какому-либо из этих множеств, оказывается параллельным хотя бы одному из принадлежащих ему. Далее следует получить кратчайшее покрытие (естественно потребовать минимум компонентных автоматов) этими множествами всех пар частичных состояний, связанных переходами. Каждое множество из полученного покрытия соответствует множеству состояний одного из компонентных последовательных автоматов, составляющих искомую сеть N .

В статье [23] описан алгоритм установления параллельности частичных состояний, который для параллельных автоматов рассматриваемого класса решает данную задачу за полиномиальное время. Как отмечается далее, в случае энергосберегающего кодирования приходится рассматривать глобальные состояния параллельного автомата и по ним легко установить параллельность частичных состояний. Для примера параллельного автомата, описание которого приведено выше и перечислены глобальные состояния, матрица отношения параллельности на множестве частичных состояний имеет вид

1	2	3	4	5	6	7	8	9	10	
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	2
0	1	0	1	1	1	0	0	1	0	3
0	0	1	0	1	1	0	0	0	0	4
0	0	1	1	0	1	0	0	1	0	5
0	0	1	1	1	0	1	1	1	0	6
0	0	0	0	0	1	0	1	0	0	7
0	0	0	0	0	1	1	0	1	0	8
0	1	1	0	1	1	0	1	0	0	9
0	0	0	0	0	0	0	0	0	0	10

Ее можно рассматривать как матрицу смежности графа, представляющего отношение параллельности. Упомянутые множества непараллельных частичных состояний соответствуют независимым множествам данного графа. Способы нахождения максимальных независимых множеств в графе описаны в работе [18]. В данном графе максимальными независимыми являются множества {1, 2, 4, 7, 10}, {1, 3, 7, 10}, {1, 2, 5, 7, 10}, {1, 2, 6, 10}, {1, 2, 4, 8, 10}, {1, 3, 8, 10}, {1, 2, 5, 8, 10} и {1, 4, 7, 9, 10}. Табл. 1 представляет собой таблицу покрытия, в которой строкам соответствуют максимальные множества взаимно непараллельных частичных состояний, а столбцам – пары частичных состояний, связанных переходами. Задача здесь решается тривиально: выбираются строки, содержащие единицу, единственную в каком-либо столбце, и они составляют искомое покрытие. В данном случае искомое покрытие составляют множества {1, 2, 6, 10}, {1, 3, 8, 10}, {1, 2, 5, 8, 10} и {1, 4, 7, 9, 10}.

Таблица 1
 Table 1

	1-6	1-8	1-9	1-10	2-5	2-6	2-10	3-8	3-10	4-7	4-9	4-10	5-8	7-9
1,2,4,7,10				1			1			1		1		
1,3,7,10				1					1					
1,2,5,7,10				1	1		1							
1,2,6,10	1			1		1	1							
1,2,4,8,10		1		1			1					1		
1,3,8,10		1		1				1	1					
1,2,5,8,10		1		1	1		1						1	
1,4,7,9,10			1	1						1	1	1		1

Состояния компонентных автоматов обозначим теми же символами, которые обозначают соответствующие им частичные состояния заданного параллельного автомата. Таким образом, искомую сеть составляют четыре автомата: A_1 с множеством состояний $Q_1 = \{1, 2, 5, 8, 10\}$, A_2 с множеством состояний $Q_2 = \{1, 2, 6, 10\}$, A_3 с множеством состояний $Q_3 = \{1, 3, 8, 10\}$ и A_4 с множеством состояний $Q_4 = \{1, 4, 7, 9, 10\}$. Переходы между состояниями определяются согласно выражению (2), как описано выше. Поведение получаемой сети N зададим в виде системы не полностью определенных функций $q^{1'}$, $q^{2'}$, $q^{3'}$ и $q^{4'}$, принимающих значения соответственно из множеств Q_1 , Q_2 , Q_3 и Q_4 . Их аргументами являются булевы переменные x_1 , x_2 и многозначные переменные q^1 , q^2 , q^3 и q^4 , принимающие значения из тех же множеств, что и указанные функции. Данную систему функций представляет табл. 2 – обычное табличное задание дискретных функций. В ней символ «–» обозначает безразличное значение переменной.

В результате кодирования состояний, т. е. замены каждой переменной q^j и, соответственно, $q^{j'}$ булевым вектором с компонентами z_i^j (соответственно $z_i^{j'}$), представляющими состояния двоичных элементов памяти, система функций, заданная табл. 2, преобразуется в систему булевых функций.

Метод противогоночного кодирования состояний асинхронного автомата. Для энерго-сберегающего противогоночного кодирования состояний асинхронного последовательного автомата используем способ, подробно описанный в статье [13].

Таблица 2
Table 2

Номер строки <i>Line number</i>	Внешний вход <i>External input</i>	Начальное состояние <i>Initial state</i>	Конечное состояние <i>Final state</i>	Номер строки <i>Line number</i>	Внешний вход <i>External input</i>	Начальное состояние <i>Initial state</i>	Конечное состояние <i>Final state</i>
	$x_1 x_2$	$q^1 q^2 q^3 q^4$	$q^{1'} q^{2'} q^{3'} q^{4'}$		$x_1 x_2$	$q^1 q^2 q^3 q^4$	$q^{1'} q^{2'} q^{3'} q^{4'}$
1	- 0	1 1 1 1	1 1 1 1	12	1 0	5 6 3 4	5 6 3 4
2	- 1	1 1 1 1	10 10 10 10	13	- 0	5 6 3 9	5 6 3 9
3	- 1	10 10 10 10	10 10 10 10	14	0 1	5 6 3 9	8 6 8 9
4	0 0	10 10 10 10	2 2 3 9	15	1 1	5 6 3 9	10 10 10 10
5	1 0	10 10 10 10	5 6 3 4	16	- 1	8 6 8 7	8 6 8 7
6	0 -	2 2 3 9	2 2 3 9	17	1 0	8 6 8 7	1 1 1 1
7	1 0	2 2 3 9	5 6 3 9	18	0 0	8 6 8 7	8 6 8 9
8	1 1	2 2 3 9	10 10 10 10	19	0 -	8 6 8 9	8 6 8 9
9	0 0	5 6 3 4	5 6 3 9	20	1 1	8 6 8 9	10 10 10 10
10	0 1	5 6 3 4	8 6 8 9	21	1 0	8 6 8 9	1 1 1 1
11	1 1	5 6 3 4	8 6 8 7				

Явление одновременного переключения элементов памяти называется *состязаниями* или *гонками* элементов памяти [3]. Принято называть состязания *неопасными*, если все промежуточные состояния, в которых автомат может оказаться при переходе из одного состояния в другое под воздействием некоторого входного сигнала, являются неустойчивыми для этого сигнала, т. е. при любом порядке переключений элементов памяти автомат из некоторого состояния q_i переходит всегда в состояние q_j , определяемое функцией переходов. Если же при этом автомат может оказаться в некотором устойчивом состоянии q_k , отличном от q_j , то состязания называются *опасными*. Кодирование состояний, обеспечивающее отсутствие опасных состязаний (гонок), называется *противогоночным*.

Рассматриваются пары переходов между состояниями при одном и том же сигнале на входе. Например, при $x_1 = 0, x_2 = 1$ согласно строкам 2 и 10 табл. 2 у автоматов A_1, A_2, A_3 и A_4 имеются пары переходов $(1 \rightarrow 10, 5 \rightarrow 8), (1 \rightarrow 10, 6 \rightarrow 6), (1 \rightarrow 10, 3 \rightarrow 8)$ и $(1 \rightarrow 10, 4 \rightarrow 9)$ соответственно. Условие отсутствия опасных состязаний на паре переходов формулируется в виде троичного вектора, компонентам которого соответствуют состояния автомата, и компоненты, соответствующие состояниям одной пары, имеют значение 1, а другой пары – значение 0. Такими векторами для перечисленных пар переходов являются $(0 - 1 1 0), (0 - 1 0), (0 1 1 0)$ и $(0 1 - 1 0)$, где единицы и нули могут меняться местами. Все условия отсутствия опасных состязаний в виде описанных векторов составляют так называемую *матрицу условий*, в которой отсутствуют имплицитруемые строки [3]. Троичный вектор **a** имплицитрует троичный вектор **b**, если **b** получается из **a** заменой некоторых нулей или единиц значением «-» и, возможно, инвертированием полученного результата. Например, вектор $(1 0 - - 1 0 1)$ имплицитрует $(1 0 - - - 0 1)$ и $(0 1 - - - 1 -)$. Смысл этого отношения в том, что условие, представленное вектором **b**, автоматически выполняется при соблюдении условия, представленного вектором **a**. Для компонентных автоматов рассматриваемой сети матрицы условий имеют следующий вид:

$$\begin{matrix}
 \begin{matrix} 1 & 2 & 5 & 8 & 10 \\ \begin{bmatrix} 0 & 1 & - & - & 1 \\ 0 & - & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & - \\ 0 & 1 & - & - & 0 \\ 0 & - & 1 & 1 & 0 \\ - & 0 & 1 & 1 & 0 \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}
 \end{matrix} &
 \begin{matrix} 1 & 2 & 6 & 10 \\ \begin{bmatrix} 0 & 1 & - & 1 \\ 0 & - & 1 & 1 \\ 0 & 1 & 1 & - \\ 0 & 1 & - & 0 \\ 0 & - & 1 & 0 \\ - & 0 & 1 & 0 \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}
 \end{matrix} &
 \begin{matrix} 1 & 3 & 8 & 10 \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} & \begin{matrix} 1 \\ 2 \end{matrix}
 \end{matrix} &
 \begin{matrix} 1 & 4 & 7 & 9 & 10 \\ \begin{bmatrix} 0 & - & - & 1 & 1 \\ 0 & 1 & 0 & - & 1 \\ 0 & 1 & - & 1 & 0 \\ 0 & - & 1 & 1 & - \\ 0 & 1 & 1 & - & 0 \\ 1 & 0 & - & 1 & 0 \\ - & 0 & 0 & 1 & 1 \\ - & 0 & 1 & 0 & - \\ 0 & - & 0 & 1 & - \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix}
 \end{matrix}
 \end{matrix}$$

Троичная матрица R *имплицирует* троичную матрицу S , если для каждой строки матрицы S в матрице R найдется имплицирующая ее строка. Задача противогоночного кодирования с минимизацией длины кода состояния сводится к нахождению матрицы с минимальным числом строк, имплицирующей матрицу условий и называемой *кратчайшей имплицирующей формой* матрицы условий. Столбцы этой матрицы будут представлять искомые коды состояний, а получаемая в результате ее транспонирования матрица называется *матрицей кодирования*. Строкам матрицы кодирования соответствуют состояния автомата, а столбцам – внутренние переменные. Строки этой матрицы представляют коды соответствующих состояний.

Кратчайшая имплицирующая форма матрицы условий находится следующим образом. Множество строк матрицы условий называется *совместимым*, если существует вектор, имплицирующий каждую строку этого множества. Совместимое множество называется *максимальным*, если оно не является собственным подмножеством другого совместимого множества. Необходимо найти кратчайшее покрытие множества строк матрицы условий максимальными совместимыми множествами. Каждому совместимому множеству соответствует вектор, имплицирующий все строки, принадлежащие этому множеству. Указанные векторы, соответствующие элементам полученного покрытия, в качестве строк составят кратчайшую имплицирующую форму заданной матрицы условий. Ниже приведены матрицы, строки которых представляют максимальные совместимые множества и соответствующие имплицирующие векторы. Справа каждой строки показаны номера имплицируемых ею строк матрицы условий.

$$\begin{array}{c}
 1 \ 2 \ 5 \ 8 \ 10 \\
 \left[\begin{array}{ccccc} 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{array} \right] \begin{array}{l} 1,2,3 \\ 1,6 \\ 3,4 \\ 4,5 \\ 5,6 \end{array} , \\
 \end{array}
 \quad
 \begin{array}{c}
 1 \ 2 \ 6 \ 10 \\
 \left[\begin{array}{cccc} 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{array} \right] \begin{array}{l} 1,2,3 \\ 1,6 \\ 3,4,5 \\ 5,6 \end{array} , \\
 \end{array}
 \quad
 \begin{array}{c}
 1 \ 3 \ 8 \ 10 \\
 \left[\begin{array}{ccc} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array} \right] \begin{array}{l} 1, \\ 2 \end{array} , \\
 \end{array}
 \quad
 \begin{array}{c}
 1 \ 4 \ 7 \ 9 \ 10 \\
 \left[\begin{array}{ccccc} 0 & 1 & 0 & 1 & 1 \\ 0 & - & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{array} \right] \begin{array}{l} 1,2,8,9 \\ 1,4 \\ 1,7,9 \\ 2,6 \\ 3,4,5 \\ 3,8,9 \\ 5,7 \end{array} . \\
 \end{array}$$

При применении описанного подхода к решению задачи противогоночного кодирования состояний асинхронного автомата для снижения интенсивности переключений элементов памяти можно использовать следующие соображения. Каждому i -му столбцу матрицы кодирования можно поставить в соответствие множество переходов. Это множество составляют переходы, которыми связаны состояния автомата. В кодах состояний переменная z_i имеет различные значения, т. е. при таких переходах i -й триггер в реальной схеме, реализующей заданный автомат, меняет свое состояние. Очевидно, интенсивность переключений элементов памяти определяется вероятностями переходов между состояниями. Столбцу матрицы кодирования состояний ставится в соответствие вероятность события. Это событие заключается в том, что происходит некоторый переход, при котором меняется значение переменной, соответствующей данному столбцу. Поскольку переходы между состояниями автомата являются несовместимыми событиями, эта вероятность равна сумме вероятностей отдельных переходов из множества, соответствующего i -му столбцу.

Каждому совместимому множеству строк матрицы условий и, соответственно, вектору, имплицирующему все строки из данного множества, приписывается вес в виде величины, пропорциональной сумме вероятностей переходов, связанных с этим вектором. Искомое решение получается в виде покрытия множества строк матрицы условий максимальными совместимыми множествами, обладающего минимальным весом. Весом покрытия является сумма весов принадлежащих ему элементов.

Вычисление вероятностей переходов. Для подсчета вероятностей переходов между состояниями последовательного автомата принимаются следующие предположения: автомат является полностью определенным; все состояния являются взаимно достижимыми, т. е. для любых

двух состояний существует последовательность входных сигналов, переводящая автомат из одного состояния в другое; автомат работает достаточно долго; время пребывания автомата в устойчивом состоянии для всех состояний одно и то же.

Вероятность перехода последовательного автомата из состояния q_i в состояние q_j , вызываемого входным сигналом $\mathbf{x} = (x_1, x_2, \dots, x_n)$, равна вероятности прихода входного сигнала \mathbf{x} . Если имеется несколько входных сигналов, переводящих автомат из состояния q_i в состояние q_j , условная вероятность p'_{ij} такого перехода равна сумме вероятностей этих сигналов как вероятность несовместимых событий. Условием вероятности является то, что автомат находится в состоянии q_i . Абсолютная вероятность p_{ij} перехода из состояния q_i в состояние q_j в течение всего времени работы автомата равна произведению $p(q_i)p'_{ij}$, где $p(q_i)$ – вероятность того, что автомат находится в состоянии q_i (это событие и приход сигналов, меняющих состояние q_i на q_j , являются независимыми событиями).

Для вычисления вероятностей $p(q_i)$, $i = 1, 2, \dots, m$, где m – число состояний автомата, можно использовать уравнения Чэпмена – Колмогорова для дискретных Марковских цепей [24]. Подобно закону Кирхгоффа из электротехники, можно сказать, что сумма вероятностей переходов в некоторое состояние равна сумме вероятностей переходов из этого состояния. На основании приведенных соображений можно вывести следующую систему уравнений с неизвестными $p(q_i)$ ($i = 1, 2, \dots, m$):

$$\sum_{i=1}^m p(q_i)p'_{ij} = p(q_j), \quad j = 1, 2, \dots, m,$$
$$\sum_{i=1}^m p(q_i) = 1.$$

Вероятности p'_{ij} должны быть известны. Таким образом, решив эту систему уравнений, получим вероятности $p(q_i)$. Как было сказано раньше, абсолютная вероятность p_{ij} определяется как $p_{ij} = p(q_i)p'_{ij}$.

Для компонентного автомата в сети входной сигнал, кроме внешних сигналов сети, могут составлять сигналы о состояниях других компонентных автоматов. Благодаря внутренним связям то, что компонентный автомат находится в определенном состоянии, и приход некоторого входного для него сигнала нельзя считать независимыми событиями. В случае когда асинхронный параллельный автомат допускает моделирование его последовательным автоматом, предлагается для вычисления вероятностей переходов между состояниями компонентных автоматов использовать такое моделирование. Состояниями моделирующего асинхронного последовательного автомата являются устойчивые глобальные состояния заданного параллельного автомата. Вероятности переходов между глобальными состояниями определяются так, как показано выше для моделирующего последовательного автомата, а вероятности переходов между частичными состояниями и, соответственно, между состояниями компонентных автоматов определяются через вероятности переходов между глобальными состояниями. Вероятность перехода между частичными состояниями q_i и q_j равна сумме вероятностей переходов между теми глобальными состояниями P_s и P_t , для которых $q_i \in P_s$ и $q_j \in P_t$ или $q_i \in P_t$ и $q_j \in P_s$. Считается, что вероятность нахождения автомата в неустойчивом состоянии пренебрежимо мала и приравнивается нулю.

Таблица переходов между устойчивыми глобальными состояниями параллельного автомата из рассматриваемого примера представлена в виде табл. 3, где строки и столбцы соответствуют глобальным состояниям, и на пересечении строки и столбца стоит условие перехода из состояния, соответствующего строке, в состояние, соответствующее столбцу. Здесь и далее пустая клетка в таблице означает, что соответствующего перехода нет. По этой таблице легко определяются условные вероятности переходов, приведенные в табл. 4. Считается, что входные двоичные сигналы x_1 и x_2 независимы и равновероятны.

Таблица 3

Table 3

	1	10	2.3.9	3.4.5.6	3.5.6.9	6.7.8	6.8.9
1	\bar{x}_2	x_2					
10		x_2	$\bar{x}_1 \bar{x}_2$	$x_1 \bar{x}_2$			
2.3.9		$x_1 x_2$	\bar{x}_1		$x_1 \bar{x}_2$		
3.4.5.6				$x_1 \bar{x}_2$	$\bar{x}_1 \bar{x}_2$	$x_1 x_2$	$\bar{x}_1 x_2$
3.5.6.9		$x_1 x_2$			\bar{x}_2		$\bar{x}_1 x_2$
6.7.8	$x_1 \bar{x}_2$					x_2	$\bar{x}_1 \bar{x}_2$
6.8.9	$x_1 \bar{x}_2$	$x_1 x_2$					\bar{x}_1

Таблица 4

Table 4

	1	10	2.3.9	3.4.5.6	3.5.6.9	6.7.8	6.8.9
1	1/2	1/2					
10		1/2	1/4	1/4			
2.3.9		1/4	1/2		1/4		
3.4.5.6				1/4	1/4	1/4	1/4
3.5.6.9		1/4			1/2		1/4
6.7.8	1/4					1/2	1/4
6.8.9	1/4	1/4					1/2

Для вероятностей попадания рассматриваемого автомата в глобальные состояния $P_1 = \{1\}$, $P_2 = \{10\}$, $P_3 = \{2, 3, 9\}$, $P_4 = \{3, 4, 5, 6\}$, $P_5 = \{3, 5, 6, 9\}$, $P_6 = \{6, 7, 8\}$ и $P_7 = \{6, 8, 9\}$ по табл. 4 получим следующую систему уравнений:

$$\begin{aligned}
 p(1) &= 1/2 p(1) + 1/4 p(6.7.8) + 1/4 p(6.8.9); \\
 p(10) &= 1/2 p(1) + 1/2 p(10) + 1/4 p(2.3.9) + 1/4 p(3.5.6.9) + 1/4 p(6.8.9); \\
 p(2.3.9) &= 1/4 p(10) + 1/2 p(2.3.9); \\
 p(3.4.5.6) &= 1/4 p(10) + 1/4 p(3.4.5.6); \\
 p(3.5.6.9) &= 1/4 p(2.3.9) + 1/2 p(3.5.6.9) + 1/4 p(3.4.5.6); \\
 p(6.7.8) &= 1/4 p(3.4.5.6) + 1/2 p(6.7.8); \\
 p(6.8.9) &= 1/4 p(3.4.5.6) + 1/4 p(3.5.6.9) + 1/4 p(6.7.8) + 1/2 p(6.8.9); \\
 p(1) + p(10) + p(2.3.9) + p(3.4.5.6) + p(3.5.6.9) + p(6.7.8) + p(6.8.9) &= 1,
 \end{aligned}$$

решением которой являются вероятности $p(1) = 15/153$, $p(10) = 48/153$, $p(2.3.9) = 24/153$, $p(3.4.5.6) = 16/153$, $p(3.5.6.9) = 20/153$, $p(6.7.8) = 8/153$, $p(6.8.9) = 22/153$ (удобно рассматривать эти дроби при одном и том же знаменателе).

Абсолютные вероятности переходов между глобальными состояниями, полученные по формуле $p_{ij} = p(q_i) p'_{ij}$, представлены в табл. 5.

Таблица 5

Table 5

	1	10	2.3.9	3.4.5.6	3.5.6.9	6.7.8	6.8.9
1	15/306	15/306					
10		24/153	12/153	12/153			
2.3.9		6/153	12/153		6/153		
3.4.5.6				4/153	4/153	4/153	4/153
3.5.6.9		5/153			10/153		5/153
6.7.8	2/153					4/153	2/153
6.8.9	11/306	11/306					11/153

Как было сказано выше, вероятности переходов между частичными состояниями определяются через вероятности переходов между глобальными состояниями. Например, в задании ав-

томата частичное состояние 3 меняется на частичное состояние 8 (переход τ_4). Это происходит при переходах из глобального состояния $\{3, 4, 5, 6\}$ в глобальное состояние $\{6, 7, 8\}$, из того же состояния в глобальное состояние $\{6, 8, 9\}$ и из $\{3, 5, 6, 9\}$ в $\{6, 8, 9\}$. Суммой вероятностей таких несовместимых событий согласно табл. 5 является $13/153$, это вероятность перехода из частичного состояния 3 в частичное состояние 8. Табл. 6 представляет вероятности переходов между частичными состояниями (не важно, в каком направлении), где удалены пустые строки и столбцы, а дроби приведены к общему знаменателю.

Таблица 6
Table 6

	5	6	7	8	9	10
1		15/306	4/306	15/306	11/306	15/306
2	12/306	12/306				36/306
3			16/306	26/306		74/306
4			8/306		16/306	24/306
5				26/306		34/306
6						45/306
7					4/306	
8						11/306
9						45/306

Вероятности переходов между частичными состояниями параллельного автомата совпадают с вероятностями переходов между соответствующими состояниями компонентных автоматов в реализующей его сети.

Кодирование состояний компонентных автоматов и получение функций возбуждения элементов памяти. Ранее отмечалось, что столбцу z_i матрицы кодирования можно приписать вес в виде величины, пропорциональной сумме вероятностей переходов, при которых меняется значение z_i . В качестве этой величины возьмем числитель данной суммы при общем знаменателе всех рассматриваемых дробей. Таким же образом припишем веса максимальным совместимым множествам.

В настоящей работе не рассматривается какой-либо конкретный метод получения минимального взвешенного покрытия, так как эта задача подробно исследована (см., например, [25]). Для автоматов A_1 , A_2 и A_3 задача взвешенного кратчайшего покрытия строк матрицы условий максимальными совместимыми множествами решается просто: обязательным элементом покрытия для A_1 является множество $\{1, 2, 3\}$, для A_2 обязательными элементами покрытия являются $\{1, 2, 3\}$ и $\{3, 4, 5\}$, а у A_3 такое покрытие совпадает с совокупностью одноэлементных максимальных совместимых множеств. Для автомата A_1 искомое покрытие составляют множества $\{1, 2, 3\}$, $\{4, 5\}$, $\{5, 6\}$, для A_2 – множества $\{1, 2, 3\}$, $\{3, 4, 5\}$, $\{5, 6\}$ и для A_3 – $\{1\}$, $\{2\}$. Таблицей покрытия для автомата A_4 является табл. 7. Ее кратчайшим покрытием с минимальным весом составляют строки, соответствующие множествам $\{1, 2, 8, 9\}$, $\{2, 6\}$, $\{3, 4, 5\}$ и $\{5, 7\}$. Полученные покрытия определяют следующие матрицы кодирования состояний:

$$\begin{array}{c}
 \begin{array}{ccc} z_1^1 & z_2^1 & z_3^1 \\ \left[\begin{array}{ccc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{array} \right] & \begin{array}{c} 1 \\ 2 \\ 5 \\ 8 \\ 10 \end{array} & , \\
 \end{array}
 \end{array}
 \begin{array}{c}
 \begin{array}{ccc} z_1^2 & z_2^2 & z_3^2 \\ \left[\begin{array}{ccc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{array} \right] & \begin{array}{c} 1 \\ 2 \\ 6 \\ 10 \end{array} & , \\
 \end{array}
 \end{array}
 \begin{array}{c}
 \begin{array}{cc} z_1^3 & z_2^3 \\ \left[\begin{array}{cc} 0 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{array} \right] & \begin{array}{c} 1 \\ 3 \\ 8 \\ 10 \end{array} & , \\
 \end{array}
 \end{array}
 \begin{array}{c}
 \begin{array}{cccc} z_1^4 & z_2^4 & z_3^4 & z_4^4 \\ \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{array} \right] & \begin{array}{c} 1 \\ 4 \\ 7 \\ 9 \\ 10 \end{array} & . \\
 \end{array}
 \end{array}$$

Качество кодирования состояний автомата можно оценить введенной в работе [11] суммой $D = \sum p_{ij}(d_{ij} - 1)$, где p_{ij} – вероятность перехода между состояниями q_i и q_j в любом направлении, d_{ij} – расстояние по Хэммингу между кодами состояний q_i и q_j , а суммирование ведется по всем парам состояний автомата. Очевидно, что чем эта сумма меньше, тем лучше решение, и $D = 0$, если любой переход между состояниями соответствует переключению только одного элемента

памяти в логической схеме, реализующей заданный автомат. Что касается автоматов A_1, A_2 и A_3 , то покрытия с минимальным весом для них либо совпадают с покрытиями, получаемыми без учета весов, либо мало отличаются от них. Для автомата A_4 $D = 86/306$. Кратчайшее покрытие, полученное по табл. 7 без учета весов, составляют множества $\{1, 2, 8, 9\}, \{1, 7, 9\}, \{2, 6\}$ и $\{3, 4, 5\}$. Соответственно, $D = 97/306$. Ясно, что предлагаемый метод снижает интенсивность переключения элементов памяти и вместе с тем потребление энергии в проектируемом устройстве.

Таблица 7

Table 7

	1	2	3	4	5	6	7	8	9	Вес Weight
1,2,8,9	1	1						1	1	38
1,4	1			1						30
1,7,9	1						1		1	70
2,6		1				1				84
3,4,5			1	1	1					84
3,8,9			1					1	1	92
5,7					1		1			48

Функции возбуждения элементов памяти компонентных автоматов получаются по табл. 2 заменой символов состояний их кодами. При этом функции определяются на интервалах пространства внутренних переменных, определяемых переходами, что необходимо для исключения опасных состязаний в асинхронных автоматах. Система полученных слабо определенных булевых функций возбуждения элементов памяти представлена в табл. 8, где функции заданы в интервальной форме.

Таблица 8

Table 8

	$x_1 x_2$	$z_1^1 z_2^1 z_3^1$	$z_1^2 z_2^2 z_3^2$	$z_1^3 z_2^3$	$z_1^4 z_2^4 z_3^4 z_4^4$	$z_1^{1'} z_2^{1'} z_3^{1'}$	$z_1^{2'} z_2^{2'} z_3^{2'}$	$z_1^{3'} z_2^{3'}$	$z_1^{4'} z_2^{4'} z_3^{4'} z_4^{4'}$
1	- 0	0 0 0	0 0 0	0 0	0 0 0 0	0 0 0	0 0 0	0 0	0 0 0 0
2	- 1	- 0 0	- 0 0	- 0	- - 0 0	1 0 0	1 0 0	1 0	1 1 0 0
3	- 1	1 0 0	1 0 0	1 0	1 1 0 0	1 0 0	1 0 0	1 0	1 1 0 0
4	0 0	1 - 0	1 - 0	1 -	1 - - 0	1 1 0	1 1 0	1 1	1 0 1 0
5	1 0	1 - -	1 - -	1 -	1 1 - -	1 1 1	1 1 1	1 1	1 1 1 1
6	0 -	1 1 0	1 1 0	1 1	1 0 1 0	1 1 0	1 1 0	1 1	1 0 1 0
7	1 0	1 1 -	1 1 -	1 1	1 0 1 0	1 1 1	1 1 1	1 1	1 0 1 0
8	1 1	1 - 0	1 - 0	1 -	1 - - 0	1 0 0	1 0 0	1 0	1 1 0 0
9	0 0	1 1 1	1 1 1	1 1	1 - 1 -	1 1 1	1 1 1	1 1	1 0 1 0
10	0 1	- 1 1	1 1 1	- 1	1 - 1 -	0 1 1	1 1 1	0 1	1 0 1 0
11	1 1	- 1 1	1 1 1	- 1	- - 1 1	0 1 1	1 1 1	0 1	0 0 1 1
12	1 0	1 1 1	1 1 1	1 1	1 1 1 1	1 1 1	1 1 1	1 1	1 1 1 1
13	- 0	1 1 1	1 1 1	1 1	1 0 1 0	1 1 1	1 1 1	1 1	1 0 1 0
14	0 1	- 1 1	1 1 1	- 1	1 0 1 0	0 1 1	1 1 1	0 1	1 0 1 0
15	1 1	1 - -	1 - -	1 -	1 - - 0	1 0 0	1 0 0	1 0	1 1 0 0
16	- 1	0 1 1	1 1 1	0 1	0 0 1 1	0 1 1	1 1 1	0 1	0 0 1 1
17	1 0	0 - -	- - -	0 -	0 0 1 1	0 0 0	0 0 0	0 0	0 0 0 0
18	0 0	0 1 1	1 1 1	0 1	- 0 1 -	0 1 1	1 1 1	0 1	1 0 1 0
19	0 -	0 1 1	1 1 1	0 1	1 0 1 0	0 1 1	1 1 1	0 1	1 0 1 0
20	1 1	- - -	1 - -	- -	1 - - 0	1 0 0	1 0 0	1 0	1 1 0 0
21	1 0	0 - -	- - -	0 -	- 0 - 0	0 0 0	0 0 0	0 0	0 0 0 0

Минимизация функций и установление межкомпонентных связей. Задача совместной минимизации функций возбуждения элементов памяти решается отдельно для каждой компоненты сети. На первом этапе производится максимальное расширение интервалов, на каждом из которых хотя бы одна функция имеет значение 1. Данная процедура выполняется с помощью построения матриц различий для каждого из интервалов и нахождения кратчайших столбцовых покрытий этих матриц, как описано в работе [3]. В результате получается система дизъюнктив-

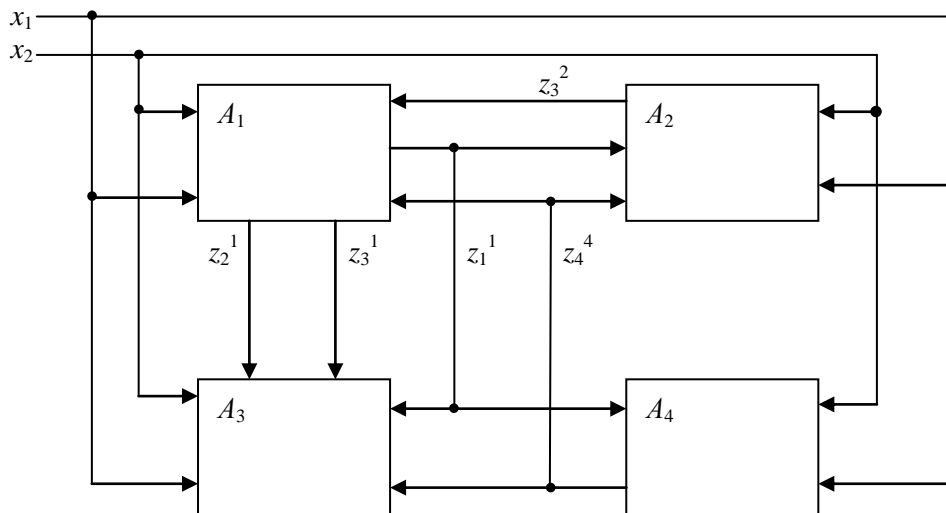
ных нормальных форм (ДНФ). При этом некоторые переменные оказываются несущественными аргументами и выбор среди них дает возможность минимизировать межкомпонентные связи, выбирая направление расширения интервала. В процессе минимизации ДНФ нельзя использовать обобщенное склеивание, так как в решение могут войти интервалы, при которых не устраняются опасные состязания. Поэтому минимизация заключается здесь в выявлении и удалении избыточных элементарных конъюнкций из полученных ДНФ. Эти конъюнкции выявляются с помощью способа, описанного в работах [18, 26], где для получения кратчайшего покрытия совокупности интервалов используется простая операция пересечения множеств.

В результате минимизации получаем следующие пары матриц, которые представляют системы ДНФ, реализующие соответственно автоматы A_1, A_2, A_3 и A_4 :

$$\begin{matrix} \begin{matrix} x_1 & x_2 & z_1^1 & z_2^1 & z_3^1 & z_3^2 & z_4^4 \\ \begin{bmatrix} - & 1 & - & 0 & - & - & - \\ 0 & 0 & 1 & - & 0 & - & - \\ 1 & 0 & 1 & - & - & - & - \\ 0 & - & - & 1 & - & 0 & - \\ - & 0 & 1 & - & 1 & - & - \\ - & 1 & - & - & 1 & - & - \\ - & 1 & - & - & - & - & 1 \\ 0 & - & 0 & - & - & - & - \\ 1 & 1 & - & - & - & - & 0 \end{bmatrix} & \begin{matrix} z_1^1 & z_2^1 & z_3^1 \\ \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \end{matrix} \end{matrix} ; & \begin{matrix} x_1 & x_2 & z_1^1 & z_1^2 & z_2^2 & z_3^2 & z_4^4 \\ \begin{bmatrix} - & 1 & - & - & 0 & - & - \\ 0 & 0 & - & 1 & - & 0 & - \\ 1 & 0 & 1 & - & - & - & - \\ 0 & - & - & - & - & 1 & - \\ - & 1 & - & - & - & - & 1 \\ 1 & 1 & - & - & - & - & 0 \end{bmatrix} & \begin{matrix} z_1^2 & z_2^2 & z_3^1 \\ \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \end{matrix} \end{matrix} ; \end{matrix}$$

$$\begin{matrix} \begin{matrix} x_1 & x_2 & z_1^1 & z_2^1 & z_3^1 & z_3^2 & z_4^4 \\ \begin{bmatrix} - & 1 & - & 0 & - & - & - \\ - & 0 & 1 & - & - & - & - \\ 0 & - & - & - & 0 & 1 & - \\ 1 & 1 & - & - & 0 & - & - \\ 0 & 1 & - & - & 1 & - & - \\ - & 1 & - & - & - & - & 1 \\ 0 & - & 0 & - & - & 1 & - \\ 1 & 1 & - & - & - & - & 0 \end{bmatrix} & \begin{matrix} z_1^3 & z_2^3 \\ \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \end{matrix} \end{matrix} ; & \begin{matrix} x_1 & x_2 & z_1^1 & z_1^4 & z_2^4 & z_3^4 & z_4^4 \\ \begin{bmatrix} - & 1 & - & - & - & 0 & 0 \\ 0 & 0 & - & 1 & - & - & - \\ 1 & 0 & - & - & 1 & - & - \\ 0 & - & - & - & - & 1 & - \\ - & 0 & 1 & - & 0 & - & - \\ 1 & 1 & - & - & - & - & 0 \\ 1 & 1 & - & - & - & - & 1 \\ - & 1 & - & 0 & - & - & 1 \end{bmatrix} & \begin{matrix} z_1^4 & z_2^4 & z_3^4 & z_4^4 \\ \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix} \end{matrix} . \end{matrix}$$

Существенные аргументы функций показывают связи между компонентами сети. Полученная сеть изображена на рисунке.



Сеть последовательных автоматов
Net of sequential automata

Заключение. Описанный метод декомпозиции параллельного автомата в сеть последовательных автоматов может быть применен при синтезе распределенных систем управления. В таких системах, управляющих некоторой совокупностью отдаленных друг от друга объектов, все блоки связаны информационно в единую сеть, а каждый блок находится непосредственно на объекте управления. Здесь же возникает задача распределения выходных сигналов по компонентам сети. В данной статье выходные сигналы не рассматривались, все внимание было сосредоточено на функции возбуждения элементов памяти. Предлагаемый метод энергосберегающего кодирования частичных состояний параллельного автомата рассчитан на использование его в автоматизированной системе логического проектирования, когда заданный параллельный автомат допускает моделирование последовательным асинхронным автоматом. Сравнение результатов применения изложенного метода с результатами кодирования частичных состояний без учета интенсивности переключений элементов памяти показывает, что применение данного метода дает лучший результат. Использование декомпозиции параллельного автомата позволяет снизить размерность трудоемкой задачи кодирования состояний.

Список использованных источников

1. Закревский, А. Д. Параллельные алгоритмы логического управления / А. Д. Закревский. – М. : УРСС, 2003. – 304 с.
2. Питерсон, Дж. Теория сетей Петри и моделирование систем : пер. с англ. / Дж. Питерсон. – М. : Мир, 1984. – 264 с.
3. Закревский, А. Д. Логические основы проектирования дискретных устройств / А. Д. Закревский, Ю. В. Поттосин, Л. Д. Черемисинова. – М. : Физматлит, 2007. – 592 с.
4. Закревский, А. Д. Блочное кодирование частичных состояний у автоматов, реализующих параллельные алгоритмы логического управления / А. Д. Закревский // Изв. АН СССР. Техническая кибернетика. – 1983. – № 5. – С. 3–11.
5. Черемисинова, Л. Д. Реализация асинхронными автоматами параллельных алгоритмов логического управления / Л. Д. Черемисинова // Автоматика и вычислительная техника. – 1985. – № 2. – С. 65–69.
6. Поттосин, Ю. В. Декомпозиционный метод кодирования состояний параллельного автомата / Ю. В. Поттосин // Автоматика и вычислительная техника. – 1987. – № 1. – С. 84–91.
7. Мурога, С. Системное проектирование сверхбольших интегральных схем : в 2 кн. / С. Мурога. – М. : Мир, 1985. – Кн. 1. – 288 с.
8. Pedram, M. Power minimization in IC design: Principles and applications / M. Pedram // ACM Trans. Design Automat. Electron. Syst. – 1996. – Vol. 1. – P. 3–56.
9. Kashirova, L. State assignment of finite state machine for decrease of power dissipation / L. Kashirova, A. Keevallik, M. Meshkov // Second Intern. Conf. Computer-Aided Design of Discrete Devices, CAD DD'97, Minsk, Republic of Belarus, 12–14 Nov. 1997. – Minsk : Institute of Engineering Cybernetics of the National Academy of Sciences of Belarus, 1997. – Vol. 1. – P. 60–67.
10. Sudnitson, A. Partition search for FSM low power synthesis / A. Sudnitson // Fourth Intern. Conf. Computer-Aided Design of Discrete Devices, CAD DD'2001, Minsk, 14–16 Nov. 2001. – Minsk : Institute of Engineering Cybernetics of the National Academy of Sciences of Belarus, 2001. – Vol. 1. – P. 44–49.
11. Закревский, А. Д. Алгоритмы энергосберегающего кодирования состояний автомата / А. Д. Закревский // Информатика. – 2011. – № 1(29). – С. 68–78.
12. Поттосин, Ю. В. Кодирование состояний дискретного автомата, ориентированное на уменьшение энергопотребления реализующей схемы / Ю. В. Поттосин // Прикладная дискретная математика. – 2011. – № 4(14). – С. 62–71.
13. Поттосин, Ю. В. Энергосберегающее противогоночное кодирование состояний асинхронного автомата / Ю. В. Поттосин // Информатика. – 2015. – № 2(46). – С. 94–101.
14. Pottosin, Yu. Race-free state assignment for low power asynchronous automaton / Yu. Pottosin // Further Improvements in the Boolean Domain / ed. B. Steinbach. – Cambridge Scholars Publishing, 2018. – P. 253–267.
15. Pottosin, Yu. V. Low power assignment of partial states of a parallel automaton / Yu. V. Pottosin // Прикладная дискретная математика. – 2022. – № 56. – С. 113–122.
16. Pottosin, Yu. Optimal state assignment of synchronous parallel automata / Yu. Pottosin // Design of Embedded Control Systems. – N. Y. : Springer, 2005. – P. 111–124.

17. Поттосин, Ю. В. Совместное энергосберегающее кодирование состояний последовательных автоматов сети, реализующей параллельный автомат / Ю. В. Поттосин // Информатика. – 2023. – Т. 20, № 1. – С. 75–90.
18. Поттосин, Ю. В. Комбинаторные задачи в логическом проектировании дискретных устройств / Ю. В. Поттосин. – Минск : Беларуская навука, 2021. – 175 с.
19. Piguët, C. Low-power and low-voltage CMOS digital design / C. Piguët // *Microelectronic Engineering*. – 1997. – No. 39. – P. 179–208.
20. Якубайтис, Э. А. Асинхронные логические автоматы / Э. А. Якубайтис. – Рига : Зинатне, 1966. – 380 с.
21. Hartmanis, J. Algebraic structure theory of sequential machines / J. Hartmanis, R.E. Stearns. – Englewood Cliffs, N. J. : Prentice Hall Inc., 1966. – 208 p.
22. Кээваллик, А. Э. Теорема декомпозиции конечных автоматов / А. Э. Кээваллик // *Автоматика и вычислительная техника*. – 1974. – № 1. – С. 17–24.
23. Ковалев, А. В. О нахождении отношения параллельности на множестве мест одного подкласса сетей Петри / А. В. Ковалев // *Вес. Акад. навук Беларускай ССР. Сер. фіз.-мат. навук*. – 1989. – № 2. – С. 106–110.
24. Macii, E. High-level power modeling, estimation and optimization / E. Macii, M. Pedram, F. Somenzi // *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*. – 1998. – Vol. 17, no. 11. – P. 1061–1079.
25. Закревский, А. Д. Оптимизация покрытий множеств / А. Д. Закревский // *Логический язык для представления алгоритмов синтеза релейных устройств*. – М. : Наука, 1966. – С. 136–148.
26. Поттосин, Ю. В. Метод минимизации системы полностью определенных булевых функций / Ю. В. Поттосин, Н. Р. Торопов, Е. А. Шестаков // *Информатика*. – 2008. – № 2(18). – С. 102–110.

References

1. Zakrevskij A. D. Parallel'nye algoritmy logicheskogo upravleniya. *Parallel Algorithms for Logical Control*. Moscow, URSS, 2003, 304 p. (In Russ.).
2. Peterson J. *Petri Net Theory and the Modeling of Systems*; first edition. Prentice Hall, 1981, 290 p.
3. Zakrevskij A. D., Pottosin Yu. V., Cheremisinova L. D. Logicheskie osnovy proektirovaniya diskretnyh ustrojstv. *Logical Fundamentals of Discrete Devices Design*. Moscow, Fizmatlit, 2007, 592 p. (In Russ.).
4. Zakrevskij A. D. *Block partial state assignment of automata that implement parallel algorithms for logical control*. *Izvestija Akademii nauk SSSR. Tehnicheskaja kibernetika [Proceedings of the Academy of Sciences of the USSR. Engineering Cybernetics]*, 1983, no. 5, pp. 3–11 (In Russ.).
5. Cheremisinova L. D. *Implementation of parallel algorithms for logical control by asynchronous automata*. *Avtomatika i vychislitel'naya tehnika [Automation and Computer Engineering]*, 1985, no. 2, pp. 65–69 (In Russ.).
6. Pottosin Yu. V. *A decomposition method for state assignment of a parallel automaton*. *Avtomatika i vychislitel'naya tehnika [Automation and Computer Engineering]*, 1987, no. 1, pp. 84–91 (In Russ.).
7. Muroga C. *Sistemnoye proektirovanie sverhbol'shih integral'nyh shem : v dvuh knigah. System Design of Super Large Integrated Circuits : In 2 Issues*. Moscow, Mir, 1985, iss. 1, 288 p. (In Russ.).
8. Pedram M. Power minimization in IC design: Principles and applications. *ACM Transactions on Design Automation of Electronic Systems*, 1996, vol. 1, pp. 3–56.
9. Kashirova L., Keevallik A., Meshkov A. M. State assignment of finite state machine for decrease of power dissipation. *Second International Conference Computer-Aided Design of Discrete Devices, CAD DD'97, Minsk, Republic of Belarus, 12–14 November 1997*. Minsk, Institute of Engineering Cybernetics of the National Academy of Sciences of Belarus, 1997, vol. 1, pp. 60–67.
10. Sudnitson A. Partition search for FSM low power synthesis. *Fourth International Conference Computer-Aided Design of Discrete Devices, CAD DD'2001, Minsk, 14–16 November 2001*. Minsk, Institute of Engineering Cybernetics of the National Academy of Sciences of Belarus, 2001, vol. 1, pp. 44–49.
11. Zakrevskij A. D. *Algorithms for low power state assignment of an automaton*. *Informatika [Informatics]*, 2011, no. 1(29), pp. 68–78 (In Russ.).
12. Pottosin Yu. V. *State assignment of a discrete automaton to decrease power consumption of the implementing circuit*. *Prikladnaya diskretnaya matematika [Discrete Applied Mathematics]*, 2011, no. 4(14), pp. 62–71 (In Russ.).

13. Pottosin Yu. V. *Low power race-free state assignment of an asynchronous automaton*. Informatika [Informatics], 2015, no. 2(46), pp. 94–101 (In Russ.).
14. Pottosin Yu. Race-free state assignment for low power asynchronous automaton. *Further Improvements in the Boolean Domain*. Ed. B. Steinbach. Cambridge Scholars Publishing, 2018, pp. 253–267.
15. Pottosin Yu. V. Low power assignment of partial states of a parallel automaton. *Prikladnaya diskretnaya matematika [Discrete Applied Mathematics]*, 2022, no. 56, pp. 113–122.
16. Pottosin Yu. Optimal state assignment of synchronous parallel automata. *Design of Embedded Control Systems*. New York, Springer, 2005, pp. 111–124.
17. Pottosin Yu. V. Joint low power state assignment of sequential automata that form a net implementing a parallel automaton. Informatika [Informatics], 2023, vol. 20, no. 1, pp. 75–90 (In Russ.).
18. Pottosin Yu. V. *Kombinatornye zadachi v logicheskom proektirovanii diskretnyh ustrojstv. Combinatorial Problems in Logical Design of Discrete Devices*. Minsk, Belaruskaya navuka, 2021, 175 p. (In Russ.).
19. Piguet C. Low-power and low-voltage CMOS digital design. *Microelectronic Engineering*, 1997, no. 39, pp. 179–208.
20. Yakubajtis E. A. *Asinhronnye logicheskie avtomaty. Asynchronous Logical Automata*. Riga, Zinatne, 1966, 380 p. (In Russ.).
21. Hartmanis J., Stearns R. E. *Algebraic Structure Theory of Sequential Machines*. Englewood Cliffs, New Jersey, Prentice Hall Inc., 1966, 208 p.
22. Keevallik A. *Decomposition theorem of finite automata*. Avtomatika i vychislitel'naya tehnika [Automation and Computer Engineering], 1974, no. 1, pp. 17–24 (In Russ.).
23. Kovalyov A. V. *About finding parallelism relation on the set of positions of a subclass of Petri nets*. Vesci Akademii nauk Belaruskaj SSR. Seria fizika-matematichnyh navuk [Proceedings of the Academy of Sciences of Byelorussian SSR. Series of Physico-Mathematical Sciences], 1989, no. 2, pp. 106–110 (In Russ.).
24. Macii E., Pedram M., Somenzi F. High-level power modeling, estimation and optimization. *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, 1998, vol. 17, no. 11, pp. 1061–1079.
25. Zakrevskij A. D. *Optimization of set covers*. Logicheskij yazyk dlya predstavlenia algoritmov sinteza relejnyh ustrojstv [Logical Language for Presenting Algorithms for Synthesis of Relay Devices], Moscow, Nauka, 1966, pp. 136–148 (In Russ.).
26. Pottosin Yu. V., Toropov N. R., Shestakov E. A. *A method for minimizing a system of completely specified Boolean functions*. Informatika [Informatics], 2008, no. 2(18), pp. 102–110 (In Russ.).

Информация об авторе

Поттосин Юрий Васильевич, кандидат физико-математических наук, ведущий научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси.
E-mail: pott@newman.bas-net.by

Information about the author

Yuri V. Pottosin, Ph. D. (Phys.-Math.), Leading Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.
E-mail: pott@newman.bas-net.by