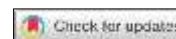


ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

LOGICAL DESIGN



УДК 519.711
<https://doi.org/10.37661/1816-0301-2023-20-1-75-90>

Оригинальная статья
Original Paper

Совместное энергосберегающее кодирование состояний последовательных автоматов сети, реализующей параллельный автомат

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. Рассматривается задача энергосберегающего кодирования частичных состояний параллельного автомата. Целью работы является исследование возможности использования приема декомпозиции при кодировании частичных состояний для снижения размерности задачи.

Методы. Заданный параллельный автомат разлагается в сеть последовательных автоматов, состояния которых кодируются затем троичными векторами. Метод кодирования использует поиск максимального разреза во взвешенном графе, представляющем пары состояний, связанных переходами. Весами ребер графа являются величины, связанные с вероятностями переходов.

Результаты. Описан способ построения сети из последовательных автоматов, реализующей заданный параллельный автомат. Вероятности переходов между состояниями вычисляются путем решения системы линейных уравнений согласно методу Чэпмена – Колмогорова. Значения внутренних переменных, кодирующих состояния каждого компонентного последовательного автомата, находятся по двухблочным разбиениям множества его состояний, которые определяются разрезами соответствующего графа переходов.

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, граф переходов, вероятность перехода между состояниями

Для цитирования. Поттосин, Ю. В. Совместное энергосберегающее кодирование состояний последовательных автоматов сети, реализующей параллельный автомат / Ю. В. Поттосин // Информатика. – 2023. – Т. 20, № 1. – С. 75–90. <https://doi.org/10.37661/1816-0301-2023-20-1-75-90>

Конфликт интересов. Автор заявляет об отсутствии конфликта интересов.

Поступила в редакцию | Received 18.12.2022
Подписана в печать | Accepted 20.01.2023
Опубликована | Published 29.03.2023

Joint low power state assignment of sequential automata of a net implementing a parallel automaton

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 problem of low power state assignment of partial states of a parallel automaton is considered. The objective of the paper is to investigate the possibilities of using the decomposition in state assignment of partial states in order to decrease the task dimension.

Methods. Parallel automaton is decomposed into a net of sequential automata whose states are assigned then with ternary vectors. The method for assignment uses searching for a maximal cut in a weighted graph that represents pairs of states connected by transitions. The edge weights of the graph are the values related to the probabilities of transitions.

Results. A method to construct a net of sequential automata that realizes the given parallel automaton is described. The probabilities of transitions between sets are calculated by means of solving a system of linear equations according to the Chapman – Kolmogorov method. The values of inner variables assigned to the states of every component sequential automaton are obtained from two-block partitions of its set of states that are determined by the cuts of corresponding transition graph.

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, transition graph, probability of transition between states

For citation. Pottosin Yu. V. *Joint low power state assignment of sequential automata of a net implementing a parallel automaton*. Informatika [Informatics], 2023, vol. 20, no. 1, pp. 75–90 (In Russ.).
<https://doi.org/10.37661/1816-0301-2023-20-1-75-90>

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

Введение. Одним из важных критериев оптимизации при проектировании дискретных устройств является величина потребляемой энергии. 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 [1, 2], 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 [3–8]. Метод энергосберегающего кодирования частичных состояний параллельного автомата описан в статье [9], где используется подход [10], сводящий данную задачу к поиску покрытия графа его полными двудольными подграфами.

Параллельный автомат является функциональной моделью дискретного устройства управления, с достаточной степенью удобства представляющей параллелизм управляемых взаимодействующих процессов [11]. Эта модель близка к широко известной сети Петри [12]. В отличие от последовательного автомата, который в отдельный 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. Задачу кодирования состояний параллельного автомата можно решать путем нахождения всех его глобальных состояний и последующего их кодирования каким-либо из методов, разработанных применительно к последовательным автоматам. Однако при сравнительно небольшом количестве частичных состояний параллельного автомата число его глобальных состояний, как показано в работе [13], может быть настолько велико, что задача их кодирования не может быть решена за приемлемое время. Кроме того, при асинхронной реализации параллельный автомат не сводится к последовательному [13] и указанный подход вообще не может быть использован. Поэтому были разработаны методы специально для кодирования частичных состояний параллельного автомата [14–16], использующие 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 автоматом, и поэтому для кодирования его состояний можно использовать какой-либо из известных методов, предназначенных для последовательных автоматов. В статье показано, как использовать для совместного кодирования состояний компонентных автоматов сети, реализующей параллельный автомат, итеративный метод, описанный в работе [17]. Все такие автоматы могут быть синтезированы на основе программируемых логических интегральных схем, как показано, например, в работе [18]. Использование приема декомпозиции параллельного автомата, т. е. разложения его в сеть последовательных автоматов, позволяет снизить размерность задачи. Разложение проектируемого устройства на блоки дает возможность снизить энергопотребление путем блокировки синхронизирующих сигналов, подаваемых на некоторые блоки [19].

Описание модели. Параллельный автомат состоит из следующих объектов: $Q = \{q_1, q_2, \dots, q_r\}$ – 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 $T = \{\tau_1, \tau_2, \dots, \tau_r\}$, имеющих вид [13]

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

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 строки (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 единица 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 строке (1). 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. В данной работе не будут рассматриваться выходные сигналы и конъюнкция K_i' будет опускаться. Так же, как для последовательного автомата, возможны синхронная и асинхронная реализации параллельного автома-

та. Далее будет рассматриваться синхронная реализация. При такой реализации время делится на фиксированные промежутки – такты, в течение одного такта автомат совершает переход между состояниями и выдает определенные выходные сигналы.

В данную модель внесены следующие ограничения.

1. Введено 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 $S_i = S_j$.

Существует ряд других ограничений, связанных с корректностью задания параллельного автомата и указанных в работе [11], которых не будем касаться, так как там решается другой круг задач. Не будем также обращать внимание на выходные сигналы, ограничимся рассмотрением переключений элементов памяти.

Примером параллельного автомата может служить следующая последовательность строк:

1: $-\bar{x}_1 x_2 \rightarrow 10;$	τ_1
10: $-\bar{x}_2 \rightarrow 2.3.4;$	τ_2
2: $\rightarrow 5;$	τ_3
3.5: $-x_2 \rightarrow 8;$	τ_4
4: $-\bar{x}_1 \rightarrow 7;$	τ_5
4: $-x_1 \rightarrow 9;$	τ_6
7: $-\bar{x}_2 \rightarrow 9;$	τ_7
8.9: $\rightarrow 6;$	τ_8
6: $-x_1 \rightarrow 1.$	τ_9

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 $x_1 = 0$ и $x_2 = 1$, то автомат выходит из состояния $\{1\}$ и в следующем такте входит в состояние $\{10\}$, которое также является глобальным. При любой другой комбинации значений x_1 и x_2 автомат остается в состоянии $\{1\}$. Из состояния $\{10\}$ при $x_2 = 0$ автомат переходит в частичные состояния 2, 3 и 4, которые составляют глобальное состояние $\{2, 3, 4\}$ (переход τ_2). В следующем такте автомат меняет частичное состояние 2 на частичное состояние 5 независимо от значений входных переменных. Поскольку рассматривается синхронная реализация, где переходы могут совершаться одновременно, переход τ_3 совершается одновременно с переходом τ_5 при $x_1 = 0$ или одновременно с переходом τ_6 при $x_1 = 1$. Глобальными состояниями при этом будут соответственно $\{3, 5, 7\}$ или $\{3, 5, 9\}$. Проследив таким образом функционирование данного параллельного автомата, получим еще глобальные состояния $\{7, 8\}$, $\{8, 9\}$ и $\{6\}$.

Построение сети последовательных автоматов, реализующей заданный параллельный автомат. 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 тех же входных булевых переменных, которые 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 автоматов A_j является автоматом без выходов, т. е. для него заданы только переходы, которые представим в форме, аналогичной (1):

$$q^j : -\alpha_j \rightarrow q^j, \quad (2)$$

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 единица 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 $S_i \subseteq \Phi(q^1, q^2, \dots, q^n)$ имеем $S_i' \subseteq \Phi(q^{1'}, q^{2'}, \dots, q^{n'})$, если $\alpha_j = 1$, и $S_i' \subseteq \Phi(q^1, q^2, \dots, q^n)$, если $\alpha_j = 0$, согласно (2). В случае синхронной реализации параллельного автомата B можно построить эквивалентный ему последовательный автомат A с состояниями, соответствующими глобальным состояниям автомата B . Описанная сеть в этом случае реализует автомат A в смысле работ [20, 21].

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 достаточное 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 , любой 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 $f_j(q^j) \in S_i$. Такое состояние в Q_j единственное, 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 состояний, принадлежащих ему. Далее следует получить покрытие 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 [22] описан алгоритм установления параллельности частичных состояний, который для параллельных автоматов рассматриваемого класса решает данную задачу за полиномиальное время. В случае энергосберегающего кодирования, как отмечено далее, 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	1	0	0	0	0	0	0	2
0	1	0	1	1	0	1	0	1	0	3
0	1	1	0	0	0	0	0	0	0	4
0	0	1	0	0	0	1	0	1	0	5
0	0	0	0	0	0	0	0	0	0	6
0	0	1	0	1	0	0	1	0	0	7
0	0	0	0	0	0	1	0	1	0	8
0	0	1	0	1	0	0	1	0	0	9
0	0	0	0	0	0	0	0	0	0	10

Такую матрицу можно рассматривать как матрицу смежности графа, представляющего отношение параллельности. Упомянутые множества непараллельных частичных состояний соот-

ветствуют независимым множествам этого графа. Способы нахождения максимальных независимых множеств в графе описаны в работе [17]. В данном графе максимальными независимыми множествами являются $\{1, 2, 5, 6, 8, 10\}$, $\{1, 3, 6, 8, 10\}$, $\{1, 4, 5, 6, 8, 10\}$, $\{1, 2, 6, 7, 9, 10\}$ и $\{1, 4, 6, 7, 9, 10\}$. Табл. 1 является таблицей покрытия. В ней строкам соответствуют максимальные множества взаимно непараллельных частичных состояний, а столбцам – пары частичных состояний, связанных переходами.

Таблица 1

Table 1

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

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

Таблица 2

Table 2

x_1x_2	q^1	q^2	q^3	$q^{1'}$	$q^{2'}$	$q^{3'}$
0 1	1	1	1	10	10	10
1 –	1	1	1	1	1	1
– 0	1	1	1	1	1	1
– 0	10	10	10	2	3	4
– 1	10	10	10	10	10	10
– –	2	–	–	5	–	–
– 1	5	3	–	8	8	–
– 0	5	3	–	5	3	–
0 –	–	–	4	–	–	7
1 –	–	–	4	–	–	9
– 0	–	–	7	–	–	9
– 1	–	–	7	–	–	7
– –	8	8	9	6	6	6
1 –	6	6	6	1	1	1
0 –	6	6	6	6	6	6

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

Метод кодирования состояний последовательного автомата. Для кодирования состояний автомата используем итеративный способ, описанный в работе [17]. В его основе лежит поиск

максимального разреза во взвешенном графе. Пусть требуется закодировать состояния некоторого автомата A , т. е. каждому его состоянию q поставить в соответствие булев вектор (z_1, z_2, \dots, z_k) , называемый кодом состояния. Различным состояниям должны быть приписаны различные коды. Текущая ситуация в процессе выполнения данного способа характеризуется частичными кодами состояний (z_1, z_2, \dots, z_j) , $j < k$, и взвешенным графом $G = (V, E)$, вершины которого соответствуют состояниям автомата. Две вершины этого графа связаны ребром, если и только если соответствующие состояния имеют один и тот же частичный код. В начальной ситуации частичные коды пустые и граф является полным. Каждое ребро $v_s v_t \in E$ имеет вес w_{st} , пропорциональный величине $1 - p_{st}$, где p_{st} – вероятность перехода между состояниями q_s и q_t , соответствующими вершинам v_s и v_t , независимо от направления перехода. Ясно, что вероятность p_{st} равна сумме вероятностей переходов от q_s к q_t и от q_t к q_s . Очевидно, что для снижения переключательной активности элементов памяти расстояние по Хэммингу между кодами состояний q_s и q_t в пространстве булевых переменных z_1, z_2, \dots, z_k должно быть как можно малым, если вероятность p_{st} высока.

Процесс кодирования состояний заданного автомата представляет собой последовательность шагов. На i -м шаге находится разбиение множества вершин V графа G на подмножества V_1 и V_2 , вводится переменная z_i , которая получает значение 0 (или 1) для состояний, соответствующих вершинам из V_1 , и значение 1 (или 0) для состояний, соответствующих вершинам из множества V_2 . Затем удаляются ребра, соединяющие вершины из V_1 с вершинами из V_2 , и выполняется следующий $(i + 1)$ -й шаг. Процесс заканчивается, когда граф G становится пустым.

Задача разбиения множества V на подмножества V_1 и V_2 сводится к нахождению максимального разреза в графе G , т. е. такого разбиения, что сумма весов ребер, соединяющих вершины из V_1 с вершинами из V_2 , была бы максимальной. На последнем шаге остаются только те ребра, которые соответствуют парам состояний, связанных переходами со сравнительно большой вероятностью. Расстояния между кодами этих состояний равны единице. Для нахождения разреза можно применять описанный в статье [23] метод, представляющий собой последовательность шагов, на каждом из которых выбирается вершина v из множества V_2 и переносится в множество V_1 . Начальными значениями этих множеств являются $V_1 = \emptyset$ и $V_2 = V$, а вершина v выбирается следующим образом. Пусть d – сумма весов ребер, инцидентных вершине v , и c – сумма весов ребер, связывающих вершину v с вершинами из множества V_1 . Перенос вершины v из V_2 в V_1 увеличивает сумму весов ребер, соединяющих вершины из множества V_1 с вершинами из множества V_2 , на величину $d - 2c$, если она положительна. На первом шаге она равна d . На каждом шаге выбирается такая вершина v , для которой величина $d - 2c$ максимальна. Процесс заканчивается, когда эта величина окажется отрицательной или нулем для всех вершин из множества V_1 .

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.

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 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 сумма вероятностей переходов в некоторое состо-

яние равно 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 ранее, абсолютная вероятность p_{ij} определяется как $p_{ij} = p(q_i) p'_{ij}$.

Совместное кодирование состояний компонентных автоматов. 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. Для вычисления вероятностей переходов между состояниями компонентных автоматов предлагается использовать моделирование синхронного параллельного автомата последовательным автоматом, состояниями которого являются глобальные состояния заданного параллельного автомата [11]. Вероятности переходов между глобальными состояниями определяются, 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_r , для которых $q_i \in P_s$ и $q_j \in P_r$ или $q_i \in P_r$ и $q_j \in P_s$.

Переходы между глобальными состояниями параллельного автомата из рассматриваемого примера представлены 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. По этой таблице легко определяются условные вероятности переходов, представленные в табл. 4. Считается, что входные двоичные сигналы x_1 и x_2 независимы и равновероятны.

Таблица 3

Table 3

	1	10	2.3.4	3.5.7	3.5.9	7.8	8.9	6
1	$x_1 \vee \bar{x}_2$	$\bar{x}_1 x_2$						
10		x_2	\bar{x}_2					
2.3.4				\bar{x}_1	x_1			
3.5.7					\bar{x}_2	x_2		
3.5.9					x_2		\bar{x}_2	
7.8						x_2	\bar{x}_2	
8.9								1
6	x_1							\bar{x}_1

Таблица 4

Table 4

	1	10	2.3.4	3.5.7	3.5.9	7.8	8.9	6
1	3/4	1/4						
10		1/2	1/2					
2.3.4				1/2	1/2			
3.5.7					1/2	1/2		
3.5.9					1/2		1/2	
7.8						1/2	1/2	
8.9								1
6	1/2							1/2

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

Матрица смежности взвешенного графа G , общего для автоматов A_1 , A_2 и A_3 , где в связи с ее симметричностью опущены значения ниже главной диагонали ввиду симметричности матрицы и сама диагональ, имеет следующий вид:

$$\begin{array}{cccccccccc}
 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
 \left[\begin{array}{cccccccccc}
 25 & 25 & 25 & 25 & 23 & 25 & 25 & 25 & 23 & \\
 & 0 & 0 & 23 & 25 & 25 & 25 & 25 & 23 & \\
 & & 0 & 0 & 25 & 0 & 23 & 0 & 23 & \\
 & & & 25 & 25 & 24 & 25 & 24 & 23 & \\
 & & & & 25 & 0 & 23 & 0 & 25 & \\
 & & & & & 25 & 23 & 23 & 25 & \\
 & & & & & & 0 & 23 & 25 & \\
 & & & & & & & 0 & 25 & \\
 & & & & & & & & & 25
 \end{array} \right] \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array}
 \end{array}$$

Нули в матрице относятся к тем парам состояний, которые принадлежат различным автоматам, и поэтому соответствующие вершины не должны быть связаны ребрами. В качестве весов ребер w_{ij} , как было сказано выше, должны быть величины, пропорциональные разностям $1 - p_{ij}$. Здесь в качестве значений w_{ij} взяты числители этих дробей при общем знаменателе 25.

При формировании разреза графа G начальными значениями множеств V_1 и V_2 являются \emptyset и $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Суммы весов ребер, инцидентных вершинам, принимают следующие значения:

$$\begin{aligned}
 d(1) &= 221, d(2) = 171, d(3) = 96, d(4) = 171, d(5) = 146, d(6) = 219, d(7) = 147, d(8) = 169, \\
 d(9) &= 145, d(10) = 217.
 \end{aligned}$$

Максимум имеет вершина 1, поэтому она переносится в множество V_1 : $V_1 = \{1\}$, $V_2 = \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Теперь для вершин из множества V_2 вычисляются значения $h(v) = d(v) - 2c(v)$, где $c(v)$ – сумма весов ребер, связывающих вершину v с вершинами из множества V_1 :

$$\begin{aligned}
 h(2) &= 171 - 2 \cdot 25 = 121, h(3) = 96 - 2 \cdot 25 = 46, h(4) = 171 - 2 \cdot 25 = 121, \\
 h(5) &= 146 - 2 \cdot 25 = 96, h(6) = 219 - 2 \cdot 23 = 173, h(7) = 147 - 2 \cdot 25 = 97, \\
 h(8) &= 169 - 2 \cdot 25 = 119, h(9) = 145 - 2 \cdot 25 = 95, h(10) = 217 - 2 \cdot 23 = 171.
 \end{aligned}$$

По максимуму значения h из V_2 в V_1 переносится вершина 6 и получается разбиение $\{1, 6\}$, $\{2, 3, 4, 5, 7, 8, 9, 10\}$. Процесс переноса вершин из V_2 в V_1 продолжается до тех пор, пока значение h для всех вершин в V_2 не станет нулевым или отрицательным. В результате получим разбиение $\{1, 2, 4, 6, 10\}$, $\{3, 5, 7, 8, 9\}$, или разрез максимального веса. (Возможно, этот максимум не является наибольшим в силу эвристического характера применяемого алгоритма.) Соответственно, вводим кодирующую переменную z_1 со следующими значениями:

$$\begin{array}{cccccccccc}
 \text{состояния компонентных автоматов:} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
 \text{значения переменной } z_1: & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0
 \end{array}$$

Из графа G удаляются ребра, принадлежащие найденному разрезу, и в полученном графе так же находится разрез. Теперь матрица смежности примет вид

$$\begin{array}{cccccccccc} 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & \\ \left[\begin{array}{cccccccccc} 25 & 0 & 25 & 0 & 23 & 0 & 0 & 0 & 23 & \\ & 0 & 0 & 0 & 25 & 0 & 0 & 0 & 23 & \\ & & 0 & 0 & 0 & 0 & 23 & 0 & 0 & \\ & & & 0 & 25 & 0 & 0 & 0 & 23 & \\ & & & & 0 & 0 & 23 & 0 & 0 & \\ & & & & & 0 & 0 & 0 & 25 & \\ & & & & & & 0 & 23 & 0 & \\ & & & & & & & 0 & 0 & \\ & & & & & & & & 0 & \end{array} \right] \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \end{array}$$

Разрез в этом графе определяет разбиение $\{1, 5, 6, 7, 8\}$, $\{2, 3, 4, 9, 10\}$, а переменная z_2 имеет следующие значения:

$$\begin{array}{l} \text{состояния компонентных автоматов: } 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ \text{значения переменной } z_2: \quad 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \end{array}$$

Для определения значений переменной z_3 необходимо удалить из графа G не только ребра найденного разреза, но и изолированные вершины. Для состояний, соответствующих изолированным вершинам, переменная z_3 приобретает неопределенное значение. Матрица смежности графа G теперь имеет вид

$$\begin{array}{cccccc} 2 & 4 & 5 & 6 & 8 & 10 \\ \left[\begin{array}{cccccc} 0 & 0 & 0 & 23 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 23 \\ & & 0 & 0 & 0 & 23 \\ & & & 0 & 23 & 0 \\ & & & & 0 & 0 \\ & & & & & 0 \end{array} \right] \begin{array}{l} 1 \\ 2 \\ 4 \\ 5 \\ 6 \\ 8 \end{array} \end{array}$$

Все ребра из этого графа составляют единственный разрез, разделяющий множество вершин на $\{1, 5, 10\}$ и $\{2, 4, 6, 8\}$, и по этому разрезу находим значения переменной z_3 :

$$\begin{array}{l} \text{состояния компонентных автоматов: } 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \\ \text{значения переменной } z_2: \quad 0 \ 1 \ - \ 1 \ 0 \ 1 \ - \ 1 \ - \ 0 \end{array}$$

Таким образом, получаем матрицы C_1 , C_2 и C_3 кодирования состояний автоматов A_1 , A_2 и A_3 соответственно:

$$C_1 = \begin{array}{ccc} z_1^1 & z_2^1 & z_3^1 \\ \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{array} \right] \begin{array}{l} 1 \\ 2 \\ 5 \\ 6 \\ 8 \\ 10 \end{array} \end{array}, \quad C_2 = \begin{array}{ccc} z_1^2 & z_2^2 & z_3^2 \\ \left[\begin{array}{ccc} 0 & 0 & 0 \\ 1 & 1 & - \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{array} \right] \begin{array}{l} 1 \\ 3 \\ 6 \\ 8 \\ 10 \end{array} \end{array}, \quad C_3 = \begin{array}{ccc} z_1^3 & z_2^3 & z_3^3 \\ \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & - \\ 1 & 1 & - \\ 0 & 1 & 0 \end{array} \right] \begin{array}{l} 1 \\ 4 \\ 6 \\ 7 \\ 9 \\ 10 \end{array} \end{array}.$$

Качество кодирования состояний автомата можно оценить введенной в работе [5] величиной $D = \sum p_{ij}(d_{ij} - 1)$, 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 меньше эта величина, тем лучше решение

и $D = 0$, если любой переход между состояниями соответствует переключению только одного элемента памяти в логической схеме, реализующей заданный автомат. Для матриц C_1 , C_2 и C_3 соответственно получаем $D_1 = 4/25$, $D_2 = 0$ и $D_3 = 3/25$. Если взять произвольное кодирование, например последовательностью натуральных чисел и нулем в двоичной системе счисления, то получим $D_1 = 12/25$, $D_2 = 2/25$ и $D_3 = 7/25$. Ясно, что предлагаемый метод снижает интенсивность переключения элементов памяти и вместе с тем потребление энергии в проектируемом устройстве.

Матрицы C_1 , C_2 и C_3 определяют систему троек не полностью определенных булевых функций $(z_1^{1'}, z_2^{1'}, z_3^{1'})$, $(z_1^{2'}, z_2^{2'}, z_3^{2'})$ и $(z_1^{3'}, z_2^{3'}, z_3^{3'})$, представляющих состояния элементов памяти автоматов A_1 , A_2 и A_3 . Их задает в интервальной форме следующая пара матриц, полученная по табл. 2 путем замены состояний их кодами:

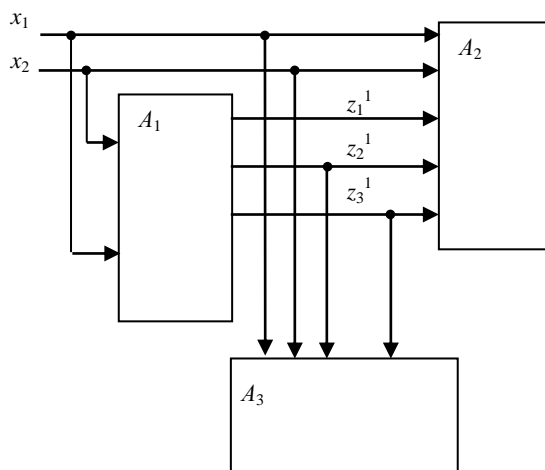
$$U = \begin{matrix} & x_1 & x_2 & z_1^1 & z_2^1 & z_3^1 & z_1^2 & z_2^2 & z_3^2 & z_1^3 & z_2^3 & z_3^3 \\ \begin{matrix} 0 \\ 1 \\ - \\ - \\ - \\ - \\ 0 \\ - \\ 0 \\ 1 \\ - \\ - \\ 1 \\ - \\ 0 \\ - \\ 1 \\ - \\ 0 \end{matrix} & \begin{matrix} 1 \\ - \\ 0 \\ 0 \\ 1 \\ 0 \\ - \\ 1 \\ 0 \\ - \\ - \\ 1 \\ 0 \\ - \\ - \\ 1 \\ - \\ 0 \\ - \\ 0 \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ - \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ - \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ - \\ 1 \\ 1 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} \end{matrix}, \quad V = \begin{matrix} & 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_3^{3'} \\ \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \\ - \end{matrix} \end{matrix}.$$

На части булева пространства входных и внутренних переменных, не охваченной интервалами, представленными матрицей U , значения всех функций не определены. Из матриц U и V видно, что функции определены на довольно небольшой части булева пространства. Это легко подсчитать, поскольку все интервалы, кроме двух, представленных второй и третьей строками матрицы U , не пересекаются между собой. Например, функция $z_1^{3'}$ (самая определенная) из всех 2024 элементов пространства определена только на 788. Поэтому для упрощения дизъюнктивных нормальных форм уместно применить метод минимизации слабо определенных булевых функций, описанный в работе [13]. При этом многие аргументы оказываются несущественными. В результате минимизации получаем пары матриц (U_1, V_1) , (U_2, V_2) и (U_3, V_3) , которые представляют системы дизъюнктивных нормальных форм, реализующие соответственно автоматы A_1 , A_2 и A_3 :

$$U_1 = \begin{matrix} & x_1 & x_2 & z_1^1 & z_2^1 & z_3^1 \\ \begin{matrix} 0 \\ - \\ - \\ - \\ - \\ - \\ 1 \end{matrix} & \begin{matrix} 1 \\ 0 \\ - \\ - \\ 1 \\ - \\ - \end{matrix} & \begin{matrix} 0 \\ 0 \\ - \\ - \\ 1 \\ - \\ - \end{matrix} & \begin{matrix} - \\ 1 \\ 1 \\ - \\ - \\ 0 \\ 0 \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ - \\ 1 \\ 1 \end{matrix} \end{matrix}, \quad V_1 = \begin{matrix} & z_1^{1'} & z_2^{1'} & z_3^{1'} \\ \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{matrix} \end{matrix}; \quad U_2 = \begin{matrix} & x_1 & x_2 & z_1^1 & z_2^1 & z_3^1 & z_2^2 \\ \begin{matrix} 0 \\ - \\ - \\ - \\ - \\ 0 \end{matrix} & \begin{matrix} 1 \\ 0 \\ - \\ - \\ 1 \\ - \end{matrix} & \begin{matrix} 0 \\ - \\ - \\ 1 \\ - \\ - \end{matrix} & \begin{matrix} - \\ 1 \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} - \\ - \\ - \\ - \\ - \\ - \end{matrix} & \begin{matrix} - \\ - \\ - \\ 1 \\ - \\ - \end{matrix} & \begin{matrix} - \\ - \\ - \\ - \\ - \\ - \end{matrix} \end{matrix}, \quad V_2 = \begin{matrix} & z_1^{2'} & z_2^{2'} & z_3^{2'} \\ \begin{matrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{matrix} \end{matrix};$$

$$U_3 = \begin{bmatrix} x_1 & x_2 & z_2^1 & z_3^1 & z_1^3 & z_2^3 & z_3^3 \\ 0 & 1 & - & - & 0 & - & 0 \\ - & 0 & 1 & - & - & - & - \\ - & - & 1 & - & - & 1 & 0 \\ - & - & - & - & 0 & 1 & 1 \\ 1 & - & - & - & 0 & 1 & - \\ - & 0 & - & - & 1 & 0 & - \\ - & - & - & - & 1 & 0 & - \\ - & - & - & - & 1 & - & - \\ 0 & - & - & 1 & - & - & - \end{bmatrix}, V_3 = \begin{bmatrix} z_1^3 & z_2^3 & z_3^3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

На рисунке показана полученная сеть последовательных автоматов, реализующая заданный параллельный автомат.



Сеть последовательных автоматов
Network of sequential automata

Заключение. Описанный подход к кодированию частичных состояний параллельного автомата 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 и каждый блок находится непосредственно на объекте управления. При этом возникает проблема распределения выходных сигналов по компонентам сети. В настоящей статье выходные сигналы не рассматривались, все внимание было сосредоточено на функции возбуждения элементов памяти. Метод энергосберегающего кодирования частичных состояний параллельного автомата рассчитан на использование его в автоматизированной системе логического проектирования. Сравнение результатов применения предложенного метода с результатами кодирования частичных состояний без учета интенсивности переключений элементов памяти показывает, что данный метод показывает лучший результат. Использование декомпозиции параллельного автомата позволяет снизить размерность трудоемкой задачи кодирования состояний.

Список использованных источников

1. Мурога, С. Системное проектирование сверхбольших интегральных схем : в 2-х кн. / С. Мурога. – М. : Мир, 1985. – Кн. 1. – 288 с.
2. Pedram, M. Power minimization in IC design: Principles and applications / M. Pedram // ACM Transactions on Design Automation of Electronic Systems. – 1996. – Vol. 1. – P. 3–56.

3. 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 NASB, 1997. – Vol. 1. – P. 60–67.
4. Sudnitson, A. Partition search for FSM low power synthesis / A. Sudnitson // *Fourth Intern. Conf. Computer-Aided Design of Discrete Devices (CAD DD'2001)*, Minsk, Republic of Belarus, 14–16 Nov. 2001. – Minsk : Institute of Engineering Cybernetics of the NASB, 2001. – Vol. 1. – P. 44–49.
5. Закревский, А. Д. Алгоритмы энергосберегающего кодирования состояний автомата / А. Д. Закревский // *Информатика*. – 2011. – № 1(29). – С. 68–78.
6. 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.
7. Поттосин, Ю. В. Энергосберегающее противогоночное кодирование состояний асинхронного автомата / Ю. В. Поттосин // *Информатика*. – 2015. – № 2(46). – С. 94–101.
8. 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.
9. Pottosin, Yu. V. Low power assignment of partial states of a parallel automaton / Yu. V. Pottosin // *Прикладная дискретная математика*. – 2022. – № 56. – С. 113–122.
10. Pottosin, Yu. Optimal state assignment of synchronous parallel automata / Yu. Pottosin // *Design of Embedded Control Systems*. – N. Y. : Springer, 2005. – P. 111–124.
11. Закревский, А. Д. Параллельные алгоритмы логического управления / А. Д. Закревский. – М. : УРСС, 2003. – 304 с.
12. Питерсон, Дж. Теория сетей Петри и моделирование систем / Дж. Питерсон. – М. : Мир, 1984. – 264 с.
13. Закревский, А. Д. Логические основы проектирования дискретных устройств / А. Д. Закревский, Ю. В. Поттосин, Л. Д. Черемисинова. – М. : Физматлит, 2007. – 592 с.
14. Закревский, А. Д. Блочное кодирование частных состояний у автоматов, реализующих параллельные алгоритмы логического управления / А. Д. Закревский // *Изв. АН СССР. Техническая кибернетика*. – 1983. – № 5. – С. 3–11.
15. Черемисинова, Л. Д. Реализация асинхронными автоматами параллельных алгоритмов логического управления / Л. Д. Черемисинова // *Автоматика и вычислительная техника*. – 1985. – № 2. – С. 65–69.
16. Поттосин, Ю. В. Декомпозиционный метод кодирования состояний параллельного автомата / Ю. В. Поттосин // *Автоматика и вычислительная техника*. – 1987. – № 1. – С. 84–91.
17. Поттосин, Ю. В. Комбинаторные задачи в логическом проектировании дискретных устройств / Ю. В. Поттосин. – Минск : Беларус. навука, 2021. – 175 с.
18. Соловьев, В. В. Проектирование цифровых систем на основе программируемых логических интегральных схем / В. В. Соловьев. – М. : Горячая линия – Телеком, 2001. – 636 с.
19. Piguet, C. Low-power and low-voltage CMOS digital design / C. Piguet // *Microelectronic Engineering*. – 1997. – No. 39. – P. 179–208.
20. Hartmanis, J. Algebraic structure theory of sequential machines / J. Hartmanis, R. E. Stearns. – Englewood Cliffs, N. J. Prentice Hall Inc., 1966. – 208 p.
21. Кээваллик, А. Э. Теорема декомпозиции конечных автоматов / А. Э. Кээваллик // *Автоматика и вычислительная техника*. – 1974. – № 1. – С. 17–24.
22. Ковалев, А. В. О нахождении отношения параллельности на множестве мест одного подкласса сетей Петри / А. В. Ковалев // *Вест. Акад. наук Беларуской ССР. Сер. фіз.-матэм. навук*. – 1989. – № 2. – С. 106–110.
23. Закревский, А. Д. Раскраска графов при декомпозиции булевых функций / А. Д. Закревский // *Логическое проектирование*. – Минск : Ин-т техн. кибернетики НАН Беларусі, 2000. – Вып. 5. – С. 83–97.
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.

References

1. Muroga S. *VLSI System Design: When and How to Design Very-Large-Scale Integrated Circuits*, 1st edition. Wiley, 1982, 496 p.
2. Pedram M. Power minimization in IC design: Principles and applications. *ACM Transactions on Design Automation of Electronic Systems*, 1996, vol. 1, pp. 3–56.
3. 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.
4. Sudnitson A. Partition search for FSM low power synthesis. *Fourth International Conference Computer-Aided Design of Discrete Devices (CAD DD'2001)*, Minsk, Republic of Belarus, 14–16 November 2001. Minsk, Institute of Engineering Cybernetics of the National Academy of Sciences of Belarus, 2001, vol. 1, pp. 44–49.
5. Zakrevskij A. D. *Algorithms for low power state assignment of an automaton*. Informatika [Informatics], 2011, no. 1(29), pp. 68–78 (In Russ.).
6. 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.).
7. Pottosin Yu. V. *Low power race-free state assignment of an asynchronous automaton*. Informatika [Informatics], 2015, no. 2(46), pp. 94–101 (In Russ.).
8. Pottosin Yu. Race-free state assignment for low power asynchronous automaton. *Further Improvements in the Boolean Domain*. In B. Steinbach (ed.). Cambridge Scholars Publishing, 2018, pp. 253–267.
9. 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.
10. Pottosin Yu. Optimal state assignment of synchronous parallel automata. *Design of Embedded Control Systems*. New York, Springer, 2005, pp. 111–124.
11. Zakrevskij A. D. Parallel'nye algoritmy logicheskogo upravleniya. *Parallel Algorithms for Logical Control*. Moscow, URSS, 2003, 304 p. (In Russ.).
12. Peterson J. L. *Petri Net Theory and the Modeling of Systems*, 1st edition. Prentice Hall, 1981, 290 p.
13. 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.).
14. Zakrevskij A. D. *Block partial state assignment of automata that implement parallel algorithms for logical control*. Izvestija Akademii nauk Sojuza Sovetskikh Socialisticheskikh Respublik. Tehnicheskaja kibernetika [Proceedings of the Academy of Sciences of the Union of Soviet Socialist Republics. Technical Cybernetics], 1983, no. 5, pp. 3–11 (In Russ.).
15. 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.).
16. 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.).
17. 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.).
18. Solov'yov V. V. Proektirovanie cifrovyyh sistem na osnove programmiruemykh logicheskikh integral'nyh shem. *The Design of Digital Systems Based on Programmable Logical Integrated Circuits*. Moscow, Goryachaya linia – Telekom, 2001, 636 p. (In Russ.).
19. Pigué C. Low-power and low-voltage CMOS digital design. *Microelectronic Engineering*, 1997, no. 39, pp. 179–208.
20. Hartmanis J., Stearns R.E. *Algebraic Structure Theory of Sequential Machines*. Englewood Cliffs, N. J. Prentice Hall Inc., 1966, 208 p.
21. Keevallik A. *Decomposition theorem of finite automata*. Avtomatika i vychislitel'naya tehnika [Automation and Computer Engineering], 1974, no. 1, pp. 17–24 (In Russ.).
22. Kovalyov A. V. *About finding parallelism relation on the set of positions of a subclass of Petri nets*. Vesci Akademiï navuk Belaruskaj Saveckaj Sacyjalistychnaj Rjespubliki. Seryja fizika-matjematychnyh navuk [Proceedings of the Academy of Sciences of the Belarusian Soviet Socialist Republic. Physics and Mathematics Series], 1989, no. 2, pp. 106–110 (In Russ.).

23. Zakrevskij A. D. *Graph coloring during the decomposition of Boolean functions*. Logicheskoe proektirovanie [*Logical Design*]. Minsk, Institut tehniczeskoj kibernetiki Nacional'noj akademii nauk Belarusi, 2000, iss. 5, pp. 83–97 (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.

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