ИНФОРМАТИКА

2015 апрель-июнь № 2

УДК 519.7

Ю.В. Поттосин

ЭНЕРГОСБЕРЕГАЮЩЕЕ ПРОТИВОГОНОЧНОЕ КОДИРОВАНИЕ СОСТОЯНИЙ АСИНХРОННОГО АВТОМАТА

Рассматривается задача противогоночного кодирования состояний асинхронного автомата, которая сводится к задаче нахождения минимального взвешенного покрытия. Предлагается метод кодирования состояний, который наряду с устранением опасных состязаний элементов памяти в реализующей схеме обеспечивает минимизацию их числа и минимизацию интенсивности их переключений.

Введение

В настоящее время при проектировании дискретных устройств управления на основе сверхбольших интегральных схем большое внимание уделяется проблеме снижения энергопотребления проектируемой схемы. Это обусловлено стремлением, с одной стороны, увеличить время действия источника энергии в портативных приборах, а с другой – снизить остроту проблемы отвода тепла при проектировании сверхбольших интегральных схем. Поэтому одним из основных критериев оптимизации при проектировании дискретных устройств является величина потребляемой схемой энергии.

Как отмечено в работах [1, 2], потребляемая мощность схемы, построенной на основе КМОП-технологии, пропорциональна интенсивности переключений логических элементов и элементов памяти, что дает возможность частично решить проблему снижения энергопотребления на уровне логического проектирования. В частности, снижения энергопотребления можно добиться при кодировании состояний автомата [3–5]. Очевидно, кодировать состояния при этом надо таким образом, чтобы при переходе автомата из одного состояния в другое меняли свое состояние как можно меньше элементов памяти.

1. Модель поведения асинхронного автомата

Моделью поведения логической схемы с памятью является конечный автомат, представляющий собой пятерку (A, B, Q, Ψ, Φ), где A, B и Q – соответственно множества входных сигналов, выходных сигналов и состояний автомата, а Ψ и Φ – функции $\Psi : A \times Q \to Q$ и $\Phi : A \times Q \to B$, называемые соответственно *функцией переходов* и *функцией выходов*. Для состояний $q_i, q_j \in Q$ и входного сигнала $a \in A$ состояние $q_j = \Psi(a, q_i)$ является тем состоянием, в которое автомат переходит из состояния q_i под воздействием входного сигнала a. Конечный автомат функционирует в дискретном времени, т. е. время разбивается на конечные промежутки, называемые *тактами*, в течение каждого из которых автомат может перейти из состояния в состояние и выдать соответствующий выходной сигнал. Рассматриваемая задача позволяет игнорировать функцию выходов Φ. Поэтому в дальнейшем она не будет упоминаться.

В настоящей работе рассматривается асинхронная реализация конечного автомата, называемая *асинхронным автоматом*, которая в отличие от синхронной реализации не имеет внешнего источника тактирующих сигналов. Переход от такта к такту происходит в момент изменения входного сигнала. При действии любого входного сигнала асинхронный автомат приходит в некоторое устойчивое состояние, из которого он не выходит до конца действия данного сигнала. При этом должно выполняться требование прямого перехода, которое формально выражается следующим образом: если $\Psi(a, q_i) = q_j$ для фиксированного входного сигнала a и некоторых состояний q_i и q_j, то $\Psi(a, q_j) = q_j$.

Задача кодирования состояний автомата заключается в присвоении каждому состоянию определенного булева вектора $(z_1, z_2, \ldots, z_k)$, называемого *кодом состояния*, который соответствует набору состояний двоичных элементов памяти (триггеров) в логической схеме, где каждый переход из состояния в состояние представляется переключением одного или нескольких триггеров. Естественно, что в реальной электронной схеме такое переключение не может происходить мгновенно и одновременно. Явление неодновременного переключения элементов памяти называется *состязаниями* или *гонками* элементов памяти [6]. 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 a, являются неустойчивыми для сигнала a, т. е. при любом порядке переключений элементов памяти автомат из некоторого состояния q_i под воздействием входного сигнала a переходит всегда в состояние $q_j = \Psi(a, q_i)$. 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*. Естественно, здесь возникает задача минимизации длины кода состояния, приводящая к наименьшему числу элементов памяти в реальной схеме.

Другим критерием оптимизации схемы, как указано выше, является величина потребляемой энергии. Проблеме энергосберегающего кодирования состояний синхронного автомата посвящено довольно много работ, одной из которых является, например, работа [5], где процесс кодирования состояний синхронного автомата представляется как размещение состояний в булевом пространстве внутренних переменных. Асинхронным автоматам давно стало уделяться большое внимание [7–9], и некоторые преимущества в определенных условиях асинхронных схем над синхронными схемами отмечены, например, в работе [7, с. 45]. Задача энергосбережения для асинхронных автоматов также может быть частично сведена к уменьшению переключательной активности элементов схемы. В предлагаемой вниманию работе рассматривается возможность учета энергосбережения при противогоночном кодировании состояний асинхронного автомата.

2. Условия отсутствия опасных состязаний

Существование опасных состязаний для пары переходов $q_i \to q_j$, $q_k \to q_l$ ($q_j \neq q_l$) при одном и том же входном сигнале a может привести к тому, что автомат вместо перехода в состояние q_j из состояния q_i может оказаться в состоянии q_l, которое также является устойчивым при входном сигнале a. Условие отсутствия опасных состязаний для этой пары можно выразить троичным вектором, в котором компоненты i и j соответствуют состояниям автомата и имеют одно значение (0 или 1), а компоненты k и l – противоположное значение [10]. Остальным компонентам приписывается значение «–». В схеме, реализующей заданный автомат, это условие выполняется триггером, который в процессе одного из переходов рассматриваемой пары хранит состояние 0, а в процессе другого перехода – состояние 1.

Пусть, например, табл. 1 представляет функцию переходов $\Psi(a, q)$ заданного автомата, т. е. является его таблицей переходов. Ее строкам соответствуют состояния автомата, а столбцам – входные сигналы. В клетках таблицы даны значения функции $\Psi(a, q)$ при соответствующих значениях аргументов a и q. Устойчивые состояния для каждого входного сигнала выделены.

Таблица 1

	a_1	a_2	a_3	a_4
q_1	$\boldsymbol{q_1}$	q_2	q_3	$\boldsymbol{q_1}$
q_2	$\boldsymbol{q_2}$	$\boldsymbol{q_2}$	q_8	q_4
q_3	q_1	q_2	$\boldsymbol{q_3}$	q_4
q_4	q_2	q_2	q_5	$\boldsymbol{q_4}$
q_5	$\boldsymbol{q_5}$	$\boldsymbol{q_5}$	$\boldsymbol{q_5}$	q_6
q_6	$\boldsymbol{q_6}$	$\boldsymbol{q_6}$	q_3	$\boldsymbol{q_6}$
q_7	$\boldsymbol{q_7}$	q_8	$\boldsymbol{q_7}$	$\boldsymbol{q_7}$
q_8	q_7	$\boldsymbol{q_8}$	$\boldsymbol{q_8}$	q_1

Условие отсутствия опасных состязаний для пары переходов $q_3 \to q_1$, $q_4 \to q_2$ при входном сигнале a_1 выражается вектором (0 1 0 1 – –) либо покомпонентной инверсией этого вектора (1 0 1 0 – –).

На множестве векторов, представляющих условия отсутствия опасных состязаний, имеется отношение импликации: троичный вектор a *имплицирует* 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) имплицирует вектор (1 0 – – – 0 1), а также вектор (0 1 – – – 1 –). Смысл этого отношения в том, что условие, представленное вектором b, автоматически выполняется при соблюдении условия, представленного вектором a.

Все условия отсутствия опасных состязаний в виде описанных векторов составляют троичную матрицу, в которой отсутствуют имплицируемые строки. Эта матрица называется *матрицей условий* [10]. Для автомата, таблицей переходов которого является табл. 1, при рассмотрении пар переходов число строк этой матрицы равно 31.

Чтобы избежать громоздких вычислений, для ускорения процесса противогоночного кодирования состояний асинхронного автомата вместо пар переходов рассматривают пары так называемых *K-множеств* [6], каждое из которых является множеством состояний асинхронного автомата, переходы из которых при некотором фиксированном входном сигнале ведут в одно и то же устойчивое состояние (также принадлежащее данному множеству). Например, для входного сигнала a_2 автомата, поведение которого задает табл. 1, K-множествами являются $\{q_1, q_2, q_3, q_4\}$, $\{q_5\}$, $\{q_6\}$ и $\{q_7, q_8\}$. Фактически K-множество представляет множество переходов в одно и то же устойчивое состояние при одном и том же входном сигнале.

Два различных K-множества, построенных для одного и того же входного сигнала, образуют *пару K-множеств*. Так же как и для пары переходов, для каждой пары K-множеств строится троичный вектор, компоненты которого соответствуют состояниям автомата. Компоненты, соответствующие состояниям, которые принадлежат одному из этих K-множеств, имеют значение 0; компоненты, соответствующие состояниям из другого K-множества, – значение 1, а компоненты, соответствующие состояниям, не принадлежащим ни одному из них, – значение «–». Так же как и для пар переходов, множество этих векторов для всех пар K-множеств, из которого удалены имплицируемые векторы, образует матрицу условий. Для автомата, поведение которого описано в табл. 1, матрица условий имеет следующий вид:

$$
\begin{array}{cc}
\begin{array}{cccccccc} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 & q_8 \end{array} & \\
\begin{bmatrix}
0 & 1 & 0 & 1 & - & - & - & - \\
- & - & - & - & 0 & - & 1 & 1 \\
- & - & - & - & - & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & - & - & - \\
0 & 0 & 0 & 0 & - & 1 & - & - \\
0 & 0 & 0 & 0 & - & - & 1 & 1 \\
0 & 1 & 0 & - & - & 0 & - & 1 \\
0 & - & 0 & 1 & 1 & 0 & - & - \\
0 & - & 0 & - & - & 0 & 1 & - \\
- & 0 & - & 1 & 1 & - & - & 0 \\
- & 0 & - & - & - & - & 1 & 0 \\
0 & 1 & 1 & 1 & - & - & - & 0 \\
0 & - & - & - & 1 & 1 & - & 0 \\
0 & - & - & - & - & - & 1 & 0 \\
- & 0 & 0 & 0 & 1 & 1 & - & - \\
- & - & - & 0 & 0 & 0 & 1 & -
\end{bmatrix}
&
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \end{array}
\end{array}.
$$

3. Минимизация длины кода состояния

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. Теперь 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. Необходимо заметить, что при рассмотрении K-множеств не всегда удается получить минимум длины кода состояния, достижимый при рассмотрении пар переходов. Этот прием следует использовать в том случае, когда важнее получить результат за более короткое время.

4. Минимизация переключательной активности элементов памяти

При применении описанного подхода к решению задачи противогоночного кодирования состояний асинхронного автомата для снижения интенсивности переключений элементов памяти можно использовать следующие предположения.

Каждому i-му столбцу матрицы кодирования можно поставить в соответствие множество переходов. Данными переходами связаны состояния автомата, в кодах которых переменная z_i имеет различные значения, т. е. при этих переходах i-й триггер в реальной схеме, реализующей заданный автомат, меняет свое состояние. Следовательно, для снижения интенсивности переключений элементов памяти надо выбрать такой вариант противогоночного кодирования состояний, который соответствует наименьшему множеству переходов между состояниями.

Если удается вычислить вероятности переходов, то столбцу матрицы кодирования состояний ставится в соответствие вероятность события, которое заключается в том, что происходит некоторый переход из множества переходов, связанных с данным столбцом матрицы кодирования состояний. Поскольку переходы между состояниями автомата являются несовместимыми событиями, эта вероятность равна сумме вероятностей отдельных переходов из данного множества. Для подсчета вероятностей переходов между состояниями в статье [5] используется метод Чэпмена – Колмогорова, где данные вероятности получаются в результате решения системы линейных уравнений с этими вероятностями в качестве неизвестных. Однако такой метод можно применять только тогда, когда автомат полностью определен, а его граф поведения является сильносвязным ориентированным графом. В противном случае столбцу матрицы кодирования состояний автомата можно, например, приписывать мощность связанного с ним множества переходов.

Таким образом, каждому совместимому множеству строк матрицы условий и, соответственно, вектору, имплицирующему все строки из данного множества, приписывается вес в виде числа переходов или 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.

Вероятность перехода из состояния q_i в состояние q_j, вызываемого входным сигналом a, когда автомат находится в состоянии q_i, равна вероятности прихода входного сигнала a. 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, поскольку поступления на вход автомата различных входных сигналов – несовместимые события. Условием перехода является то, что автомат находится в состоянии q_i. Нахождение автомата в состоянии q_i и приход сигнала, переводящего в состояние q_j, являются независимыми событиями. Поэтому абсолютная вероятность p_{ij} перехода из состояния q_i в состояние q_j в течение всего времени работы автомата равна $P_i p'_{ij}$, где P_i – вероятность того, что автомат находится в состоянии q_i.

Вероятности P_i ($i = 1, 2, \ldots, |Q|$) находятся путем решения системы уравнений Чэпмена – Колмогорова, которые имеют следующий вид:

$$\sum_{i=1}^{|Q|} P_i p'_{ij} = P_j , \quad j = 1, 2, \ldots, |Q|;$$

$$\sum_{i=1}^{|Q|} P_i = 1.$$

Вероятности p'_{ij} должны быть известны. Таким образом, решив данную систему уравнений, получим вероятности P_i. Как было сказано раньше, абсолютная вероятность $p_{ij} = P_i p'_{ij}$.

Асинхронный автомат, поведение которого описано в табл. 1, является полностью определенным, а граф его поведения является сильносвязным ориентированным графом. Следовательно, вероятности переходов между состояниями можно определять по методу Чэпмена – Колмогорова. Допустим, что вероятности входных сигналов данного автомата имеют равномерное распределение. Тогда условные вероятности p'_{ij} переходов (из состояния q_i в состояние q_j, когда автомат находится в состоянии q_i) представлены в табл. 2, где строки и столбцы соответствуют состояниям автомата и на пересечении строки q_i и столбца q_j расположена вероятность p'_{ij}. Пустые клетки означают отсутствие перехода между соответствующими состояниями (вероятность такого перехода равна нулю).

Таблица 2

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
q_1	1/2	1/4	1/4					
q_2		1/2		1/4				1/4
q_3	1/4	1/4	1/4	1/4				
q_4		1/2		1/4	1/4			
q_5					3/4	1/4		
q_6			1/4			3/4		
q_7							3/4	1/4
q_8	1/4						1/4	1/2

Для нахождения вероятностей состояний (вероятностей попадания автомата в те или иные состояния) надо решить следующую систему линейных уравнений (для упрощения вычислений используем величины, пропорциональные условным вероятностям):

$$2 P_1 + P_3 + P_8 = 4 P_1;$$

$$P_1 + 2 P_2 + P_3 + 2 P_4 = 4 P_2;$$

$$P_1 + P_3 + P_6 = 4 P_3;$$

$$P_2 + P_3 + P_4 = 4 P_4;$$

$$P_4 + 3\,P_5 = 4\,P_5;$$

$$P_5 + 3\,P_6 = 4\,P_6;$$

$$3\,P_7 + P_8 = 4\,P_7;$$

$$P_2 + P_7 + 2\,P_8 = 4\,P_8;$$

$$P_1 + P_2 + P_3 + P_4 + P_5 + P_6 + P_7 + P_8 = 1.$$

В результате решения данной системы уравнений получаем $P_1 = 4/27$, $P_2 = P_7 = P_8 = 5/27$, $P_3 = P_4 = P_5 = P_6 = 2/27$. Абсолютные вероятности переходов, приведенные к общему знаменателю, представлены в табл. 3, где строки и столбцы соответствуют состояниям автомата и на пересечении строки q_i и столбца q_j расположена вероятность p_{ij}. Пустые клетки, так же как и в табл. 2, показывают нулевые вероятности.

Таблица 3

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
q_1	2/27	1/27	1/27					
q_2		5/54		5/108				5/108
q_3	1/54	1/54	1/54	1/54				
q_4		1/27		1/54	1/54			
q_5					3/54	1/54		
q_6			1/54			3/54		
q_7							15/108	5/108
q_8	5/108						5/108	5/54

Если рассматривать пары переходов, то максимальных совместимых множеств строк матрицы условий окажется 55, что затруднит демонстрацию предлагаемого подхода. Поэтому обратимся к менее громоздким вычислениям при рассмотрении пар *K*-множеств. В табл. 4 представлены полученные для рассматриваемого примера максимальные совместимые множества строк матрицы условий, обозначенные их номерами, и векторы, имплицирующие все строки из соответствующих множеств. В качестве весов полученных множеств представлены величины, пропорциональные суммам соответствующих вероятностей (числители этих сумм при общем знаменателе 108).

Таблица 4

Совместимые множества	Имплицирующие векторы	Веса множеств
{1,2,3,7,9}	(0 1 0 1 0 0 1 1)	15
{1,2,8}	(0 1 0 1 1 0 0 0)	15
{1,2,13,16}	(0 1 0 1 1 1 0 0)	15
{1,3,7,8,9}	(0 1 0 1 1 0 1 1)	15
{1,7,8,11}	(0 1 0 1 1 0 0 1)	25
{1,8,9,14}	(0 1 0 1 1 0 1 0)	25
{1,13,14}	(0 1 0 1 1 1 1 0)	25
{2,3,4,5,13,15}	(0 0 0 0 1 1 0 0)	4
{2,3,6,9,16}	(0 0 0 0 0 0 1 1)	10
{2,3,7,9,10,16}	(0 1 0 0 0 0 1 1)	16
{2,3,10,13,16}	(0 0 0 1 1 1 0 0)	13
{2,3,12,13,16}	(0 1 1 1 1 1 0 0)	15
{2,5,6}	(0 0 0 0 0 1 1 1)	14
{3,4,6,9}	(0 0 0 0 1 0 1 1)	14
{4,5,6,15}	(0 0 0 0 1 1 1 1)	14
{4,5,11,13,14,15}	(0 0 0 0 1 1 1 0)	14

Окончание табл. 4

Совместимые множества	Имплицирующие векторы	Веса множеств
{4,9,11,14}	(0 0 0 0 1 0 1 0)	14
{7,10,11}	(0 1 0 0 0 0 0 1)	30
{8,9,10,11,14}	(0 0 0 1 1 0 1 0)	23
{9,11,14,16}	(0 0 0 0 0 0 1 0)	10
{12,13,14}	(0 1 1 1 1 1 1 0)	25

В настоящей работе не рассматривается какой-то конкретный метод получения минимальных взвешенных покрытий. Эта задача известна давно и достаточно подробно исследована (см., например, [11]). Заметим только, что если не учитывать веса полученных максимальных совместимых множеств, то в качестве покрытия может быть найдена совокупность совместимых множеств {1, 8, 9, 14}, {2, 3, 12, 13, 16}, {4, 5, 6, 15}, {7, 10, 11} с весом 84. Минимальный вес, равный 67, имеет покрытие, которое состоит из множеств {1, 3, 7, 8, 9}, {2, 3, 12, 13, 16}, {4, 5, 6, 15}, {8, 9, 10, 11, 14} и согласно принятому критерию является лучшим. Соответственно имеем следующие матрицы кодирования:

$$
\begin{array}{c} \\ q_1\\ q_2\\ q_3\\ q_4\\ q_5\\ q_6\\ q_7\\ q_8 \end{array}
\begin{array}{c} \begin{array}{cccc} z_1 & z_2 & z_3 & z_4 \end{array} \\
\begin{bmatrix} 0&0&0&0\\ 1&1&0&1\\ 0&1&0&0\\ 1&1&0&0\\ 1&1&1&0\\ 0&1&1&0\\ 1&0&1&0\\ 0&0&1&1 \end{bmatrix} \end{array}
;\quad
\begin{array}{c} \\ q_1\\ q_2\\ q_3\\ q_4\\ q_5\\ q_6\\ q_7\\ q_8 \end{array}
\begin{array}{c} \begin{array}{cccc} z_1 & z_2 & z_3 & z_4 \end{array} \\
\begin{bmatrix} 0&0&0&0\\ 1&1&0&0\\ 0&1&0&0\\ 1&1&0&1\\ 1&1&1&1\\ 0&1&1&0\\ 1&0&1&1\\ 1&0&1&0 \end{bmatrix} \end{array} .
$$

Оценить преимущество первого варианта матрицы кодирования над вторым можно еще следующим образом. Кодирование состояний автомата можно представить как размещение состояний автомата в пространстве внутренних переменных $z_1, z_2, \ldots, z_k$ [5], т. е. по вершинам булева гиперкуба, представляющего это пространство. В статье [5] введен критерий качества такого размещения с точки зрения интенсивности переключений элементов памяти. Этот критерий выражается формулой $D = \sum w_{ij}(d_{ij} - 1)$, где d_{ij} – расстояние по Хэммингу между кодами состояний q_i и q_j, w_{ij} в данном случае – число переходов между состояниями q_i и q_j или величина, пропорциональная вероятности перехода между состояниями q_i и q_j ($i \neq j$). Суммирование ведется по всем парам состояний, соответствующим парам вершин в гиперкубе. Очевидно, чем меньше значение D, тем лучше результат размещения, и $D = 0$, если всем парам состояний, связанным переходами, соответствуют ребра гиперкуба. Тогда при любом переходе из состояния в состояние переключается ровно один элемент памяти.

Для первого варианта кодирования $D = 35$. Второй вариант кодирования дает $D = 18$. Сравнение по критерию D результатов решения примеров показывает целесообразность использования предлагаемого метода.

Заключение

Предлагаемый метод энергосберегающего противогоночного кодирования состояний асинхронного автомата рассчитан на использование его в автоматизированной системе логического проектирования. Сравнение результатов кодирования состояний изложенным методом и кодирования состояний без учета интенсивности переключений элементов памяти показывает, что применение данного метода дает лучший результат.

Список литературы

1. Мурога, С. Системное проектирование сверхбольших интегральных схем. В 2-х кн. Кн. 1 / С. Мурога. – М. : Мир, 1985. – 288 с.

2. Pedram, M. Power minimization in IC design: Principles and applications / M. Pedram // ACM Trans. Design Automat. Electron. Syst. – 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. – Minsk : Institute of Engineering Cybernetics NAS of Belarus, 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. – Minsk : Institute of Engineering Cybernetics NAS of Belarus, 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. Закревский, А.Д. Алгоритмы синтеза дискретных автоматов / А.Д. Закревский. – М. : Наука, 1971. – 512 с.

7. Ангер, С. Асинхронные последовательностные схемы / С. Ангер. – М. : Наука, 1977. – 400 с.

8. Синтез асинхронных автоматов на ЭВМ / под ред. А.Д. Закревского. – Минск : Наука и техника, 1975. – 184 с.

9. Автоматизированное проектирование цифровых устройств / под ред. С.С. Бадулина. – М. : Радио и связь, 1981. – 240 с.

10. Закревский, А.Д. Логические основы проектирования дискретных устройств / А.Д. Закревский, Ю.В. Поттосин, Л.Д. Черемисинова. – М. : Физматлит, 2007. – 592 с.

11. Закревский, А.Д. Оптимизация покрытий множеств / А.Д. Закревский // Логический язык для представления алгоритмов синтеза релейных устройств. – М. : Наука, 1966. – С. 136–148.

Поступила 07.05.2015

Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: pott@newman.bas-net.by

Yu.V. Pottosin

LOW POWER RACE-FREE STATE ASSIGNMENT OF AN ASYNGHRONOUS AUTOMATON

The problem of a race free state assignment of an asynchronous automaton is considered. A method for the state assignment is suggested that provides the minimization of the number and the switching activity of the memory elements along with the elimination of the critical races between them.