ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ ДИСКРЕТНЫХ АВТОМАТОВ

УДК 519.7 DOI 10.17223/20710410/56/7

LOW POWER ASSIGNMENT OF PARTIAL STATES OF A PARALLEL AUTOMATON

Yu. V. Pottosin

United Institute of Informatics Problems, National Academy of Sciences of Belarus, Minsk, Belarus

E-mail: pott@newman.bas-net.by

The problem of a low-power assignment of the partial states of a parallel automaton is considered. A method to solve that problem is suggested that provides minimizing the number of memory elements in the implementing circuit of the automaton and minimization of their switching activity. The problem is reduced to finding a minimal weighted cover of a graph with its complete bipartite sub-graphs (bi-cliques).

Keywords: *parallel automaton, partial state, state assignment, complete bipartite sub-graph, weighted cover problem.*

ЭНЕРГОСБЕРЕГАЮЩЕЕ КОДИРОВАНИЕ ЧАСТИЧНЫХ СОСТОЯНИЙ ПАРАЛЛЕЛЬНОГО АВТОМАТА

Ю. В. Поттосин

Объединенный институт проблем информатики НАН Беларуси, г. Минск, Беларусь

Рассматривается задача кодирования частичных состояний параллельного автомата. Предложен метод решения, который обеспечивает минимизацию числа элементов памяти в схеме, реализующей автомат, и минимизацию интенсивности их переключений. Задача сводится к нахождению минимального взвешенного покрытия графа его полными двудольными подграфами (бикликами).

Ключевые слова: *параллельный автомат, частичное состояние, кодирование состояний, полный двудольный подграф, задача о взвешенном покрытии.*

1. Introduction

At present time, a great attention is paid to decreasing power consumption in designing discrete devices based on CMOS technology. It is caused by the tendency to increase the working time of power supply for portable devices and, on the other hand, by the tendency to lower acuity of the problem of heat rejection in designing VLSI circuits. Therefore, one of the main optimization criteria in designing discrete devices is amount of power consumption.

As it is said in [1, 2], the power consumption of a circuit built on the base of CMOS technology is proportional to switching activity of its logical and memory elements. It allows solving this problem at the level of logical design. In particular, decreasing power consumption can be achieved in the stage of state assignment of an automaton when the states of a given automaton are assigned with binary or ternary vectors (codes) in order to obtain a system of Boolean functions necessary for constructing a logical net of a designed device. This problem was solved in [3 – 8] for classical sequential automata at synchronous and asynchronous realizations. The states of an automaton must be encoded in such a way that during the transition between its states as few memory elements as possible change their state. In this paper, a method for low-power assignment of partial states of a parallel automaton is suggested. The method uses the approach described in [9] that reduces the problem to the search for a cover of a graph with its complete bipartite sub-graphs (bicliques).

2. The used model

A parallel automaton is a functional model of a discrete device that gives a convenient description of the parallelism of controlled interactive processes [10]. This model is close to the widely known Petri net [11]. It consists of the following objects: a set of *partial states* $Q = \{q_1, q_2, \dots, q_\gamma\}$, a set of input Boolean variables $X = \{x_1, x_2, \dots, x_n\}$, a set of output Boolean variables $Y = \{y_1, y_2, \dots, y_m\}$ and a set of transitions $T = \{\tau_1, \tau_2, \dots, \tau_t\}$ that is a sequence of lines of the form:

$$\tau_i = S_i : -\ K'_i \to K''_i \to F_i, \tag{1}$$

where $S_i, F_i \subseteq Q$, K'_i is an elementary conjunction of variables from the set X, and K''_i is an elementary conjunction of variables from the set Y.

Unlike the classical finite sequential automaton, the parallel automaton can be in several states simultaneously. Those states are called above as partial. The set of all partial states that a parallel automaton can be in at some time is called *global state*. The sense of the line (1) is the following. If a partial automaton is in states forming the set S_i and Boolean variables took values that convert K'_i to 1, then K''_i takes value 1 and the automaton comes to states in F_i from states composing S_i. In other words, let $P = \{P_1, P_2, \dots, P_p\}$ be the set of all reachable global states of a given parallel automaton. Then if $S_i \subseteq P_g$, where P_g is a current global state of the automaton, and the automaton has received binary signals that turned the conjunction K'_i into 1, then the global set will be $P_h = (P_g \setminus S_i) \cup F_i$ at the next time and the automaton will produce binary signals that turn K''_i into 1. Any conjunction, K'_i and K''_i, can be absent in the line. The absence of K'_i means its identical equality to 1. The absence of K''_i means, according to interpretation of the model, that either all the variables in Y are equal to 0 or the values of them do not change. As well as for a sequential automaton, the synchronous or asynchronous implementation can be for a parallel automaton. Further, the synchronous implementation is considered. At that implementation, the time is divided into fixed units, during which the automaton goes from one state to another.

The following restrictions are introduced in the model:

1) The initial global state is a one-element set. For the sake of determinacy, it can be $\{q_1\}$.
2) For two different lines, i-th and j-th, $S_i = S_j$ if $S_i \cap S_j \neq \varnothing$.

There are a number of other restrictions given in [10] and connected with correctness of an automaton description. They will not be considered here, since the correctness problem is not regarded. Also, the output signals will not be considered here.

Example 1. The following sequence of lines can be an example of parallel automaton:

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

Here, $Q = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and $X = \{x_1, x_2\}$. Let us take one-element set $\{1\}$ as an initial global state. The first line (transition τ_1) means that the automaton goes from state $\{1\}$ and comes in state $\{10\}$ in the next time unit if $x_1 = 0$ and $x_2 = 1$. The state $\{10\}$ is global one as well. The automaton stays in state $\{1\}$ at any other value combination of x_1 and x_2. The automaton goes from global state $\{10\}$ to partial states 2, 3 and 4 at $x_2 = 0$ (transition τ_2). Those partial states constitute the global state $\{2, 3, 4\}$. At the next time unit, the automaton changes the partial state 2 for partial states 5 and 6 independently on the values of input variables. As far as we consider the synchronous implementation, where transitions can happen simultaneously, the transitions τ_3 and τ_5 happen simultaneously at $x_1 = 0$, while τ_3 and τ_6 at $x_1 = 1$. Correspondingly, the automaton will be in global states $\{3, 5, 6, 7\}$ and $\{3, 5, 6, 9\}$. Having observed the functioning of the automaton in this way we get global states $\{6, 7, 8\}$ and $\{6, 8, 9\}$.

3. The problem of assignment of partial states

Any two partial states, in which the automaton can be simultaneously, are called *parallel*. The state assignment of an automaton consists in assigning its states with binary or ternary vectors of the space of introduced inner variables $z_1, z_2, \ldots, z_l$ (codes of states). The components of a ternary vector have values 0, 1 and "–". Parallel partial states of a parallel automaton are assigned with non-orthogonal ternary vectors and non-parallel partial states with orthogonal ones [12]. Two ternary vectors are orthogonal if there is a component that has value 0 in one vector and 1 in the other [13].

One of the way to establish parallelism between partial states of an parallel automaton is to obtain the set $P = \{P_1, P_2, \ldots, P_p\}$ of all reachable global states. Two partial states are parallel if there is a reachable global state $P_j \subseteq Q$ containing these states. The parallelism relation is considered to be irreflexive, i.e., a partial state is not parallel to itself. Indeed, the requirement of correctness of the automaton description does not allow intersection of sets $P_g \setminus S_i$ and F_i in transition τ_i, where P_g is the current global state of the automaton.

For the Example 1, the global states $P_1 = \{1\}$, $P_2 = \{10\}$, $P_3 = \{2, 3, 4\}$, $P_4 = \{3, 5, 6, 7\}$, $P_5 = \{3, 5, 6, 9\}$, $P_6 = \{6, 7, 8\}$, and $P_7 = \{6, 8, 9\}$ are obtained that yield the following matrix of partial states parallelism:

$$\begin{array}{cccccccccc|c}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 3 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 5 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 6 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 7 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 8 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 9 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10
\end{array}$$

4. The method for partial state assignment

The suggested method for the assignment of partial states of parallel automaton supposes considering the non-parallelism graph G of partial states. Its vertices correspond to the partial states of a given automaton, and edges to the pairs of non-parallel partial states. Complete bipartite subgraphs (*bi-cliques*) are extracted from G. Each bi-clique B can be given enough by the pair $\langle V^1, V^2\rangle$ of its vertices, as every vertex in V^1 is connected by edges with all the vertices in V^2.

Let a family of bi-cliques $B_1, B_2, \ldots, B_m$ of graph G, where $B_i = \langle V_i^1, V_i^2\rangle$, $i = 1, 2, \ldots, m$, be a cover of G, i.e., for each edge of G there is a bi-clique in this family that has this edge. Every $B_k = \langle V_k^1, V_k^2\rangle$, $k = 1, 2, \ldots, m$, is put in correspondence to variable z_k with $z_k = 0$ for the partial states in V_k^1 and $z_k = 1$ for partial states in V_k^2 (or vice versa). The value of z_k is "$-$" if the corresponding partial state is neither in V_k^1 nor in V_k^2. It is natural to demand minimum of m. Having this minimum we obtain the assignment of partial states with codes of minimal length that relates to the minimum number of memory elements in the designed device. Evidently, it is enough to consider only maximal bi-cliques, i.e., any of them is not a proper subgraph of any other bi-clique.

Thus, the partial state assignment of a parallel automaton is reduced to the search for maximal bi-cliques in the non-orthogonality graph G of partial states of the given automaton and covering G with the found bi-cliques.

When applying this approach to lower the memory element activity, the *problem of minimal weighted cover* must be solved and the following reasons can be used.

Each inner variable z_i can be put into correspondence to a set of transitions between partial states. That set consists of the transitions which connect the partial states whose codes have different values of z_i. During these transitions the i-th memory element in a real circuit implementing the given automaton changes its state. Hence, in order to lower the activity of memory elements, such a variant of the assignment of partial states must be chosen that the number of variables changing their values during the transitions between states would be as small as possible.

If we manage to calculate the probabilities of transitions between partial states, then the probability of the transitions changing the value of z_i is put in correspondence to z_i. The more probability of the transition, the less inner variables have to change their values at that. We connect the transitions between partial states with the transitions between global states. Since the transitions between global states of a parallel automaton are incompatible events, the probability of the transition between partial states q_i and q_j is equal to the sum of the probabilities of the transitions between those global states where there is exchange of q_i and q_j. When all the reachable global states of a given parallel automaton are determined, it is easy to construct the sequential automaton that is equivalent to the given one. Its states are the global states of the given parallel automaton. The Chapman — Kolmogorov method is used in [5] to calculate the probabilities of transitions between states of a sequential automaton, where the probabilities are found as a result of solving a system of linear equations with those probabilities as unknowns. This method can be applied only when the automaton is completely specified and its behavior graph is strongly connected directed graph. Otherwise, the number of transitions related to changing z_i, $i = 1, 2, \dots, m$, can be put in correspondence to z_i.

Each maximal bi-clique of the non-parallelism graph G of partial states is weighted with a value proportional to the probability of transitions related to the corresponding inner variable. The weight of a cover is the sum of weights of bi-cliques constituting the cover.

Let us consider the parallel automaton from the Example 1. The non-parallelism matrix of partial states of it (the inverse of the parallelism matrix with retained main diagonal) that is the adjacency matrix of the non-orthogonality graph G of the partial state codes of the automaton is as follows:

$$\begin{array}{cccccccccc|c}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & \\
\hline
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 3 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 4 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 5 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 6 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 7 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 8 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 9 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 10
\end{array}$$

A method for constructing all the maximal bi-cliques in a graph is described in [14]. The following maximal bi-cliques are obtained for graph G:

$B_1 = \langle\{1\}, \{2,3,4,5,6,7,8,9,10\}\rangle$, $B_9 = \langle\{1,2,4,8,10\}, \{5\}\rangle$, $B_{17} = \langle\{1,7\}, \{2,4,9,10\}\rangle$,
$B_2 = \langle\{1,2,3,4,5\}, \{8,10\}\rangle$, $B_{10} = \langle\{1,2,4,9\}, \{7,10\}\rangle$, $B_{18} = \langle\{1,7,10\}, \{2,4,9\}\rangle$,
$B_3 = \langle\{1,2,3,4,5,6,7,8,9\}, \{10\}\rangle$, $B_{11} = \langle\{1,2,4,9,10\}, \{7\}\rangle$, $B_{19} = \langle\{1,8\}, \{2,3,4,5,10\}\rangle$,
$B_4 = \langle\{1,2,3,4,5,10\}, \{8\}\rangle$, $B_{12} = \langle\{1,2,4,10\}, \{5,6,7,8,9\}\rangle$, $B_{20} = \langle\{1,8,10\}, \{2,3,4,5\}\rangle$,
$B_5 = \langle\{1,2,4\}, \{5,6,7,8,9,10\}\rangle$, $B_{13} = \langle\{1,5\}, \{2,4,8,10\}\rangle$, $B_{21} = \langle\{1,9\}, \{2,4,7,10\}\rangle$,
$B_6 = \langle\{1,2,4,7\}, \{9,10\}\rangle$, $B_{14} = \langle\{1,5,6,7,8,9\}, \{2,4,10\}\rangle$, $B_{22} = \langle\{1,9,10\}, \{2,4,7\}\rangle$,
$B_7 = \langle\{1,2,4,7,10\}, \{9\}\rangle$, $B_{15} = \langle\{1,5,6,7,8,9,10\}, \{2,4\}\rangle$, $B_{23} = \langle\{1,10\}, \{2,3,4,5,6,7,8,9\}\rangle$.
$B_8 = \langle\{1,2,4,8\}, \{5,10\}\rangle$, $B_{16} = \langle\{1,5,10\}, \{2,4,8\}\rangle$,

5. Calculating probabilities of transitions between partial states and weighting bi-cliques

As it was said above, the probability of changing a partial state for another is equal to the sum of probabilities of transitions between global states, where this changing happens. Considering the sequential automaton equivalent to a given parallel automaton, whose global states in $P = \{P_1, P_2, \dots, P_\gamma\}$ are in one-to-one correspondence to the states in $S = \{s_1, s_2, \dots, s_\gamma\}$ of the sequential automaton, is appropriate here.

The probability of transition of a sequential automaton from a state s_i to a state s_j caused by an input signal $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is equal to the probability of coming $\mathbf{x}$. If there are several input signals transferring the automaton from s_i to s_j, the conditional probability p'_{ij} of such transfer is equal to the sum of the probabilities of those signals. The condition is that the automaton is in the state s_i. The absolute probability p_{ij} of the transition from s_i to s_j for all the time of the automaton working is equal to the product $p(s_i)p'_{ij}$, where $p(s_i)$ is the probability that the automaton is in the state s_i — this event and coming signals that transfer the automaton from s_i to s_j, are independent events. The following system of equations has to be solved:

$$\sum_{i=1}^{\gamma} p(s_i)p'_{ij} = p(s_j), \quad j = 1, \ldots, \gamma,$$
$$\sum_{i=1}^{\gamma} p(s_i) = 1.$$

The probability p'_{ij} must be known. This is the probability of coming an input signal that transfers the automaton from state s_i into s_j. Further, we assume that the probabilities of input signals are uniformly distributed. Having solved this system of equations we obtain the probabilities $p(s_i) = p(P_i)$, $i = 1, 2, \ldots, \gamma$.

The transitions between global states of the parallel automaton in the Example 1 are shown in Table 1, where rows and columns correspond to global states, and its entry is the condition of the transition from the global state corresponding to the row into the global state corresponding to the column.

Table 1

States	1	10	$\{2,3,4\}$	$\{3,5,6,7\}$	$\{3,5,6,9\}$	$\{6,7,8\}$	$\{6,8,9\}$
1	$x_1 \vee \overline{x}_2$	$\overline{x}_1 x_2$					
10		x_2	$\overline{x}_2$				
$\{2,3,4\}$				$\overline{x}_1$	x_1		
$\{3,5,6,7\}$					$\overline{x}_2$	x_2	
$\{3,5,6,9\}$					$\overline{x}_2$		x_2
$\{6,7,8\}$						x_2	$\overline{x}_2$
$\{6,8,9\}$	x_1						$\overline{x}_1$

For the probabilities of global states of the considered parallel automaton according to Table 1, the following system of equations is formed:

$$\begin{aligned}
p(1) &= 3/4\,p(1) + 1/2\,p(\{6,8,9\}),\\
p(10) &= 1/2\,p(10) + 1/4\,p(1),\\
p(\{2,3,4\}) &= 1/2\,p(10),\\
p(\{3,5,6,7\}) &= 1/2\,p(\{2,3,4\}),\\
p(\{3,5,6,9\}) &= 1/2\,p(\{3,5,6,9\}) + 1/2\,p(\{2,3,4\}) + 1/2\,p(\{3,5,6,7\}),\\
p(\{6,7,8\}) &= 1/2\,p(\{6,7,8\}) + 1/2\,p(\{3,5,6,7\}),\\
p(\{6,8,9\}) &= 1/2\,p(\{6,8,9\}) + 1/2\,p(\{3,5,6,9\}) + 1/2\,p(\{6,7,8\}),
\end{aligned}$$
$$p(1) + p(10) + p(\{2,3,4\}) + p(\{3,5,6,7\}) + p(\{3,5,6,9\}) + p(\{6,7,8\}) + p(\{6,8,9\}) = 1.$$

The solution of this system gives the probabilities $p(1) = 8/23$, $p(10) = 4/23$, $p(\{2,3,4\}) = = 2/23$, $p(\{3,5,6,7\}) = 1/23$, $p(\{3,5,6,9\}) = 3/23$, $p(\{6,7,8\}) = 1/23$, $p(\{6,8,9\}) = 4/23$.

Table 2 represents the absolute probabilities of transitions between global states. It is similar to Table 1, but contains probabilities instead of transition conditions. Table 3 shows the probabilities of transitions between partial states of the parallel automaton under consideration. As it is said above, the probability of the transition between partial states q_i and q_j is equal to the sum of the probabilities of the transitions between those global states, where there is exchange of q_i and q_j, because the transitions between global states of a parallel automaton are incompatible events. For example, the automaton goes from partial state 2 into partial states 5 and 6 when it goes from global state $\{2, 3, 4\}$ to global state $\{3, 5, 6, 7\}$ or $\{3, 5, 6, 9\}$. Then the probabilities of transitions from 2 to 5 and from 2 to 6 equal $1/23 + 1/23 = 2/23$.

Table 2

States	1	10	$\{2,3,4\}$	$\{3,5,6,7\}$	$\{3,5,6,9\}$	$\{6,7,8\}$	$\{6,8,9\}$
1		2/23					
10			2/23				
$\{2,3,4\}$				1/23	1/23		
$\{3,5,6,7\}$					1/46	1/46	
$\{3,5,6,9\}$							3/46
$\{6,7,8\}$							1/46
$\{6,8,9\}$	2/23						

Table 3

States	1	2	3	4	5	6	7	8	9	10
1										2/23
2					2/23	2/23				
3								2/23		
4							1/23		1/23	
5								2/23		
6	2/23									
7	2/23								1/23	
8	2/23									
9										
10		2/23	2/23	2/23						

The edges of the non-orthogonality graph G of the partial state codes of the automaton correspond to the pairs of non-parallel partial states of the automaton. We weight the edges of G by the numbers proportional to the probabilities of transitions in the corresponding pairs of partial states. As these numbers, we take the numerators of the fractional probabilities with a common denominator.

6. Solving the minimal weighted cover problem

The method for solving the problem of minimal weighted cover is described in [15]. The table of covering graph G under consideration is given by Table 4, where some columns are removed according to the reduction rule: a column a can be excluded from consideration if it has ones everywhere a column b has ones [13]. The table cannot be represented completely because of restricted size of the page.

Table 4

Maximal	Edges of G														Weight
bi-cliques	(1,3)	(1,6)	(1,10)	(2,6)	(3,8)	(3,10)	(4,8)	(4,9)	(5,8)	(6,10)	(7,9)	(7,10)	(8,10)	(9,10)	
B_1	1	1	1												16
B_2			1		1	1	1		1						28
B_3			1			1				1		1	1	1	16
B_4					1		1		1				1		12
B_5		1	1	1			1	1							36
B_6			1					1			1	1			20
B_7								1			1			1	8
B_8			1						1				1		16
B_9									1						8
B_{10}			1								1			1	16
B_{11}											1	1			4
B_{12}		1		1			1	1		1		1	1	1	24
B_{13}			1						1						16
B_{14}			1	1			1			1		1	1	1	18
B_{15}				1			1	1							20
B_{16}									1				1		16
B_{17}			1								1	1			12
B_{18}														1	16
B_{19}	1		1		1										12
B_{20}	1				1	1	1		1						20
B_{21}			1					1			1			1	10
B_{22}								1			1	1			12
B_{23}	1	1				1				1		1	1	1	28

The family of bi-cliques $\{B_{12}, B_{20}, B_{21}\}$ is the solution with the weight 54 for Table 4. To appreciate the quality of the solution, let we do not take into consideration the weights of bi-cliques. In that case the solution would be $\{B_6, B_{12}, B_{20}\}$ with the weight 64. The matrices of coding constructed according to these families, where the rows represent the codes of the partial states, are as follows:

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

The quality of the state assignment problem solution for a sequential automaton can be appreciated by the value $D = \sum p_{ij}(d_{ij} - 1)$, where p_{ij} is the probability of the transition between states s_i and s_j in both directions, d_{ij} is the Hamming distance between the codes of s_i and s_j $(i \neq j)$, and summation is made over all the pairs of states. This value was introduced in [5]. Evidently, the less value of D, the better solution, and $D = 0$ if any transition between states corresponds to switching only one memory element in the circuit implementing the automaton.

In the case of a parallel automaton, such an appreciation of the quality of state assignment can be used considering the global states. The code of a global state can be easily obtained by intersection of ternary vectors coding the partial states that constitute the global one. The result of intersection of non-orthogonal ternary vectors **u** and **v** is the vector **w** obtained in the following way: a component w_i has value "$-$" if both u_i and v_i have this value, and $w_i = 0$ (or 1) if at least one of u_i and v_i has this value. The codes of the global states of the parallel automaton under consideration are represented by the following matrices, where the left matrix is obtained taking into account the weights of bi-cliques, and the right one without taking into account the weights:

$$\begin{array}{c} \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \end{array} \begin{array}{l} \\ 1 \\ 10 \\ \{2,3,4\} \\ \{3,5,6,7\} \\ \{3,5,6,9\} \\ \{6,7,8\} \\ \{6,8,9\} \end{array} , \qquad \begin{array}{c} \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \end{array} \begin{array}{l} \\ 1 \\ 10 \\ \{2,3,4\} \\ \{3,5,6,7\} \\ \{3,5,6,9\} \\ \{6,7,8\} \\ \{6,8,9\} \end{array} .$$

In the first case $D = 1/23$, in the second case $D = 5/23$. This shows the advantage of taking into account weights of bi-cliques. If we take the sequential automaton equivalent to the parallel automaton from Example 1 and apply, for instance, the method for low power state assignment [14], then $D = 3/23$. It allows one to say about utility of the suggested method.

7. Conclusion

The suggested method for low power state assignment of a parallel automaton is intended for using in a computer aided logical design system. Comparison of the results of applying the described method with the results of the partial state assignment without taking into account the switching activity of memory elements shows that the described method gives the better result. It is confirmed in considering 6 parallel automata with the numbers of partial states from 8 to 13, input variables from 2 to 3, and transitions 5, 6, 8, 9 and 13. Moreover, the Boolean function systems obtained after applying the described method for the state assignment turn out to be not worse than ones obtained without taking into account the switching activity of memory elements. This comparison is made in terms of the number of elementary conjunctions in disjunctive normal forms of the functions and the sum of their ranks.

REFERENCES

1. *Muroga S.* VLSI System Design. When and How to Design Very-Large-Scale Integrated Circuits. N.Y., John Wiley & Sons, 1982.
2. *Pedram M.* Power minimization in IC design: Principles and applications. ACM Trans. Design Automat. Electron. Syst., 1996, vol. 6, pp. 3–56.
3. *Kashirova L., Keevallik A., and Meshkov M.* State assignment of finite state machine for decrease of power dissipation. Second Intern. Conf. CAD DD'97, Minsk, Republic of Belarus, November 12–14, 1997. Minsk, NASB, Institute of Engineering Cybernetics, 1997, vol. 1, pp. 60–67.

4. *Sudnitson A.* Partition search for FSM low power synthesis. Fourth Intern. Conf. CAD DD'2001, Minsk, November 14–16, 2001. Minsk, NASB, Institute of Engineering Cybernetics, 2001, vol. 1, pp. 44–49.
5. *Zakrevskiy A. D.* Algoritmy energosberegayushchego kodirovaniya sostoyaniy avtomata [Algorithms for low power state assignment of an automaton]. Informatika, 2011, no. 1(29), pp. 68–78. (in Russian)
6. *Pottosin Yu. V.* Kodirovanie sostoyaniy diskretnogo avtomata, orientirovannoe na umen'shenie energopotrebleniya realizuyushchey skhemy [State assignment in a discrete automaton targeting an implementing low power circuit]. Prikladnaya Diskretnaya Matematika, 2011, no. 4(14), pp. 62–71. (in Russian)
7. *Pottosin Yu. V.* Energosberegayushchee protivogonochnoe kodirovanie sostoyaniy asinkhronnogo avtomata [Low power race-free state assignment of an asynchronous automaton]. Informatika, 2015, no. 2(46), pp. 94–101. (in Russian)
8. *Pottosin Yu.* Race-free state assignment for low power asynchronous automaton. Further Improvements in the Boolean Domain. Ed. B. Steinbach. Cambridge Scholars Publ., 2018, pp. 253–267.
9. *Pottosin Yu.* Optimal state assignment of synchronous parallel automata. Design of Embedded Control Systems. Eds. M. A. Adamski, A. Karatkevich, and M. Wegrzyn. N.Y., Springer, 2005, pp. 111–124.
10. *Zakrevskiy A. D.* Parallel'nye Algoritmy Logicheskogo Upravleniya [Parallel Algorithms for Logical Control]. Moscow, URSS, 2003, 304 p. (in Russian)
11. *Peterson J. L.* Petri Net Theory and the Modeling of Systems. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1981.
12. *Zakrevskiy A. D., Pottosin Yu. V., and Cheremisinova L. D.* Design of Logical Control Devices. Tallinn, TUT Press, 2009.
13. *Zakrevskiy A. D., Pottosin Yu. V., and Cheremisinova L. D.* Combinatorial Algorithms of Discrete Mathematics. Tallinn, TUT Press, 2008.
14. *Pottosin Yu. V.* Kombinatornye Zadachi v Logicheskom Proektirovanii Diskretnykh Ustroystv [Combinatorial Problems in Logical Design of Discrete Devices]. Minsk, Belaruskaja Navuka, 2021, 175 p. (in Russian)
15. *Zakrevskiy A. D.* Optimizatsiya pokrytiy mnozhestv [Optimization of set covering]. Logicheskiy Yazyk dlya Predstavleniya Algoritmov Sinteza Releynykh Ustroystv [Logical Language for Representation of Algorithms for Relay Devices Synthesis]. Ed. M. A. Gavrilov. Moscow, Nauka, 1966, pp. 136–148. (in Russian)