ON USING THE PROBABILISTIC APPROACH TO SYNTHESIS OF LOGICAL CIRCUITS WITH LOW POWER CONSUMPTION

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 state assignment of a discrete automaton aimed to decreasing the switching activity of memory elements in the implemented circuits is considered. To solve this problem the probabilities of transitions between states of the automaton is taken into account.

Keywords: Discrete automaton, state assignment, probability of transition between states.

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 state encoding, i. e. when abstract symbols of states are assigned with Boolean vectors [3 – 6]. The states of an automaton must be encoded in such a way that during the transition between its states, as few as possible memory elements change their state.

In this paper, the approach suggested in [6] is used. It is similar to that of [7], which is used in [3, 4] as well. The method proposed below is named iterative because its fulfilling consists of a sequence of iterations, and at each iteration an intermediate variable is introduced that is a component of the state code of the automaton.

DESCRIPTION OF THE METHOD

One of behavioral models of a discrete device is a finite automaton that consists of a set of inputs A, a set of outputs B, a set of states Q and two functions, output function $\Phi(a, q) = b$ and transition function $\Psi(a, q) = q^+$ where $a \in A$, $b \in B$, $q, q^+ \in Q$ and q^+ is the state, to which the automaton goes from q at the input a.

In the synthesis of a logical circuit, the functions Φ and Ψ are transformed into a system of Boolean functions by introducing Boolean vectors instead of abstract symbols a, b and q. Often, inputs and outputs are presented in the functional description of the synthesized circuit as binary signals. The problem is to assign Boolean vectors $(z_1, z_2, \dots, z_k)$ to abstract symbols q of states of a given automaton according to some optimization criterion. In this case, any state of the automaton will be presented in the circuit as a set of states of binary memory elements where the state of i-th memory element is the value of intermediate variable z_i. A Boolean vector $(z_1, z_2, \dots, z_k)$ assigned to a state of an automaton is called the code of the state. Among numerous variants of that encoding, such a variant must be chosen that guarantee as low switching activity of memory elements as possible.

We consider the probabilities of transitions between states, and the larger probability of transition between any pair of states, the less number of different components in the codes of that states, no matter what is the direction of the transition. The way of calculating probabilities will be described in the next section. Assume that the probabilities of transitions between states are known.

The values of internal variables $z_1, z_2, \dots, z_k$ are specified in the following way.

The current situation in this process is characterized by partial codes of states $(z_1, z_2, \dots, z_j)$, $j < k$, and a weighted graph $G = (V, E)$ whose vertices correspond to the states of the automaton. Two vertices of the graph are adjacent if the corresponded states have the same partial codes. Each edge $v_s v_t \in E$ is weighted with $w_{st} = 1 - p^*_{st}$ where p^*_{st} is the probability of the transition between the states q_s and q_t corresponding to vertices v_s and v_t, no matter in which direction, i. e. $p^*_{st} = p_{st} + p_{ts}$ where p_{st} is the probability of the transition from q_s to q_t. Evidently, to lower the switching activity of memory elements, the short Hamming distance between the codes of states q_s and q_t should be made if the probability p^*_{st} is high.

The process of state assignment of a given automaton is a sequence of steps. At the j-th step, a partition of the vertex set V of G into two subsets, A and B, is obtained, the variable z_j is introduced and receives the value 0 (or 1) for the states corresponding to vertices in A and value 1 (or 0) for the states corresponding to vertices in B. Then, the edges connecting vertices in A with vertices in B are removed, and the next step, $(j + 1)$-th one is fulfilled. The process is over when graph G becomes empty.

The problem to partition V into A and B is reduced to finding the maximum cut of G, i. e. finding such a partition that the sum of weights of the edges connecting the vertices of A with the vertices of B would be maximum. At the last step, the graph G is bipartite, and its edges connect the vertices corresponding to the states, the probabilities of transition between which are comparatively high. So, the Hamming distance between the codes of those states will be equal to one.

How to solve that problem of partition exactly is described in [8]. But because of heuristic approach to the state assignment problem considered here, it is enough to use the "greedy" algorithm from [9] that is a sequence of steps, at each of which a vertex v is selected in B and carried to A. The initial meanings are $A = \varnothing$ and $B = V$, and the vertex v is selected in the following way.

Let d be the sum of weights of the edges incident with v and c the sum of weights of edges connecting v with the vertices of A. The transfer of the vertex v from B to A changes the sum of weights of edges connecting the vertices of A with the vertices of B by $d - 2c$. At the first step, this value is equal to d. At the subsequent steps it can be negative. At any step, the vertex v with maximum value of $d - 2c$ is selected. The process comes to the end when this value is not positive for all the vertices in B.

CALCULATION OF TRANSITION PROBABILITIES

To calculate the probabilities of transitions between states of an automaton, the following assumption are accepted: the automaton must be completely specified; all the states are mutually accessible, i. e. for any two states there exists an input sequence, which transfers the automaton from one of them to the other. The automaton is supposed to be working infinitely long.

The probability of transition from state q_i to state q_j caused by input a is equal to the probability of input a. If there are several inputs causing the transition from state q_i to state q_j, the conditional probability p'_{ij} of this transition is equal to the sum of those probabilities. The condition is that the automaton is at the set q_i. The absolute probability p_{ij} of transition from state q_i to state q_j during all the time the automaton working is equal to $P_i p'_{ij}$ where P_i is the probability of the automaton being at the state q_i.

To calculate the probabilities P_i, $i = 1, 2, \ldots, |Q|$, the Chapmann – Kolmogorov equations for discrete-time Markov Chains [10] can be used. This method is used also in [5]. Similarly to Kirchhoff's low of electrical engineering, one may say that the sum of the transition probabilities to some state is equal to the sum of the transition probabilities from this state. Based on the considerations above, the following equations can be derived:

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

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

The probabilities p'_{ij} must be known. So, having solved this system of equations the probabilities P_i will be obtained. As it was said above, the absolute probability p_{ij} is defined as $p_{ij} = P_i p'_{ij}$.

EXAMPLE

The model of automaton with abstract state [11] is used here, which is described by one multi-valued variable q and many Boolean input and output variables, $x_1, x_2, \ldots, x_n$ and $y_1, y_2, \ldots, y_m$ instead of a and b, respectively. Let the transitions between states of a given automaton be shown in Table 1 where rows and columns correspond to the states of the automaton, and the entry at i-th row and j-th column is the condition of transition from q_i to q_j.

Table 1

Conditions of transitions between states

	q_1	q_2	q_3	q_4	q_5	q_6
q_1			– – – 0			– – – 1
q_2			1 1 – 1	0 – – –	1 – – 0	1 0 – 1
q_3				– – 1 –		– – 0 –
q_4	– 0 0 1		– 1 0 1			– – 0 0 – – 1 –
q_5			1 – 0 1	0 – 0 1	– – 0 0	– – 1 –
q_6		1 1 – 1	1 1 – 0	0 – – –	1 0 – –	

The automaton has six states and four input variables. The output variables are not taken into consideration in this task. For example, the entry at the row q_2 and column q_3 shows that the automaton goes from q_2 to q_3 when $x_1 = 1$, $x_2 = 1$, $x_4 = 1$ and no matter what is x_3 equal to. An empty entry means that the corresponding transition does not exist.

Assume that the probabilities of input signals are distributed uniformly. Then the conditional probabilities p'_{ij} of transitions (the transition from q_i to q_j when the automaton is at the state q_i) are represented in Table 2.

Table 2

Conditional probabilities of transitions

	q_1	q_2	q_3	q_4	q_5	q_6
q_1	0	0	0.5	0	0	0.5
q_2	0	0	0.125	0.5	0.25	0.125
q_3	0	0	0	0.5	0	0.5
q_4	0.125	0	0.125	0	0	0.75
q_5	0	0	0.125	0.125	0.25	0.5
q_6	0	0.125	0.125	0.5	0.25	0

To find out the probabilities of states of the automaton, the following equation system must be solved:

$$0.125\, P_4 = P_1;$$
$$0.125\, P_6 = P_2;$$
$$0.5\, P_1 + 0.125\, P_2 + 0.125\, P_4 + 0.125\, P_5 + 0.125\, P_6 = P_3;$$
$$0.5\, P_2 + 0.5\, P_3 + 0.125\, P_5 + 0.5\, P_6 = P_4;$$
$$0.25\, P_2 + 0.25\, P_5 + 0.25\, P_6 = P_5;$$
$$0.5\, P_1 + 0.125\, P_2 + 0.5\, P_3 + 0.75\, P_4 + 0.5\, P_5 = P_6;$$
$$P_1 + P_2 + P_3 + P_4 + P_5 + P_6 = 1.$$

Having solved these equations with allowed accuracy we obtain $P_1 = 0.0351$, $P_2 = 0.0469$, $P_3 = 0.1230$, $P_4 = 0.2808$, $P_5 = 0.1507$ and $P_6 = 0.3752$. The absolute probabilities p_{ij} of transitions are shown in Table 3.

Table 3

Absolute probabilities of transitions

	q_1	q_2	q_3	q_4	q_5	q_6
q_1	0	0	0.0175	0	0	0.0175
q_2	0	0	0.0059	0.0234	0.0117	0.0059
q_3	0	0	0	0.0615	0	0.0615
q_4	0.0351	0	0.0351	0	0	0.2106
q_5	0	0	0.0188	0.0188	0.0377	0.0753
q_6	0	0.0469	0.0469	0.1876	0.0938	0

The edge weights $w_{st} = 1 - p^*_{st}$ of graph G where p^*_{st} is the probability of the transition in both directions between the states q_s and q_t respective to vertices v_s and v_t are shown in Table 4.

Table 4

Edge weights of graph *G*

	v_2	v_3	v_4	v_5	v_6
v_1	1	0.9825	0.9649	1	0.9825
v_2		0.9941	0.9766	0.9883	0.9472
v_3			0.9034	0.9812	0.8916
v_4				0.9812	0.6018
v_5					0.8309

In the description of the further process of state assignment of the given automaton, the following designations are used:

$d(v_i)$ is the sum of weights of the edges incident with vertex v_i;

$h_j(v_i) = d(v_i) - 2c_j(v_i)$ where $c_j(v_i)$ is the sum of weights of the edges connecting $v_i \in B$ with the vertices of the set A at the beginning of the j-th step of partitioning the set V;

$z_r(q_s)$ is the value of the r-th component of the code of the state q_s.

According to Table 4 we have $d(v_1) = 4.9299$, $d(v_2) = 4.9062$, $d(v_3) = 4.7528$, $d(v_4) = 4.4279$, $d(v_5) = 4.5525$ and $d(v_6) = 4.254$. As $A = \varnothing$ at the beginning of the first step of partitioning the set V, $c_1(v_i) = 0$ and $h_1(v_i) = d(v_i) - 2c_1(v_i) = d(v_i)$ for any $v_i \in B$. The maximum is $d(v_1) = 4.9299$ and the result of the first step is $A = \{v_1\}$ and $B = \{v_2, v_3, v_4, v_5, v_6\}$.

At the second step we have $h_2(v_2) = 2.9062$, $h_2(v_3) = 2.7878$, $h_2(v_4) = 2.4981$, $h_2(v_5) = 2.5525$ and $h_2(v_6) = 2.289$. The maximum is $h_2(v_2) = 2.9062$ and then $A = \{v_1, v_2\}$ and $B = \{v_3, v_4, v_5, v_6\}$.

At the third step we have $h_3(v_3) = 0.7996$, $h_3(v_4) = 0.5449$, $h_3(v_5) = 0.5759$ and $h_3(v_6) = 0.3946$. The maximum is $h_3(v_3) = 0.7996$ and then $A = \{v_1, v_2, v_3\}$ and $B = \{v_4, v_5, v_6\}$.

The third step is the final one, because all the values of $h_j(v_i)$ are negative after it.

The internal variable z_1 is introduced with values $z_1(q_1) = z_1(q_2) = z_1(q_3) = 0$ and $z_1(q_4) = z_1(q_5) = z_1(q_6) = 1$. After deleting the edges connecting the vertices of A with the vertices of B, Table 4 is transformed into Table 5.

Table 5

Edge weights of graph *G*

	v_2	v_3	v_4	v_5	v_6
v_1	1	0.9825			
v_2		0.9941			
v_3					
v_4				0.9812	0.6018
v_5					0.8309

According to Table 5 we have $d(v_1) = 1.9825$, $d(v_2) = 1.9941$, $d(v_3) = 1.9766$, $d(v_4) = 1.583$, $d(v_5) = 1.8121$ and $d(v_6) = 1.4327$. The maximum is $d(v_2) = 1.9941$, and the result of the first step of partitioning the set V is $A = \{v_2\}$ and $B = \{v_1, v_3, v_4, v_5, v_6\}$.

Then, the vertex v_5 is transferred from B to A because of the values $h_2(v_1) = -0.0175$, $h_2(v_3) = -0.0116$, $h_2(v_4) = 1.583$, $h_2(v_5) = 1.8121$ and $h_2(v_6) = 1.4327$. All the values of $h_j(v_i)$ becomes negative after this action, and the result of partitioning the set V is $A = \{v_2, v_5\}$ and $B = \{v_1, v_3, v_4, v_6\}$. The internal variable z_2 is introduced with values $z_2(q_2) = z_2(q_5) = 1$ and $z_2(q_1) = z_2(q_3) = z_2(q_4) = z_2(q_6) = 0$.

After deleting the edges between A and B, only two edges, v_1v_3 and v_4v_6, with the weights 0.9825 and 0.6018, respectively, remain in G, and the set V is partitioned into $A = \{v_1, v_4\}$ and $B = \{v_2, v_3, v_5, v_6\}$. As a result of the process of state assignment is the following encoding of the states: q_1 – 000, q_2 – 011, q_3 – 001, q_4 – 100, q_5 – 111, q_6 – 101.

The quality of a solution of the state assignment problem can be appreciated by the value of $D = \sum p^*_{ij}(d_{ij} - 1)$ where p^*_{ij} is the probability of the transition between states q_i and q_j in both directions, d_{ij} is the Hamming distance between the codes of q_i and q_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.

For the variant of encoding obtained above we have $D = 0.2513$. If we take arbitrary encoding, i. e. assign to the states q_1, q_2, q_3, q_4, q_5, q_6 the sequence of natural numbers with zero in binary system – 000, 001, 010, 011, 100, 101 – then we obtain $D = 0.7944$. It is clear that the first variant is better then the second one.

REFERENCES

1. *Muroga, S.* VLSI System Design. When and How to Design Very-Large-Scale Integrated Circuits / S. Muroga. New York : John Viley & Sons, Inc., 1982.

2. *Pedram, M.* Power minimization in IC design: Principles and applications / M. Pedram // ACM Trans. Design Automat. Electron. Syst. 1996. V. 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 International Conference Computer-Aided Design of Discrete Devices, CAD DD'97, Minsk, Republic of Belarus, November 12–14, 1997. V. 1. Minsk : National Academy of Sciences of Belarus, Institute of Engineering Cybernetics, 1997. P. 60–67.

4. *Sudnitson, A.* Partition search for FSM low power synthesis / A. Sudnitson // Fourth International Conference Computer-Aided Design of Discrete Devices, CAD DD'2001, Minsk, November 14–16, 2001. V. 1. Minsk : National Academy of Sciences of Belarus, Institute of Engineering Cybernetics, 2001.P. 44–49.

5. *Zakrevskij, A. D.* Algorithms for low power state assignment of an automaton / A. D. Zakrevskij // Informatika. 2011. No. 1(29). P. 68–78 (in Russian).

6. *Pottosin, Yu. V.* An iterative method for low power state assignment of a discrete automaton / Yu. V. Pottosin // Informatika. 2012. No. 4(36). P. 93–99 (in Russian).

7. *Hartmanis, J.* Algebraic Structure Theory of Sequential Machines / J. Hartmanis, R. E. Stearns. N.-Y. : Prentis-Hall Inc., 1966. 208 p.

8. *Pottosin, Yu. V.* Extracting a maximal bipartite part in an undirected graph / Yu. V. Pottosin // Automation of Solving Logical Combinatorial Problems. Minsk : ITC NAS Belarus, 1985. P. 22–28 (in Russian).

9. *Zakrevskij, A. D.* Graph coloring in decomposition of Boolean functions / A. D. Zakrevskij // Logical Design. Minsk : ITC NAS Belarus, 2000. Issue 5. P. 83–97 (in Russian).

10. *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. V. 17. No. 11. P. 1061–1079.

11. *Zakrevskij, A.* Design of Logical Control Devices / A. Zakrevskij, Yu. Pottosin, L. Cheremisinova. Tallinn : TUT Press, 2009. 304 p.