Alexander BARKALOV, Larysa TITARENKO, **Sławomir CHMIELEWSKI**
UNIWERSYTET ZIELONOGÓRSKI, INSTYTUT INFORMATYKI I ELEKTRONIKI, ul. licealna 9, 65-417 Zielona Góra

Synthesis of Finite State Machines with using pseudoequivalent states

prof. zw. dr hab. inż. Alexander BARKALOV

Prof. Alexander A. Barkalov w latach 1976 - 1996 był pracownikiem dydaktycznym w Instytucie Informatyki Narodowej Politechniki Donieckiej. Współpracował aktywno z Instytutem Cybernetyki im. V. M. Glushkova w Kijowie, gdzie uzyskał tytuł doktora habilitowanego ze specjalnością informatyka. W latach 1996 - 2003 pracował jako profesor w Instytucie Informatyki Narodowej Politechniki Donieckiej. Od 2004 pracuje jako profesor na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego.

e-mail: A.Barkalov@iie.uz.zgora.pl

mgr inż. Sławomir CHMIELEWSKI

Mgr inż. Sławomir Chmielewski absolwent Uniwersytetu Zielonogórskiego. Ukończył studia informatyczne o specjalizacji inżynieria komputerowa. Zainteresowania naukowe koncentrują się wokół nowoczesnych metod projektowania specjalistycznych układów cyfrowych..

e-mail: S.Chmielewski@weit.uz.zgora.pl

Abstract

A new two-stage method of FSMs synthesis for PAL-based CPLD is proposed. It is based on both the wide fan-in of PAL cells and existence of the classes of pseudoequivalent states of Moore FSM. The first step targets decreasing for the number of PAL cells used for implementing the input memory functions. The second step targets decreasing for the number of PAL cells in the block of microoperations. An example of application of the proposed method is given, as well as, results of experiments carried out for standard benchmarks.

Keywords: Moore FSM, Logic Synthesis, State assignment, CPLD

Synteza skończonych automatów stanów z wykorzystaniem pseudorównoważnych stanów

Streszczenie

W artykule przedstawiono metody syntezy mikroprogramowalnego układu sterującego z użyciem wbudowanych bloków pamięci. Metody są ukierunkowane na zmniejszenie rozmiaru układu sterującego poprzez zastosowanie transformacji kodów klas pseudorównoważnych w pamięci. Podejście takie pozwala uzyskać uproszczoną formę funkcji przejścia części adresowej układu, dzięki któremu możliwa jest redukcja zasobów sprzętowych potrzebnych do implementacji jednostki sterującej w układach programowalnych typu CPLD bez zmniejszenia wydajności systemu cyfrowego. W artykule zamieszono wprowadzenie teoretyczne, przykład oraz wyniki badań uzyskanych podczas syntezy testowych sieci działań.

Słowa kluczowe: automat Moore'a, Synteza logiczna, CPLD

1. Introduction

The overwhelming majority of digital systems include a control unit responsible for interplay of all other systems' blocks [1]. In many practical cases, the model of Moore finite state machine (FSM) is used for representing a control unit [2]. Nowadays, two families of programmable logic devices are used mostly for implementing logic circuits of FSMs: the field programmable gate arrays (FPGA) and the complex programmable logic devices (CPLD) [3]. As a rule, FPGAs are used for implementing rather complex FSMs, whereas CPLDs target fast FSMs [4,5].

The majority of CPLD are based on PAL cells connected with programmable flip-flops [6,7]. Each cell can be viewed as q s-input AND gates connected with an OR gate. The general model of the PAL – based cell is shown, for example, in [5]. The number q determines the upper limit for product terms to be implemented by a cell. This value is rather small, and $q \le 6$ in the most common cases [6,7]. Obviously, Boolean functions having more than q terms in their sum-of-products (SOP) forms need more than one cell for implementation. In the case of functions with more than q terms, the decomposition methods should be applied [8,9,10]. It leads to increasing for the number of circuit layers and, therefore, for the total delay of the resulting circuit's performance. To avoid this negative effect, it is important to decrease the numbers of terms in SOP of Boolean functions representing an FSM circuit. The next specific of PAL cells is their wide fan-in (the number of cell inputs $S > 20$) [6,7].

dr hab. inż. Larysa TITARENKO, prof. UZ

Dr hab. Larysa Titarenko w 2004 roku obroniła rozprawę habilitacyjną i uzyskała tytuł doktora habilitowanego ze specjalnością telekomunikacja. W latach 2004 - 2005 pracowała jako profesor w Narodowym Uniwersytecie Radioelektroniki w Charkowie. Od 2007 pracuje jako profesor na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego.

e-mail: L.Titarenko@iie.uz.zgora.pl

2. Theoretical background

Let the control algorithm of a digital system be specified by a GSA $\Gamma = (B, E)$, where $B = \{b_0, b_E\} \cup E_1 \cup E_2$ is a set of the vertices and E is a set of edges. Here b_0 is an initial vertex, b_E is a final vertex, E_1 is a set of operational vertices, and E_2 is a set of conditional vertices. The logic circuit of the Moore FSM U_1 is represented by the following systems of Boolean functions:

$$\Phi = \Phi(T, X), \quad (1)$$

$$Y = Y(T), \quad (2)$$

where $T = \{T_1, ..., T_R\}$ is a set of internal variables encoding the states $a_m \in A$, $\Phi = \{D_1, ..., D_R\}$ is a set of the FSM input memory functions. The value of R is determined as

$$R = \lceil \log_2 M \rceil, \quad (3)$$

where $\lceil a \rceil$ is a minimum integer not less than a. The structure diagram of Moore FSM U_1 is shown in Fig. 1.

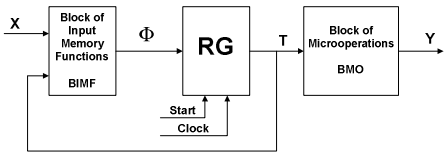

Fig. 1. Structure diagram of Moore FSM U_1
Rys. 1. Schemat blokowy automatu Moore'a FSM U_1

Let us consider a GSA Γ_1 (Fig. 2a) marked by the states of Moore FSM using the rules [1].

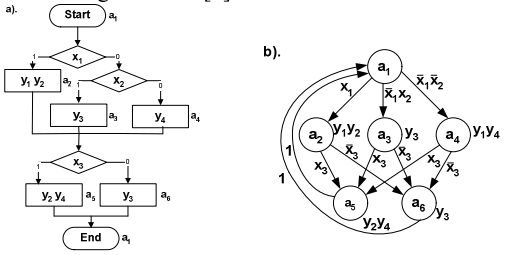

Fig. 2. Initial GSA Γ_1 (a) and corresponding state transition graph (b) of Moore FSM
Rys. 2. Sieć działań Γ_1 (a) graf (b) automatu Moore'a

The Moore FSM $U_1(\Gamma_1)$ has the following characteristics: $A = \{a_1, ..., a_6\}$, $M = 6$, $X = \{x_1, x_2, x_3\}$, $L = 3$, $Y = \{y_1, ..., y_4\}$, and $N = 4$. The FSM $U_1(\Gamma_1)$ can be represented by a state transition graph (STG) shown in Fig. 2b. This STG has $U_1(\Gamma_1) = 11$ edges. Each edge corresponds to one row of the structure table [1].

In turn, each row of the structure table determines up to R product terms in the system (1). So, it is important to diminish the number of rows, $H_1(\Gamma)$. The states $a_m, a_s \in A$ are pseudoequivalent states if identical inputs result in identical next states for both $a_m, a_s \in A$. This is possible if the outputs of the operational vertices marked by these states are connected with the input of the same vertex of the GSA Γ [3]. Let $\Pi_A = \{B_1,...,B_I\}$ be a partition of the set A by the classes of PSs ($I \le M$).

In the case of FSM $U_1(\Gamma_1)$, the partition $\Pi_A = \{B_1, B_2, B_3\}$ can be found, where $B_1 = \{a_1\}$, $B_2 = \{a_2, a_3, a_4\}$, and $B_3 = \{a_5, a_6\}$. Let us replace the states $a_m \in B_i$ by blocks $B_i \in \Pi_A$ having $M_i = |B_i|$ inputs. It allows transforming an initial GSA Γ into a block GSA $B(\Gamma)$. The block GSA $B(\Gamma_1)$ is shown in Fig. 3a. Obviously, the initial STG of Moore FSM can be transformed, too. To transform an STG, the vertices corresponding to $a_m \in B_i$ should be replaced by a single vertex $B_i \in \Pi_A$. Let us name the resulting STG as a block transition graph (BTG) (Fig. 3b). This graph determines a reduced structure table (RST) of Moore FSM.

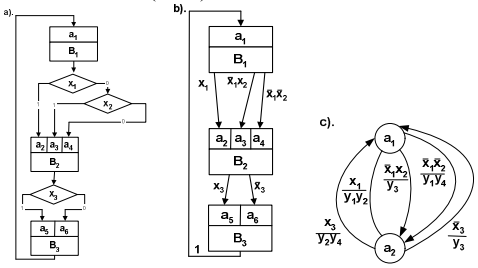

Fig. 3. Block GSA B(Γ1) (a), block transition graph of Moore FSM U1(Γ1) (b) and state transition graph of Mealy FSM U0(Γ1) (c)
Rys. 3. Blok GSA B(Γ1) (a) blok połączeń automatu Moore'a (b) graf automatu Mealy'ego (c)

Let us point out that there are only two states in the Mealy FSM $U_0(\Gamma_1)$ (Fig. 3c). The comparison of graphs Fig. 3b and Fig. 3c allows finding the following relations:

$$\left.\begin{aligned} I &= M_0 + 1, \\ H_2 &= H_0 + 1. \end{aligned}\right\} \qquad (4)$$

In (4), M_0 is the number of states for Mealy FSM $U_0(\Gamma_1)$, whereas H_2 is the number of edges for BTG $B(\Gamma)$.

Therefore, the replacement of internal states of Moore FSM by the classes of PS leads to diminishing for the number of product terms in the system (1) up to $H_2 \approx H_0$. It can result in decreasing for the number of PAL cells in the logic circuit of BIMF.

Two methods can be used for representing the classes $B_i \in \Pi_A$ [6]: optimal state assignment, transformation of the codes of states into the codes of classes of pseudoequivalent states.

In the first case, the states $a_m \in A$ are encoded in such a way that the codes of the states $a_m \in B_i$ $(i = 1,...,I)$ belong to a single generalized interval of the R-dimensional Boolean space. This leads to a Moore FSM U_2 that has the same structure as the Moore FSM U_1. The algorithm from [11] can be used for such a state assignment. In the case of $U_2(\Gamma_1)$, there are $R = 3$ and $T = \{T_1, T_2, T_3\}$. One of the possible variants of state assignment (optimal encoding) is shown in Fig. 4.

As follows from Fig. 4, the class B_1 is determined by the interval *00, the class B_2 by **1, and the class B_3 by *10. These intervals are treated as the class codes.

T_1 \ T_2T_3	00	01	11	10
0	a_1	a_2	a_3	a_5
1	*	a_4	*	a_6

Fig. 4. Optimal state codes for Moore FSM U2(Γ1)
Rys. 4. Kodowanie stanów automatu Moor'a U2(Γ1)

But such an encoding is not always possible [10]. In the second case, the classes $B_i \in \Pi_A$ are encoded by the binary codes $K(B_i)$ with $R_1 = \lceil \log_2 I \rceil$ bits. The variables $\tau_r \in \tau$ are used for such an encoding, where $|\tau| = R_1$. This approach leads to a Moore FSM U_3 (Fig. 5) including a block of code transformer (BCT).

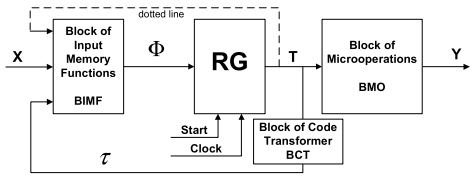

Fig. 5. Structure diagram of Moore FSM U3 without dotted line, U4 with dotted line
Rys. 5. Schemat blokowy automatu U3 bez linii kreskowanej, U4 z linią kreskowaną

In the FSM U_3, the BIMF generates the functions

$$\Phi = \Phi(\tau, X) \qquad (5)$$

and the block of code transformer implements the functions

$$\tau = \tau(T). \qquad (6)$$

The number of transitions of Moore finite-state-machine U_3 is equal to $H_0(\Gamma)$. The drawback of FSM U_3 is the existence of BCT that consumes additional resources of a chip (in comparison with U_1).

3. Conclusion

In this article, the design method is proposed targeting Moore FSM logic circuit and PAL-based CPLD chips. The proposed method is based on the wide fan-in of PAL cells, as well as on the existence of pseudoequivalent states of Moore FSM. The main idea of proposed method is using two sources of codes of the classes of pseudoequivalent states.

Our strategy is oriented towards the area optimization. The positive back effect of the proposed method is possible decrease for the number of layers of PAL cells in the FSM logic circuits. But we did not research this property of the proposed method. The method includes two stages. First of all, the optimal state assignment is executed based of JEDI approach. Next, the codes are rearranged to optimize the number of cells in the circuit implementing the system of microoperations. We called the second stage as "the refined state assignment".

4. Literature

[1] Anderson J., Brown S., Technology mapping for large complex PLDs, In Proceedings of Design Automation Conference, 1998, pp. 698-703.
[2] Baranov S., Logic Synthesis for Control Automata, Boston, Kluwer Academic Publishers, 1994, pp. 405.
[3] Barkalov A., Titarenko L., Logic Synthesis for FSM-Based Control Units, Berlin, Springer, 2009, pp. 234.
[4] Kania D., The logic synthesis for the PAL-based complex programmable logic devices, Gliwice, Silesian University of Technology 2004, pp. 212. (in Polish).
[5] Kania D., Czerwinski R., Area and speed oriented synthesis of FSMs for PAL-based CPLDs, Microprocessors and Microsystems, 2012, pp. 45-61, V.36, N:1.
[6] Czerwinski R., Kania D., Synthesis of finite state machines for CPLDs, International Journal of Applied Mathematics and Computer Science, 2009, pp. 647-659,V. 19, N:4.
[7] Kania D., Milek A., Logic Synthesis Based on Decomposition for CPLDs, Microprocessors and Microsystems, 2010, pp. 25-38, V.34.
[8] Barkalov A., Titarenko L., Chmielewski S., Reduction in the number of PAL macrocells in the circuit of a Moore FSM, International Journal of Applied Mathematics and Computer Science, 2007, pp. 565-575, V.17, N:4.
[9] www.altera.com.
[10] Barkalov A., Principles of logic optimization for Moore microprogrammed automaton, Cybernetics and Systems Analysis, 1998, pp. 54-63, V. 34, N:1.
[11] www.xilinx.com.