

Alexander BARKALOV, Larysa TITARENKO, Sławomir CHMIELEWSKI

UNIwersytet Zielonogórski, Instytut Informatyki i Elektroniki

Decrease of the number of PAL macrocells for Moore FSM

prof. dr hab. inż. Alexander BARKALOV, prof. UZ

Prof. Alexander A. Barkalov w latach 1976-1996 był pracownikiem dydaktycznym w Instytucie Informatyki Narodowej Politechniki Donieckiej. Współpracował aktywnie 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



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



mgr inż. Sławomir CHMIELEWSKI

Mgr inż. Sławomir Chmielewski absolwent Uniwersytetu Zielonogórskiego. Ukończył studia informatyczne o specjalizacji inżynieria komputerowa. Obecnie student trzeciego roku na studiach doktoranckich.

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



of pseudoequivalent states and regular character of system of microoperations that permits its implementation using EMBs [12, 13]. In this article we propose the method of optimization of the amount of PAL macrocells in the logic circuit of Moore FSM using the above mentioned considerations.

2. Background of Moore FSM

Let the behavior of digital system be specified by algorithmic state machine (ASM) $\Gamma = (B, E)$, where $B = \{b_0, b_E\} \cup E_1 \cup E_2$ is set of vertices and E is the set of arcs [1]. Here b_0 is an initial vertex, b_E is a final vertex, E_1 is the set of operational vertices, E_2 is the set of conditional vertices. The vertex $b_q \in E_1$ contains a set of microoperations $Y(b_q) \subseteq Y$, where $Y = \{y_1, \dots, y_N\}$ is the set of microoperations implemented by the data-path of the digital system [3]. Vertex $b_q \in E_2$ contains logic condition $x_e \in X$, where $X = \{x_1, \dots, x_L\}$ is the set of logic conditions (flags) [4]. The initial and final vertices of ASM correspond to initial state $a_1 \in A$, where $A = \{a_1, \dots, a_M\}$ is the set of states of Moore FSM. Each operational vertex $b_q \in E_1$ corresponds to the unique state $a_m \in A$ and collection $Y(b_q) = Y(a_m)$. The logic circuit of Moore FSM U_1 is specified by the following systems of Boolean functions:

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

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

Here $T = \{T_1, \dots, T_R\}$ is the set of state variables encoding the internal states $a_m \in A$, $R = \lceil \log_2 M \rceil$; $\Phi = \{D_1, \dots, D_R\}$ is the set of input memory functions. The systems (1) – (2) are formed on the base of direct structure table (DST) [1] with columns: a_m is a current state of FSM; $K(a_m)$ is a code of state a_m ; a_s is the next state of FSM; $K(a_s)$ is a code of state a_s ; X_h is a product of input variables – some elements of the set X (or their complements) determining the transition $\langle a_m, a_s \rangle$; Φ_h is the set of input memory functions equal to 1 to switch the memory from $K(a_m)$ into $K(a_s)$; $h = 1, \dots, H_1(\Gamma)$ is a number of line.

The column a_m contains collection of microoperations $Y(a_m) \subseteq Y$, equal to one in the state $a_m \in A$. It is clear that $Y(a_m) = Y(b_q)$, where a vertex $b_q \in E_1$ is marked by an internal state $a_m \in A$.

As a rule, the number of transitions $H_1(\Gamma)$ exceeds the number of transitions $H_2(\Gamma)$ in the equivalent Mealy FSM [5]. It can increase the number of PAL macrocells in the circuit of Moore FSM as to equivalent Mealy FSM. The value $H_1(\Gamma)$ can be decreased by taking into account the pseudoequivalent states (PES) of Moore FSM [4]. The states $a_m, a_s \in A$ are PES, if outputs of corresponding operational vertices are connected to the input of the same vertex of ASM Γ . Let $\prod_A = \{B_1, \dots, B_I\}$ be the partition of set A by the classes of PES ($I \leq M$). Let us encode each class $B_i \in \prod_A$ by binary code $K(B_i)$ with $R_1 = \lceil \log_2 I \rceil$ bits and let use the variables

Abstract

Method of decrease in the number of PAL macrocells in logic circuit of Moore FSM is proposed. This method is based on the implementation of free outputs of embedded memory blocks to represent the code of the class of the pseudoequivalent states. The proposed approach allows minimizing hardware without decreasing of the digital system performance. An example of application of the proposed method is given.

Keywords: Moore finite-state-machine, PAL macrocells, CPLD, embedded memory blocks, algorithmic state machine

Zmniejszenie zużycia makrokomórek PAL w automatach Moore'a

Streszczenie

W pracy przedstawiona została metoda zmniejszania zużycia makrokomórek w układach typu PAL przy pomocy automatów Moore'a FSM. Metoda ta bazuje na wyznaczeniu odpowiednich stanów i ich przekształceniu. Zaproponowane podejście pozwala zmniejszyć wykorzystanie zużycia sprzętowego bez zmniejszenia wydajności systemów cyfrowych. Podany również jest przykład aplikacji zaproponowanego rozwiązania.

Słowa kluczowe: automat Moore'a, PAL makro-komórka, CPLD, wbudowany blok pamięci, algorytmiczna maszyna stanów

1. Introduction

A control unit (CU) is a very important block of any digital system, its function is the coordination of other blocks interplay [1, 2, 3]. In many cases, a Moore finite-state-machine (FSM) is used to represent the CU [4, 5, 6]. The current state of electronics permits to implement a complex digital system on a single chip – “system-on-a-chip” (SoC) [7, 8]. An arbitrary logic of a digital system can be constructed using PAL (programmable array logic) macrocells of SoC, if they used CPLD (complex programmable logic devices) approach [3]. The tabular functions can be implemented with embedded memory blocks (EMB) of the SoC. One of important problems is the decrease of the chip area of CU [7-11]. The peculiarities of both PAL macrocells and the model of CU should be taking into account to solve this problem [12,13]. The peculiarities of PAL are a wide fan-in of macrocells and a very limited number of terms per macrocell The specific features of Moore FSM are the existence

$\tau_r \in \tau$ for this encoding, where $|\tau| = R_1$. In this case, Moore FSM can be presented as structure U_1 in Fig. 1.

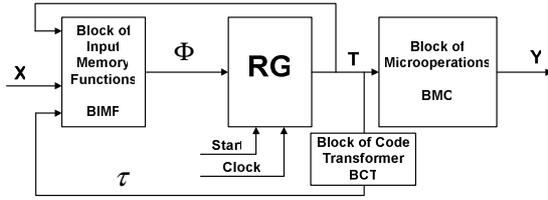


Fig. 1. Structural diagram of Moore FSM U_1
Rys. 1. Struktura diagramu Moore'a FSM U_1

In the Moore FSM U_1 circuit BIMF realizes the input memory functions

$$\Phi = \Phi(\tau, X), \quad (3)$$

and circuit BMO implements the system (2). The register RG represents the memory of the states. The pulse Start is used to set up FSM into initial state $a_1 \in A$. The content of RG is changed by pulse Clock. The code transformer BCT implements the system

$$\tau = \tau(T), \quad (4)$$

and code $K(B_i)$ is formed on the base of the code $K(a_m)$, where $a_m \in B_i$. It is shown in the work [4] that the number of transitions of Moore FSM U_1 is reduced up to $H_2(\Gamma)$. The drawback of U_1 is the existence of BCT that consumes additional PAL macrocells or EMBs. In our article we propose the method for design of Moore FSM permitting to decrease of the logic circuit in the block BCT. Sometimes this block can be even eliminated. The proposed method is based on the following features of CPLD technology [2, 8, 10]:

- the fan-in of PAL macrocells significantly exceeds the maximal number of literals in the terms of the system (1);
- the number of outputs of EMB can be chosen from some restricted area $\{1, 2, 4, 8\}$.

3. Main idea of proposed method

Let us use the idea of optimal encoding of the states of Moore FSM [5]. In this case the states are encoded in such a manner where maximal possible number of classes $B_i \in \Pi_A$ corresponds to the unique single interval of R -dimensional Boolean space. Let $\Pi_A = \Pi_B \cup \Pi_C$, where $B_i \in \Pi_A$, if

$$|B_i| > 1. \quad (5)$$

If condition (5) is violated, then $B_i \in \Pi_C$. It is clear that circuit BCT should form only the codes $K(B_i)$, where $B_i \in \Pi_B$. Let us encode the states $a_m \in A$ in the optimal way [5] and let $\Pi_B = \Pi_D \cup \Pi_E$. Here $B_i \in \Pi_D$, if the codes of states $a_m \in B_i$ belong to the single generalized interval of Boolean space. Now only codes of states $a_m \in A(\Pi_E)$ should be transformed, where $A(\Pi_j)$ is a set of states $a_m \in B_i$, where $B_i \in \Pi_j$ ($j = A, B, C, D, E$). It is enough R_2 binary variables to encode classes $B_i \in \Pi_E$:

$$R_2 = \lceil \log_2 (|\Pi_E| + 1) \rceil. \quad (6)$$

Let these variables form the set Z , where $|Z| = R_2$, and t_F is a fixed number of outputs of the EMB block and let q is the amount of its words, if $t_F = 1$. The value t_F for FSM U_1 is determined as

$$t_F = \lceil q/M \rceil. \quad (7)$$

The total amount of the outputs t_S of all EMBs in the circuit BMO is determined as

$$t_S = \lceil N/t_F \rceil * t_F. \quad (8)$$

In this case

$$\Delta_r = t_S - N \quad (9)$$

outputs can be used to represent the variables $z_r \in Z$. If

$$\Delta_r \geq R_2, \quad (10)$$

then ASM Γ can be interpreted by proposed Moore FSM U_2 (Fig. 2). In this structure, circuit BIMF implements functions

$$\Phi = \Phi(T, Z, X), \quad (11)$$

and circuit BMO implements functions (2) and functions

$$Z = Z(T). \quad (12)$$

In FSM U_2 the block BCT is absent and variables $T_r \in T$ represent the states $a_m \in A(\Pi_C)$ and the classes $B_i \in \Pi_D$. The classes $B_i \in \Pi_E$ are represented by circuit BMO.

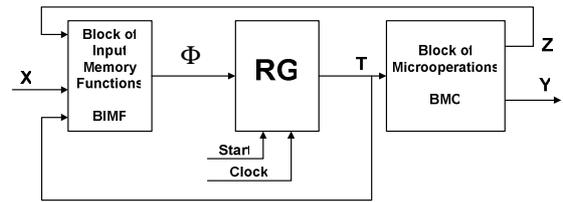


Fig. 2. Structural diagram of Moore FSM U_2
Rys. 2. Struktura diagramu Moore'a FSM U_2

In this case, the number of inputs in the PAL macrocells is increased from $L + R_1$ (FSM U_1) to $L + R + R_2$ (FSM U_2), but it does not increase the circuit BIMF as compare to FSM U_1 . The cycle time of both U_1 and U_2 is the same even in the worst case. In the best case, the circuit BIMF of U_2 has less amount of levels, than circuit BIMF of U_1 . It means that the delay of U_2 can be less than the delay of U_1 . Therefore, the proposed approach permits to decrease the hardware amount without decrease of performance of digital system.

The proposed method of design of Moore FSM U_2 includes the following steps:

1. Construction of marked ASM Γ .
2. Construction of partition $\Pi_A = \Pi_B \cup \Pi_C$.
3. Optimal state encoding and construction of sets Π_D and Π_E .
4. Encoding of classes $B_i \in \Pi_E$.
5. Construction of table for circuit BMO.
6. Construction of modified DST of U_2 .
7. Implementation of the logic circuit of FSM U_2 .

4. Example of proposed method application

Let for some marked ASM Γ_1 we have a Moore FSM with the set of states $A = \{a_1, \dots, a_{15}\}$ and let we have constructed the partition $\Pi_A = \{B_1, \dots, B_8\}$, where $B_1 = \{a_1\}$, $B_2 = \{a_2, a_3, a_4\}$, $B_3 = \{a_5, a_6\}$, $B_4 = \{a_7, a_8, a_9\}$, $B_5 = \{a_{10}, a_{11}, a_{12}\}$, $B_6 = \{a_{13}\}$, $B_7 = \{a_{14}\}$, $B_8 = \{a_{15}\}$. Therefore, we have $\Pi_B = \{B_2, \dots, B_5\}$ and $\Pi_C = \{B_1, B_6, B_7, B_8\}$. Let us use the method of optimal state encoding from [4, 5]. Such encoding in our example for four is shown in the Fig. 3.

$T_3 T_4$	00	01	11	10
00	a_1	a_5	a_6	a_7
01	a_2	a_3	a_4	*
11	a_{10}	a_{11}	a_{12}	a_8
10	a_{13}	a_{14}	a_{15}	a_9

Fig. 3. Optimal state encoding for Moore FSM $U_2(\Gamma_1)$
Rys. 3. Stany automatu Moore'a FSM $U_2(\Gamma_1)$

As before, $U_i(\Gamma_j)$ means that FSM U_i implements ASM Γ_j . For FSM $U_2(\Gamma_1)$, we have the set of state variables $T = \{T_1, \dots, T_4\}$. It is clear from Fig. 3, that $\Pi_D = \{B_2, B_3, B_4\}$ and $\Pi_E = \{B_5\}$. The classes $B_i \in \Pi_A$ have the following codes: $K(B_1) = 0000$, $K(B_2) = 01**$, $K(B_3) = 00*1$, $K(B_4) = **10$, $K(B_5) = 1000$, $K(B_6) = 1001$, $K(B_7) = 1011$. Thus, we have $|\Pi_E| = 1$, $R_2 = 1$ and $Z = \{z_1\}$. Let ASM Γ_1 includes $N = 15$ microoperations and let us use the EMB with $t_F = 4$ for $q = 16$. In this case, we have $t_s = 4 * 4 = 16$ and $\Delta_s = 1$. Therefore, condition (10) is satisfied and the application of the proposed method has sense. Let $K(B_5) = 1$, in this case the value $z_1 = 0$ means that the current state of FSM $a_m \notin B_5$. Let transitions between the states of Moore FSM $U_2(\Gamma_1)$ be specified by the following system of generalized formulae of transitions [1]:

$$\begin{aligned} B_1 &\rightarrow a_2; B_2 \rightarrow x_1 a_{10} + /x_1 x_2 a_{11} + /x_1 /x_2 a_{12}; \\ B_3 &\rightarrow x_1 a_{13} + /x_1 a_{14}; B_4 \rightarrow x_1 a_5 + /x_1 x_3 a_6 + /x_1 /x_3 a_7; \\ B_5 &\rightarrow x_4 a_2 + /x_4 x_3 a_3 + /x_4 /x_3 a_4; \\ B_6 &\rightarrow x_5 a_8 + /x_5 a_9; B_7 \rightarrow a_{15}; B_8 \rightarrow x_3 a_{10} + /x_3 a_1. \end{aligned} \quad (13)$$

Let microoperations $y_n \in Y$ are distributed among the states of FSM $U_2(\Gamma_1)$ in the following manner: $Y(a_1) = \emptyset$, $Y(a_2) = Y(a_6) = \{y_1, y_3\}$, $Y(a_3) = \{y_2, y_4, y_6\}$, $Y(a_4) = Y(a_8) = Y(a_{12}) = \{y_1, y_7, y_8, y_{15}\}$, $Y(a_5) = \{y_3, y_5, y_9\}$, $Y(a_7) = \{y_{10}, y_{11}\}$, $Y(a_9) = \{y_{10}, y_{12}\}$, $Y(a_{10}) = \{y_1, y_{13}, y_{14}\}$, $Y(a_{11}) = Y(a_{15}) = \{y_4, y_{13}\}$, $Y(a_{13}) = \{y_7, y_9\}$, $Y(a_{14}) = \{y_2, y_{12}\}$.

The table of BMO circuit includes the columns a_m , $K(a_m)$, $Y(a_m)$, $K(B_i)$, m , where $K(a_m)$ is an address of the EMB word. In case of $U_2(\Gamma_1)$, this table is presented in Fig. 4.

$T_3 T_4$	00	01	11	10
00	—	$y_3 y_5 y_9$	$y_1 y_3$	$y_{10} y_{11}$
01	$y_1 y_3$	$y_2 y_4 y_6$	$y_1 y_7 y_8 y_{15}$	*
11	$y_1 y_3 y_4 z_1$	$y_1 y_{13} z_1$	$y_1 y_7 y_8 y_{15} z_1$	$y_1 y_7 y_8 y_{15}$
10	$y_7 y_9$	$y_2 y_{12}$	$y_4 y_{13}$	$y_{10} y_{12}$

Fig. 4. The distribution of microoperations for the BMO circuit of $U_2(\Gamma_1)$
Rys. 4. Tabela mikrooperacji dla układu BMO $U_2(\Gamma_1)$

It is clear from Fig. 4, that variable z_1 is included into the set of microoperations for the states $a_m \in B_5$.

The modified DST (MDST) of FSM U_2 includes the columns B_i , $K(B_i)$, a_s , $K(a_s)$, X_h , Φ_h , h , where code $K(B_i) = \langle Z, T \rangle$. For FSM $U_2(\Gamma_1)$ this table has $H_2(\Gamma_1) = 17$ lines; the value of this parameter is equal to the number of the terms in the system (13). The transitions for the classes $B_2, B_5, B_6 \in \Pi_A$ are shown in the Table I.

The system (11) can be constructed from this table. For example, from Table I, we can get the part of DNF of function D_3 :

$$D_3 = /z_1 /T_1 T_2 /x_1 /x_2 + z_1 /x_4 /x_3 + /z_1 T_1 /T_2 /T_3 /T_4.$$

Let us point out that the number of terms in system (1) of the Moore FSM U_0 (without optimal encoding of the states) is equal to $H_0(\Gamma_1) = 37$.

The implementation of the logic circuit of Moore FSM U_2 is reduced to implementation of the system (11) using PAL macrocells and implementation of the systems (2) and (12) using EMBs. These tasks are well-known and effective methods are existed for their solution [2].

Tab. 1. Fragment of the MDST of Moore FSM $U_2(\Gamma_1)$

Tab. 1. Fragment tabeli MDST Moore'a FSM $U_2(\Gamma_1)$

B_i	$K(B_i)$	a_s	$K(a_s)$	X_h	Φ_h	h
	$z_1 T_1 T_2 T_3 T_4$		$T_1 T_2 T_3 T_4$			
B_2	001**	a_{10}	1100	x_1	$D_1 D_2$	1
		a_{11}	1101	$/x_1 x_2$	$D_1 D_3 D_4$	2
		a_{12}	1111	$/x_1 /x_2$	$D_1 D_2 D_3 D_4$	3
B_5	1****	a_2	0100	x_4	D_2	4
		a_3	0101	$/x_4 x_3$	$D_2 D_4$	5
		a_4	0111	$/x_4 /x_3$	$D_2 D_3 D_4$	6
		a_8	1110	x_5	$D_1 D_2 D_3$	7
B_6	01000	a_9	1010	$/x_5$	$D_1 D_3$	8

5. Conclusion

The proposed method allows to decrease the number of PAL macrocells in the circuit implementing input memory functions of Moore FSM. Our researches showed that this decrease is proportional to coefficient

$$\eta_1 = H_0(\Gamma) / H_2(\Gamma). \quad (14)$$

If condition (10) takes place, then block BCT is eliminated from the circuit of FSM. It leads to decrease of number of EMBs in the circuit of U_2 as compare to equivalent Moore FSM U_1 . This decrease is proportional to the factor

$$\eta_2 = 1 + R_1 / m, \quad (15)$$

where m is the bit capacity of the field $Y(a_m)$, which depends on the microoperations encoding [4]. In the example of ASM Γ_1 , we have $\eta_1 = 37 / 17 \approx 2,18$ and $\eta_2 = 1 + 3 / 15 = 1,2$. It means that the number of elements in the circuit of $H_2(\Gamma_1)$ is 56% less than in case of $H_0(\Gamma_1)$ and 17% less than in case of $H_1(\Gamma_1)$. We would like to underline that such minimization does not lead to decrease of performance of a digital system with FSM U_2 .

Our future research is connected with exploration of possibility for the proposed method application when a control unit is implemented using technology of FPGA. In this case we are going to use some standard benchmarks from [14].

6. References

- [1] S. Baranov, "Logic Synthesis for Control Automata," Boston: Kluwer, 1994.
- [2] D. Kania, "Logic Synthesis Oriented on Programmable Logic Devices of the PAL type," Gliwice: Silesian Technical University (in Polish), 2004.
- [3] G. De Micheli, "Synthesis and Optimization of Digital Circuits," New York: McGraw Hill, 1994.
- [4] A. Barkalov and M. Węgrzyn, "Design of Control Units with Programmable Logic," Zielona Gora: University of Zielona Gora Press, 2006.
- [5] A. Barkalov, "Principles of Optimization of logical circuit of Moore FSM. Cybernetics and system analysis," No.1, 1998, pp. 65-72 (in Russian).
- [6] V. Solovjev, "Design of Digital System Using the Programmable Logic Integrated Circuits," Moscow: Hotline – Telecom (in Russian), 2001.
- [7] T. Kam, T. Villa, R. Brayton, A. Sangiovanni-Vincentelli, "Synthesis of Finite State Machines: Functional Optimization," Kluwer Academic Publishers, Boston/London/Dordrecht, 1998.
- [8] T. Villa, T. Kam, R. Brayton, A. Sangiovanni-Vincentelli, "Synthesis of Finite State Machines: Logic Optimization," Kluwer Academic Publishers, Boston/London/Dordrecht, 1998.
- [9] S. Devadas, H.-K. Ma, R. Newton, A. Sangiovanni-Vincentelli, "State Assignment of Finite State Machines Targeting Multilevel Logic Implementations," *IEEE Transactions on Computer-Aided Design*, 1988, pp.1290-1300.
- [10] S. Chattopadhyay, "Area Conscious State Assignment with Flip-Flop and Output Polarity Selection for Finite State Machine Synthesis," *A Genetic Algorithm Approach*, The Computer Journal, vol. 48, No. 4, 2005, pp. 443-450.
- [11] Y. Xia and A. Almaini, "Genetic algorithm based state assignment for power and area optimization," *IEEP. – Comput. Dig. T.*, 149, 2002, pp. 128-133.
- [12] A. Barkalov, L. Titarenko, S. Chmielewski, "Optimization of Moore FSM on CPLD," *Computer – Aided design of Discrete Devices*, Proceedings of the Sixth International conference, Minsk, vol. 2, pp. 39-45, November 2007.
- [13] A. Barkalov, L. Titarenko, S. Chmielewski, "Optimization of Moore FSM on System-on-Chip," *IEEE East-West Design & Test Symposium*, Yerevan, Armenia, Kharkov, 2007, pp. 105-109.
- [14] S. Yang, "Logic Synthesis and Optimization Benchmarks User Guide," Microelectronics Center of North Carolina, Research Triangle Park, North Carolina, 1991.