

UDC 004.27

Larysa Titarenko (Doctor of Sc., Professor)¹,
 Sergey Kovalyov (PhD, Associate Professor)²,
 Sergey Tsololo (PhD, Associate Professor)²

¹University of Zielona Gora (Poland),

²Donetsk National Technical University (Ukraine)

E-mail: L.Titarenko@iie.uz.zgora.pl, kovalevsa55@gmail.com, s.solos@gmail.com

REDUCING THE NUMBER OF PAL MACROCELLS IN MOORE FSM

A method of Moore's circuit optimization is proposed. The method is based on the features of CPLD architecture and Moore's FSM model. The example of the offered method is given. The carried out researches have shown that the method reduces hardware expenses up to 30%. The scientific novelty of the proposed method is reduced to usage of peculiarities of both the Moore FSM (the existence of pseudoequivalent states) and CPLD (the wide fan-in of PAL macrocells). These peculiarities are used for decreasing the number of PAL macrocells in the Moore FSM's logic circuit. The practical meaning of the proposed method is determined by the diminishing the area of SoC occupied by the logic circuit of a control unit in comparison with known approaches.

Keywords: Moore FSM, pseudoequivalent states, SoC, CPLD, PAL, macrocells.

Introduction. The systems-on-chips (SoC) are widely used for implementing digital systems. Using this basis allows implementing a complex digital system using only a single chip [2]. Two main elements can be found in CPLD-based SoC, namely, macrocells of programmable array logic (PAL) and embedded memory blocks (EMB). The macrocells are used for implementing random logic, whereas the EMBs for implementing tabular functions [3].

Control units (CU) are very often implanted as parts of SoC [4, 5]. These devices are very important because they coordinate the interplay of all other system blocks. The model of Moore finite state machine (FSM) is very popular for synthesis of CUs [6]. To improve many characteristics of a digital system, it is necessary to optimize the number of PALs in the FSM's logic circuit.

To solve this problem, it is necessary to take into account the specifics of both the Moore FSM and a complex programmable logic device (CPLD) used for implementing a CU. Two peculiarities of Moore FSM can be used for the pseudoequivalent states [7]. A regular nature of output variables (microoperations) is the second specific. It allows using EMBs for implementing the system of microoperations [3]. The main specific of CPLD is a wide fan-in of PAL macrocells (up to 30) [8]. Also, the limited amount of product terms let a macrocells should be taken into account (up to 8) [4]. The aim of research is the reduction of the number PALs into the Moore FSM's logic circuit. The main task of the research is development of synthesis method using more than one source of state codes.

Peculiarities of Moore FSM. Let a control algorithm of a digital system be represented by a graph-scheme of algorithm (GSA) $\Gamma = \Gamma(B, E)$ where $B = \{b_0, b_E\} \cup E_1 \cup E_2$ is a set of vertices, $E = \{ \langle b_q, b_t \rangle | b_q, b_t \in B \}$ is a set of arcs. Here b_0 is a start vertex of GSA, b_E is an end vertex, E_1 is a set of operator vertices, E_2 is a set of conditional vertices. Operator vertices $b_q \in E_1$ contain collections of microoperations $Y(b_q) \subseteq Y$ where $Y = \{y_1, \dots, y_N\}$ is a set of microoperations [9]. Conditional vertices $b_q \in E_2$ include elements of the set of logical conditions $X = \{x_1, \dots, x_L\}$. Both vertices b_0 and b_E correspond to the initial state $a_1 \in A$, where $A = \{a_1, \dots, a_M\}$ is a set of

internal states. Each operator vertex $b_q \in E_1$ corresponds to a single state $a_m \in A$ [5]. The logic circuit of Moore FSM is represented by the following systems:

$$\Phi = \Phi(T, X), \tag{1}$$

$$Y = Y(T). \tag{2}$$

In (1)-(2) the set $T = \{T_1, \dots, T_R\}$ is a set of state variables used for state assignment ($R = \lceil \log_2 M \rceil$); the set $\Phi = \{D_1, \dots, D_R\}$ is a set of input memory functions. The systems (1)-(2) are constructed on the base of structure table (ST). It has the following columns: a_m is a current state; $K(a_m)$ is a code of the state $a_m \in A$; a_s is a state of transition; $K(a_s)$ is a code of the state a_s ; X_h is a conjunction of some elements $x_i \in X$ (or their complements) determining the $\langle a_m, a_s \rangle$; Φ_h is a collection of input memory functions equal to 1 to replace the code $K(a_m)$ by the code $K(a_s)$; $h = \overline{1, H_1(\Gamma)}$ is the number of transition. The column a_m of ST includes the collection $Y(a_m) \subseteq Y$ of microoperations generated in the state $a_m \in A$. Naturally, it is $Y(a_m) = Y(b_q)$ where the vertex $b_q \in E_1$ is marked by the state $a_m \in A$.

The number of transitions $H_1(\Gamma)$ is, as a rule, bigger than this number $H_2(\Gamma)$ for the equivalent Mealy FSM [5]. It results in the increasing of the number of PALs in the logic circuit of Moore FSM in comparison with its counterpart of equivalent Mealy FSM. The value of $H_1(\Gamma)$ could be decreased due to the use of pseudoequivalent if outputs of operator vertices marked by these states are connected with an input of the same vertex. Let $\Pi_A = \{B_1, \dots, B_I\}$ be a partition of the set A by classes of pseudoequivalent states (PES). Let us encode a class $B_i \in \Pi_A$ by a binary code $K(B_i)$ having $R_1 = \lceil \log_2 I \rceil$ bits. Let us use the variables $\tau_r \in \tau$ for the encoding, where $|\tau| = R_1$. In this, the following model U_1 is used (Fig. 1).

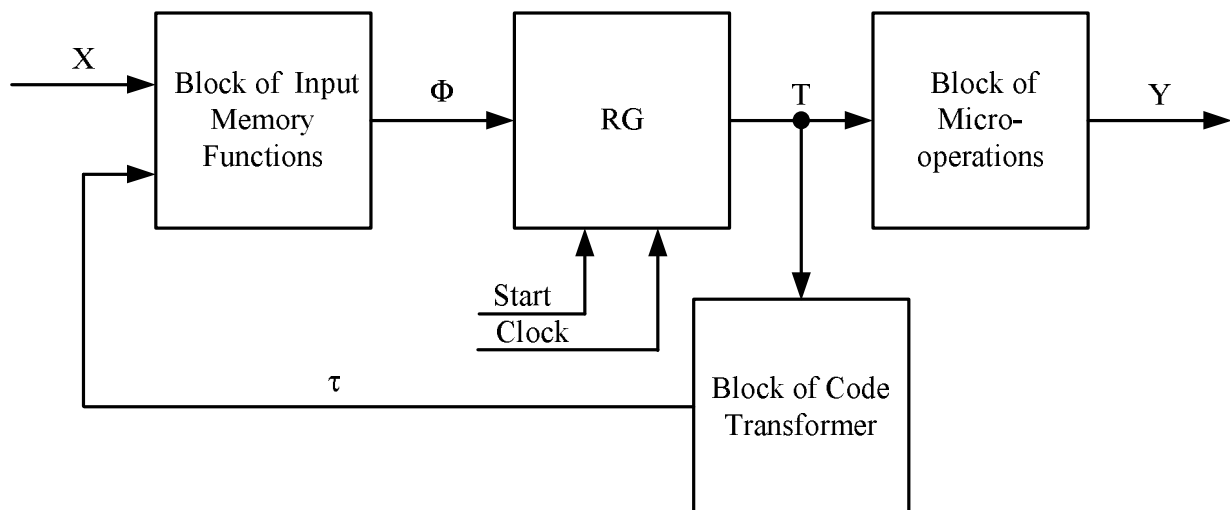


Figure 1 – Structural diagram of Moore FSM U_1

In FSM U_1 , the block of input memory functions (BIMF) implements the functions:

$$\Phi = \Phi(\tau, X). \tag{3}$$

The block of microoperations (BMO) implements the system (2). The register RG represents a state memory. If Start = 1 then zero code of the initial state $a_1 \in A$ is loaded into RG. The pulse Clock causes the changing of state codes into RG. The block of code transformer (BCT) implements the system:

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

So, the BCT generates a code $K(B_i)$ on the base of codes $a_m \in B_i$.

At it is shown in [10], The number of transitions is existence of the BCT requiring some resources of the chip. In U_1 , the circuit of BIMF is implement with PALs, whereas the circuits of BCT and BMO by EMBs. A method is proposed in this article allowing the hardware reductions in the circuit of BCT.

Two specifics of CPLD are used in the proposed method [6, 7]:

- the fan-in of PAL macrocells exceeds tremendously the value $L+R$, which is equal the maximal possible number of literals in terms of (1);
- the number of EMB outputs can be taken from some set $v = \{1, 2, 4, 8, 16\}$.

The main idea of proposed method. Let us use the method of optimal state assignment from [10]. The main idea of the assignment is to represent each class $B_i \in \Pi_A$ by minimal possible amount of generalized intervals of R-dimensional Boolean space. Let us represent the set Π_A as $\Pi_A = \Pi_B \cup \Pi_C$, where $\Pi_B \cap \Pi_C = \emptyset$. The class $B_i \in \Pi_B$ if the flowing condition takes place:

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

If the condition (5) is violated, then $B_i \in \Pi_C$. Obviously, the BCT should generate only the codes of $B_i \in \Pi_B$.

Let us encode the state by optimal codes [10]. Let us represent the set Π_B as $\Pi_B = \Pi_D \cup \Pi_E$. Let $B_i \in \Pi_D$ if the codes of states $a_m \in B_i$ are represented by single generalized interval of R-dimensional Boolean space. Only codes of states $a_m \in A(\Pi_E)$ should be transformed. Here the set $A(\Pi_E) \subseteq A$ states from the classes $B_i \in \Pi_E$. It is enough R_2 bits for encoding of the classes $B_i \in \Pi_E$:

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

The variables $z_r \in Z$ could be used for encoding of the classes, where $|Z| = R_2$.

Let us use the following symbols: t_F is a fixed number of EMB; q is a the number of cells of EMB for $t_F = 1$. There are t_S outputs in all EMBs of BMO:

$$t_S = \lceil N/t_F \rceil \cdot t_F. \quad (7)$$

The value of t_F is determined for U_1 using the following expression:

$$t_F = \lceil q/M \rceil. \quad (8)$$

Obviously, the following amount of outputs are free:

$$\Delta_t = 0. \tag{9}$$

These outputs could be used for representing the variables $z_r \in Z$. Let us discuss the case, when the following condition takes place:

$$R_2 = \lceil \log_2(|\Pi_E| + 1) \rceil. \tag{10}$$

In this case, we propose to use the model of Moore FSM U_2 (Fig. 2).

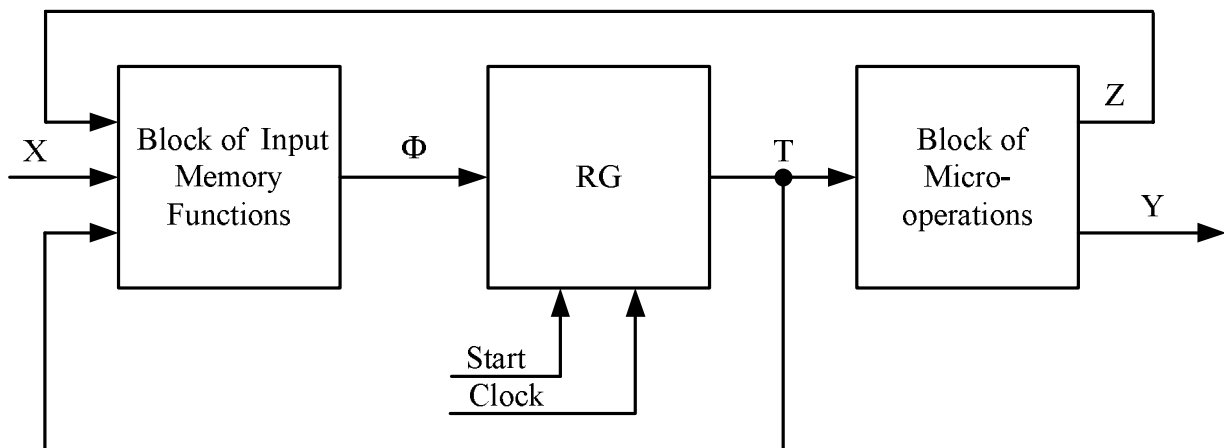


Figure 2 – Structural diagram of Moore FSM U_2

The model U_2 has some distinguishing features:

- the BIMF implements the functions:

$$\Phi = \Phi(T, Z, X); \tag{11}$$

- the BMO implements not only the system (2) but also the system

$$Z = Z(T); \tag{12}$$

- there is no a code transformer;
- the variables $T_r \in T$ represent the states $a_m \in A(\Pi_C)$, as well as the classes $B_i \in \Pi_D$.
The set $A(\Pi_C)$ includes the states $a_m \in B_i$ such that $B_i \in \Pi_C$.

In the case of U_2 , the number of required inputs of macrocells is increased to $L + R + R_2$. In the case of U_1 , only $L + R_1$ inputs are enough. But it does not cause the growth of hardware because of the high value of fan-in of modern CPLD (up to 30 [7]). The propagation times equal for equivalent FSMs U_1 and U_2 . Therefore, the proposed method allows the hardware reduction without the loss of performance.

The proposed design method for U_2 includes the following steps:

1. Constructing the marked GSA Γ .
2. Finding the partition $\Pi_A = \Pi_B \cup \Pi_C$.

3. Optimal state assignment and finding the sets Π_D, Π_E .
4. Encoding of the classes $B_i \in \Pi_E$.
5. Constructing the table of BMO.
6. Constructing the modified structure table of FSM U_2 .
7. Implementing the FSMs logic circuit.

Example of application of proposed method. Let us form the sets $A = \{a_1, \dots, a_{15}\}$ and $\Pi_A = \{B_1, \dots, B_8\}$ for some GSA Γ_1 . Let it be $B_1 = \{a_1\}$, $B_2 = \{a_2, a_3, a_4\}$, $B_3 = \{a_5, a_6\}$, $B_4 = \{a_7, a_8, a_9\}$, $B_5 = \{a_{10}, a_{11}, a_{12}\}$, $B_6 = \{a_{13}\}$, $B_7 = \{a_{14}\}$, $B_8 = \{a_{15}\}$. So, there are $\Pi_B = \{B_2, B_3, B_4, B_5\}$ and $\Pi_C = \{B_1, B_6, B_7, B_8\}$. Let us use the approach from [10] and encode the states $a_m \in A$ in the optimal manner (Fig. 3). In the discussed case, there is $R = 4$ and, therefore, $T = \{T_1, \dots, T_4\}$.

		$T_3 T_4$			
		00	01	11	10
$T_1 T_2$	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

Figure 3 – Optimal state codes for Moore FSM $U_2(\Gamma_1)$

The symbol $U_i(\Gamma_j)$ means that the model U_i is used for designing the logic circuit for a GSA Γ_j . As follows from Fig. 3, there are $\Pi_D = \{B_2, B_3, B_4\}$ and $\Pi_E = \{B_5\}$. The codes of $B_i \in \Pi_D$ are the following: $K(B_2) = 01**$, $K(B_3) = 00*1$, $K(B_4) = **10$. The codes of classes $B_i \in \Pi_C$ coincide with corresponding codes of the states $a_m \in B_i$. In the discussed example, there are $K(B_1) = 0000$, $K(B_6) = 1000$, $K(B_7) = 1001$, $K(B_8) = 1011$. So, there are is $|\Pi_E| = 1$. Using (6), we can find that there are $R_2 = 1$ and $Z = \{z_1\}$.

Let it be $N = 15$ for the GSA Γ_1 . Let the EMBs in use have $t_F = 4$ for $q = 16$. So, there are $t_S = 4 \cdot 4 = 16$ and $\Delta_t = 16 - 15 = 1$. It means that the condition (9) takes place. So, the proposed design method can be applied. Let it be $K(B_5) = 1$. If $Z_1 = 0$, then the FSM is some state $a_m \notin B_5$.

Let the interstate transitions be represented by the following system of generalized formula of transitions [6]:

$$\begin{aligned}
 B_1 &\rightarrow a_2; & B_2 &\rightarrow x_1 a_{10} \vee \bar{x}_1 x_2 a_{11} \vee \bar{x}_1 \bar{x}_2 a_{12}; \\
 B_3 &\rightarrow x_1 a_{13} \vee \bar{x}_1 a_{14}; & B_4 &\rightarrow x_1 a_5 \vee \bar{x}_1 x_3 a_6 \vee \bar{x}_1 \bar{x}_3 a_7; \\
 B_5 &\rightarrow x_4 a_2 \vee \bar{x}_4 x_3 a_3 \vee \bar{x}_4 \bar{x}_3 a_4; & B_6 &\rightarrow x_5 a_8 \vee \bar{x}_5 a_9; \\
 B_7 &\rightarrow a_{15}; & B_8 &\rightarrow x_3 a_{10} \vee \bar{x}_3 a_1.
 \end{aligned}
 \tag{13}$$

Let the microoperations $y_n \in Y$ be distributed among the states of FSM $U_2(\Gamma_1)$ in the fol-

lowing 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 includes the columns a_m , $K(a_m)$, $Y(a_m)$, $K(B_i)$, m , where $K(a_m)$ is an address of some cell. In the case of $U_2(\Gamma_1)$, this table has $M = 15$ rows (Fig. 4). As follows from Fig.4, the variable $Z_1 = 1$ is added to the collections of microoperations for states $a_m \in B_5$. It is connected with the relation $B_5 \in \Pi_E$. The modified structure table of FSM U_2 includes the columns B_i, \dots, h . The code of any class B_i is represented as $B_i, K(B_i), a_s, K(a_s), X_h, \Phi_h, h$. In the case of FSM $U_2(\Gamma_1)$, the table includes $H_2(\Gamma_1) = 17$ rows. This number is determined by the total amount of terms in the system (13). The transitions for classes B_2, B_5, B_6 are represented by Table 1.

$T_1 T_2$		$T_3 T_4$			
		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		$y_1 y_3 y_4 z_1$	$y_4 y_{13} z_1$	$y_1 y_7 y_8 y_{15} z_1$	$y_1 y_7 y_8 y_{15}$
10		$y_7 y_9$	$y_2 y_{12}$	$y_4 y_{13}$	$y_{10} y_{12}$

Figure 4 – Content of cells for block BMO of FSM $U_2(\Gamma_1)$

Table 1 – The part of modified ST of Moore 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	0	0	1	*	*	a_{10}	1	1	0	0	x_1	$D_1 D_2$	1
						a_{11}	1	1	0	1	$\bar{x}_1 x_2$	$D_1 D_2 D_3$	2
						a_{12}	1	1	1	1	$\bar{x}_1 \bar{x}_2$	$D_1 D_2 D_3 D_4$	3
B_5	1	*	*	*	*	a_2	0	1	0	0	x_4	D_2	4
						a_2	0	1	0	1	$\bar{x}_4 x_3$	$D_2 D_3$	5
						a_4	0	1	1	1	$\bar{x}_4 \bar{x}_3$	$D_2 D_3 D_4$	6
B_6	0	1	0	0	0	a_8	1	1	1	0	x_5	$D_1 D_2 D_3$	7
						a_9	1	0	1	1	\bar{x}_5	$D_1 D_3$	8

This table is the base for constructing the system (11). For example, the following part of equation can be derived from the Table 1: $D_3 = z_1 \bar{T}_1 \bar{T}_2 \bar{x}_1 \bar{x}_2 \vee z_1 \bar{x}_4 \bar{x}_3 \vee \bar{z}_1 T_1 \bar{T}_2 \bar{T}_3 \bar{T}_4$. Let us point out that this equation is minimized. It is interesting that there are 37 rows in the structure table of Moore FSM with the arbitrary state encoding (for the GSA Γ_1).

The last step of the method is connected with development of VHDL – models and use of standard tools. These questions are thoroughly discussed in literature [1, 7]. We do not discuss this step in our article.

Conclusion. The proposed method allows decreasing the hardware amount in the logic circuit of CPLD – based Moore FSM. The comparison is made with the Moore FSM U_1 including the transformer of state codes into the codes of classes of pseudoequivalent states. If the condition (10) takes place, then there is no need in the BCT. It decreases the number of PALs in the logic circuit of Moore FSM.

The scientific novelty of the proposed method is reduced to usage of peculiarities of both the Moore FSM (the existence of pseudoequivalent states) and CPLD (the wide fan-in of PAL macrocells). These peculiarities are used for decreasing the number of PAL macrocells in the Moore FSM's logic circuit.

The practical meaning of the proposed method is determined by the diminishing the area of SoC occupied by the logic circuit of a control unit in comparison with known approaches. We conducted the investigations to compare the models U_1 and U_2 . It turns out that the proposed approach always gives the gain if the condition (10) takes place. The maximal gain can be up to 38%. Besides, both FSMs have equal propagation times. Therefore, the proposed method gives a gain in hardware without the loss of performance.

Bibliography

1. Barkalov, A. Logic Synthesis for FSM-based Control Units / A. Barkalov, L. Titarenko. – Berlin: Springer, 2009. – 234 p.
2. Грушницький, Р.И. Проектирование систем с использованием микросхем программируемой логики / Р.И. Грушницький, А.Х. Мурсаев, Е.П. Угрюмов. – СПб.: БХВ. – Петербург, 2002. – 608 с.
3. Maxfield, C. The Design Warriors Guide to FPGAs / C. Maxfield. – Elsevier, 2004. – 541 p.
4. Соловьев, В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем / В.В. Соловьев. – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с.
5. Баркалов, А.А. Синтез микропрограммных устройств управления / А.А. Баркалов, А.В. Палагин. – К.: Институт кибернетики НАН Украины, 1997. – 136 с.
6. Baranov, S. Logic Synthesis for Control Automata / S. Baranov. – Netherlands: Kluwer Academic Publishers, 1994. – 312 p.
7. Баркалов, А.А. Синтез устройств управления на программируемых логических устройствах / А.А. Баркалов. – Донецк: ДНТУ, 2002. – 262 с.
8. Kania, D. Synteza logiczna przeznaczona dla matrycowych struktur programowalnych typu PAL / D. Kania. – Gliwice: Zeszyty naukowe Politechniki Śląskiej, 2004. – 240 p.
9. DeMicheli, G. Synthesis and Optimization of Digital Circuits / G. DeMicheli. – McGraw-Hill, 1994. – 636 p.
10. Баркалов, А.А. Принципы оптимизации логической схемы микропрограммного автомата Мура / А.А. Баркалов // Кибернетика и системный анализ. – 1998, №1. – С. 65 – 72.

References

1. Barkalov, A. and Titarenko L. (2009), [Logic Synthesis for FSM-based Control Units], Springer, Berlin, Germany.
2. Grushnickij, R.I., Mursaev A.H. and Ugrjumov E.P. (2002), *Proektirovanie sistem s ispol'zovaniem mikroshem programmiruemoj logiki* [Designing systems using programmable logic chips], BHV, Saint-Petersburg, Russian Federation.
3. Maxfield, C (2004), [The Design Warriors Guide to FPGAs], Elsevier.
4. Solov'ev, V.V. (2001), *Proektirovanie cifrovyh shem na osnove programmiruemyh logicheskikh integral'nyh shem* [Designing digital circuits based on programmable logic integrated circuits], Hot Line-TELECOM, Moscow, Russian Federation.
5. Barkalov, A.A. and Palagin A.V. (1997), *Sintez mikroprogrammnyh ustrojstv upravlenija*

- [Synthesis of microprogram control devices], Institute of Cybernetics of NAS of Ukraine, Kiev, Ukraine.
6. Baranov, S (1994), [Logic Synthesis for Control Automata],: Kluwer Academic Publishers, Netherlands.
 7. Barkalov, A.A (2002), *Sintez ustrojstv upravlenija na programmiruemym logicheskim ustrojstvah* [Synthesis of control devices on programmable logic devices], DNTU, Donetsk: Ukraine.
 8. Kania, D. (2004), [Synteza logiczna przeznaczona dla matrycowych struktur programowalnych typu PAL], Zeszyty naukowe Politechniki Śląskiej, Gliwice.
 9. DeMicheli, G (1994), [Synthesis and Optimization of Digital Circuits], McGraw-Hill.
 10. Barkalov, A. A. (1998) «Principles of optimization logic circuit of Moore FSM», Cybernetics and systems analysis. – №1. – pp. 65 – 72.

Надійшла до редакції:
23.04.2016

Рецензент:
д-р техн. наук, доц. Є.Є. Федоров

Л.О. Титаренко, С.О. Ковальов, С.О. Цололо

¹Зеленогурський університет, ²Донецький національний технічний університет

Зменшення кількості макроосередків PAL у схемі МПА Мура при реалізації на CPLD.
Пристрій керування координує взаємодію всіх блоків цифрової системи і на практиці часто реалізується з використанням моделі мікропрограмного автомата Мура. У наш час прогрес в області мікроелектроніки привів до появи «систем-на-кристалі» (SoC, system-on-chip), функціональні можливості яких є достатніми для реалізації складної цифрової системи на одному кристалі. У SoC логіка може реалізовуватися з використанням макроосередків Programmable Array Logic (PAL), а табличні функції реалізуються за допомогою блоків пам'яті Embedded Memory Blocks (EMB). Однією з актуальних завдань в цьому випадку є зменшення витрат апаратури в схемі мікропрограмного автомата Мура. Вирішення цього завдання дозволяє зменшити площу кристала, яку займає схема пристрою керування, при цьому можливо збільшення функціональних можливостей системи в рамках одного кристала. В роботі запропоновано метод зменшення апаратних витрат у схемі автомата Мура, що заснований на використанні як особливостей елементного базису, так й можливих модифікацій структурної схеми мікропрограмного автомата Мура. Так, вільні виходи вбудованих блоків пам'яті можуть бути використані для представлення кодів класів псевдоеквівалентних станів мікропрограмного автомата, при цьому зменшується кількість макроосередків PAL завдяки великому коефіцієнту об'єднання за входом. Це дозволяє уникнути використання перетворювача кодів при використанні псевдоеквівалентних станів мікропрограмного автомата Мура. Сукупність цих підходів дозволяє зменшити площу кристала SoC, що займає комбінаційна схема мікропрограмного автомата Мура та отримати схему, яка володіє меншою вартістю, ніж відомі з літератури аналоги.

Ключові слова: CPLD, витрати апаратури, МПА Мура, PAL, макроосередки.

Л.А. Титаренко, С.А. Ковалев, С.А. Цололо

¹Зеленогурський університет, ²Донецький національний технічний університет

Уменьшение количества макроячеек PAL в схеме МПА Мура при реализации на CPLD.

Устройство управления координирует взаимодействие всех блоков цифровой системы и на практике часто реализуется с использованием модели микропрограммного автомата Мура. В настоящее время прогресс в области микроэлектроники привел к появлению «систем-на-кристалле» (SoC, system-on-chip), функциональные возможности которых достаточны для реализации сложной цифровой системы на одном кристалле. В SoC произвольная логика может реализовываться с использованием макроячеек Programmable Array Logic (PAL), а

табличные функции реализуются с помощью блоков памяти *Embedded Memory Blocks (EMB)*. Одной из актуальных задач в этом случае является уменьшение аппаратных затрат в схеме микропрограммного автомата Мура. Решение этой задачи позволяет уменьшить площадь кристалла, занимаемую схемой устройства управления, при этом возможно увеличение функциональных возможностей системы в рамках одного кристалла. В работе предлагается метод, который позволяет уменьшить аппаратные затраты в схеме автомата Мура, реализуемого в базисе CPLD, по сравнению с МПА $U_1(\Gamma)$, включающим преобразователь кодов псевдоэквивалентных состояний в коды классов псевдоэквивалентных состояний. При наличии достаточного количества свободных выходов блоков EMB отпадает необходимость в использовании преобразователя, что уменьшает число блоков EMB по сравнению с МПА $U_1(\Gamma)$. Суть предложенного метода заключается в использовании особенностей автомата Мура (наличие классов псевдоэквивалентных состояний) и элементного базиса (большой коэффициент объединения по входу) для уменьшения числа макроячеек PAL в логической схеме автомата. Это позволяет уменьшить площадь кристалла SoC, занимаемую комбинационной схемой МПА $U_2(\Gamma)$, и получить схему, которая обладает меньшей стоимостью, чем известные из литературы аналоги. При этом автоматы $U_1(\Gamma)$ и $U_2(\Gamma)$ имеют одинаковое быстродействие, то есть выигрыш по аппаратуре не приводит к потере производительности.

Ключевые слова: CPLD, аппаратные затраты, МПА Мура, PAL, макроячейки.



Larisa Titarenko, Ukraine, graduated from the Kharkov Institute of Radio Electronics, Doctor of Engineering Sciences, Professor at the Telecommunication Equipment Department of SHEE «Kharkiv National University of Radioelectronics» (Nauka Avenue, 14, Kharkiv, Kharkiv region, 61000) and at the Institute of Informatics and Electronics of the University of Zielona Góra (ul prof Z. Szafrana 2, 65-516 Zielona Góra, Poland), key research area is Research and development of synthesis methods of control devices for telecommunication systems



Sergey Kovalev, Ukraine, graduated from Donetsk Polytechnic Institute, Docent, Candidate of Engineering Sciences, Associate Professor at the Computer Engineering Department of SHEE "Donetsk National Technical University" (Shibankova Square, 2, Krasnoarmiysk, 85300, Ukraine). Key research areas are development, modeling and research of design tools for computer systems and components, intelligent monitoring and control systems.



Sergey Tsololo, Ukraine, graduated from Donetsk National Technical University, Docent, Candidate of Engineering Sciences, Associate Professor at the Computer Engineering Department of SHEE "Donetsk National Technical University" (Shibankova Square, 2, Krasnoarmiysk, 85300, Ukraine). Key research areas are development, modeling and research of synthesis methods of control devices in the bases of modern programmable logic integrated circuits, as well as development, modeling and research of distributed web-based computer systems