International Journal of Software Engineering and Computer Systems (IJSECS)
ISSN: 2289-8522, Volume 4 Issue 1, pp. 75-83, February 2018
©Universiti Malaysia Pahang
https://doi.org/10.15282/ijsecs.4.1.2018.6.0039

IMPLEMENTING COMBINED FSM WITH CPLDS

Barkalov A.[1], Titarenko L.[1], Zeleneva I.[2], Hrushko S.[2]

[1]Institute of Metrology, Electrotechnics and Informatics, University of Zielona Gora, Poland.
[2]Information Science and Computer Engineering Department, Zaporizhzhya National Technical University, Ukraine.
Email: a.barkalov@imei.uz.zgora.pl

ABSTRACT

The subject of the research in this article is the logic circuit of the combined finite state machine (CFSM), which combines the functions of the both FSM Mealy and Moore. In practice, such a model of control automata is used widely, but in the literature there is almost no theoretical description CFSM models and ways to optimize them. The article considers the problem of optimizing the logic of the combined finite state machine implemented in complex programmable logic devices (CPLD) basis. The CFSM circuit using programmable array logic (PAL) macrocells is implemented. The number of CPLD components, required to implement the logic of the automaton circuit, depends on the CFSM parameters and characteristics of element basis. Obviously, the reduction of necessary number of components leads to a decrease area occupied CFSM scheme in CPLD, thereby leads to reducing the hardware amount and power consumption in the circuit, and as result, increases the efficiency of the whole project. To solve the problem of CFSM optimization for a criterion of hardware expenses in this article it's proposed to use the structural features of the basis CPLD, as well as the method of pseudoequivalent states. The FSM states are defined as a pseudoequivalent, if they mark some vertices linked with the input of the same next vertex in flow-chart. The proposed method includes the following steps: forming a plurality of CFSM states; encoding of states; forming a set of classes pseudoequivalent states; formation PALer blocks and tables; implementation of the CFSM scheme in a given element basis, such as CPLD. As a result, it's possible to reduce the necessary number of PAL macro cells for implementing CFSM circuit in CPLD. The general result is a decreasing of in the total area of CFSM circuit on a chip. The advisability conditions of this method are discussed.

Keywords: combined FSM, CPLD, PAL, pseudoeguivalent states.

INTRODUCTION

Today, the use of programmable logic integrated circuits and very large scale integrated (VLSI) circuits is a common practice in the production of electronics in a wide variety of industries. The use of modern VLSI, such as CPLD (complex programmable logic devices) and FPGA (field programmable gate array) in the design of any complex digital devices, including embedded systems, provides a gain in size, power consumption and functionality of the final product in comparison with the use of not only standard logic chips, but also microcontrollers, microprocessors, signal processors. This article considers CPLD, on the basis of which the problem of synthesis and optimization of a control device logic circuit, that is an important part of almost any digital system (Baranov, 2008; De Micheli, 1994), is solved. Any digital device can be structurally divided into a control part and an executive part, i.e. control and operational automaton. In this case, the operating machine provides the actual processing of information according to a given algorithm, and the control automaton just ensures the order of operation of the operating part in strict accordance with this algorithm.

At present, when designing complex digital devices, the model of a combined control automaton is widely used, which functions both as a Mealy and Moore finite state machines (FSM) (Solovyov & Klimovich, 2008; Czerwinski & Kania, 2013). In practice, the need for such a combined model is explained by the complexity and variety of functions performed by modern digital systems. However, theoretical materials devoted to the design and optimization of logic circuits of combined FSM (CFSM) are practically absent among the publications. This explains the interest of the authors of the article to the topic of optimization of CFSM. The control unit in the form of a combined finite state machine, implemented on the basis of CPLD, ensures the performance of all the necessary functions for solving the tasks at moderate material costs. Let us point out that CPLD-based circuits are very popular for implementing control units (Czerwinski & Kania, 2013; Barkalov, Titarenko & Chmielewski, 2007).

The actual scientific and practical problem of optimizing the CFSM scheme by the criterion of hardware costs is considered, namely, at the cost minimizing of the CPLD chips used in the implementation (Sklyarov, Sklyarova, Barkalov & Titarenko, 2014; Sklyarova, Sklyarov & Sudnitson, 2012). As a rule, the solution of this problem allows decreasing the consumed power and increasing the FSM performance (Sklyarova, Sklyarov & Sudnitson, 2012). Methods of this problem`s solution depend strongly on peculiarities of both an FSM model and logic elements used for implementing an FSM circuit (Solovyov & Klimovich, 2008; Czerwinski & Kania, 2013).

The basis of the CPLD structure is the macrocell PAL or PLA, which depends on the manufacturer. The cost of the chip depends on the number of these macrocells inside the CPLD. Thus, the aim of the research is to reduce the number of internal macrocells of the PAL type required for the implementation of the SMPA logic circuit.

The optimization method proposed in the article is a continuation of the earlier proposed methods by the authors for other hardware bases. This article represents a development of FPGA-based design methods (Barkalov, Titarenko, & Zelenjova, 2015; Barkalov, Zelenjova & Hrushko, 2015) for the case of CPLD-based CFSMs.

The method is based on the use of pseudoequivalent states in the automaton graph. The FSM states are defined as a pseudoequivalent, if they mark some vertices linked with the input of the same next vertex in flow-chart. The proposed method includes the following steps: forming a plurality of CFSM states; encoding of states; forming a set of classes pseudoequivalent states; formation PALer blocks and tables; implementation of the CFSM scheme in a given element basis, such as CPLD. Application of the proposed method makes it possible to reduce the required number of macrocells by almost half, thereby achieving the goal of reducing costs when implementing the device circuit.

SPECIFICS OF CFSM AND CPLD

The main specific of CFSM is existence of two types of output variables (Baranov, 2008). There is a set Y^1 including output variables of Mealy FSM. The set

Y^2 includes output variables of Moore FSM. The set $Y = Y^1 \cup Y^2$ is a set of output variables of CFSM, where $|Y^1| = N_1$, $|Y^2| = N_2$ and N1+N2 = N.

The CFSM can be represented as the vector (Baranov, 2008):

$$S = <A, X, Y^1, Y^2, \delta, \lambda_1, \lambda_2, \alpha_1> \quad (1)$$

In (1) there are the following components: $A=\{\alpha_1, \dots, \alpha_M\}$ is a set of internal states; $X=\{x_1, \dots, x_L\}$ is a set of input variables; δ is a transition function; λ_1 is an output function of Mealy FSM; λ_2 is an output function of Moore FSM; a_1 is an initial state, $a_1 \in$ A . We can define the functions δ, λ_1 and λ_2 as the following:

$$a_s = \delta(a_m, X), \text{ where } a_m, a_s \in A; \quad (2)$$
$$y_n = \lambda_1(a_m, X), \text{ where } y_n \in Y^1; \quad (3)$$
$$y_n = \lambda_2(a_m), \text{ where } y_n \in Y^2. \quad (4)$$

The function δ determines the state of transition $a_s \in A$ as a function of the current state $a_m \in A$ and input variables X. As follows from (3), outputs of Mealy FSM depend on current states and inputs. As follows from (4), outputs of Moore FSM depend only on current states.

Different nature of function (3) and (4) allows using the optimization methods of both Mealy and Moore FSMs for optimizing the circuit of CFSM. It is the main peculiarity of CFSM.

CPLD includes PAL macrocells (Altera documentation, 2017; Xilinx browse documentation, 2017). Each macrocell can be viewed as q programmable AND gates having S inputs. The AND gates are connected with OR gate. The OR output can be either connected with D flip-flop or not. A flip-flop has inputs of synchronization (Clock) and clearing (Start). Therefore, it is possible to get either combinational or registered output of a macrocell. Moreover macrosells are connected through the matrix of programmable interconnections. This matrix also connect marocells with input-output blocks.

The main CPLD specific is a rather small value of q ($q \leq 8$). It leads to disjoint minimization of Boolean functions representing CFSM circuit (Czerwinski & Kania, 2013; Barkalov, Titarenko & Chmielewski, 2007). The main aim of minimization is to decrease the number of terms in Boolean functions.

In this article we propose one of possible methods for solving this problem. We use a graph-scheme of algorithm (GSA) for representing of CFSM (Baranov, 2008).

IMPLEMENTING CFSM USING GSA

There are some steps of design independent on logic elements. They are the following (Baranov, 2008):

1. Marking initial GSA G by states.
2. Encoding of states $a_m \in A$ by binary codes $K(a_m)$ having R bits, where

$$R = \lceil log_2 M \rceil. \quad (5)$$

3. Constructing direct structural table (DST) of FSM.
4. Developing systems of Boolean functions corresponding to (2) – (4).

The special register RG, which includes R flip-flops, saves state codes. In this case of CPLD, the RG is distributed among the macrocells. Internal variables $T_r \in T$ are used for representing the state codes, where $T = \{T_1, \dots, T_R\}$.

Excitation functions $D_r \in \varphi$ can change the content of RG, where $\varphi = \{D_1, \dots, D_R\}$. The variable D_r enters the D input of the r-th flip-flop of RG ($r = \overline{1,R}$).

The following columns can be found in a DST of CFSM (Baranov, 2008): a_m is a current state; $K(a_m)$ is a code of state $a_m \in A$; a_s is a state of transition; $K(a_s)$ is a code of state $a_s \in A$; X_h is an input signal determining the transition $<a_m, a_s>$; Y_h^1 is a set of output variables $y_n \in Y^1$ produced during the transition $<a_m, a_s>$; φ_h is a set of input memory functions $D_r \in \varphi$ equal to 1 to load the code $K(a_s)$ into RG; h is a number of transition ($h = \overline{1,H}$). Besides, there is output variables $y_n \in Y^2$ produced in the state $a_m \in A$ written in the column of current state.

The following functions are developed during the step 4:

$$\varphi = \varphi(T, X); \tag{6}$$
$$Y^1 = Y^1(T, X); \tag{7}$$
$$Y^2 = Y^2(T). \tag{8}$$

System (6) determines the function (2), systems (7) and (8) determines the functions (3) and (4) correspondently.

Let the symbol PALer be used for a circuit based on PALs. It follows from (6) and (7) that the functions (2) and (3) determine a circuit with inputs $x_l \in X$ and $T_r \in T$. Let us name this as PALer1. The function (8) corresponds to a circuit PALer2 having inputs $T_r \in T$. Also there is the distributed register RG in the PALer1. Thus, there are the inputs Clock and Start entering the PALer1. The systems (6) – (8) determine the structural diagram of CFSM U$_1$ (Fig. 1). Let us point out the state variables $T_r \in T$ are the outputs of the PALer1.

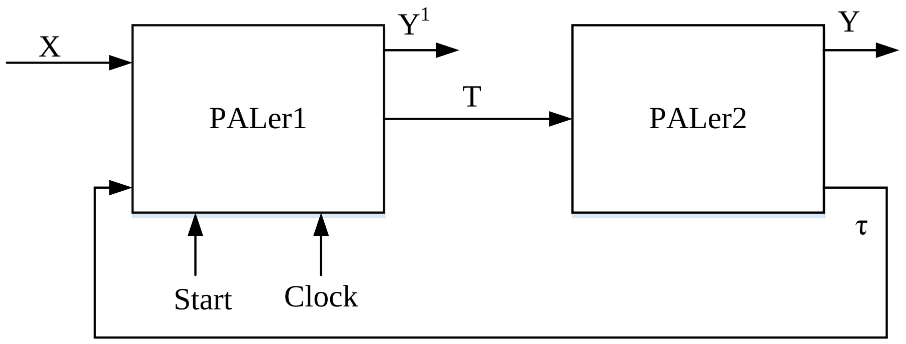

Fig. 1. Structural diagram of CFSM U$_1$

Functions (6) – (7) depend on conjunctive terms F_h corresponding to lines of DST. Each term is determined as

$$F_h = A_m \cdot X_h, \ (h = \overline{1,H}). \tag{9}$$

In (9) the symbol A_m stands for a conjunction of variables $T_r \in T$ corresponding to the state $a_m \in A$ from the h-th line of DST. The function (8) depends on the terms A_m ($m = \overline{1,M}$).

As a rule, different methods of state assignment are used for optimizing functions $D_r \in \varphi$ and $y_n \epsilon Y$ (De Micheli, 1994). In this article we propose a method based on existence of pseudoequivalent states (PES) of Moore FSM (Yang, 1991).

THE PROPOSED DESIGN METHOD

States $(a_m, a_s) \in$ A are PES if correspond to operator vertices of GSA G which outputs are connected with the same vertex of GSA (Sklyarov, Sklyarova, Barkalov & Titarenko, 2014). This definition allows constructing a partition $\pi_A = \{B_1, \dots, B_I\}$ on the set A. Each element of π_A is a class of the PES.

Let us encode states $a_m \in$ A in such a way that each class $B_i \epsilon \pi_A$ corresponds to minimal possible number of generalized intervals of R-dimensional Boolean space. These intervals can be viewed as codes $K(B_i)$ of the classes. It means that the same variables $\tau_r \epsilon T$ are used for both $K(a_m)$ and $K(B_i)$.

So, in this article, we propose a CFSM U_2 based on this approach for the state assignment. Let us point out that the desirable state assignment can be executed by the method JEDI from SIS (De Micheli, 1994).

There are the same structural diagrams for U_1 and U_2. The difference between U_1 and U_2 is reduced to the formulae of terms F_h. In both cases the conjunction A_m is used in the terms:

$$A_m = \wedge_{r=1}^{R} T_r^{l_{mr}}, (m = \overline{1, M}). \quad (10)$$

In (10) l_{mr} is a value of the r-th bit of the code $K(a_m)$. In the case of U_1, $l_{mr} \in \{0, 1\}$ and $T_r^0 = \overline{T_r}$, $T_r^1 = T_r$ $(r = \overline{L, R})$. In the case of U_2, $l_{mr} \epsilon \{0, 1, *\}$, where $T_r^* = 1$ $(r = \overline{1, R})$.

This difference leads to decreasing the number of DST lines up to H_1 in the case of U_2. Conversely, it decreases the number of terms in functions (6) – (7) for U_2 in comparison with the equivalent CFSM U_1. Let us point out that U_1 and U_2 are equivalent if they are synthesized using the same GSA **G**.

In this article we propose a method for synthesis CFSM U_2. It includes the following steps:

1. Marking initial GSA by the states of Moore FSM.
2. Constructing the partition π_A on the set A.
3. Executing the state assignment.
4. Constructing the direct structure table of U_2.
5. Developing systems (6) – (8).
6. Implementing CFSM circuit with given CPLD.

Let us discuss this approach for the case of GSA **G₁** (Fig. 2).

EXAMPLE OF SYNTHESIS

The vertices of G_1 are marked by the states of Moore FSM using the rules (Baranov, 2008). It is possible to find the following sets and their parameters from $\mathbf{G_1}$: $A=\{\alpha_1, \dots, \alpha_9\}$, $M = 9$, $X=\{x_1, \dots, x_4\}$, $L = 4$, $Y^1 = \{y_1, \dots, y_5\}$, $N_1 = 5$, $Y^2 = \{y_6, \dots, y_{10}\}$, $N_2 = 5$, $N = 10$. Let us point out that variables $y_n \in Y^2$ are placed inside the vertices, whereas variables $y_n \in Y^1$ are shown above the edges of GSA G_1.

Using the definition of PES, it is possible to find the partition π_A. In the discussed case, there is $\pi_A = \{B_1,...,B_4\}$ where $B_1 = \{\alpha_1\}$, $B_2 = \{\alpha_2\}$, $B_3 = \{\alpha_{3,...,}\alpha_6\}$ and $B_4 = \{\alpha_{7,}\alpha_{8,}\alpha_9\}$.

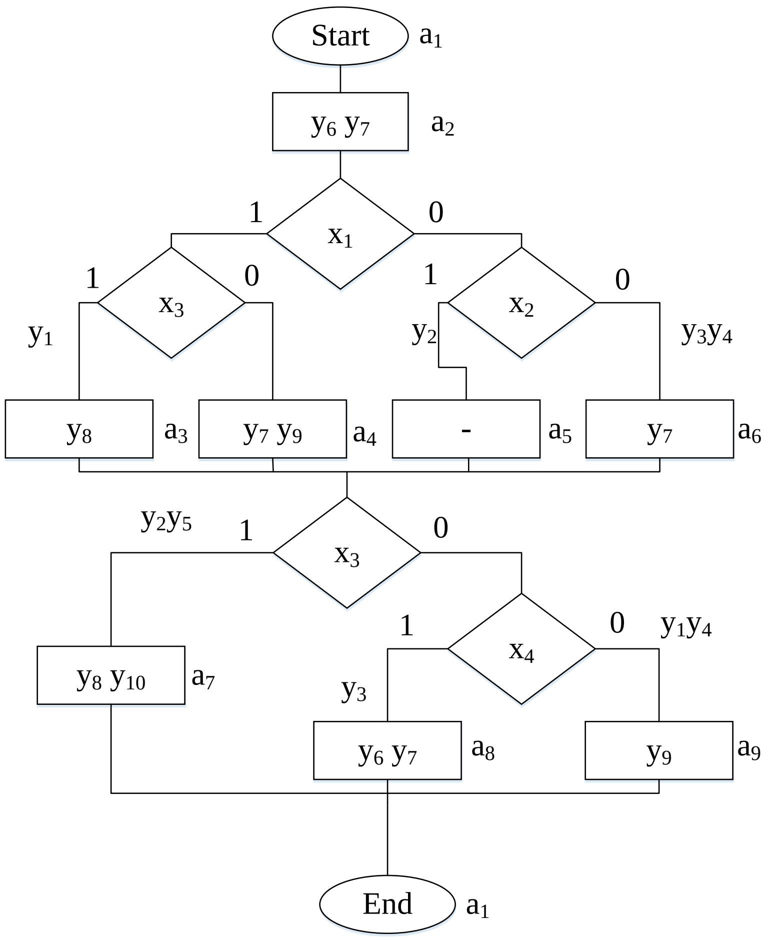

Fig. 2. Initial GSA G_1

To execute the state assignment, let us find dependence among $A_{m,}$ B_i and $y_n \in Y^2$. It is represented by the following system:

$$\begin{array}{ll} B_1 = A_1; & y_6 = A_2 \vee A_8; \\ B_2 = A_2; & y_7 = A_2 \vee A_4 \vee A_6 \vee A_8; \\ B_3 = A_3 \vee A_4 \vee A_5 \vee A_6; & y_8 = A_3 \vee A_7; \\ B_4 = A_7 \vee A_8 \vee A_9; & y_9 = A_4 \vee A_9; \\ & y_{10} = A_7. \end{array} \qquad (11)$$

First of all, the state assignment should be executed in such a way that functions $B_1 - B_3$ will be represented by single intervals. There are no transitions from $a_m \in B_4$ in DST of U_2. So, we do not care about the code of B_4.Then if it is possible, each function $y_n \in Y^2$ should be represented by minimal amount of generalized intervals. The fist rule gets reducing the amount of terms in functions (6) – (7), the second in (8).

Using (5) and M = 9, we can find R = 4. It gives the sets $T = \{T_1, \dots, T_4\}$ and $\varphi = \{D_1, \dots, D_4\}$. One of the possible variants of state assignment is shown in Fig. 3.

$T_3T_4 \backslash T_1T_2$	00	01	11	10
00	a_1	a_9	a_2	a_8
01	a_3	a_5	a_4	a_6
11	*	*	*	*
10	a_7	*	*	*

Fig. 3. State codes for CFSM U_2

Using information (Fig. 3) and system (11), the following Boolean system can be found (after minimization):

$$\begin{aligned} B_1 &= \overline{T_1} \cdot \overline{T_2} \cdot \overline{T_3} \cdot \overline{T_4}; & y_7 &= T_1; \\ B_2 &= T_1T_2\overline{T_4}; & y_8 &= \overline{T_1} \cdot \overline{T_2}T_4 \vee T_3; \\ B_3 &= T_4; & y_9 &= T_1T_2\,T_4 \vee \overline{T_1}T_2\overline{T_4}; \\ y_6 &= T_1\overline{T_4}; & y_{10} &= T_3. \end{aligned} \tag{12}$$

Analysis of (12) gives $K(B_1)$ = 0000, $K(B_2)$ = 11*0 and $K(B_3)$ = ***1. If $q \geq 2$, it is enough 3 macrocells PAL for implementing the PALer2. There is no need in PALs for functions y_7 and y_{10}. These functions are implemented by the PALer1 of U_2.

The DST of U_2 is similar to the DST of U_1. The only difference is: the columns a_m and $K(a_m)$ are replaced by B_i and $K(B_i)$, respectively. It is Table 1 in the discussed case.

Table 1. Direct structure table of CFSM U_2

B_i	$K(B_i)$	a_s	$K(a_s)$	X_h	Y_h^1	φ_h	h
B_1	0000	a_2	1100	1	-	D_1 D_2	1
B_2	11*0	a_3	0001	$x_1\,x_3$	y_1	D_4	2
		a_4	1101	$x_1\overline{x_3}$	-	D_1 D_2 D_4	3
		a_5	0101	$\overline{x_1}\,x_2$	y_2	D_2 D_4	4
		a_6	1001	$\overline{x_1} \cdot \overline{x_2}$	y_3y_4	D_1D_4	5
B_3	***1	a_7	0010	x_3	y_2y_5	D_3	6
		a_8	1000	$\overline{x_3}x_4$	y_3	D_1	7
		a_9	0100	$\overline{x_3} \cdot \overline{x_4}$	y_1y_4	D_2	8

There is H_1 = 8. In the case of equivalent U_1 there is H = 14, the DST contains more lines, and more PALs are necessary for PALer1. In the discussed case there is H/H_L=1,75. So, we should expect such a saving in PALs due to replacement U_1 to U_2. Of course, the economy depends on the value of q.

The DST is used for deriving the functions (6) – (7). In the discussed case, these systems are following:

$$\begin{array}{ll} D_1 = F_1 \vee F_3 \vee F_5 \vee F_7; & y_1 = F_2 \vee F_8; \\ D_2 = F_1 \vee F_3 \vee F_4 \vee F_8; & y_2 = F_4 \vee F_6; \\ D_3 = F_6 = T_4; & y_3 = F_5 \vee F_7; \\ D_4 = F_2 \vee F_3 \vee F_4 \vee F_5 = T_1 T_2; & y_4 = F_5 \vee F_8; \\ & y_5 = T_4. \end{array} \quad (13)$$

Let it be S = 10, q = 3. In this case it is necessary 9 PALs for implementing the circuit of PALer1. As follows from (13), there is no need in PALs for implementing functions D_3 and y_5.

So, it is necessary 12 PALs for implementing the circuit of U$_2$ in the discussed case. Let us point out that this number can be diminished. As follows from (13), there is a mutual part $F_1 \vee F_3$ in functions D_1 and D_2. So, the circuit for D_1 and D_2 can be implemented as the following (Fig. 4).

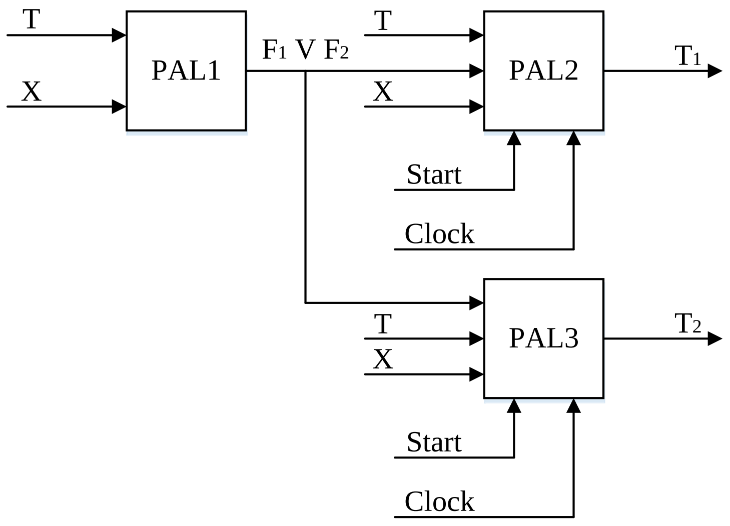

Fig. 4 Logic circuit for functions D_1 and D_2

The PAL1 implements the disjunction $F_1 \vee F_2$. It is used by both PAL2 and PAL3. Outputs D$_1$ and D$_2$ are connected with D flip-flops. Because of it, pulses Start and Clock enter PAL2 and PAL3.

To compare the circuits for U$_1$ and U$_2$ let consider the circuit of U$_1$ for GSA **G$_1$**. We use the same PALs with S = 10, q = 3 and the following state codes: $K(a_1)$ = 0000, $K(a_2)$ = 0001, …, $K(a_9)$ = 1000. It is necessary 26 PALs to implement the circuit of U$_1$. So, our approach allows a two-fold reduction in the number of macrocells for given particular case.

CONCLUTION

In this work we propose a method for synthesis the combined FSM targeting CPLD with PAL macrocells. An FSM is specified by a GSA G. The method is based on using of the classes of pseudoequivalent states.

The proposed method allows diminishing for the number of terms in systems of input memory functions $D_r \in \varphi$ and output variables of Moore FSM. Due to using PES, the number of terms is the same as for equivalent Mealy FSM.

It is possible to decrease the number of PALs using mutual parts of functions. This approach is similar to functional decomposition proposed in 1960th (Baranov, 2008). So, these "old" approaches can be used for optimization circuits with modern VLSI.

There are two directions of our future research. The first is connected with complex FSMs having $R+L>S$. The second targets CPLD with macrocells based on PLAs (Czerwinski & Kania, 2013). Such macrosells are used in CPLD Cool RunnerII by Xilinx (Xilinx browse documentation, 2017).

REFERENCES

Baranov, S. (2008). *Logic and System Desing of Digital Systems*. Tallinn: TUT Press.

De Micheli, G. *Synthesis and Optimization of Digital Circuits*. (1994). New York: Mc Graw-Hill.

Sklyarov, V., Sklyarova, I., Barkalov, A. & Titarenko, L. (2014). *Synthesis and Optimization of FPGA-based Systems*. Berlin: Springer.

Sklyarova I., Sklyarov, V. & Sudnitson, A. (2012). *Design of FPGA-based circuits using Hierarchical Finite State Machines* / Tallinn: TUT Press.

Solovyov, V.V. & Klimovich, A. (2008). *The logical design of digital systems based on programmable logic integrated circuits.* [*Logicheskoe proektirovanie tsifrovyih sistem na osnove programmiruemyih logicheskih integralnyih shem*]. Moscow: Goryachaya Liniya.

Czerwinski, R. & Kania, D. (2013). *Finite State Machine Logic Synthesis for Complex Programmable Logic Devices*. Berlin: Springer.

Barkalov, A., Titarenko, L. & Chmielewski, S. (2007). Reduction the number of PAL macrocells in the circuit of the Moore FSM. *International Journal of Applied Mathematics and Computer Science, 17*, 101-112.

Barkalov, A.A., Titarenko, L.A. & Zelenjova, I.J. (2015). Implementing of combined finite state machine with FPGA. *Proceedings of Donetsk National Technical University. Series «Informatics, Cybernetics and Computer Science",2(21)*, 84-88.

Barkalov, A.A., Zelenjova, I.J. & Hrushko, S.S. (2015). Optimization of combined automaton circuit on base FPGA using the method of input variables changing. *Scientific Herald of Chernivtsi University: Computer systems and components. Volume 6, Issue 2,* 49-54.

Altera documentation. (2017). Retrieved from www.altera.com/ support/literature /lit-index. html.

Xilinx browse documentation. (2017). Retrieved from https://www.xilinx.com/ support.html# documentation.

Yang, S. (1991). *Logic Synthesis and optimization benchmarks user guide*. North Carolina: Microelectronics Centre of North Carolina.