ISSN 2074-7888, Наукові праці Донецького національного технічного університету,
серія: «Проблеми моделювання та автоматизації проектування» № 1 (12)-2(13), 2013

UDC 004.274

A.A. Barkalov (Doctor of Sc., Professor)[1],
L.A. Titarenko (Doctor of Sc., Professor)[2],
S.A. Tsololo (PhD)[1], A.N. Miroshkin (PhD) [1]
[1]Donetsk National Technical University (Ukraine),
[2]University of Zielona Gora (Poland)
A.Barkalov@iie.uz.zgora.pl

HARDWARE REDUCTION IN CPLD-BASED MOORE FSM

Method of hardware reduction in Moore FSM with code transformer is proposed. Method is reduced to use of the specifics of both Moore FSM and CPLD for reducing the number of PAL macrocells in the FSM's logic circuit.

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. Two main elements can be found in CPLD-based SoC, namely, macrocells of PAL and EMB. The macrocells are used for implementing random logic, whereas the EMBs for implementing tabular functions.

Control units are very often implanted as parts of SoC. These devices are very important because they coordinate the interplay of all other system blocks. The model of Moore FSM is very popular for synthesis of control units. 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 CPLD used for implementing a control units. Two peculiarities of Moore FSM can be used for the pseudoequivalent states. A regular nature of output variables (microoperations) is the second specific. It allows using EMBs for implementing the system of microoperations. The main specific of CPLD is a wide fan-in of PAL macrocells (up to 30). Also, the limited amount of product terms let a macrocells should be taken into account (up to 8).

The maximum gain in hardware can reach up to 28%. Let us point out that both proposed and basic FSMs have the same performance. It means that the gain in hardware does not lead to the loss performance.

The scientific novelty of the proposed method is reduced to use of the specifics of both Moore FSM and CPLD for reducing the number of PAL macrocells in the FSM's logic circuit.The practical meaning of the method is reduced to decreasing the chip area occupied by an FSM's logic circuit in comparison with known approaches.

Keywords: CPLD, hardware reduction, Moore FSM, 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.

Specifics 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, $E = \{< b_q, b_t > | b_q, b_t \in B\}$ 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 $Y = \{y_1, ..., y_N\}$ is a set of microoperations [9]. Conditional vertices $b_q \in E_2$ include elements of the set of logical conditions $X = \{x_1, ..., x_L\}$. Both vertices b_0 and b_E correspond to the initial state $a_1 \in A$, where $A = \{a_1, ..., a_M\}$ 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:

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

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

In (1)-(2) the set $T = \{T_1, ..., T_R\}$ is a set of state variables used for state assignment ($R = \lceil \log_2 M \rceil$); the set $\Phi = \{D_1, ..., D_R\}$ 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 $x_l \in X$ (or their complements) determining the $< a_m, a_s >$; 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 $\Pi_A = \{B_1, ..., B_I\}$ 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 encodeing, where $|\tau| = R_1$. In this, the following model U_1 is used (Fig. 1).

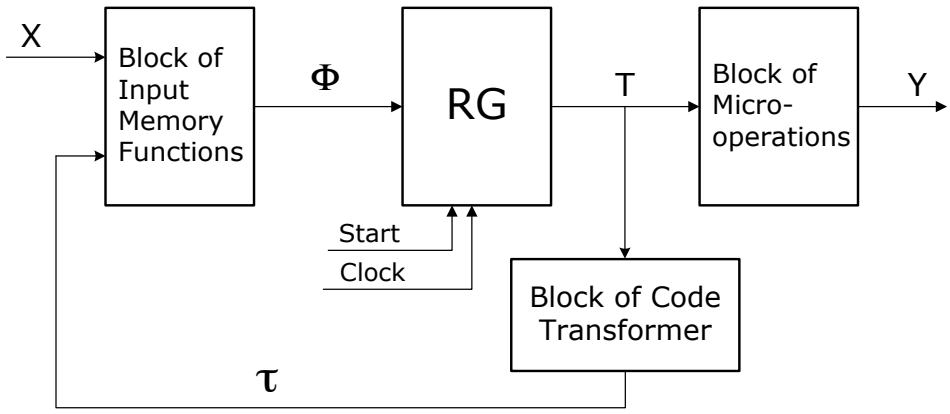

Figure 1 – Structural diagram of Moore FSM U_1

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

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

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 $\upsilon = \{1,2,4,8,16\}$.

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 $\Pi_B \cap \Pi_C = \varnothing$. 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 t_F be a fixed number of outputs of EMB. Let q be the number of block cells if there is $t_F = 1$. For the BMO, the value of t_F is determine as

$$t_F = \lceil q / M \rceil \quad (6)$$

There are t_S outputs in all blocks of BMO:

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

Obviously, Δ_t outputs are redundant, where there is

$$\Delta_t = t_S - N \quad (8)$$

Let us consider the situation when the following condition takes place:

$$\Delta_t = 0 \qquad (9)$$

It means that all outputs of EMBs are used for implementing microoperations.

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. Obviously, only the codes of states $a_m \in A(\Pi_E)$ should be transformed. Here $A(\Pi_E) \subseteq A$ is a set of states belonging to classes $B_i \in \Pi_E$. It is enough R_2 variables for encoding the classes $B_i \in \Pi_E$:

$$R_2 = \lceil \log_2(|\Pi_E| + 1) \rceil \qquad (10)$$

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

In comparison with U_1, the FSM U_2 has the following specifics:

– the BIMF generates functions

$$\Phi = \Phi(T, \tau, X) \qquad (11)$$

– the BCT generates the codes of classes $B_i \in \Pi_E$;

– the variables $T_r \in T$ represent the states a_m from the classes $B_i \in \Pi_C$ and the classes $B_i \in \Pi_D$.

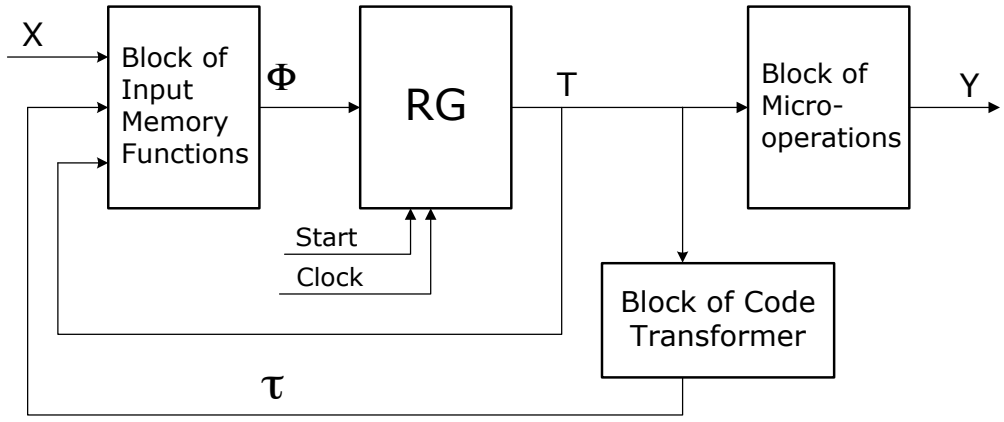

Figure 2 – Structural diagram of Moore FSM U_2

The number of required inputs of PAL is increased from $L + R_1$ (for U_1) to $L + R + R_2$ (for U_2). But it does not result in increasing the required number of PALs due their high fan-in [7]. There are equal times of cycles for both models. Therefore, the proposed method allows preserving performance and decreasing the hardware amount.

The proposed design method for FSM 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 constructing sets Π_D and 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 logic circuit of FSM U_2.

Investigation of efficiency of proposed method

The base of conducted investigation is the transition from a single GSA to the classes of GSAs. The method is based on the probabilistic approach [7, 11]. To estimate the hardware amount, the transition from standard PALs to matrix structures is executed [6]. To find the general tendencies, the transition is made from absolute to relative values.

The matrix circuit of FSM U_1 is shown in Fig. 3. Each combinatorial element includes two matrices, namely a conjunctive matrix (the symbol “&”) and disjunctive (the symbol “V”).

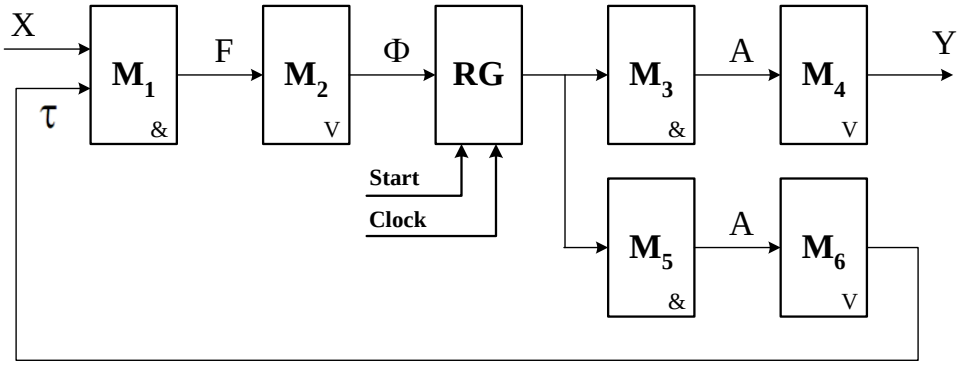

Figure 3 – Matrix implementation of FSM U_1

The matrices M_1 and M_2 form the BIMF. It has the numbers of terms and feedback variables the same as for equivalent Mealy FSM. The matrices M_3 and M_4 form the BMO. The outputs of M_3 correspond to the conjunctions $A_m (m = \overline{1,M})$ determined by the codes $K(a_m)$. The matrices M_5 and M_6 form the BCT. The sizes (areas) $S(M_i)$ of matrices $M_i (i = \overline{1,6})$ are determined in some abstract units of area. They could be found as a following:

$$S(M_1) = 2(L + R_1)H_0; \quad S(M_2) = H_0R;$$
$$S(M_3) = S(M_5) = 2R \cdot 2^R; \quad S(M_4) = 2^R \cdot N; \qquad (12)$$
$$S(M_6) = 2^R \cdot R_1.$$

The variable P_1 is used for estimation the classes of GSAs. It is equal to the part of operator vertices in GSA [11]. Using P_1 together with results [7], the following expressions could be found for arguments of (12):

$$L \approx 0.75 \cdot (1 - P_1)K;$$
$$R_1 =]\log_2(3.55 + 0.3 \cdot P_1 \cdot K)[; \qquad (13)$$
$$H_0 = 4.44 + P_1 \cdot K;$$
$$R =]\log_2 P_1 \cdot K[.$$

In (13), the symbol K stands for the number of vertices in GSA Γ. Let us introduce the coefficient k_p. It determines the ratio of one abstract area unit for the PAL macrocells respectively the same parameter of EMB. It allows findings the following formula for the area of matrix circuit of FSM U_1:

$$S(U_1) = k_pH_0(2L + 2R_1 + R) + 2^R(4R + N + R_1) \qquad (14)$$

Taking into account the formulas (13), the expression (14) could be represented as some function of arguments K, P_1, k_p, N.

Using the same approach, the matrix circuit can be constructed for FSM U_2 (Fig.4).

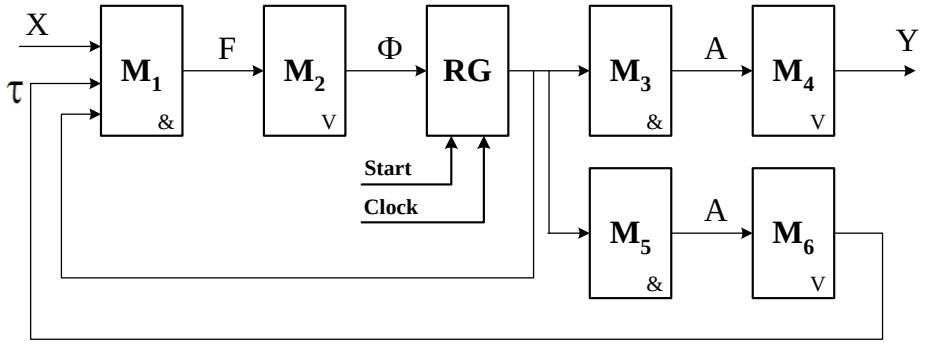

Figure 4 – Matrix implementation of FSM U_2

It's area is determined as the following:

$$S(U_2) = k_pH_0(2L + 2R_1 + R) + 2^R(2R + N + R_1) \qquad (15)$$

This formula also could be represented as some functions of arguments k_1, P_1, k_p, N

To find the area of effective use of the proposed method. It is enough to find the area where the following condition takes place:

$$f_1 = \frac{S(U_2)}{S(U_1)} < 1 \qquad (16)$$

Some results of our investigations are shown in Fig. 5 and Fig 6. They are conducted for $K = \overline{100,1000}, k_p = 0.2,\ N \in \{10,50,100\}$ and $P_1 \in \{0.3;0.7\}$.

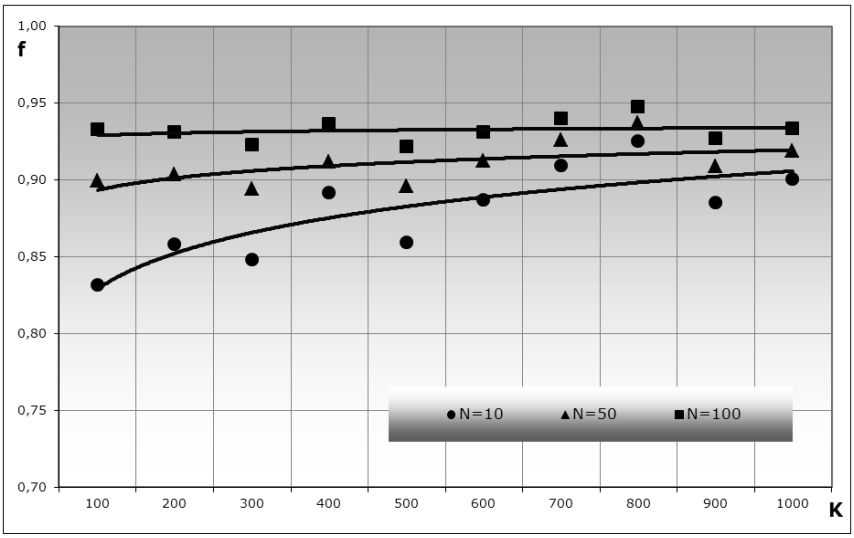

Figure 5 – Function f_1 for $P_1 = 0.3, k_p = 0.2$

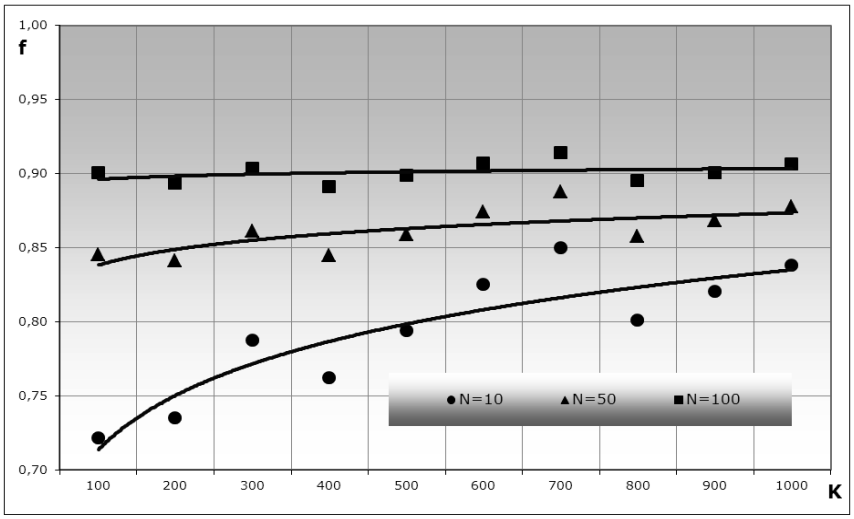

Figure 6 – Function f_1 for $P_1 = 0.7, k_p = 0.2$

The following conclusions can be made on the base of the investigation. The FSM U_2 always requires less amount of hardware than the equivalent FSM U_1. The gain is increased with the decreasing of the numbers of microoperations and vertices, as well as with increasing the part of operator vertices. Maximum gain is equal 28% ($K = 100, N = 10,\ P_1 = 0.7$).

Conclusion

The proposed method allows hardware reduction in comparison with the Moore FSM having a code transformer. The maximum gain in hardware can reach up to 28%. Let us point out that both FSMs $U_1(\Gamma)$ and $U_2(\Gamma)$ have the same performance. It means that the gain in hardware does not lead to the loss performance.

The scientific novelty of the proposed method is reduced to use of the specifics of both Moore FSM and CPLD for reducing the number of PAL macrocells in the FSM's logic circuit.

The practical meaning of the method is reduced to decreasing the chip area occupied by an FSM's logic circuit in comparison with known approaches.

Literature

1. Barkalov A., Titarenko L. Logic Synthesis for FSM-based Control Units. – Berlin: Springer, 2009. – 234 pp.
2. Грушницкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем с использованием микросхем программируемой логики. – СПб: БХВ. – Петербург, 2002. – 608 с.
3. Maxfield C. The Design Warriors Guide to FPGAs. – Elseveir, 2004. – 541p.
4. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем. – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с.
5. Баркалов А.А., Палагин А.В. Синтез микропрограммных устройств управления. – Киев: Институт кибернетики НАН Украины, 1997. – 136с.
6. Baranov S. Logic Synthesis for Control Automata. – Kluwer Academic Publishers, 1994. – 312 pp.
7. Баркалов А.А. Синтез устройств управления на программируемых логических устройствах. – Донецк: ДНТУ, 2002. – 262 с.
8. Kania D. Synteza logiczna przeznaczona dla matrycowych struktur programowalnych typuPAL. – Zeszyty naukowe Politechniki Śląskiej, Gliwice, 2004. – 240 pp.
9. DeMicheli G. Synthesis and Optimization of Digital Circuits. – McGraw-Hill, 1994. – 636p.
10. Баркалов А. А. Принципы оптимизации логической схемы микропрограммного автомата Мура. // Кибернетика и системный анализ. – 1998, №1.
11. Новиков Г.И. оценка эффективности параллельной выборки микрокоманд // Известия вузов СССР. Приборостроение. – 1974, №2. – с.71-73

Надійшла до редакції 12.10.2013 р. Рецензент: к.т.н., доц.Приходько Т.О.

О.О. Баркалов[1], Л.О. Титаренко[2], С.О. Цололо[1], О.М. Мірошкін[1]
[1]Донецький національній технический університет, [2]Зеленогурьский університет

Зменшення витрат апаратури у схемі МПА Мура в базисі CPLD. В роботі запропоновано метод зменшення апаратурних витрат у схемі автомата Мура, що заснований на використанні вільних виходів вбудованих блоків пам'яті для представлення кодів класів псевдоеквівалентних станів. Запропонований підхід дозволяє зменшити число використовуваних блоків PAL при збереженні швидкодії, при цьому проведені дослідження показали, що максимальна економія в апаратурних витратах для запропонованої моделі автомата Мура досягає 28%.
Ключові слова: CPLD, витрати апаратури, МПА Мура, PAL, макроосередки

А.А. Баркалов[1], Л.А. Титаренко[2], С.А. Цололо[1], А.Н. Мирошкин[1]
[1]Донецкий национальный технический университет, [2]Зеленогурский университет

Уменьшение аппаратурных затрат в схеме МПА Мура в базисе CPLD. В работе предлагается метод уменьшения аппаратурных затрат в схеме автомата Мура, основанный на использовании свободных выходов встроенных блоков памяти для представления кодов классов псевдоэквивалентных состояний. Предлагаемый подход позволяет уменьшить число используемых блоков PAL при сохранении быстродействия на прежнем уровне. При этом проведенные исследования показали, что максимальный выигрыш в аппаратурных затратах для предлагаемой модели автомата Мура достигает 28%.
Ключевые слова: CPLD, аппаратурные затраты, МПА Мура, PAL, макроячейки