

# КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

A.A. Barkalov, L.A. Titarenko,  
Y.E. Vizor, A.V. Matvienko

## ENCODING OUTPUT VARIABLES IN COMBINED AUTOMATION

*The method is proposed for reducing the number of LUT elements in the circuit of combined automation. Conditions allowing for the circuit to be implemented with a single EMB and minimum number of LUTs are shown.*

*Key words: combined FSM, FPGA, LUT, EMB, encoding of collections of micro-operations, synthesis.*

*Запропоновано метод зменшення числа елементів LUT у схемі суміщеного автомата. Показані умови реалізації автомата на одному блоці пам'яті EMB і мінімальній кількості елементів LUT.*

*Ключові слова: суміщений автомат, FPGA, LUT, EMB, кодування наборів мікрооперацій, синтез.*

*Предложен метод уменьшения числа элементов LUT в схеме совмещенного автомата. Показаны условия реализации автомата на одном блоке памяти EMB с минимальным числом элементов LUT.*  
*Ключевые слова: совмещенный автомат, FPGA, LUT, EMB, синтез.*

© А.А. Баркалов, Л.А. Титаренко,  
Я.Е. Визор, А.В. Матвиенко,  
2018

УДК 004.274

А.А. БАРКАЛОВ, Л.А. ТИТАРЕНКО  
Я.Е. ВИЗОР, А.В. МАТВИЕНКО

## КОДИРОВАНИЕ ВЫХОДНЫХ ПЕРЕМЕННЫХ В СОВМЕЩЕННОМ АВТОМАТЕ

**Введение.** При реализации цифровых систем часто возникает задача уменьшения аппаратных затрат как в системе в целом, так и устройстве управления (УУ) системы [1]. Методы оптимизации схемы УУ в значительной степени зависят от: 1) модели, представляющей поведение УУ и 2) особенностей элементного базиса [2, 3]. В настоящей работе мы рассматриваем случай, когда для синтеза схемы УУ используется модель совмещенного микропрограммного автомата (СМПА).

В совмещенном автомате существуют выходные сигналы двух типов [3]. Выходные сигналы автомата Мура существуют в течение такта работы автомата. Выходные сигналы автомата Мили существуют при переходе между состояниями. Это и есть основная особенность совмещенного автомата.

Для реализации схем цифровых систем в настоящее время широко используются СБИС типа *FPGA* (*field-programmable logic arrays*) [4, 5]. Два типа логических элементов, входящих в *FPGA*, могут использоваться для реализации схемы МПА. Первый из них – логические элементы типа *LUT* (*look-up table*), выходы которых могут быть связаны с входами триггеров. Элементы *LUT* имеют ограниченное число входов ( $S \leq 6$ ) и только один выход. Второй тип логических элементов – встроенные блоки памяти типа *EMB* (*embedded memory blocks*). Их важной характеристикой является реконфигурация, при которой меняется число выходов ( $t_F$ ) и ячеек памяти ( $V$ ). При этом общая емкость ( $V_0$ ) является константой:

$$V_0 = 2^{S_A} \times t_F, \quad (1)$$

где  $S_A$  – число адресных входов при данном количестве выходов  $t_F$ . Как правило, существуют следующие конфигурации *EMB*: 32К×1, 16К×2, 8К×4, 4К×8, 2К×16, 1К×32, 512×64 (битов) [4, 5]. Это определяет следующие пары вида  $\langle S_A, t_F \rangle$ :  $\langle 15, 1 \rangle$ ;  $\langle 14, 2 \rangle$ ;  $\langle 13, 4 \rangle$ ;  $\langle 12, 8 \rangle$ ;  $\langle 11, 16 \rangle$ ;  $\langle 10, 32 \rangle$ ;  $\langle 9, 64 \rangle$ .

Наилучшими характеристиками по быстродействию, потребляемой мощности и площади схемы обладают автоматы, реализуемые в базе *EMB* [6, 7]. Однако блоки *EMB* широко используются в операционных автоматах цифровых систем [8, 9]. Поэтому часто для реализации схемы УУ можно использовать только один блок *EMB*. Если его ресурсов не хватает для реализации автомата, необходимо использовать и элементы *LUT*. Как правило, элементы *LUT* применяются для реализации блока замены входных переменных [8]. В настоящей работе мы предлагаем использовать кодирование выходных переменных (микроопераций) автомата Мура для уменьшения числа элементов *LUT* в схеме СМПА.

**Особенности совмещенного автомата и FPGA.** Математическая модель СМПА – вектор  $S = \langle A, X, Y^1, Y^2, \delta, \lambda_1, \lambda_2, a_1 \rangle$ , где  $A = \{a_1, \dots, a_M\}$  – множество внутренних состояний;  $X = \{x_1, \dots, x_L\}$  – множество входных переменных;  $Y^1$  – множество выходных переменных автомата Мили;  $Y^2$  – множество выходных переменных автомата Мура;  $\delta$  – функция переходов;  $\lambda_1$  – функция выходов автомата Мили;  $\lambda_2$  – функция выходов автомата Мура;  $a_1 \in A$  – начальное состояние автомата [6].

Множества  $Y^1$  и  $Y^2$  образуют множество выходных переменных  $Y$ . Для этих множеств справедливы следующие отношения:  $Y = Y^1 \cup Y^2$ ;  $Y^1 \cap Y^2 = \emptyset$ . Введем следующие обозначения:  $|Y^1| = N_1$ ,  $|Y^2| = N_2$  и  $N_1 + N_2 = N$ .

Функция  $\delta$  определяет состояние перехода  $a_s \in A$  на основе текущего состояния  $a_m \in A$  и входных переменных:

$$a_s = \delta(a_m, X). \quad (2)$$

Функции  $\lambda_1$  и  $\lambda_2$  имеют следующий вид:

$$y_n = \lambda_1(a_m, X). \quad (3)$$

$$y_n = \lambda_2(a_m). \quad (4)$$

При синтезе схемы МПА состояния  $a_m \in A$  представляются двоичными кодами  $K(a_m)$  разрядности  $R$ , где  $\lceil \log_2 M \rceil \leq R \leq M$ . Коды состояний представляются внутренними переменными, которые образуют множество  $T = \{T_1, \dots, T_R\}$ . Для задания кода состояния перехода (2) используют функции возбуждения памяти, образующие множество  $\Phi = \{D_1, \dots, D_R\}$ .

Для синтеза схемы СМПА необходимо получить функции (2) – (4). Эти функции определяются следующими системами булевых функций:

$$\Phi = \Phi(T, X); \quad (5)$$

$$Y^1 = Y^1(T, X); \quad (6)$$

$$Y^2 = Y^2(T). \quad (7)$$

Системы функций (5) – (7) определяют структурную схему СМПА, которая подробно рассмотрена в [6, рис. 1].

Как уже отмечалось, особенностью *FPGA* является наличие реконфигурируемых блоков *EMB* и элементов *LUT*, имеющих один выход. Блоки *EMB* часто используются для реализации табличных функций, памяти и других блоков цифровых систем [9, 10]. Поэтому, как правило, схемы автоматов в основном реализуются на *LUT*. Однако, при наличии свободных блоков *EMB*, целесообразно использовать эти блоки совместно с элементами *LUT*. Такой подход позволяет уменьшить площадь схемы СМПА и число межсоединений в ней.

**Реализация схемы СМПА в базисе *EMB* и *LUT*.** В настоящей работе мы рассматриваем случай, когда для реализации схемы СМПА можно использовать только один блок *EMB*. Пусть выполняется следующее условие:

$$2^{L+R} (R+N) > V_0. \quad (8)$$

В этом случае одного блока *EMB* недостаточно для реализации схемы СМПА. Следовательно, схема должна быть реализована в гетерогенном базисе, под которым понимается совместное использование блока *EMB* и элементов *LUT*.

Пусть для реализации схемы СМПА используются микросхемы *FPGA*, для блоков *EMB*, которых имеется конфигурация  $\langle S_A, t_F \rangle$  такая, что:

$$S_A = L+R; \quad (9)$$

$$N_1+R < t_F < N+R. \quad (10)$$

Из условий (9) – (10) следует, что системы (5) – (6) могут быть реализованы на одном блоке *EMB*. Однако систему (7) на этом же блоке реализовать нельзя. Используем для кодирования состояний  $a_m \in A$  коды, разрядность которых определяется формулой:

$$R = \lceil \log_2 M \rceil. \quad (11)$$

Пусть число входов  $S_L$  элемента *LUT* и параметр  $R$  находятся в следующем отношении:

$$R > S_L. \quad (12)$$

Из условий (9) – (10) следует, что система (7) должна реализовываться в виде блока *LUTer*, то есть блока, состоящего из блоков *LUT*. Из условия (12) следует, что для реализации схемы *LUTer* необходимо больше, чем  $N_2$  элементов. Для функций  $y_n \in Y^2$  необходимо применять методы функциональной декомпозиции [12, 13]. Это приводит к увеличению числа уровней в схеме и числа межсоединений. Все это имеет отрицательное влияние на быстродействие и потребляемую мощность схемы СМПА [8].

Пусть в операторных вершинах граф-схемы алгоритма (ГСА)  $\Gamma$  находятся  $Q$  наборов выходных переменных (НВП) автомата Мура. Поставим в соответствие каждому НВП  $Y_q \subseteq Y^2$  двоичный код  $K(Y_q)$  разрядности

$$R_Q = \lceil \log_2 Q \rceil. \quad (13)$$

Используем для кодирования НВП переменные  $z_r \in Z$ , где  $|Z| = R_Q$ .

Пусть выполняется следующее условие:

$$t_F - (N_1 + R) \geq R_Q. \quad (14)$$

Пусть при этом выполняются условия (9) – (10). Тогда блок *EMB* имеет как минимум  $R_Q$  свободных выходов.

Мы предлагаем использовать эти выходы для формирования переменных  $z_r \in Z$ . Это приводит к автомату  $U_1$  (рис. 1).

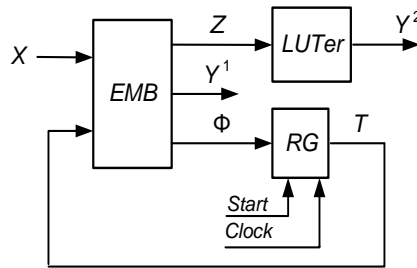


РИС. 1. Структурная схема СМПА  $U_1$

В автомате  $U_1$  блок *EMB* реализует систему функций (5), (6) и систему

$$Z = Z(T). \quad (15)$$

Функции  $z_r \in Z$  используются как входные переменные системы

$$Y^2 = Y^2(Z). \quad (16)$$

Одиночный импульс *Start* устанавливает в *RG* код начального состояния  $a_1 \in A$ . Сигнал *Clock* разрешает изменение содержимого *RG*.

Пусть для некоторого автомата и базиса *FPGA* выполняется условие

$$R_Q \leq S_L. \quad (17)$$

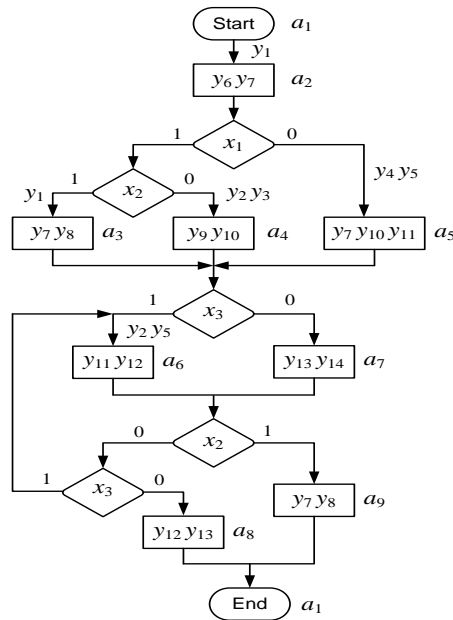
В этом случае для реализации схемы блока *LUTer* достаточно  $N_2$  элемента *LUT*. Это минимальное возможное число для автомата  $U_1$ .

В данной работе предлагается метод синтеза схемы автомата  $U_1$  по ГСА  $\Gamma$ .

**Предлагаемый метод синтеза и пример его применения.** Пусть выражение  $U_i(\Gamma_j)$  означает, что автомат  $U_i$  синтезируется по ГСА  $\Gamma_j$ . Предлагаемый метод синтеза автомата  $U_i(\Gamma_j)$  включает этапы.

1. Отметку ГСА  $\Gamma$  состояниями автомата Мура.
2. Кодирование состояний  $a_m \in A$ .
3. Формирование наборов  $Y_q \subseteq Y^2$ .
4. Кодирование наборов  $Y_q \subseteq Y^2$ .
5. Формирование прямой структурной таблицы автомата.
6. Формирование таблиц блоков *EMB* и *LUTer*.
7. Реализацию схемы в заданном элементном базисе.

Проиллюстрируем этот метод на примере синтеза автомата  $U_1(\Gamma_1)$ , где ГСА  $\Gamma_1$  показана на рис. 2.

РИС. 2. Отмеченная ГСА  $\Gamma_1$ 

Отметка ГСА  $\Gamma_1$  выполнена с использованием правил [1]. Как следует из рисунка, синтезируемый автомат характеризуется множествами:  $A = \{a_1, \dots, a_9\}$ ,  $X = \{x_1, x_2, x_3\}$ ,  $Y^1 = \{y_1, \dots, y_5\}$ ,  $Y^2 = \{y_6, \dots, y_{14}\}$ . Эти множества имеют следующие параметры:  $M = 9$ ,  $L = 3$ ,  $N_1 = 5$ ,  $N_2 = 9$ ,  $N = 14$ . Из условия (11) имеем  $R = 4$ , что дает множества  $T = \{T_1, \dots, T_4\}$  и  $\Phi = \{D_1, \dots, D_4\}$ .

Пусть для реализации схемы автомата используется *FPGA*, для которых: 1) для блока *EMB* имеется конфигурация  $2^7 \times 12$ , то есть  $S_A = 7$ ,  $t_F = 12$ ; 2) элементы *LUT* имеют  $S_L = 3$  входа. При этом для реализации схемы УУ имеется только один блок *EMB*.

Проанализируем условия (9), (10), (12). Условие (9) выполняется, так как  $L + R = 3 + 4 = S_A$ . Поскольку  $N_1 + R = 9$  и  $N + R = 18$ , условие (10) также выполняется. Так как  $R = 4$ , а  $S_L = 3$ , то выполняется условие (12).

Найдем НВП  $Y_q \subseteq Y^2$ . Анализ ГСА  $\Gamma_1$  позволяет найти следующие восемь НВП:  $Y_1 = \emptyset$ ,  $Y_2 = \{y_6, y_7\}$ ,  $Y_3 = \{y_7, y_8\}$ ,  $Y_4 = \{y_9, y_{10}\}$ ,  $Y_5 = \{y_7, y_{10}, y_{11}\}$ ,  $Y_6 = \{y_{11}, y_{12}\}$ ,  $Y_7 = \{y_{13}, y_{14}\}$ ,  $Y_8 = \{y_{12}, y_{13}\}$ . Таким образом,  $Q = 8$  и из (13) имеем  $R_Q = 3$ . Следовательно,  $t_F - (N_1 + R) = 3 \geq R_Q$ , то есть выполняется условие (14). Таким образом, модель  $U_1(\Gamma_1)$  может быть использована для данного элементного базиса.

Коды состояний  $a_m \in A$  не влияют на аппаратные затраты, поэтому мы выбираем тривиальное кодирование состояний:  $K(a_1) = 0000$ , ...,  $K(a_9) = 1000$ .

Поскольку  $S_L = R_Q = 3$ , выполняется условие (17). Поэтому коды НВП  $Y_q \subseteq Y^2$  не влияют на число элементов *LUT* в схеме *LUTer* и наборы  $Y_q \subseteq Y^2$  можно закодировать тривиальным образом (рис. 3).

		$z_1 z_2$			
		00	01	11	10
$z_3$	0	$Y_1$	$Y_3$	$Y_7$	$Y_5$
	1	$Y_2$	$Y_4$	$Y_8$	$Y_6$

РИС. 3. Коды наборов выходных переменных автомата  $U_1(\Gamma_1)$

Прямая структурная таблица (ПСТ) СМПА  $U_1$  формируется по правилам [6, 7]. Она включает столбцы:  $a_m$  – исходное состояние и его код  $K(a_m)$ ;  $a_s$  – состояние перехода и его код  $K(a_s)$ ;  $X_h$  – входной сигнал, инициирующий переход  $\langle a_m, a_s \rangle$ ;  $Y^1_h$  – набор переменных  $y_n \in Y^1$ , формируемый на переходе  $\langle a_m, a_s \rangle$ ;  $\Phi_h$  – набор функций  $D_r \in \Phi$ , равных 1 для записи в  $RG$  кода состояния перехода;  $h$  – номер перехода ( $h = \overline{1, H}$ ). Кроме того, имеется следующее отличие от [6, 7]: в столбце  $a_m$  записывается набор переменных  $z_r \in Z$ , равных 1 в коде  $K(Y_0)$ , формируемом в состоянии  $a_m \in A$ .

В данном примере ПСТ имеет  $H=18$  строк. Часть ПСТ приведена в табл. 1.

Поясним заполнение столбца  $a_m$  переменными  $z_r \in Z$ . Для состояния  $a_1$  код  $K(Y_1) = 000$ . Следовательно, переменные  $z_r \in Z$  для  $a_1$  отсутствуют. Для состояния  $a_2$  код  $K(Y_2) = 001$ , следовательно, в столбце  $a_m$  записана переменная  $z_3$  для состояния  $a_2 \in A$ . Остальная информация для столбца  $a_m$  находится аналогичным образом.

Таблица блока  $EMB$  имеет столбцы:  $K(a_m)$ ,  $X$  (адрес ячейки памяти),  $Y^1$ ,  $Z$ ,  $\Phi$  (содержимое ячейки),  $g$  (номер ячейки). Переходы из каждого состояния занимают  $H(a_m)$  ячеек памяти, где  $H(a_m) = 2^L$ . При этом таблица имеет  $G$  строк, где  $G = 2^{R+L}$ .

ТАБЛИЦА 1. Фрагмент ПСТ автомата  $U_1(\Gamma_1)$

$a_m$	$K(a_m)$	$a_s$	$K(a_s)$	$X_h$	$Y^1_h$	$\Phi_h$	$h$
$a_1$	0000	$a_2$	0001	1	—	$D_4$	1
$a_2$ ( $z_3$ )	0001	$a_3$	0010	$x_1 x_2$	$y_1$	$D_3$	2
		$a_4$	0011	$\overline{x_1 x_2}$	$y_2 y_3$	$D_3 D_4$	3
		$a_5$	0100	$\overline{x_1}$	$y_4 y_5$	$D_2$	4
$a_3$ ( $z_2$ )	0010	$a_6$	0101	$x_3$	$y_2 y_5$	$D_2 D_4$	5
		$a_7$	0110	$\overline{x_3}$	—	$D_2 D_3$	6
$a_4$ ( $z_2 z_3$ )	0011	$a_6$	0101	$x_3$	$y_2 y_5$	$D_2 D_4$	7
		$a_7$	0110	$\overline{x_3}$	—	$D_2 D_3$	8
$a_5$ ( $z_1$ )	0100	$a_6$	0101	$x_3$	$y_2 y_5$	$D_2 D_4$	9
		$a_7$	0110	$\overline{x_3}$	—	$D_2 D_3$	10

В рассматриваемом случае  $G = 128$  и  $H(a_m) = 8$ . В табл. 2 приведена часть таблицы блока *EMB* для нашего примера. В таблице рассмотрены переходы из состояния  $a_2 \in A$ . Поэтому столбец  $g$  начинается с номера 9. В таблице добавлен столбец  $h$ , чтобы было видно соответствие между табл. 1 и 2.

ТАБЛИЦА 2. Фрагмент таблицы блока *EMB* автомата  $U_1(\Gamma_1)$ 

$K(a_m)$	$X$	$Y^1$	$\Phi$	$Z$	$q$	$h$
$T_1 T_2 T_3 T_4$	$x_1 x_2 x_3$	$y_1 y_2 y_3 y_4 y_5$	$D_1 D_2 D_3 D_3$	$z_1 z_2 z_3$		
0 0 0 1	0 0 0	0 0 0 1 1	0 1 0 0	0 0 0	9	4
0 0 0 1	0 0 1	0 0 0 1 1	0 1 0 0	0 0 0	10	4
0 0 0 1	0 1 0	0 1 1 0 0	0 0 1 1	0 0 0	11	3
0 0 0 1	0 1 1	0 1 1 0 0	0 0 1 1	0 0 0	12	3
0 0 0 1	1 0 0	1 0 0 0 0	0 0 1 0	0 0 0	13	2
0 0 0 1	1 0 1	1 0 0 0 0	0 0 1 0	0 0 0	14	2
0 0 0 1	1 1 0 0	1 0 0 0 0	0 0 1 0	0 0 0	15	2
0 0 0 1	1 1 1	1 0 0 0 0	0 0 1 0	0 0 0	16	2

Таблица блока *LUTer* имеет столбцы  $K(Y_q)$ ,  $Y^2$ ,  $q$  и строится тривиальным образом (табл. 3) на основе карты Карно (рис. 3) и ГСА  $\Gamma_1$  (рис. 2).

ТАБЛИЦА 3. Таблица блока *LUTer* автомата  $U_1(\Gamma_1)$ 

$K(Y_q)$	$Y^2$	$q$
$z_1 z_2 z_3$	$y_6 y_7 y_8 y_9 y_{10} y_{11} y_{12} y_{13} y_{14}$	
0 0 0	0 0 0 0 0 0 0 0 0	1
0 0 1	1 1 0 0 0 0 0 0 0	2
0 1 0	0 1 1 0 0 0 0 0 0	3
0 1 1	0 0 0 1 1 0 0 0 0	4
1 0 0	0 0 0 0 1 1 0 0 0	5
1 0 1	0 0 0 0 0 1 1 0 0	6
1 1 0	0 0 0 0 0 0 0 1 1	7
1 1 1	0 0 0 0 0 0 1 1 0	8

Условие (17) выполняется, поэтому табл. 3 соответствует  $N_2 = 9$  таблицам элементов *LUT*. Каждая таблица имеет столбцы  $K(Y_q)$ ,  $y_n$ ,  $q$ . Если условие (17) не выполняется, для реализации схемы *LUTer* необходимо применять методы функциональной декомпозиции [12, 13].

Последний этап предлагаемого метода синтеза связан с применением стандартных средств САПР [4, 5] и в данной статье не рассматривается.

**Выводы.** Предложенный в работе метод позволяет синтезировать схемы СМПА с одним блоком *EMB*. Это достигается за счет кодирования наборов микроопераций, соответствующих входным сигналам автомата Мура. Для реализации микроопераций используются элементы *LUT*. В статье рассмотрен случай,

при котором коды состояний и наборов микроопераций не влияют на число элементов *LUT* в схеме автомата.

Анализ библиотеки [14] показал, что данный метод необходимо применять для 18 % всех примеров из этой библиотеки. Для остальных 82 % достаточно одного блока *EMB* для реализации схемы СМПА. При этом условия (9), (10), (14), (17) выполняются для 92 % всех примеров. В этих случаях наборы микроопераций представляются унитарными кодами.

Дальнейшее направление наших исследований связано с адаптацией предложенного метода для автоматов, у которых нарушаются условия (9), (10), (14), (17). Для этого необходимо разработать методы кодирования состояний и/или наборов микроопераций, позволяющие уменьшить число элементов *LUT* в схеме СМПА.

### СПИСОК ЛИТЕРАТУРЫ

1. Baranov S. Logic Synthesis for Control Automata. *Dordrecht: Kluwer Academic Publishers*. 1994. 312 p.
2. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем. М.: Горячая линия – ТЕЛЕКОМ, 2001. 636 с.
3. Sklyarov V., Skliarova I., Barkalov A., Titarenko L. Synthesis and Optimization of FPGA-based Systems. Berlin: Springer. 2014. 432 p.
4. [www.altera.com](http://www.altera.com).
5. [www.xilinx.com](http://www.xilinx.com).
6. А.А. Баркалов, Л.А. Титаренко, Я.Е. Визор, А.В. Матвиенко. Реализация схемы совмещенного автомата в базе FPGA. *Комп'ютерні засоби, мережі та системи. Збірник наукових праць. Ін-т кібернетики імені В.М. Глушкова НАН України*. Київ. 2016. Випуск 15. С. 10–19.
7. Баркалов А.А., Титаренко Л.А., Визор Я.Е., Матвиенко А.В., В.В. Горина. Уменьшение числа LUT элементов в схеме совмещенного автомата. *Управляющие системы и машины*. 2016. № 3. С. 16–22.
8. Garcia-Vargas L., Senhadji-Navarro R., M. Civit-Balcells A. and Guerra-Gutierrez P. ROM-Based Finite State Machine Implementation in Low Cost FPGAs. *IEEE International Symposium on Industrial Electronics*, Vigo. 2007. P. 2342–2347.
9. Sklyarov V., Skliarova I., Barkalov A., Titarenko L. Synthesis and Optimization of FPGA-based Systems. Berlin: Springer. 2014. 432 p.
10. Грушницкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем с использованием микросхем программируемой логики. СПб: БХВ. Петербург, 2002. 608 с.
11. Баркалов А.А., Палагин А.В. Синтез микропрограммных устройств управления. Киев: ИК НАН Украины, 1997. 136 с.
12. Nowicka M., Łuba T. and Rawski V. FPGA-based decomposition of boolean functions: algorithms and implementations. *Advanced Computer Systems*. 1999. P. 502 – 509.
13. Rawski M., Selvaraj H. and Łuba T. An application of functional decomposition in ROM-based FSM implementation in FPGA devices. *Journal of System Architecture*. 2005. 51(6–7): P. 424 – 434.
14. Yang S. Logic synthesis and optimization benchmarks user guide. *Microelectronics Center of North Carolina*. 1991. 43 p.

Получено 10.07.2018