КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

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

REDUCING NUMBER OF LUTs OF COMBINED MICROPROGRAM AUTOMAT

A synthesis method is proposed for FPGA-based combined automat. The method allows obtaining a circuit with minimum number of LUTs. The optimization is achieved due to transformation of state codes into codes of classes of pseudoequivalent states. An example is given for application of the proposed method.
Key words: combined FSM, FPGA, LUT, EMB, synthesis, structural reduction.

Запропоновано метод синтезу суміщеного мікропрограмного автомата на FPGA. Метод мінімізує число елементів LUT.
Ключові слова: суміщений автомат, FPGA, LUT, EMB, синтез, структурна редукція.

Предложен метод синтеза совмещенного микропрограммного автомата в базисе FPGA. Метод позволяет получить схему с минимальным числом элементов LUT.
Ключевые слова: совмещенный автомат, FPGA, LUT, EMB, синтез, структурная редукция.

УДК 004.274

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

УМЕНЬШЕНИЕ ЧИСЛА ЭЛЕМЕНТОВ ТАБЛИЧНОГО ТИПА В СХЕМЕ СОВМЕЩЕННОГО МИКРОПРОГРАММНОГО АВТОМАТА

Введение. Микропрограммный автомат (МПА) предназначен для осуществления управления в цифровых системах [1, 2]. При синтезе схем МПА возникает актуальная задача уменьшения аппаратурных затрат [3]. Решение этой задачи позволяет уменьшить площадь кристалла СБИС, занимаемую схемой МПА. В свою очередь это позволяет уменьшить энергию, потребляемую схемой [4], что особо важно в мобильных и автономных устройствах. Методы решения этой задачи во многом зависят от типа МПА и элементов базиса, используемого для реализации схемы автомата. В настоящей работе предлагается метод уменьшения аппаратурных затрат в схеме совмещенного МПА (СМПА), реализуемого в базисе СБИС типа *FPGA* (*field-programmable logic arrays*).

Особенностью СМПА является наличие выходных сигналов двух типов [1]. Выходные сигналы автомата Мили зависят от входных переменных и состояний, а автомата Мура – только от состояний [1, 2]. Это позволяет использовать методы оптимизации автоматов Мили и Мура для оптимизации схемы СМПА [5 – 7].

Базис *FPGA* [8, 9] на сегодня широко применяется для проектирования цифровых систем [10, 11]. Для реализации схемы СМПА можно использовать логические элементы типа *LUT* (*look-up table*), программируемые триггера и блоки памяти *EMB* (*embedded memory blocks*). Для соединения элементов схемы и ее связи с другими блоками системы используется программируемая матрица межсоединений [8, 9].

Одним из подходов к решению задачи уменьшения аппаратурных затрат в схемах МПА является увеличение количества структурных уровней [10, 12]. В случае использования *FPGA* такой подход позволяет уменьшить требования к числу адресных входов *LUT* и *EMB*. В свою очередь это позволяет уменьшить число элементов по сравнению с одноуровневыми схемами.

Наибольший эффект наблюдается при одновременном применении нескольких методов оптимизации схем. В данной статье предлагаем использовать три метода для оптимизации схемы СМПА. Это методы замены входных переменных, разделения множества входных переменных и преобразования кодов состояний в коды классов кодов псевдоэквивалентных состояний (ПЭС) [7, 10, 11, 13].

Анализ модели и элементного базиса. Математической моделью СМПА является восьмикомпонентный вектор $S = < A, X, Y^1, Y^2, \delta, \lambda_1, \lambda_2, a_1 >$.

Вектор S включает следующие компоненты: $A=\{a_1,\ldots, a_M\}$ – множество внутренних состояний; $X=\{x_1,\ldots, x_L\}$ – множество входных переменных; Y^1 – множество выходных переменных автомата Мили; Y^2– множество выходных переменных автомата Мура; δ – функция переходов; λ_1 – функция выходов автомата Мили; λ_2 – функция выходов автомата Мура; $a_1 \in A$ – начальное состояние СМПА.

Множества Y^1 и Y^2 образуют множество выходных переменных Y. При этом $Y^1 \cup Y^2 = Y$ и $Y^1 \cap Y^2 = \varnothing$. Введем следующие обозначения: $|Y|=N=N_1+N_2$; $|Y^1|=N_1$; $|Y^2|=N_2$.

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

$$a_s = \delta(a_m, X). \tag{1}$$

Функции λ_1 и λ_2 имеют следующий вид:

$$y_n = \lambda_1(a_m, X). \tag{2}$$

$$y_n = \lambda_2(a_m). \tag{3}$$

Для реализации схемы СМПА состояния $a_m \in A$ необходимо закодировать двоичными кодами $K(a_m)$. Коды состояний хранятся в специальном регистре *RG*, состоящим из R триггеров с общими входами обнуления (*Start*) и синхронизации (*Clock*). Как правило, триггера имеют входы типа D [3]. Число бит R кода $K(a_m)$ находится в интервале $\lceil \log_2 M \rceil \le R \le M$. Будем рассматривать случай, когда

$$R = \lceil \log_2 M \rceil. \tag{4}$$

Для кодирования состояний используются внутренние переменные $T_r \in T$, где $T=\{T_1,\ldots,T_R\}$. Для изменения содержимого RG используются функции возбуждения памяти, образующие множество $\Phi = \{D_1,\ldots, D_R\}$.

Для синтеза схемы СМПА необходимо найти функции (1) – (3). Эти функции определяются, соответственно, следующими системами функций:

$$\Phi = \Phi\,(T, X); \tag{5}$$

$$Y^1=Y^1(T, X); \quad (6)$$
$$Y^2=Y^2(T). \quad (7)$$

Системы функций (5) – (7) определяют структурную схему СМПА (рис. 1).

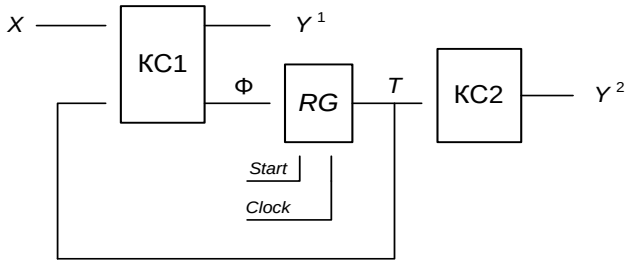

РИС. 1. Структурная схема совмещенного МПА

В схеме на рис. 1 блок КС1 реализует системы (5) и (6), а КС2 – функции (7). Сигнал *Start* обнуляет регистр *RG*, устанавливая код начального состояния $K(a1)$. Сигнал *Clock* инициирует переключение *RG*, соответствующее функции (1).

Особенность *FPGA* – это наличие элементов памяти двух типов. Первый тип – элементы *LUT* , имеющие S адресных входов и один выход. При этом параметр S относительно мал ($S \le 6$) [8, 9]. Выходы *LUT* могут быть связаны с входом триггера. Таким образом, регистр *RG* является распределенным. Второй тип *RAM* – элементы *EMB* . Их важной характеристикой является возможность реконфигурации. При реконфигурации меняется число выходов (t_F) и адресных входов (S_A). При этом общая емкость (V_0) является константой: $V_0 = 2^{S_A} \times t_F$.

Типичные конфигурации *EMB* – это следующие: 64K×1, 32K×2, 16K×4, 8K×8, 4K×16, 2K×32, 1K×64 (битов) [8, 9]. Здесь первый элемент пары определяет число ячеек памяти ($V = 2^{S_A}$), а второй – число выходов. Итак, для *EMB* следующие пары вида $< S_A, t_F >$: $< 16, 1 >, < 15, 2 >, ..., < 10, 64 >$. Следовательно, *EMB* можно "настраивать" на системы (5) – (7). Это позволяет уменьшить число блоков памяти в схеме СМПА [5 – 7].

Реализация совмещенных автоматов в базисе *FPGA*. Как показано в работах [5 – 7], существуют две тривиальные схемы совмещенного МПА в базисе *FPGA*. В первом случае (модель U_1) блоки КС1 и КС2 реализуются в виде блока *LUTer*. При этом под *LUTer* понимается схема, состоящая из элементов *LUT*. Недостатком модели U_1 является большое число уровней *LUT* и межсоединений между ними [3]. Во втором случае (модель U_2) блоки КС1 и КС2 реализуются на одном блоке *EMB*. Это приводит к схеме с наименьшей площадью, наибольшим быстродействием и наименьшей потребляемой энергией (по сравнению с другими возможными схемами) [11]. Однако эта модель может применяться только для достаточно простых автоматов, для которых выполняется условие

$$2^{R+L} \cdot (N_1 + N_2 + R) \le V_0 . \quad (8)$$

При нарушении условия (8) необходимо использовать методы структурной редукции [12]. Наиболее часто используют метод замены входных переменных [10]. В этом случае множество X заменяется множеством дополнительных переменных $P=\{p_1,\ldots,p_G\}$, где $G << L$. Параметр G определяется, как минимум из $|X(a_m)|$, где $X(a_m) \subseteq X$ множество входных переменных, определяющих переходы из состояния $a_m \in A$.

Для замены входных переменных (ЗВП) необходимо найти систему функций

$$P = P(T,X)\,. \qquad (9)$$

Система (9) реализуется на *LUT* элементах, что определяет модель U_3 (рис. 2).

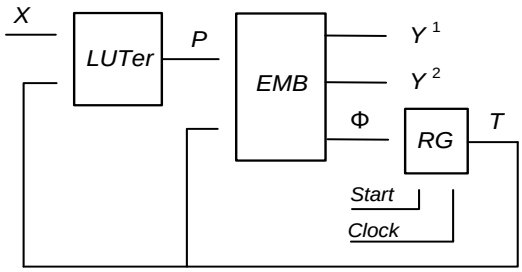

РИС. 2. Структурная схема СМПА U_3

В автомате U_3 блок *LUTer* реализует систему (9), а блок *EMB* – систему (7) и системы

$$\Phi = \Phi(T,P); \qquad Y^1 = Y^1(T,P)\,.$$

Модель U_3 применима, если выполняется условие

$$2^{G+R} \cdot (N_1 + N_2 + R) \le V_0\,. \qquad (10)$$

Как показал анализ библиотеки [14] условие (10) выполняется для 82 % имеющихся в ней автоматов.

Для уменьшения числа элементов *LUT* в блоке *LUTer* необходимо уменьшить число аргументов в системе функций (9). В настоящей работе предлагается один из методов решения этой задачи. При этом алгоритм управления представляется в виде граф-схемы алгоритма (ГСА) [1].

Основная идея предлагаемого метода. Пусть состояния a_m, $a_s \in A$ отмечают вершины ГСА Г, выходы которых связаны с входом одной и той же вершины ГСА. Такие состояния называются псевдо эквивалентными [13]. Пользуясь этим определением, можно найти разбиение $\Pi_A = \{B_1,\ldots, B_I\}$ множества состояний A на классы ПЭС.

Закодируем классы $B_i \in \Pi_A$ двоичными кодами $K(B_i)$ разрядности R_1, где:

$$R_1 = \lceil \log_2 I \rceil. \qquad (11)$$

Используем для кодирования классов $B_i \in \Pi_A$ элементы множества $\tau = \{\tau_1,...,\tau_{R1}\}$.

Пусть для ГСА Г найдены величины параметров R, R_1, N_1, N_2, G. Пусть для реализации схемы СМПА используются микросхемы *FPGA*, включающие блоки *EMB* с конфигурацией $< S_A, t_F >$, для которой выполняются следующие условия:

$$S_A > G + R; \quad (12)$$

$$t_F \geq N_1 + N_2 + R + R_1. \quad (13)$$

Условие (12) свидетельствует о наличии свободных адресных входов *EMB* при замене входных переменных. Условие (13) свидетельствует о том, что на одном блоке *EMB* можно реализовать следующие системы функций:

$$Y^1 = Y^1(T,P); \quad (14)$$

$$\Phi = \Phi(T,P); \quad (15)$$

$$\tau = \tau(T). \quad (16)$$

Кроме того, на этом блоке реализуется и система функций (7).

Определим величину параметра

$$\Delta_S = S_A - (G + R). \quad (17)$$

Очевидно, переменные $p_g \in P$ и $x_l \in X^2$, где $|X^2| = \Delta_S$, можно подать на входы блока *EMB*. Тогда блок ЗВП должен преобразовать переменные $x_l \in X^1$, где $|X^1| = L - \Delta_S$.

При выполнении условий (12) – (13) предлагаем модель U_4 (рис. 3).

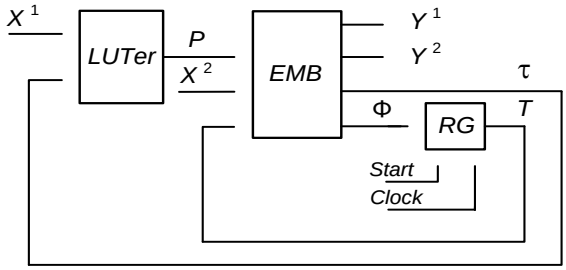

РИС. 3. Структурная схема СМПА U_4

В автомате U_4 блок *LUTer* реализует систему функций

$$P = P(\tau, X^1). \quad (18)$$

Блок *EMB* реализует системы функций (7), (14) – (16).

В настоящей работе предлагается метод синтеза СМПА U_4 по ГСА Г. Метод включает этапы.

1. Формирование множеств A, Π_A, Y^1 и Y^2.
2. Формирование множества P.
3. Разбиение множества X на классы Y^1 и Y^2.
4. Кодирование состояний и классов ПЭС.
5. Формирование таблиц *LUT* блока *LUTer*.

6. Формирование прямой структурной таблицы СМПА U_4.
7. Формирование таблицы блока *EMB*.
8. Реализация схемы СМПА в заданном элементном базисе.

Пример применения предложенного метода. Рассмотрим пример синтеза СМПА U_4 по ГСА Γ_1 (рис. 4). Для отметки состояний мы использовали подход, предложенный в работах [5 – 7]. Операторные вершины отмечаются одинаковыми состояниями, если: 1) их выходы связаны с входом одной и той же вершины ГСА, 2) в этих операторных вершинах нет выходных переменных $y_n \in Y^2$.

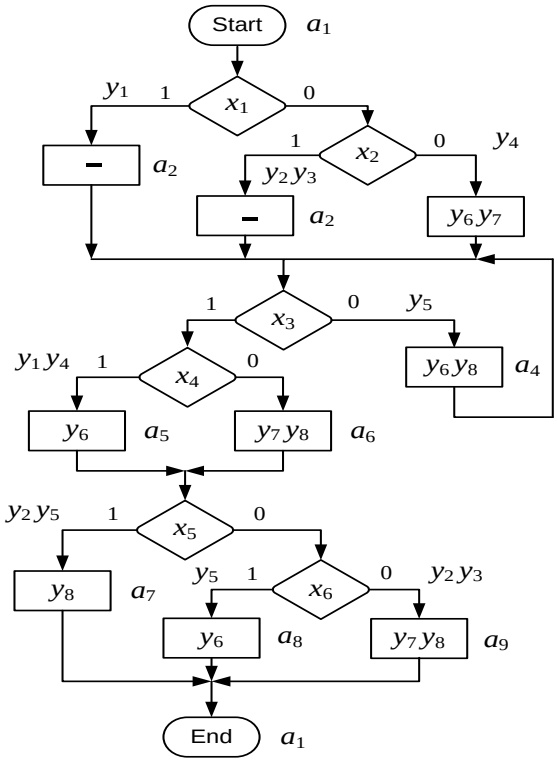

РИС. 4. Отмеченная ГСА Γ_1

Из рис. 4 можно найти следующие множества: $A = \{a_1,..., a_6\}$, $X = \{x_1,..., x_6\}$, $Y = \{y_1,..., y_8\}$, $Y^1 = \{y_1,..., y_5\}$ и $Y^2 = \{y_6, y_7, y_8\}$. Это дает следующие параметры: $M = 9$, $L = 6$, $N = 8$, $N_1 = 5$, $N_2 = 3$. Из (4) имеем $R = 4$, что дает множества $T = \{T_1,..., T_4\}$ и $\Phi = \{D_1,..., D_4\}$.

Используя определение ПЭС [13], можно найти множество $\Pi_A = \{B_1,..., B_4\}$, где $B_1 = \{a_1\}$, $B_2 = \{a_2, a_3, a_4\}$, $B_3 = \{a_5, a_6\}$ и $B_4 = \{a_7, a_8, a_9\}$. Итак, $I = 4$, что дает $R_1 = 2$. Для ГСА Γ_1 выполняются условия (11) и (13).

Анализ ГСА Γ_1 показывает, что $G = 2$ (переходы из состояний $a_m \in A$ зависят от не более, чем двух переменных $x_l \in X$). Таким образом, имеем множество $P = \{p_1, p_2\}$.

Пусть среди конфигураций *EMB* имеется конфигурация <8, 16> с $S_A = 8$ и $t_F = 16$. Для данной ГСА имеем: $G + R = 6 \le S_A$ и $N + R + R_1 = 14 \le t_F$. Таким образом, модель U_4 может быть использована, так как выражение (17) дает $\Delta_S = 2$. Следовательно, множество X можно разбить так, что $|X^2| = 2$ и $|X^1| = 4$.

Разбиение множества X целесообразно выполнить так, чтобы уменьшить количество столбцов с входными переменными в таблице ЗВП. Например, для «удаления» столбца B_3 достаточно выполнить разбиение следующим образом: $X^1=\{x_1, \ldots, x_4\}$ и $X^2=\{x_5, x_6\}$. Такой подход позволяет получить следующую таблицу ЗВП (табл. 1).

ТАБЛИЦА 1. Таблица ЗВП автомата U_4

B_1	B_1	B_2	B_3	B_4
p_1	x_1	x_3	–	–
p_2	x_2	x_4	–	–

Очевидно, коды $K(B_3)$ и $K(B_4)$ могут быть использованы для минимизации уравнений (18). При этом возможно дальнейшее уменьшение числа аргументов в функциях (18).

Коды состояний не влияют на число блоков *EMB* в схеме U_4. Поэтому состояния $a_m \in A$ можно закодировать тривиальным образом: $K(a_1)$=0000, … , $K(a_9)$ = 1000. Классы $B_i \in \Pi_A$ необходимо закодировать так, чтобы классы $B_1, B_2 \in \Pi_A$ имели соседние коды. Например, можно закодировать классы следующим образом: $K(B_1) = 00$, $K(B_2) = 01$, $K(B_3) = 10$ и $K(B_4) = 11$.

Для формирования таблиц элементов *LUT* блока *LUTer* необходимо найти уравнения (18). С учетом соседних кодов классов $B_1, B_2 \in \Pi_A$ из табл. 1 имеем:

$$p_1 = \overline{\tau_2} x_1 \vee \tau_2 x_3; \qquad p_2 = \overline{\tau_2} x_2 \vee \tau_2 x_4. \tag{19}$$

Пусть элементы *LUT* для данной микросхемы *FPGA* имеют S = 3. Тогда любое уравнение системы (19) реализуется на одном элементе *LUT*, что дает схему (рис. 5).

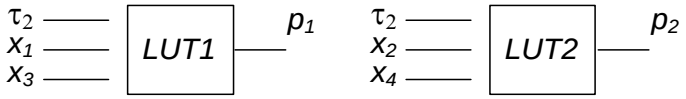

РИС. 5. Схема блока *LUTter* автомата U_4

Каждый из элементов *LUT* задается тривиальной таблицей истинности. Например, элемент *LUT* 1 задается табл. 2. Столбец q содержит номер строки таблицы.

При $\tau_2 = 0$, имеем $p_1 = x_1$ (строки 1 – 4), при $\tau_2 = 1$, имеем $p_1 = x_3$ (строки 5 – 8). Аналогичным образом строится таблица элемента $LUT2$.

ТАБЛИЦА 2. Таблица элемента $LUT1$

$\tau_2\ x_1 x_3$	p_1	q	$\tau_2 x_1 x_3$	p_1	q
0 0 0	0	1	1 0 0	0	5
0 0 1	0	2	1 0 1	1	6
0 1 0	1	3	1 1 0	0	7
0 1 1	1	4	1 1 1	1	8

Таблица ПСТ автомата U_4 включает следующие столбцы: a_m – исходное состояние; $K(a_m)$ – код состояния $a_m \in A$; a_S – состояние перехода; $K(a_S)$ – код состояния $a_S \in A$; P_h – конъюнкция переменных $p_g \in P$, определяющая переход $<a_m, a_S>$; $Y^1{}_h$ – набор выходных переменных $y_n \in Y^1$, формируемый на переходе $<a_m, a_S>$; Φ_h – набор функций $D_r \in \Phi$, необходимых для записи в RG кода $K(a_S)$; h – номер перехода ($h \in \overline{1, H}$). Кроме того, в столбце a_m записываются переменные $y_n \in Y^2$, формируемые в состоянии $a_m \in A$. Для нашего примера ПСТ имеет H=21. В табл. 3 приведены первые строки ПСТ для автомата U_4.

ТАБЛИЦА 3. Фрагмент ПСТ автомата U_4

a_m	$K(a_m)$	a_S	$K(a_S)$	X_h	P_h	$Y^1{}_h$	Φ_h	h
a_4 ($y_6\, y_8$)	0011	a_5	0100	–	$p_1 p_2$	$y_1\, y_2$	D_2	10
		a_6	0101	–	$p_1 \overline{p_2}$	–	$D_2\, D_4$	11
		a_4	0011	–	$\overline{p_1}$	y_5	$D_3\, D_4$	12
a_5 (y_6)	0100	a_7	0110	x_5	–	$y_2\, y_5$	$D_2\, D_3$	13
		a_8	0111	$\overline{x_5} x_6$	–	y_5	$D_2\, D_3\, D_4$	14
		a_9	1000	$\overline{x_5}\,\overline{x_6}$	–	$y_2\, y_3$	D_1	15

В табл. 3 заданы переходы из состояний a_4 и a_5. Выходные переменные $y_n \in Y^2$ берутся из операторных вершин ГСА Г1 (рис. 4). Переходы из состояния a_4 зависят от переменных x_3, $x_4 \in X^1$. Поэтому в столбце X_h стоит пропуск (строки 10 – 12). Так как $p_1 = x_3$ и $p_2 = x_4$ для состояний класса $B_2 \in \Pi_A$, то $x_3 x_4$ соответствует $p_1 p_2$, $x_3 \overline{x_4} = p_1 \overline{p_2}$ и $\overline{x_3} = \overline{p_1}$. Переходы из состояния $a_5 \in A$ зависят от переменных x_5, $x_6 \in X^2$. Поэтому в строках 13 – 15 табл. 3 стоит пропуск в столбце P_h. Аналогичным образом заполняются все строки ПСТ.

Таблица блока *EMB* включает следующие столбцы: $K(a_m)$, X^2, P (адрес ячейки памяти), Y^1, Y^2, Ф, τ (содержимое ячейки памяти), q – номер ячейки памяти ($q = \overline{1, Q}$). Параметр Q определяется как: $Q = 2^{R+G+\Delta S}$. Переходы из состояний $a_m \in A$ задаются с помощью $H(a_m)$ строк таблицы, где $H(a_m) = 2^{G+\Delta S}$.

В рассматриваемом примере Q = 265 и $H(a_m)$ = 16. В табл. 4 приведен фрагмент таблицы блока *EMB* для нашего примера.

ТАБЛИЦА 4. Фрагмент таблицы блока *EMB* автомата U_4

$K(a_m)$	X^2	P	Y^1	Y^2	Ф	τ	q	h
$T_1 T_2 T_3 T_4$	x_5, x_6	$p_1 p_2$	$y_1 y_2 y_6 y_4 y_5$	$y_6 y_7 y_8$	$D_1 D_2 D_3 D_4$	$\tau_1 \tau_2$		
0 0 1 1	0 0	0 0	0 0 0 0 1	1 0 1	0 0 1 1	0 1	49	12
0 0 1 1	0 0	0 1	0 0 0 0 1	1 0 1	0 0 1 1	0 1	50	12
0 0 1 1	0 0	1 0	0 0 0 0 0	1 0 1	0 1 0 1	0 1	51	11
0 0 1 1	0 0	1 1	1 0 0 1 0	1 0 1	0 1 0 0	0 1	52	10
0 0 1 1	0 1	0 0	0 0 0 0 1	1 0 1	0 0 1 1	0 1	53	12

Таблица блока *EMB* строится на основе ПСТ. Табл. 4 задает часть переходов из состояния $a_4 \in A$. Переходы из состояний $a_1 - a_3$ занимают 48 строк. Поэтому первая строка табл. 4 имеет q = 49. Так как $a_4 \in B_2$ с кодом $K(B_2)$ = 01, то код 01 записан в столбце τ во всех строках табл. 4. В состоянии $a_4 \in A$ формируются переменные $y_6\, y_7\, y_8$. Поэтому код 101 записан в столбце Y^2 во всех строках табл. 4. Столбец h добавлен, чтобы показать соответствие между ПСТ и таблицей блока *EMB*.

Последний этап предлагаемого метода связан с использованием стандартных пакетов для реализации схем в базисе *FPGA*. При этом таблицы блоков *LUTer* и *EMB* преобразовываются в битовые потоки (*bit-stream*). В данной статье этот этап не рассматривается.

Выводы. Предложенный в работе метод позволяет уменьшить число элементов *LUT* в схеме СМПА по сравнению с известными методами. Это достигается за счет преобразования кодов состояний МПА в коды классов псевдоэквивалентных состояний. Такой подход позволяет уменьшить число адресных входов в блоке замены входных переменных. Дополнительная оптимизация возможна за счет соответствующего кодирования классов ПЭС. Такая возможность реализована в данной статье.

Метод целесообразно использовать, если замена входных переменных позволяет получить схемы с одним блоком *EMB*. Анализ библиотеки [14] показал,

что к этому классу относится 82 % стандартных автоматов. Кроме того, замена кодов $K(a_m)$ кодами $K(B_i)$ позволяет в среднем на 54 % уменьшить число элементов *LUT* в блоке *LUTer*.

Дальнейшее направление наших исследований связано с адаптацией подходов [3, 11] к особенностям совмещенного автомата.

1. Baranov S. Logic Synthesis for Control Automata. *Dordrecht: Kluwer Academic Publishers.* 1994. 312 p.
2. DeMicheli G. Synthesis and Optimization of Digital Circuits. *New York: McGraw-Hill*. 1994. 636 p.
3. Sklyarov V., Skliarova I., Barkalov A., Titarenko L. Synthesis and Optimization of FPGA-based Systems. *Berlin: Springer*. 2014. 432 p.
4. Tiwari A. and Tomko K. Saving power by mapping finite state machines into embedded memory blocks in FPGAs. *Proceedings of Design Automation and Test in Europe.* 2004. Vol. 2. P. 916–921.
5. Баркалов А.А., Титаренко Л.А., Визор Я.Е., Матвиенко А.В. Синтез совмещенного микропрограммного автомата в базисе FPGA. *Комп'ютерні засоби, мережі та системи. Збірник наукових праць*. Ін-т кібернетики імені В.М. Глушкова НАН України. Київ. 2015. Вип. 14. С. 32–39.
6. Баркалов А.А., Титаренко Л.А., Визор Я.Е., Матвиенко А.В. Реализация схемы совмещенного микропрограммного автомата в базисе FPGA. *Проблеми інформатизації та управління. Збірник наукових праць.* Національний авіаційний університет. Київ. 2015. Вип. 3 (51). С. 5–13.
7. Баркалов А.А., Титаренко Л.А., Визор Я.Е., Матвиенко А.В., Горина В.В. Уменьшение числа LUT элементов в схеме совмещенного автомата. *Управляющие системы и машины*. 2016. № 3. С. 16–22.
8. www.altera.com.
9. www.xilinx.com.
10. Barkalov A., Titarenko L. Logic Synthesis for FSM-based Control Units. Berlin: Springer, 2009. 233 p.
11. Barkalov A., Titarenko L., Kolopenczyk M., Mielcarek K., Bazydlo G. Logic Synthesis for FPGA-based Finite State Mashines. Berlin: Springer, 2016. 280 p.
12. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем. М.: Горячая линия – ТЕЛЕКОМ, 2001. 636 с.
13. Баркалов А.А. Принципы оптимизации логической схемы микропрограммного автомата Мура. *Кибернетика и системный анализ*. 1998. № 1. С. 65–72.
14. Yang S. Logic synthesis and optimization benchmarks user guide. Microelectronics Center of North Carolina. 1991. 43 p.

Получено 10.07.2017