

DOI <https://doi.org/10.15407/usim.2018.03.042>

УДК 004.274

А.А. БАРКАЛОВ, д-р техн. наук, проф.,

Университет Зеленогурский (Польша), ул. Подгорная, 50, 65-246, Зеленая Гора (Польша),

a.barkalov@iie.uz.zgora.pl

Л.А. Титаренко, д-р техн. наук, проф.,

Университет Зеленогурский (Польша), ул. Подгорная, 50, 65-246, Зеленая Гора (Польша),

Я.Е. Визор, к.т.н., ст. научн. сотр.,

Институт кибернетики НАНУ, 03187, г. Киев, пр. Глушкова, 40,

yaviz@ukr.net

А.В. Матвиенко, научн. сотр.,

Институт кибернетики НАНУ, 03187, г. Киев, пр. Глушкова, 40,

matv@online.ua

Н.П. Кадет, ст. преп. прикладной информатики,

Национальный авиационный университет, 03058, г. Киев, пр. Космонавта Комарова, 1,

kadet-nata@ukr.net

КОДИРОВАНИЕ НАБОРОВ МИКРООПЕРАЦИЙ В СОВМЕЩЕННОМ АВТОМАТЕ

Предложен метод синтеза совмещенного автомата, ориентированный на базис FPGA. Метод позволяет получить схему с одним встроенным блоком памяти EMB и минимальным числом элементов LUT. В лучшем случае число элементов LUT совпадает с числом микроопераций. Получены условия применения предложенного метода. Приведен пример синтеза совмещенного микропрограммного автомата с использованием данного метода.

Ключевые слова: совмещенный автомат, FPGA, LUT, EMB, кодирование наборов микроопераций, синтез.

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:

- модели, представляющей поведение УУ;
- особенностей элементного базиса [2, 3].

В статье рассмотрен 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 — только 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 микропрограммных автоматов (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 важная характеристика — реконфигурация, при которой меняется число выходов (t_F) и ячеек памяти (V). При этом общая емкость (V_0) есть 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*: $32K \times 1$, $16K \times 2$, $8K \times 4$, $4K \times 8$, $2K \times 16$, $1K \times 32$, 512×64 (битов) [4, 5]. Это определяет следующие пары вида $\langle S_A, t_F \rangle$: $\langle 15, 1 \rangle$, $\langle 14, 2 \rangle$, $\langle 13, 4 \rangle$, $\langle 12, 8 \rangle$, $\langle 11, 16 \rangle$, $\langle 10, 32 \rangle$ и $\langle 9, 64 \rangle$.

В работах [6, 7] предлагаются методы синтеза схем СМПА в базисе *FPGA*. В основе этих методов лежит метод замены логических условий [8]. При этом часть схемы реализуется на блоках *EMB*, а часть — на элементах *LUT*. Однако существуют и другие методы оптимизации схем МПА, которые не рассматривались ранее применительно к СМПА. Авторами рассмотрена возможность использования кодирования наборов микроопераций (НМО) автомата Мили для уменьшения числа блоков *EMB* в схеме СМПА. Эта задача актуальна, так как очень часто только ограниченное число *EMB* можно использовать для реализации схемы УУ [9]. Кроме того, это связано с тем, что *EMB* в первую очередь используются для реализации операционных автоматов цифровых систем [9].

Особенности модели автомата и элементного базиса

Математической моделью СМПА является восьмикомпонентный вектор

$$S = \langle A, X, Y^1, Y^2, \delta, \lambda_1, \lambda_2, a_1 \rangle.$$

Вектор S включает следующие компоненты: $A = \{a_1, \dots, a_M\}$ — множество внутренних состояний; $X = \{x_1, \dots, x_L\}$ — множество вход-

ных 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 . 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 δ 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 λ_1 и λ_2 имеют следующий вид:

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

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

Для реальных устройств переменные $x_i \in X$ и $y_n \in Y$ есть физические объекты, принимающие значения ноль и единица. В то же время состояния $a_m \in A$ — абстрактные объекты. Для синтеза 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 хранятся в регистре памяти *RG*. Как правило, триггеры регистра *RG* имеют входы типа D .

Закодируем состояния $a_m \in A$ двоичными кодами $K(a_m)$ разрядности $R = \lceil \log_2 M \rceil$. Для кодирования состояний используем внутренние переменные, образующие множество $T = \{T_1, \dots, T_R\}$. Задать код состояния перехода (2) можно функциями возбуждения памяти, образующими множество $\Phi = \{D_1, \dots, D_R\}$.

Для синтеза схемы СМПА необходимо получить функции (2)–(4). Эти функции определяются, соответственно, 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 (рис. 1).

В этой схеме блок КС1 генерирует функции (5)–(6), а блок КС2 — функции (7). Регистр *RG*, хранящий коды $K(a_m)$, управляется сигналами *Start* и *Clock*. Сигнал *Start* устанавливает в *RG* код начального состояния $a_1 \in A$. Импульс *Clock* разрешает изменение содержимого *RG* в соответствии с функциями $D_f \in \Phi$.

Как уже отмечалось, особенность *FPGA* — 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 реализуются в основном на элементах *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 гетерогенном базисе

В статье авторами рассмотрен случай, когда для реализации схемы СМПА можно использовать только один блок *EMB*. Пусть выполняется следующее условие

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

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

Пусть имеет место условие:

$$2^{L+R} \cdot (N_1 + R) \leq V_0. \quad (9)$$

В этом случае СМПА может быть реализован, как автомат U_1 , показанный на рис. 2.

В автомате U_1 используется блок *LUTer*, состоящий из элементов *LUT*. При этом блок *EMB* реализует системы (5)–(6), а *LUTer* — систему (7).

Если условие (9) не выполняется, то система (6) не может быть реализована на *EMB*.

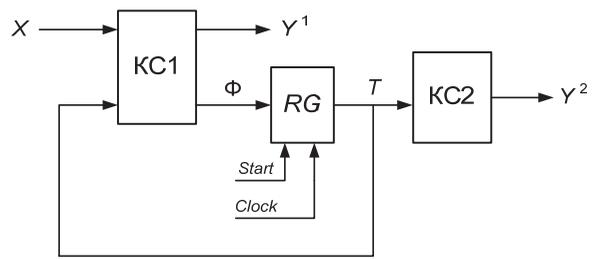


Рис. 1. Структурная схема совмещенного МПА

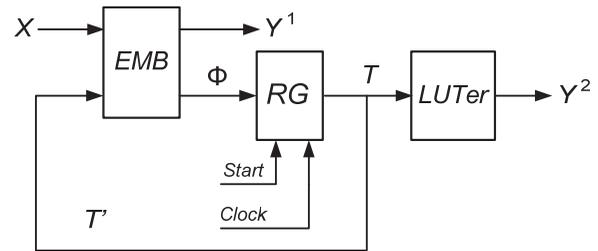


Рис. 2. Структурная схема СМПА U_1

Для реализации (6) необходимы дополнительные элементы *LUT*. Система (6) зависит от переменных $x_i \in X$, поэтому число аргументов в функциях $y_n \in Y^1$ может значительно превышать число S_L входов элемента *LUT*. Это приводит к значительному росту числа элементов и межсоединений в схеме СМПА.

В статье предложен метод решения этой проблемы, основанный на кодировании наборов входных переменных автомата Мили. Идея метода взята из [11], где подобный подход называется кодированием наборов микроопераций. Условимся в дальнейшем называть микрооперациями (МО) выходные сигналы $y_n \in Y^1 \cup Y^2$.

Предлагаемый метод синтеза

Пусть Q число НМО для некоторого СМПА (рассматриваем только МО $y_n \in Y^1$). Поставим в соответствие каждому НМО $Y_q \subseteq Y^1$ двоичный код $K(Y_q)$ разрядности R_q , где

$$R_q = \lceil \log_2 Q \rceil. \quad (10)$$

Используем элементы множества $Z = \{z_1, \dots, z_R\}$ для кодирования НМО.

Пусть для некоторой граф-схемы алгоритма (ГСА) Γ и блока EMB выполняется следующее условие:

$$2^{L+R}(R+R_0) \leq V_0. \quad (11)$$

При этом условие (8) имеет место, а условие (9) не выполняется. Очевидно, в этом случае нельзя использовать модель U_1 . Предлагаем использовать модель U_2 (рис. 3). Автомат U_2 основан на кодировании НМО.

В автомате U_2 блок EMB реализует систему функций возбуждения памяти (5) и систему функций

$$Z = Z(T, X). \quad (12)$$

Блок $LUTer1$ формирует функции $y_n \in Y^1$, представленные в виде:

$$Y^1 = Y^1(Z). \quad (13)$$

Блок $LUTer2$ реализует систему (7).

Так как условие (11) выполняется, то для реализации систем (5) и (12) достаточно одного блока EMB . Число элементов LUT в схеме зависит от отношений между парами $\langle R, S_L \rangle$ и $\langle R_0, S_L \rangle$.

В статье рассмотрен простейший случай, когда имеют место следующие соотношения:

$$R \leq S_L; \quad (14)$$

$$R_0 \leq S_L. \quad (15)$$

Отношение (14) позволяет утверждать, что блок $LUTer2$ имеет не более N_2 элементов LUT . Отношение (15) свидетельствует, что блок $LUTer1$ имеет не более N_1 элементов LUT . Таким образом, схема автомата U_2 имеет один блок EMB и не более N элементов LUT . Естественно, R элементов LUT необходимо для реализации регистра RG .

Предложен метод синтеза автомата U_2 , состоящий из следующих шагов:

- отметка исходной ГСА Γ ;
- кодирование состояний $a_m \in A$;
- кодирование наборов МО $Y_q \subseteq Y$;
- формирование таблиц блоков EMB , $LUTer1$ и $LUTer2$;

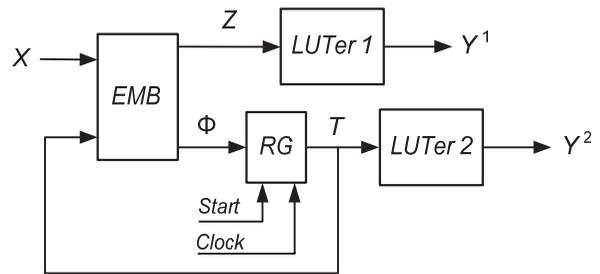


Рис. 3. Структурная схема СМПА U_2

- формирование систем функций (7) и (13);
- реализация схемы СМПА в заданном базисе.

Пусть символ $U_i(\Gamma_j)$ означает, что автомат, основанный на модели U_i , синтезируется по ГСА Γ_j . Рассмотрим пример синтеза автомата $U_2(\Gamma_1)$. Исходная ГСА Γ_1 (рис. 4) отмечена состояниями автомата Мура.

Пример применения предложенного метода

Пусть СМПА задан ГСА Γ_1 (см. рис. 4), на котором показаны множества: $A = \{a_1, \dots, a_8\}$, $X^1 = \{x_1, \dots, x_4\}$, $Y^1 = \{y_1, \dots, y_9\}$, $Y^2 = \{y_{10}, \dots, y_{15}\}$.

Дано: $M = 8$, $L = 4$, $N_1 = 9$, $N_2 = 6$, $N = 15$. Пусть $FPGA$ содержит блоки EMB с конфигурацией $\langle 7, 6 \rangle$, что дает множества $T = \{T_1, T_2, T_3\}$ и $\Phi = \{D_1, D_2, D_3\}$. Условие (9) не выполняется, так как $R + N_1 = 12 > t_F = 6$. Следовательно, модель U_1 в данном случае использовать нельзя.

Микрооперации $y_n \in Y^1$ записаны на дугах ГСА Γ_1 . Анализ ГСА Γ_1 позволяет определить следующие $Q = 8$ НМО: $Y_1 = \emptyset$, $Y_2 = \{y_1, y_2\}$, $Y_3 = \{y_3, y_8, y_9\}$, $Y_4 = \{y_2, y_4\}$, $Y_5 = \{y_3, y_5\}$, $Y_6 = \{y_4, y_6\}$, $Y_7 = \{y_7, y_8\}$, $Y_8 = \{y_1, y_6\}$. Из (10) следует, что $R_0 = 4$ и $Z = \{z_1, z_2, z_3\}$.

Предположим, что $S_L = 3$, тогда условия (14)–(15) выполняются. Следовательно, состояния $a_m \in A$ и НМО $Y_q \subseteq Y^1$ можно закодировать произвольно. Пусть для нашего примера состояния и НМО закодированы тривиально: $K(a_1) = 000, \dots, K(a_8) = 111$, $K(Y_1) = 000, \dots, K(Y_8) = 111$.

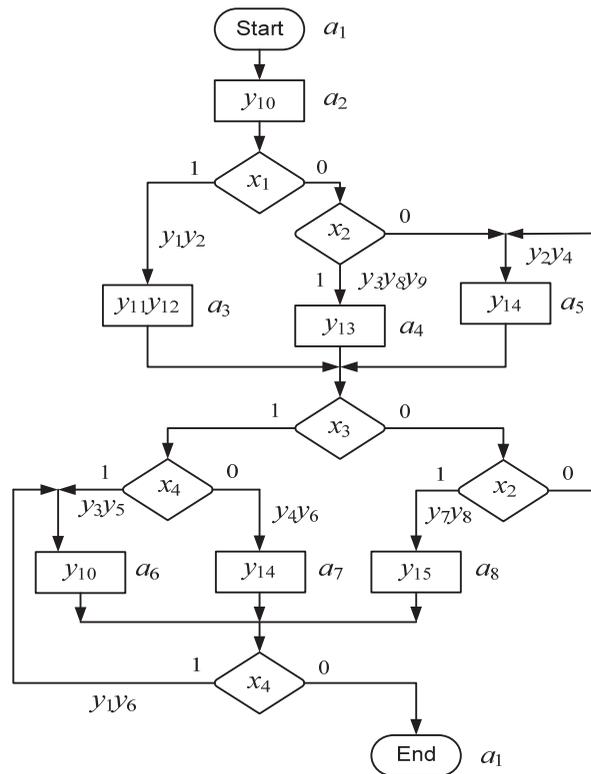


Рис. 4. Отмеченная ГСА Γ

В качестве исходной информации для формирования таблиц блоков *EMB*, *LUTer1* и *LUTer2* предлагаем использовать прямую структурную таблицу (ПСТ) автомата U_2 . Она строится по отмеченной ГСА Γ и состоит из следующих столбцов: $a_m \in A$ код исходного состояния; $K(a_m)$ — код состояния $a_m \in A$; a_s — состояние перехода; $K(a_s)$ — код состояния $a_s \in A$; X_h — конъюнкция входных переменных, определяющая переход из a_m в a_s ; Y_h — набор МО $y_n \in Y^1$, формируемый на переходе номер h ; $K(Y_h)$ — код НМО Y_h ; Z_h — набор переменных $z_r \in Z$, равных единице в коде $K(Y_h)$; Φ_h — набор функций $D_r \in \Phi$, равных единице для записи в *RG* кода состояния перехода; h — номер перехода ($h = 1, H$). Кроме того, в столбце a_m записываются НМО $y_n \in Y^2$, формируемые в состоянии $a_m \in A$.

Для нашего примера ПСТ имеет $H = 22$ строки. Первые восемь из них представлены в табл. 1.

Таблица 1. Фрагмент ПСТ автомата $U_2(\Gamma_1)$

a_m	$K(a_m)$	a_s	$K(a_s)$	X_h	Y_h	$K(Y_h)$	Z_h	Φ_h	h
a_1	000	a_2	001	1	Y_1	000	—	D_3	1
a_2 (y_{10})	001	a_3	010	x_1	Y_2	001	z_1	D_2	2
		a_4	011	$\overline{x_1 x_2}$	Y_3	010	z_2	$D_2 D_3$	3
		a_5	100	$\overline{x_1 x_2}$	Y_4	011	$z_2 z_3$	D_1	4
a_3 ($y_{11} y_{12}$)	010	a_6	101	$x_3 x_4$	Y_5	100	z_1	$D_1 D_3$	5
		a_7	110	$\overline{x_3 x_4}$	Y_6	101	$z_1 z_3$	$D_1 D_2$	6
		a_8	111	$\overline{x_3 x_2}$	Y_7	110	$z_1 z_2$	$D_1 D_2 D_3$	7
		a_5	100	$\overline{x_3 x_2}$	Y_4	011	$z_2 z_3$	D_1	8

Таблица 2. Блок *EMB* автомата $U_2(\Gamma_1)$

$K(a_m)$				X				Z			Φ			q	
T_1	T_2	T_3	T_4	x_1	x_2	x_3	x_4	z_1	z_2	z_3	D_1	D_2	D_3		
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	
0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	2
0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	3
0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	4
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	5
0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	6
0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	7
0	0	0	0	0	1	1	1	0	0	0	0	0	0	1	8

Таблица 3. Блок *LUTer1* автомата $U_2(\Gamma_1)$

$K(Y_q)$	Y^1								q	
$z_1 z_2 z_3$	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8		y_9
0	0	0	0	0	0	0	0	0	0	1
0	0	1	1	0	0	0	0	0	0	2
0	1	0	0	0	1	0	0	0	0	3
0	1	1	0	1	0	0	0	0	0	4
1	0	0	0	1	0	1	0	0	0	5
1	0	1	0	0	0	1	0	0	0	6
1	1	0	0	0	0	0	0	1	1	7
1	1	1	0	0	0	0	0	1	0	8

Таблица блока *EMB* состоит из столбцов $K(a_m)$, X (адрес ячейки памяти), Z , Φ (содержимое ячейки), g (номер ячейки). В случае СМПА $U_2(\Gamma_1)$ эта таблица имеет $G = 128$ строк (так как $R + L = 7$). Первые восемь строк представлены в табл. 2, где показан также переход из состояния a_1 (строка первая табл. 1). При этом переход описывается еще восемью строками, но они идентичны по содержанию столбцов $K(a_m)$, Z , и Φ .

В общем случае переход из состояния $a_m \in A$ соответствует $H(a_m)$ строкам таблицы блока *EMB*:

$$H(a_m) = 2^L. \quad (16)$$

Таблица 4. Блок *LUTer 2* автомата $U_2(\Gamma_1)$

$K(a_m)$	Y^2	m
$T_1 T_2 T_3$	$y_{10} y_{11} y_{12} y_{13} y_{14} y_{15}$	
0 0 0	0 0 0 0 0 0	1
0 0 1	1 0 0 0 0 0	2
0 1 0	0 1 1 0 0 0	3
0 1 1	0 0 0 1 0 0	4
1 0 0	0 0 0 0 1 0	5
1 0 1	0 0 0 0 0 0	6
1 1 0	0 0 0 0 1 0	7
1 1 1	0 0 0 0 0 1	8

Блок *LUTer1* задается таблицей истинности, имеющей столбцы: $K(Y_q)$ (адрес ячейки памяти), Y^1 (содержимое ячейки), q . В рассматриваемом примере табл. 3 имеет $Q = 8$ строк.

Табл. 3 соответствует девяти таблицам истинности для каждого элемента *LUT* схемы *LUTer 1*, так как условие (15) выполняется, *LUTer 1* содержит $N_1 = 9$ элементов *LUT*.

Блок *LUTer 2* задается табл. 4 истинности, имеющей столбцы: $K(a_m)$ (адрес ячейки памяти), Y^2 (содержимое ячейки), m . В рассматриваемом примере эта таблица имеет $M = 8$ строк.

Непосредственно по ГСА Γ_1 строится табл. 4. Строка m содержит единицы, соответствующие микрооперациям $y_n \in Y^2$ из вершины, отмеченной состоянием $a_m \in A$. Так как условие (14) выполняется, то схема *LUTer 2* имеет $N = 6$ элементов *LUT*. Таким образом, табл. 4 соответствует $N = 6$ таблицам истинности для каждого элемента *LUT*.

Поскольку условия (14)–(15) выполняются, пункт 5 в данном случае не рассматривается. Если какое-либо из условий (14) или (15) не выполняется, необходимо:

- получить уравнение для функции $y_n \in Y$;

- преобразовать уравнение так, чтобы каждая из его частей имела не больше S_L аргументов [12, 13];

- построить таблицы истинности для каждой из частей уравнения.

Последний этап предлагаемого метода синтеза связан с применением стандартных средств САПР [4, 5]. На этом этапе решаются задачи размещения, трассировки и получения таблиц «*bit-stream*» для всех элементов схемы. В данной статье этот этап не рассматривается.

Заключение

Предложенный метод позволяет синтезировать схемы СМПА с одним блоком *EMB*. Это достигается кодированием наборов микроопераций, соответствующих входным сигналам автомата Мили. 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 процентов всех примеров из этой библиотеки. Для остальных 82 процентов достаточно одного блока *EMB* для реализации схемы СМПА. При этом условия (14), (15) выполняются для 96 процентов всех примеров. В этих случаях наборы микроопераций представляются унитарными кодами.

Дальнейшее направление исследований связано с адаптацией предложенного метода для автоматов, у которых нарушаются условия (14), (15). Для этого необходимо разработать методы кодирования состояний и/или наборов микроопераций, позволяющие сократить число элементов *LUT* в схеме СМПА.

СПИСОК ЛИТЕРАТУРЫ

1. *Baranov S.* Logic Synthesis for Control Automata. Dordrecht: Kluwer Acad. Publ., 1994. 312 p.
2. *Соловьев В.В.* Проектирование цифровых схем на основе программируемых логических интегральных схем. М.: Горячая линия — ТЕЛЕКОМ, 2001. 636 с.
3. *Skliarova I., Skliarov V., Sudnitson A.* Design of FPGA-based circuits using Hierarchical Finite State Machines. Tallinn: TUT Press, 2012. 240 p.

4. www.altera.com.
5. www.xilinx.com.
6. Баркалов А.А., Титаренко Л.А., Визор Я.Е., Матвиенко А.В. Реализация схемы совмещенного микропрограммного автомата в базе *FPGA*. Проблемы информатизации та управління. Зб. наук. праць. Національний авіаційний університет. К., 2015, 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. УСиМ. 2016. № 3, с. 16–22.
8. Garcia-Vargas L., Senhadji-Navarro R., M. Civit-Balcells A., Guerra-Gutierrez P. 2007. ROM-Based Finite State Machine Implementation in Low Cost *FPGAs*, p. 2342–2347.
9. Sklyarov V., Skliarova I., Barkalov A., Titarenko L. IEEE Int. Simp. on Industrial Electronics, Vigo, P Synthesis and Optimization of *FPGA*-based Systems. Berlin: Springer, 2014. 432 p.
10. Грушницкий Р.И., Мурсаев А.Х., Урюмов 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. К.: ИК НАН Украины, 1997, 136 с.
12. Nowicka, M., Łuba T., Rawski, V. *FPGA*-based decomposition of boolean functions: algorithms and implementations, *Advanced Computer Systems*, 1999, p. 502–509.
13. Rawski, M., Selvaraj, H., Łuba, T. (2005). 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.

Поступила 20.06.2018

REFERENCES

1. Baranov, S., 1994. *Logic Synthesis for Control Automata*. Dordrecht: Kluwer Academic Publishers, 312 p.
2. Soloviev, V.V., 2001. Designing digital circuits based on programmable logic integrated circuits. М.: Hotline – TELECOM, 636 p. (In Russian).
3. Skliarova, I., Sklyarov, V., Sudnitson, A., 2012. Design of *FPGA*-based circuits using Hierarchical Finite State Machines. Tallinn: TUT Press, 240 p.
4. Fpga, [online] Available at: <<http://www.altera.com>> [Accessed 10 Apr. 2017].
5. Xilinx, [online] Available at: <<http://www.xilinx.com>> [Accessed 15 Apr. 2017].
6. Barkalov, A.A., Titarenko, L.A., Vzor, Ya.E., Matvienko, A.V., 2015. “Implementation of a combined microprogramme automaton scheme in the *FPGA* basis”. Problem of informatization i ta management. Zbirnik naukovih Prats. National Aviation University. Kyiv, 3 (51), pp. 5–13. (In Russian).
7. Barkalov, A.A., Titarenko, L.A., Vzor, YA.E., Matvienko, A.V., Gorina, V.V., 2016. “Reducing the number of *LUT* elements in the combined automaton scheme”. *Upravlyńsye sistemy i mańyny*, 3, pp. 16–22. (In Russian).
8. Garcia-Vargas, L., Senhadji-Navarro, R., M. Civit-Balcells, A., Guerra-Gutierrez, P., 2007. “ROM-Based Finite State Machine Implementation in Low Cost *FPGAs*”. IEEE International Symposium on Industrial Electronics, Vigo, pp. 2342–2347.
9. Sklyarov, V., Skliarova, I., Barkalov, A., Titarenko, L., 2014. IEEE International Symposium on Industrial Electronics, Synthesis and Optimization of *FPGA*-based Systems. Berlin: Springer, 432 pp.
10. Grushnitsky, R.I., Mursaeв, A.Kh., Ugryumov, E.P., 2002. Designing systems using programmable logic chips. SPb: BHW. Petersburg, 608 p. (In Russian).
11. Barkalov, A.A., Palagin, A.V., 1997. Synthesis of firmware controls. Kiev: IC NAS of Ukraine, 136 p. (In Russian).
12. Nowicka, M., Luba, T., Rawski, V., 1999. “*FPGA*-based decomposition of boolean functions: algorithms and implementations”, *Advanced Computer Systems*, pp. 502–509.
13. Rawski, M., Selvaraj, H., Luba, T., 2005. “An application of functional decomposition in ROM-based FSM implementation in *FPGA* devices”, *Journal of System Architecture*, 51(6–7), pp. 424–434.
14. Yang, S., 1991. Logic Synthesis and optimization benchmarks user guide. Microelectronics Center of North Carolina, 43 p.

Received 20.06.2018

A.A. Barkalov, Doctor in Techn. Sciences, Professor, University of Zielona Gora,
Podgorna str., 50, Zielona Gora, 65246, Poland,
a.barkalov@iie.uz.zgora.pl

L.A. Titarenko, Doctor of Technical Sciences, Professor of the University of Zelenogorsk (Poland),
Institute of Informatics and Electronics. Zielonogorski University, ul. Podgorna, 50, Zielona Gora, 65246, Poland

Y.E. Visor, Ph.D., Senior Researcher of the Institute of Cybernetics of NASU,
03187, Kiev, Glushkov Avenue, 40,
yaviz@ukr.net

O.V. Matvienko, Researcher of the Institute of Cybernetics of NASU,
03187, Kiev, Glushkov Avenue, 40,
matv@online.ua

N.P. Kadet, Senior Lecturer at the Department of Applied Informatics,
National Aviation University, Kyiv, Ukraine,
kadet-nata@ukr.net

ENCODING OF COLLECTIONS OF MICROOPERATIONS IN COMBINED AUTOMATION

Introduction. A method allowing synthesize the schemes of a combined microprogram automaton (CMPA) with one EMB block is proposed. A case in which the codes of states and sets of micro operations do not affect the number of LUT elements in the automaton circuit is proposed.

Purpose. The purpose of the article is the development of a new method for synthesizing the scheme of a combined microprogram.

Methods. The methods and models used in the research are based on models depending on the behavior of the control device (PU) and on the features of the element basis. In this paper, cases are considered when the synthesis of PU schemes uses the model of the combined microprogram automatic machine (CMPA). There are two types of output signals in the connected machine. Output signals of Moore's automaton exist during the operation of the machine. The output signals of the Miles are present only in the transition between the states. This main feature of the combined automaton is a basis for the developed method application.

Results. The proposed method allows to synthesize the schemes of the combined microprogram automaton (SMPA) with one block of EMB. This is achieved by encoding sets of micro operations corresponding to the input signals of the Mill machine. To implement micro operations, LUT elements are used. The article deals with the case in which the codes of states and sets of micro operations do not affect the number of LUT elements in the automaton circuit.

Conclusion. The analysis of the specialized library showed that this method should be used for 18% of all examples from this library. For the other 82%, one EMB block is enough to implement the SMPA scheme. Thus, the specified conditions are fulfilled for 96% of all examples. In these cases, sets of micro operations are represented by unitary codes.

The further direction of research is related to the adaptation of the proposed method for automata, which violates the given conditions. To do this, it is necessary to develop methods for encoding states of AND / OR sets of micro operations, which allow to reduce the number of LUT elements in the SMPA scheme.

Keywords: combined automaton, FPGA, LUT, EMB, encoding of microoperations, synthesis.

O.O. Баркалов, д-р техн. наук, проф.,
Університет Зеленогурський (Польща), вул. Підгірна, 50, Зелена Гура, 65246, Польща,
a.barkalov@imej.uz.zgora.pl

Л.О. Титаренко, д-р техн. наук, проф.,
Університет Зеленогурський (Польща), вул. Підгірна, 50, Зелена Гура, 65246, Польща,

Я.Є. Візор, к.т.н., ст. наук. співробітник,
Інститут кібернетики НАНУ, 03187, м. Київ, пр. Глушкова, 40,
yaviz@ukr.net

О.В. Матвієнко, наук. співробітник, Інститут кібернетики НАНУ, 03187, м. Київ, пр. Глушкова, 40,
matv@online.ua

Н.П. Кадет, ст. викладач прикладної інформатики,
Національний авіаційний університет, 03058, м. Київ, пр. Космонавта Комарова, 1,
kadet-nata@ukr.net

КОДУВАННЯ НАБОРІВ МІКРООПЕРАЦІЙ В СУМІЩЕНОМУ АВТОМАТІ

Вступ. В роботі запропоновано метод, який дозволяє синтезувати схеми суміщеного мікропрограмного автомата (СМПА) з одним блоком *EMB*. Розглянуто випадок, при якому коди станів і наборів мікрооперацій не впливають на число елементів *LUT* в схемі автомата.

Мета статті — розробка нового методу синтезу схеми суміщеного мікропрограмного автомата.

Методи. Методи і моделі, використані в розробці, ґрунтуються на залежності від моделі, що представляє поведінку пристрою управління (ПУ) та від особливостей елементного базису. Розглянуто випадки, коли для синтезу схем ПУ використовується модель суміщеного мікропрограмного автомата (СМПА). У об'єднаному автоматі існують вихідні сигнали двох типів. Вихідні сигнали автомата Мура існують протягом такту роботи автомата. Вихідні сигнали автомата Мілі існують лише при переході між станами. Ця основна особливість суміщеного автомата є базисом для застосування розробленого методу.

Результат. Запропонований метод дозволяє синтезувати схеми суміщеного мікропрограмного автомата (СМПА) з одним блоком *EMB*, що досягається шляхом кодування наборів мікрооперацій, відповідних вхідних сигналів автомата Мілі. Для реалізації мікрооперацій використовують елементи *LUT*. Розглянутий випадок, при якому коди станів і наборів мікрооперацій не впливають на число елементів *LUT* в схемі автомата.

Висновок. Аналіз спеціалізованої бібліотеки показав, що даний метод необхідно застосовувати для 18 відсотків всіх прикладів з цієї бібліотеки. Для інших 82 відсотків досить одного блоку *EMB* для реалізації схеми СМПА. При цьому задані умови виконуються для 96 відсотків всіх прикладів. У цих випадках набори мікрооперацій представляються унітарними кодами.

Подальший напрямок досліджень пов'язаний з адаптацією запропонованого методу для автоматів, у яких порушуються задані умови. Для цього необхідно розробити методи кодування станів *AND/OR* наборів мікрооперацій, що дозволяють зменшити число елементів *LUT* в схемі СМПА.

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