

DOI: <https://doi.org/10.15407/usim.2018.05.038>
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, канд. техн. наук, ст. научн. сотруд., Ин-т кибернетики им. В.М. Глушкова НАН Украины, 03187, г. Киев, пр. Академика Глушкова, 40, Украина yaviz@ukr.net

А.В. МАТВИЕНКО, научн. сотрудник, Ин-т кибернетики им. В.М. Глушкова НАН Украины, 03187, г. Киев, пр. Академика Глушкова, 40, Украина matv@online.ua

СМЕШАННОЕ КОДИРОВАНИЕ НАБОРОВ МИКРООПЕРАЦИЙ В СОВМЕЩЕННОМ АВТОМАТЕ

Предложенный в работе метод ориентирован на базис FPGA. Для реализации схемы автомата используется только один блок встроенной памяти типа EMB. Целью метода является уменьшение числа элементов LUT в схеме совмещенного микро программного автомата. Для этого часть микроопераций представляется унитарным кодом, генерируемый EMB. Это позволяет уменьшить число переменных, кодирующих наборы микроопераций автомата Мили. Метод позволяет синтезировать схемы совмещенного микропрограммного автомата с одним блоком ЭМИ. Это достигается путем кодирования наборов микроопераций, соответствующие входным сигналам автомата Мили. Для осуществления микроопераций используются элементы LUT. Рассматривается ситуация, когда коды состояний и наборов микроопераций не влияют на количество элементов LUT в цепи автоматов.

Ключевые слова: комбинированный автомат, FPGA, LUT, EMB, кодирования микроопераций, синтез.

Введение

В настоящее время для реализации цифровых систем широко используются программируемые пользователем микросхемы FPGA (*field-programmable logic arrays*) [1]. Для реализации схем СМПА можно использовать такие блоки FPGA, как элементы табличного типа LUT (*look-up table*), программируемые триггер, встроенные блоки памяти типа EMB (*embedded memory blocks*) и программируемую матрицу межсоединений [2]. Как правило, наилучшими характеристиками по площади, быстродействию и потребляемой энергии обладают схемы УУ, реализуемые на блоках EMB [3, 4].

Однако EMB интенсивно используются для реализации операционных устройств цифровых систем [1]. В этой связи схемы УУ реализуются в гетерогенном базисе, состоящем из EMB и LUT [5]. Именно такая задача рассматривается применительно к СМПА в данной статье. Для задания поведения автомата используется язык граф-схем алгоритма (ГСА) [6]. Особенности СМПА и базиса FPGA представлены в [7].

Реализация совмещенного МПА в базисе FPGA

Рассматривается случай, когда в распоряжении проектировщика УУ имеется только один

блок *EMB*. Пусть *EMB* имеет емкость V_0 (бит) при $t_F = 1$. Пусть для данного СМПА и блока *EMB* выполняется следующее условие:

$$2^{L+R} (R+N) > V_0. \quad (1)$$

Из условия (1) следует, что *EMB* не имеет достаточно ресурсов для реализации схемы СМПА. Следовательно, часть схемы должна быть реализована на элементах *LUT*. Пусть выполняется условие:

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

В этом случае системы (4)–(5) из [7], реализуются на *EMB*, а для реализации системы (6) из [7] необходимо использовать элементы *LUT*. Это ведет к модели U_1 (рис.1).

Блок *LUTer* автомата U_1 состоит из элементов *LUT*. Число элементов *LUT* зависит от соотношения параметров R и S_L .

Если условие (2) не выполняется, то система (5), которая представлена в [7], не может быть реализована на *EMB*. Для ее реализации необходимы дополнительные элементы *LUT*. Эта система 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*, что приводит к значительному росту числа элементов и межсоединений в схеме СМПА.

В статье авторы предлагают метод решения этой проблемы. Метод основан на кодировании наборов выходных переменных автомата Мили. Идея метода взята из [5], где подобный подход называется кодированием наборов микроопераций. Условимся в дальнейшем называть микрооперациями (МО) выходные сигналы $y_n \in Y^1UY^2$.

Основная идея предложенного метода

Пусть для данных СМПА и *EMB* условие (2) нарушается, но выполняется следующее условие:

$$t_F > R. \quad (3)$$

При этом t_F — число выходов для конфигурации *EMB*, для которой $S_A = L+R$. Поскольку

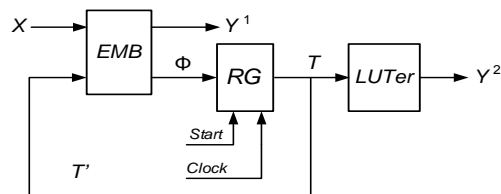


Рис.1. Структурная схема СМПА U_1

условие (2) нарушено, выполняется условие

$$\Delta_t = t_F - R < N_1. \quad (4)$$

Рассмотрим ГСА Γ_1 (рис.2), отмеченную состояниями автомата Мура. В операторных вершинах ГСА Γ_1 записаны наборы микроопераций (НМО) автомата Мура. Над дугами ГСА Γ_1 записаны НМО автомата Мили.

Из ГСА Γ_1 можно найти следующие множества и их параметры:

$$\begin{aligned} A &= \{a_1, \dots, a_{12}\}, R = 4, \\ X &= \{x_1, \dots, x_4\}, L = 4, \\ Y^1 &= \{y_1, \dots, y_9\}, N_1 = 9, \\ Y^2 &= \{y_{10}, \dots, y_{14}\}, N_2 = 5, \\ N &= N_1 + N_2 = 14. \end{aligned}$$

Обозначим символом Y_q набор НМО автомата Мили. В ГСА Γ_1 имеется $Q = 17$ НМО $Y_q \subseteq Y^1$ (табл.1). В табл.1 не представлен набор $Y_1 = \emptyset$.

Поставим в соответствие каждому набору Y_q двоичный код $K(Y_q)$ разрядности

$$R_Q = \lceil \log_2 Q \rceil. \quad (5)$$

Используем для кодирования НМО переменные $z_r \in Z$, где $|Z| = R_Q$. Пусть выполняется условие:

$$\Delta_t = R_Q. \quad (6)$$

Таблица 1. Наборы микроопераций автомата Мили для ГСА Γ_1

q	Y_q	q	Y_q	q	Y_q	q	Y_q
2	$y_1 y_2 y_3$	6	$y_3 y_5 y_9$	10	$y_1 y_2 y_7$	14	y_1
3	$y_3 y_4 y_6$	7	$y_3 y_5 y_6$	11	$y_2 y_3$	15	y_6
4	$y_2 y_7$	8	$y_1 y_4 y_6$	12	$y_2 y_6 y_7$	16	$y_1 y_6$
5	$y_4 y_8$	9	$y_2 y_3 y_6$	13	$y_1 y_4$	17	$y_1 y_3 y_4$

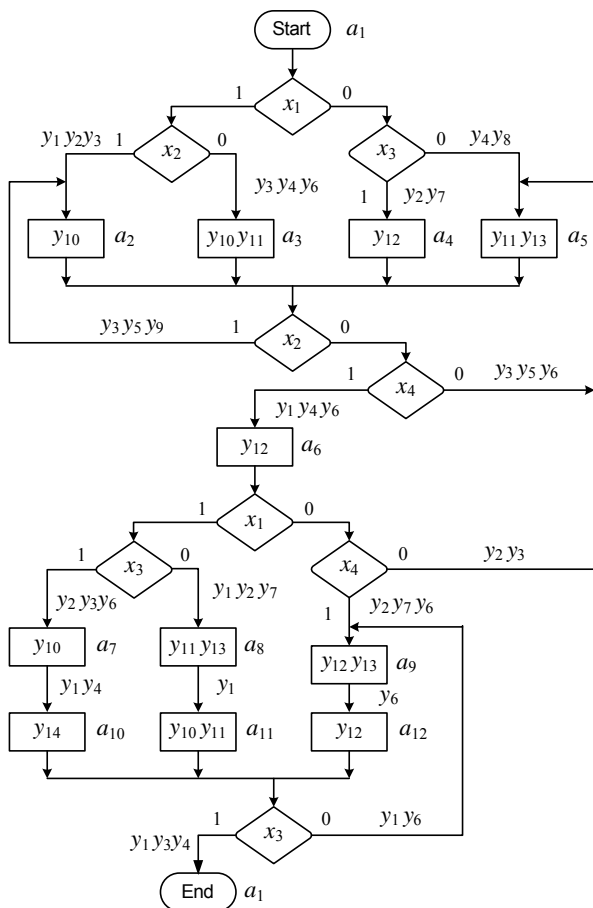


Рис.2. Исходная ГСА Γ_1

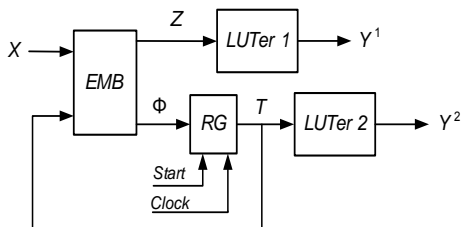


Рис.3. Структурная схема СМПА U_2

В этом случае СМПА может быть реализован в виде U_2 (рис.3).

В автомате U_2 блок *EMB* реализует системы (4) В автомате U_2 блок *EMB* реализует системы (4) [7] и [7] и

$$Z = Z(T, X). \tag{7}$$

Блок *LUTer*₁ реализует функции

$$Y^1 = Y^1(Z), \tag{8}$$

В данной статье, авторы предлагают метод уменьшения числа элементов *LUT* в схеме *LUTer*₁. Метод применим при выполнении условий (3), (4), (7) и (9):

$$R_Q > S_L. \tag{9}$$

Условие (6) означает, что блок *EMB* имеет свободные выходы. Условие (9) свидетельствует о том, что система (8) реализуется в виде многоуровневой схемы.

Для оптимизации схемы *LUTer*₁ предлагается использовать свободные выходы блока *EMB* для реализации функций $y_n \in Y^{11}$, где $Y^{11} \subset Y^1$. В этом случае часть микроопераций представляется унитарными кодами. Назовем такой подход смешанным кодированием НМО.

Поясним эту идею. Удалим микрооперацию y_1 из НМО Y_q . Как следует из табл.1, набор Y_2 теперь совпадает с $Y_{11} = \{y_2, y_3\}$, набор Y_4 — с Y_{10} , набор Y_{15} — с Y_{16} , набор Y_{14} — с $Y_1 = \emptyset$. И так, включение y_1 в Y_{11} позволяет уменьшить число НМО с $Q=17$ до $Q_1=13$. Для кодирования Q_1 наборов необходимо R_1 бит кода:

$$R_1 = \lceil \log_2 Q_1 \rceil. \tag{10}$$

При выполнении условия

$$R_1 < R_Q. \tag{11}$$

число элементов *LUT* в схеме *LUTer*₁ уменьшается.

Пусть $Y^{12} = Y^1 \setminus Y^{11}$. Тогда *LUTer*₁ реализует только функции $y_n \in Y^{12}$. Это ведет к СМПА U_3 (рис.4).

В СМПА U_3 блок *EMB* реализует системы функций (4) [3], (7) и

$$Y^{11} = Y^{11}(T, X). \tag{12}$$

Блок *LUTer*₁ реализует систему функций

$$Y^{12} = Y^{12}(Z). \tag{13}$$

Метод смешанного кодирования всегда приводит к упрощению схемы *LUTer*₁. Это связано с тем, что некоторые функции $y_n \in Y^1$ реализуются на *EMB* и не используют элементы *LUT*. Наилучший результат достигается, если выполняется условие (9) и условие

$$R_1 < S_L. \tag{14}$$

В этом случае для реализации любой функции $y_n \in Y^{12}$ достаточно одного элемента *LUT*.

В статье предлагается метод синтеза СМПА U_3 . В следующем разделе представлен предлагаемый метод и пример синтеза автомата $U_3(\Gamma_1)$, где выражение $U_i(\Gamma_j)$ означает, что ГСА Γ_j используется для синтеза СМПА U_i .

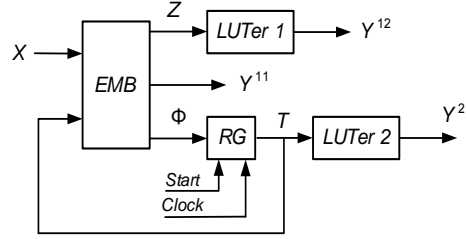


Рис. 4. Структурная схема СМПА U_3

Предлагаемый метод и пример синтеза СМПА U_3

Предлагаемый метод синтеза включает следующие этапы:

- формирование множества состояний A ;
- кодирование состояний;
- формирование наборов микроопераций автомата Мили;
- смешанное кодирование наборов МО $Y_q \subseteq Y^1$;
- формирование таблиц блоков *EMB*, *LUTer1* и *LUTer2*;

▪ реализация схемы СМПА в заданном базисе. Рассмотрим применение этого метода для ГСА Γ_1 . Первый этап синтеза уже выполнен и получено множество A с $M=12$.

Метод кодирования состояний зависит от условия

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

Когда условие (15) выполняется, то число элементов *LUT* в блоке *LUTer2* не зависит от кодов $K(a_m)$. Если условие (15) нарушается, то кодирование необходимо выполнить так, чтобы уменьшить число аргументов в функциях $y_n \in Y^2$. Для решения этой задачи можно использовать алгоритм из [5].

Пусть для данного примера $S_L=3$. Для определения разрядности кода $K(a_m)$ используем формулу

$$R = \lceil \log_2 M \rceil. \quad (16)$$

Для примера это дает $R=4$, т.е. $T = \{T_1, \dots, T_4\}$ и $\Phi = \{D_1, \dots, D_4\}$. Итак, условие (15) не выполняется.

Найдем из ГСА Γ_1 систему функций $Y^2=Y^2(A)$. Пусть A_m — конъюнкция переменных $T_r \in T$, соответствующая коду $K(a_m)$. Искомая система

		T_1	T_2		
T_3	T_4	00	01	11	10
00		a_1	a_2	*	a_9
01		a_4	a_3	*	a_5
11		a_{12}	a_{11}	*	a_8
10		a_6	a_7	*	a_{10}

Рис.5. Кодирование состояний СМПА $U_3(\Gamma_1)$

выглядит следующим образом:

$$\begin{aligned} y_{10} &= A_2 \vee A_3 \vee A_7 \vee A_{11}; \\ y_{11} &= A_3 \vee A_5 \vee A_8 \vee A_{11}; \\ y_{12} &= A_4 \vee A_6 \vee A_9 \vee A_{12}; \\ y_{13} &= A_5 \vee A_8 \vee A_9; \\ y_{14} &= A_{10}; \end{aligned} \quad (17)$$

Закодируем состояния $a_m \in A$ так, как показано на рис.5.

Используя коды из карты Карно (рис.5), можно получить следующую систему (6) [7] для нашего примера:

$$\begin{aligned} y_{10} &= T_2; \quad y_{11} = \overline{T_1}T_4 \vee T_2T_4; \\ y_{12} &= \overline{T_1}\overline{T_2}T_4 \vee \overline{T_1}T_2T_3 \vee T_1\overline{T_3}\overline{T_4}; \\ y_{13} &= T_1\overline{T_3} \vee T_1T_4; \quad y_{14} = T_1T_3\overline{T_4}. \end{aligned} \quad (18)$$

Из (18) следует, что: y_{10} является выходом T_2 блока *EMB*; функции y_{11} , y_{13} , y_{14} имеют по три аргумента; для реализации функции y_{12} требуется два элемента *LUT*.

Этап 3 для рассматриваемого примера выполнен. Наборы выходных переменных $Y_q \subseteq Y^1$ представлены в табл.1.

Пусть используемый блок *EMB* имеет конфигурацию 256×9 , т.е. $S_A=8$, $t_F=9$. Условие (3)

Таблица 2. Наборы микроопераций после удаления y_1

q	Y_q	q	Y_q	q	Y_q	q	Y_q
2	$y_2 y_3$	5	$y_4 y_6$	8	$y_3 y_5 y_6$	11	$y_3 y_4$
3	$y_2 y_3 y_6$	6	$y_4 y_8$	9	$y_2 y_7$	12	$y_3 y_4 y_6$
4	y_4	7	$y_3 y_5 y_9$	10	$y_2 y_7 y_6$	13	y_6

Таблица 3. Наборы микроопераций после удаления y_6

q	Y_q	q	Y_q
2	$y_2 y_3$	6	$y_3 y_5$
3	y_4	7	$y_2 y_7$
4	$y_4 y_8$	8	$y_3 y_4$
5	$y_3 y_5 y_9$	1	-

Таблица 4. Фрагмент ПСТ автомата $U_3(\Gamma_1)$

a_m	$K(a_m)$	a_s	$K(a_s)$	X_h	Y_h^{11}	Y_h^{12}	Φ_h	h
a_1 (-)	0000	a_2	0100	$x_1 x_2$	y_4	Y_2	D_2	1
		a_3	0101	$\overline{x_1 x_2}$	y_4	Y_8	$D_2 D_4$	2
		a_4	0001	$\overline{x_1 x_3}$	-	Y_7	D_4	3
		a_5	1001	$\overline{x_1 x_3}$	-	Y_4	$D_1 D_4$	4
a_2 (y_{11}, y_{12})	0100	a_2	0100	x_2	-	Y_5	D_2	5
		a_6	0010	$\overline{x_2 x_4}$	$y_4 y_6$	Y_3	D_3	6
		a_5	1001	$\overline{x_2 x_4}$	y_4	Y_6	$D_1 D_4$	7

Таблица 5. Фрагмент таблицы блока EMB автомата $U_3(\Gamma_1)$

$K(a_m)$	X	Y^{11}	$K(Y^{12})$	Φ	q	h
$T_1 T_2 T_3 T_4$	$x_1 x_2 x_3 x_4$	$y_1 y_6$	$z_1 z_2 z_3$	$D_1 D_2 D_3 D_4$		
0 0 0 0	0 0 0 0	0 0	0 1 0	0 0 0 1	1	4
0 0 0 0	0 0 0 1	0 0	0 1 0	0 0 0 1	2	4
0 0 0 0	0 0 1 0	0 0	0 1 1	1 0 0 1	3	3
0 0 0 0	0 0 1 1	0 0	0 1 1	1 0 0 1	4	3
0 0 0 0	0 1 0 0	0 0	0 1 0	0 0 0 1	5	4
0 0 0 0	0 1 0 1	0 0	0 1 0	0 0 0 1	6	4
0 0 0 0	0 1 1 0	0 0	0 1 1	1 0 0 1	7	3
0 0 0 0	0 1 1 1	0 0	0 1 1	1 0 0 1	8	3

выполняется и $\Delta_i = 5$. Итак, блок EMB имеет пять свободных выходов, а $N_1=9$. Следовательно, условие (4) выполняется. Так как $Q=17$, то $R_Q=5$ и выполняется условие (8). Поэтому модель $U_3(\Gamma_1)$ может быть использована для данного элементного базиса.

Для данного примера имеет место равенство

$$\Delta_i = R_Q. \tag{19}$$

Из (19) следует, что удаление МО $y_1 \in Y^1$ имеет смысл, если это приводит к уменьшению параметра R_Q . В противном случае в блоке EMB нет свободных выходов для $y_1 \in Y^1$.

Решаем эту задачу следующим образом. Удаляем из Y^1 МО y_n . Если это приводит к уменьшению R_Q , то включаем y_n в Y^1 и исключаем из $Y^1 : Y^{12} = Y^1 \setminus \{y_n\}$. Затем пытаемся удалить из Y^{12} следующую МО. Процесс заканчивается, если удаление любой МО из Y^{12} не приводит к уменьшению величины R_Q .

После удаления МО $y_1 \in Y^1$ остается $Q_1=13$ наборов МО (табл.2). Так как $R_1=4$, то $Y^1=\{y_1\}$ и процесс продолжается.

Удаление микроопераций y_2, y_3, y_4 и y_5 не ведет к уменьшению R_1 . Удаление y_6 из табл.2 приводит к $Q_1=8$ (табл. 3).

После удаления y_6 из Y^{12} получаем $R_1=3$. Следовательно, условие (14) выполняется и дальнейший процесс удаления МО не имеет смысла. Итак, имеем два множества: $Y^1=\{y_1, y_6\}$ и $Y^{12}=\{y_2, y_3, y_4, y_5, y_7, y_8, y_9\}$. Так как $R_1=3$, то $Z=\{z_1, z_2, z_3\}$.

Закодируем НМО $Y_q \subseteq Y^{12}$ как показано на рис.6.

Построим прямую структурную таблицу (ПСТ) автомата $U_3(\Gamma_1)$. Эта таблица (табл.4) включает следующие столбцы: a_m — исходное состояние (этот столбец содержит набор микро-операций $y_n \in Y^2$, формируемом в состоянии $a_m \in A$); $K(a_m)$ — код состояния $a_m \in A$; a_s — состояние перехода; $K(a_s)$ — код состояния $a_s \in A$; X_h — входной сигнал, определяющий переход $\langle a_m, a_s \rangle$; Y_h^{11} — выходные переменные, формируемые на переходе $\langle a_m, a_s \rangle$; Y_h^{12} — символ НМО, формируемого на переходе $\langle a_m, a_s \rangle$; Φ_h — набор функций $D_r \in \Phi$, равных единице для записи в RG кода состояния перехода; h — номер перехода ($h = \overline{1, H}$).

Таблица блока EMB (см. табл.5) включает следующие столбцы: $K(a_m)$, X (адрес ячейки памяти) Y^{11} , $K(Y^{12})$, Φ , (содержимое ячейки), q (номер ячейки). В общем случае эта таблица имеет G строк:

$$G = 2^{L+R}. \quad (20)$$

Переходы из состояния $a_m \in A$ требуют $H(a_m)$ строк:

$$H(a_m) = 2^L. \quad (21)$$

В данном примере из (20)—(21) следует, что $G = 256$, $H(a_m) = 16$. Итак, для задания переходов из каждого состояния $a_m \in A$ необходимо 16 ячеек памяти.

В табл. 5 приведены первые 8 строк блока EMB для СМПА $U_3(\Gamma_1)$. Столбец h добавлен, чтобы показать соответствие между табл.4 и 5.

Таблица блока $LUTer1$ (табл.6) имеет следующие столбцы: $K(Y_q)$, Y_q и q .

Из карты Карно (рис.6) и уравнения $y_3 = Y_2 \vee Y_5 \vee Y_6 \vee Y_8$ (табл.3) следует, что $z_1 = y_3$. Таким образом, МО $y_3 \in Y^1$ формируется блоком EMB . Поэтому y_3 отсутствует в табл.6, как и МО $y_1, y_6 \in Y^{11}$.

Эта таблица соответствует шести таблицам истинности для каждого из элементов LUT блока $LUTer1$.

Таблицы блока $LUTer2$ строятся на основе системы (18). При этом функция y_{12} должна быть представлена в виде двух подфункций, зависящих не более чем от $S_L=3$ аргументов.

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

Рис.6. Коды НМО $Y_q \subseteq Y^{12}$

Таблица 6. Таблица блока $LUTer1$ автомата $U_3(\Gamma_1)$

$K(Y_q)$	Y_q	q
$z_1 z_2 z_3$	$y_4 y_5 y_6 y_7 y_8 y_9$	
000	0 0 0 0 0 0	1
001	1 0 0 0 0 0	3
010	1 0 0 0 1 0	4
011	0 0 1 1 0 0	7
100	0 0 1 0 0 0	2
101	0 1 0 0 0 1	5
110	0 1 0 0 0 0	6
111	1 0 0 0 0 0	8

Этот этап в статье не рассматривается. Отметим только, что для реализации блока $LUTer2$ требуется пять элементов LUT .

Последний этап предлагаемого метода синтеза связан с применением стандартных средств САПР [8, 9]. На этом этапе решаются задачи размещения, трассировки и получения таблиц «bit-stream» для всех элементов схемы. В данной статье этот этап не рассматривается. Итак, применение предложенного метода позволило получить схему с одним блоком EMB и 11 элементами LUT . Отметим, что для автомата $U_2(\Gamma_1)$ необходимо 28 элементов LUT с $S_L=3$. Таким образом, применение предложенного метода позволяет уменьшить число LUT в 2,5 раза (для данного автомата и базиса $FPGA$).

Заключение

Предлагаемый в статье метод ориентирован на базис $FPGA$. При этом для реализации схемы автомата используется только один блок встроенной памяти типа EMB . Цель метода — уменьшение числа элементов LUT в схеме СМПА. Для этого часть микроопераций представляется унитарным кодом, генерируемым

EMB. Это позволяет уменьшить число переменных, кодирующих наборы микроопераций автомата Мили.

Анализ библиотеки [10] показал, что 82 процента всех примеров допускают реализацию схемы автомата на одном блоке *EMB*. Для остальных 18 процентов необходимо использовать элементы *LUT*. Исследования показали, что модель U_3 может быть применена для этих

18 процентов. При этом количество элементов *LUT* уменьшается в среднем на 48 процентов в сравнении с автоматом U_2 .

Дальнейшее направление исследований связано с разработкой метода нахождения микроопераций для унитарного кодирования. Генерирование этих микроопераций блоком *EMB* должно привести к минимизации числа элементов *LUT* в схеме автомата.

СПИСОК ЛИТЕРАТУРЫ

1. Grout I. Digital Systems Design with FPGAs and CPLDs. Amsterdam: Elsevier, 2008. 784 p.
2. Maxfield C. The Design Warrior's Guide to FPGAs. Orlando: Academic Press, 2004. 542 p.
3. Garcia-Vargas, L., Senhadji-Navarro R.M., Civit-Balcells A. Guerra-Gutierrez P. ROM-Based Finite State Machine Implementation in Low Cost FPGAs, IEEE International Symposium on Industrial Electronics, 2007, Vigo, p. 2342–2347.
4. Rawski, M., Tomaszewicz, P., Borowski, G., Łuba, T. (2011). Logic Synthesis Method of Digital Circuits Designed for Implementation with Embedded Memory Blocks on FPGAs, Design of Digital Systems and Devices. LNEE 70, Springer, Berlin, P. 121–144.
5. Barkalov A., Titarenko L. Logic Synthesis for FSM-based Control Units, Berlin: Springer, 200, 233 p.
6. Baranov S. Logic Synthesis for Control Automata, Dordrecht: Kluwer Academic Publishers, 1994, 312 p.
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. Skliarova I., Sklyarov V., Sudnitson A. Design of FPGA-based circuits using Hierarchical Finite State Machines. Tallinn: TUT Press, 2012. 240p.
9. Баркалов А.А., Титаренко Л.А., Визор Я.Е., Матвиенко А.В. Реализация схемы совмещенного микропрограммного автомата в базе FPGA. Проблеми інформатизації та управління: Зб. науков. Праць, Національний авіаційний університет, К., 2015. 3(51), С. 5–13.
10. Yang S. Logic Synthesis and optimization benchmarks user guide. Microelectronics Center of North Carolina. 1991, 43 p.

Поступила 27.11.2018

REFERENCE

1. Grout, I., 2008. Digital Systems Design with FPGAs and CPLDs. Amsterdam: Elsevier, 784 p.
2. Maxfield, C., 2004. The Design Warrior's Guide to FPGAs. — Orlando: Academic Press, 542 p.
3. 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.
4. Rawski, M., Tomaszewicz, P., Borowski, G., Łuba, T., 2011. "Logic Synthesis Method of Digital Circuits Designed for Implementation with Embedded Memory Blocks on FPGAs", Design of Digital Systems and Devices. LNEE 70, Springer, Berlin, pp. 121–144.
5. Barkalov, A., Titarenko, L., 2000. Logic Synthesis for FSM – based Control Units, Berlin: Springer, 233 p.
6. Baranov, S., 1994. Logic Synthesis for Control Automata, Dordrecht: Kluwer Academic Publishers, 312 p.
7. Barkalov, A.A., Titarenko, L.A., Vzor, Ya.Ye., Matvienko, A.V., Gorina, V.V., 2016. "Synthesis of Combined Finite State Machine with FPGAs", Upravlausie sistemy i masiny, 2016, 3, pp. 16–22. (In Russian).
8. Skliarova, I., Sklyarov, V., Sudnitson, A., 2012. Design of FPGA-based circuits using Hierarchical Finite State Machines. Tallinn: TUT Press, 240 p.
9. Barkalov, A.A., Titarenko, L.A., Vzor, Ya.Ye., Matvienko, A.V., 2015. "Realizatsiya skhemy sovmeshchennogo mikroprogrammno avtomata v bazise FPGA". Problemy informatyzatsiyi ta upravlinnya: Zb. naukov. Prats, Natsional'nyy aviatsiynyy universytet, K., 3(51), pp. 5–8. (In Russian).
10. Yang, S., 1991. Logic Synthesis and optimization benchmarks user guide. Microelectronics Center of North Carolina, 43 p.

Received 27.11.2018

О.О. 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,
a.barkalov@imej.uz.zgora.pl

Я.Є. Візор, канд. техн. наук, ст. наук. співробітник, Ін-т кібернетики ім. В.М. Глушкова
НАН України, 03187, м. Київ, пр. Академіка Глушкова, 40, Україна,
yaviz@ukr.net

О.В. Матвієнко, наук. співробітник, Ін-т кібернетики ім. В.М. Глушкова НАН України,
03187, м. Київ, пр. Академіка Глушкова, 40, Україна,
matv@online.ua

ЗМІШАНЕ КОДУВАННЯ НАБОРУ МІКРООПЕРАЦІЙ В СУМІЩЕНОМУ АВТОМАТІ

Вступ. Запропонований в статті метод орієнтований на базис *FPGA*. При цьому для реалізації схеми автомата використовується тільки один блок вбудованої пам'яті типу *EMB*.

Мета — зменшення числа елементів *LUT* в схемі суміщеного мікро програмного автомата (СМПА).

Методи. Частина мікрооперацій представляється унітарним кодом, що генерується *EMB* і дозволяє зменшити число змінних, що кодують набори мікрооперацій автомата Мілі.

Результати. Запропонований метод дозволяє синтезувати схеми суміщеного мікропрограмного автомата (СМПА) з одним блоком *EMB*. Це досягається шляхом кодування наборів мікрооперацій, що відповідають вхідним сигналам автомата Мілі. Для здійснення мікрооперацій використовуються елементи *LUT*. Розглянуто ситуацію, коли коди станів та наборів мікрооперацій не впливають на кількість елементів *LUT* в ланцюзі автоматів.

Висновок. Аналіз спеціалізованої бібліотеки показав, що 82 відсотки всіх прикладів допускають реалізацію схеми автомата на одному блоці *EMB*. Для решти 18 відсотків необхідно використовувати елементи *LUT*. Дослідження показали, що нова модель автомата може бути застосована для цих 18 відсотків. При цьому кількість елементів *LUT* зменшується в середньому на 48 відсотків в порівнянні з попереднім автоматом.

Подальший напрям досліджень пов'язаний з розробкою методу пошуку мікрооперацій для унітарного кодування. Генерування цих мікрооперацій блоком *EMB* має привести до мінімізації числа елементів *LUT* в схемі автомата.

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

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

L.A. Titarenko, Doctor of Techn. Sciences, Professor, University of Zielona Gora, (Poland),
Podgorna street, 50, Zielona Gora, 65246, Poland,
a.barkalov@iie.uz.zgora.pl

Ya. Ye. Visor, Ph.D in Techn. Sciences, Institute of Cybernetics of NAS of Ukraine,
03187, Kiev, Glushkov Avenue, 40, Ukraine,
yaviz@ukr.net

O.V. Matvienko, Researcher Associate, Institute of Cybernetics of NAS of Ukraine,
03187, Kiev, Glushkov Avenue, 40, Ukraine,
matv@online.ua

ENCODING OF MICROOPERATION COLLECTIONS IN COMBINED AUTOMATA

Introduction. The method for the synthesis of control unit targeting the *FPGA* chips use is proposed. To synthesize the final circuit, a combined automaton model is used. The purpose of the method is reducing the number of *LUT* elements.

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

Methods. The method is based on the representation of certain microoperations of the Mile automaton by unitary codes that are generated by embedded memory block (*EMB*). This allows reducing the number of look-up table (*LUT*) elements and using only a single *EMB* block in the circuit of the automaton.

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. An analysis of a specialized library showed that for 82% of all examples it is enough only a single *EMB* block for the implementation of a circuit of an automaton. For the remaining 18%, it is necessary use the *LUT* elements. Our research proves that the proposed model of the automaton can be applied for these 18%. At the same time, the number of *LUT* elements decreases by an average of 48% as compared with automata with a trivial structure.

Our further research is related to the development of a method for finding micro-operations for unitary coding. The generation of these microoperations by the *EMB* block should lead to a minimization for the number of *LUT* elements in the combined automaton's circuit.

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