

**А.А. Баркалов** (докт. тех. наук, проф.)<sup>1</sup>, **Л.А. Титаренко**  
(докт. тех. наук, проф.)<sup>1</sup>, **К.Н. Ефименко** (канд. тех. наук, доц.)<sup>2</sup>,  
**Я.М. Липински** (магистр тех. наук, асс.)<sup>1</sup>  
Университет Зеленогурский (Польша)<sup>1</sup>,  
Донецкий национальный технический университет<sup>2</sup>  
A.Barkalov@iie.uz.zgora.pl

## **ОПТИМИЗАЦИЯ СХЕМЫ КМУУ С ПРЕОБРАЗОВАТЕЛЕМ АДРЕСА МИКРОКОМАНД**

Предлагается метод уменьшения аппаратурных затрат в схеме КМУУ с преобразователем адреса микрокоманд, ориентированный на технологию FPGA. Метод основан на использовании двух источников кодов классов псевдоэквивалентных ОЛЦ и мультиплексора, позволяющего выбрать один из этих источников. Такой подход позволит уменьшить число LUT элементов в схеме адресации КМУУ. Приведен пример применения предложенного метода.

**КМУУ, ГСА, ОЛЦ, FPGA, логическая схема**

### ***Введение***

Если алгоритм управления некоторой системы представлен линейной граф-схемой алгоритма (ГСА) [1], то для реализации схемы устройства управления (УУ) может быть использована модель композиционного микропрограммного устройства управления (КМУУ) с преобразователем адреса микрокоманд [2,3]. В настоящее время для реализации схем УУ широко используются программируемые логические интегральные схемы (ПЛИС) вида FPGA (field-programmable gate arrays) [4,5]. Основу FPGA представляют макроячейки LUT (look-up table), имеющие ограниченное число входов (4-6) [6,7]. Для оптимизации схемы УУ на FPGA необходимо уменьшить количество аргументов и термов в реализуемых системах булевых функций [8]. В настоящей работе предлагается метод решения этой задачи для КМУУ с преобразователем адреса микрокоманд [9]. Метод основан на использовании двух источников кодов классов псевдоэквивалентных операторных линейных цепей (ОЛЦ) и мультиплексора источников кодов (МИК).

Целью исследования является оптимизация схемы КМУУ с преобразователем адреса микрокоманд за счет мультиплексирования источников кодов псевдоэквивалентных ОЛЦ.

Задачей исследования является разработка метода синтеза КМУУ с преобразователем адреса микрокоманд, позволяющего уменьшить число LUT-элементов в комбинационной части КМУУ.

## Композиционное МУУ с преобразователем адреса микрокоманд

Пусть ГСА  $\Gamma = \Gamma(V, E)$  представлена множествами вершин  $V$  и соединяющих их дуг  $E$ . Пусть  $V = b_0 \cup b_E \cup E_1 \cup E_2$ , где  $b_0$  – начальная вершина,  $b_E$  – конечная вершина,  $E_1$  – множество операторных вершин и  $E_2$  – множество условных вершин ГСА  $\Gamma$ . В операторных вершинах  $b_q \in E_1$  записываются наборы микроопераций  $Y(b_q) \subseteq Y$ , где  $Y = \{y_1, \dots, y_N\}$  – множество микроопераций. В условных вершинах  $b_q \in E_2$  записываются элементы множества логических условий  $X = \{x_1, \dots, x_L\}$ . Введем ряд определений, взятых из [2].

Определение 1. Операторной линейной цепью ГСА  $\Gamma$  называется конечная последовательность операторных вершин  $\alpha_g = \langle b_{g1}, \dots, b_{gF_g} \rangle$  такая, что для любой пары соседних компонент  $b_{gi}, b_{gi+1}$ , где  $i$  – номер компоненты кортежа  $\alpha_g$ , существует дуга  $\langle b_{gi}, b_{gi+1} \rangle \in E$ .

Определение 2. Вершина  $b_q \in D^g$ , где  $D^g$  – множество вершин, входящих в ОЛЦ  $\alpha_g$ , называется входом ОЛЦ  $\alpha_g$ , если существует дуга  $\langle b_t, b_q \rangle \in E$ , где  $b_t \notin D^g$ .

Определение 3. Вершина  $b_q \in D^g$ , называется выходом ОЛЦ  $\alpha_g$ , если существует дуга  $\langle b_q, b_t \rangle \in E$ , где  $b_t \notin D^g$ .

Определение 4. ОЛЦ  $\alpha_i, \alpha_j$  называются псевдоэквивалентными ОЛЦ, если их выходы связаны со входом одной и той же вершины  $b_q \in V$ .

Пусть для некоторой ГСА  $\Gamma$  сформировано множество ОЛЦ  $C = \{\alpha_1, \dots, \alpha_G\}$ , определяющее разбиение на множестве  $E_1$  [3], и пусть  $|E_1| = M$ . Поставим в соответствие каждой вершине  $b_q \in E_1$  микрокоманду  $MI_q$  с адресом  $A(b_q)$ , имеющим разрядность

$$R = \lceil \log_2 M \rceil. \quad (1)$$

Используем для адресации микрокоманд переменные  $T_r \in T$ , где  $|T| = R$ . Адресация выполняется таким образом, чтобы выполнялось условие

$$A(b_{gi+1}) = A(b_{gi}) + 1, \quad (2)$$

где  $b_{gi}, b_{gi+1} \in D^g$  и  $\langle b_{gi}, b_{gi+1} \rangle \in E$ .

Пусть  $\Pi_C = \{V_1, \dots, V_I\}$  – разбиение множества ОЛЦ  $C$  на классы псевдоэквивалентных ОЛЦ. Поставим в соответствие каждому классу  $V_i \in \Pi_C$  двоичный код  $K(V_i)$  разрядности

$$R_C = \lceil \log_2 I \rceil. \quad (3)$$

Используем для кодирования классов  $K(B_i)$  элементы множества  $\tau = \{\tau_1, \dots, \tau_{R_C}\}$ . В этом случае УУ может быть реализовано в виде КМУУ с преобразователем адреса [2,3], обозначаемом в дальнейшем символом  $U_1$  (рис. 1).

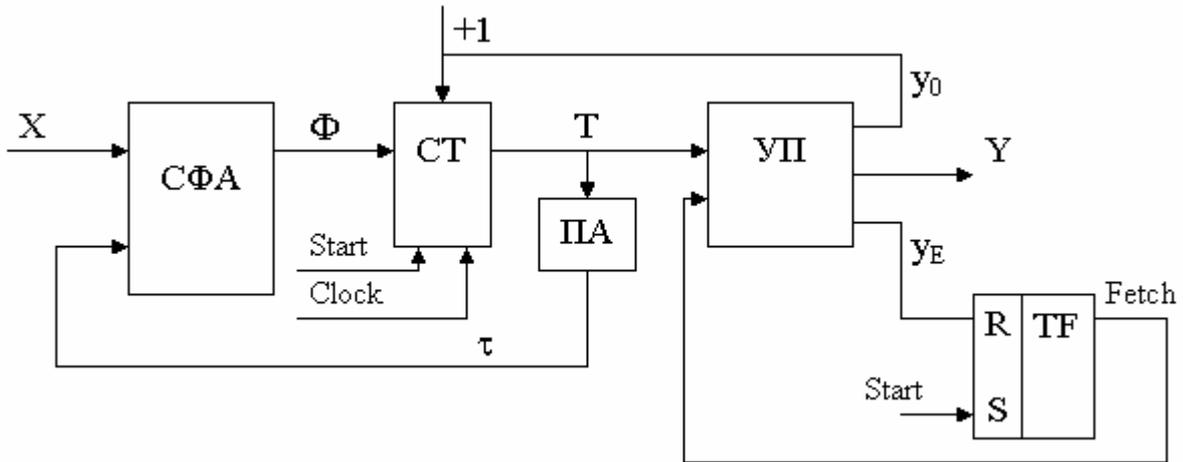


Рисунок 1 – Структурная схема КМУУ  $U_1$

По сигналу Start в счетчик СТ заносится адрес первой микрокоманды (МК) микропрограммы, соответствующей ГСА Г. При этом триггер TF устанавливается в единичное состояние и сигнал Fetch = 1 разрешает выборку МК управляющей памяти (УП). Если очередная МК  $MI(b_q)$  соответствует любой компоненте ОЛЦ  $\alpha_g \in C$  кроме ее выхода, то формируется сигнал  $y_0 = 1$ . При этом содержимое СТ увеличивается на 1, что соответствует режиму (2). Если  $b_q$  – выход ОЛЦ, то  $y_0 = 0$  и адрес перехода формируется схемой СФА. Для этого формируются функции

$$\Phi = \Phi(\tau, X). \quad (4)$$

Переменные  $\tau_r \in \tau$  формируются схемой преобразователя адреса (ПА), для чего реализуется система

$$\tau = \tau(T). \quad (5)$$

Таким образом, схема Па преобразовывает адреса выходов ОЛЦ  $\alpha_g \in B_i$  в код класса  $V_i \in \Pi_C$ .

Если  $\langle b_q, b_E \rangle \in E$ , то формируется переменная  $y_E = 1$ , триггер TF обнуляется и функционирование КМУУ  $U_1$  прекращается.

Такой подход позволяет уменьшить число термов и входных переменных в системе функций  $\Phi$  по сравнению с базовой схемой КМУУ с общей памятью [2]. Это приводит к уменьшению числа LUT-элементов и их уровней в схеме КМУУ. Однако недостатком такого подхода является наличие схемы ПА, которая потребляет некоторые ресурсы кристалла FPGA.

## Основная идея предлагаемого метода

Выполним адресацию МК таким образом, чтобы при выполнении условия (2) максимально возможное число классов  $V_i \in \Pi_C$  представлялось одним обобщенным интервалом R-мерного булева пространства. Пусть  $n_i$  – число интервалов для класса  $V_i \in \Pi_C$ . Разделим множество  $\Pi_C$  на множества  $\Pi_1$  и  $\Pi_2$ . При этом  $V_i \in \Pi_1$ , если  $n_i > 1$ , и  $V_i \in \Pi_2$ , если  $n_i = 1$ .

Закодируем классы  $V_i \in \Pi_1$  двоичными кодами  $K(V_i)$  разрядности

$$R_1 = \lceil \log_2 I_1 \rceil, \quad (6)$$

где  $I_1 = |\Pi_1|$ . Используем для кодирования переменные  $\tau_r \in \tau$ , где  $|\tau| = R_1$ .

При реализации КМУУ в базисе FPGA управляющая память реализуется на блоках ЕМВ. Эти блоки имеют фиксированную ёмкость, однако их конфигурация может меняться. Для реализации системы  $Y \cup \{y_0, y_E\}$  необходимо

$$n_f = \left\lceil \frac{N+2}{t_F} \right\rceil \quad (7)$$

блоков ЕМВ, имеющих не менее M слов. При этом параметр  $t_F$  (число выходов блока) определяется как

$$t_F = \left\lceil \frac{V}{2^R} \right\rceil. \quad (8)$$

Этот параметр затем уточняется как ближайшее меньшее целое к одному из элементов множества фиксированных выходов  $O_F = \{1, 2, 4, 8, 16\}$  [6,7]. Общее число выходов памяти можно найти как

$$t_O = \lceil n_f \cdot t_F \rceil. \quad (9)$$

Если выполняется условие

$$t_O - (N+3) \geq 1, \quad (10)$$

то для реализации УУ предлагается модель КМУУ  $U_2$  (рис. 2).

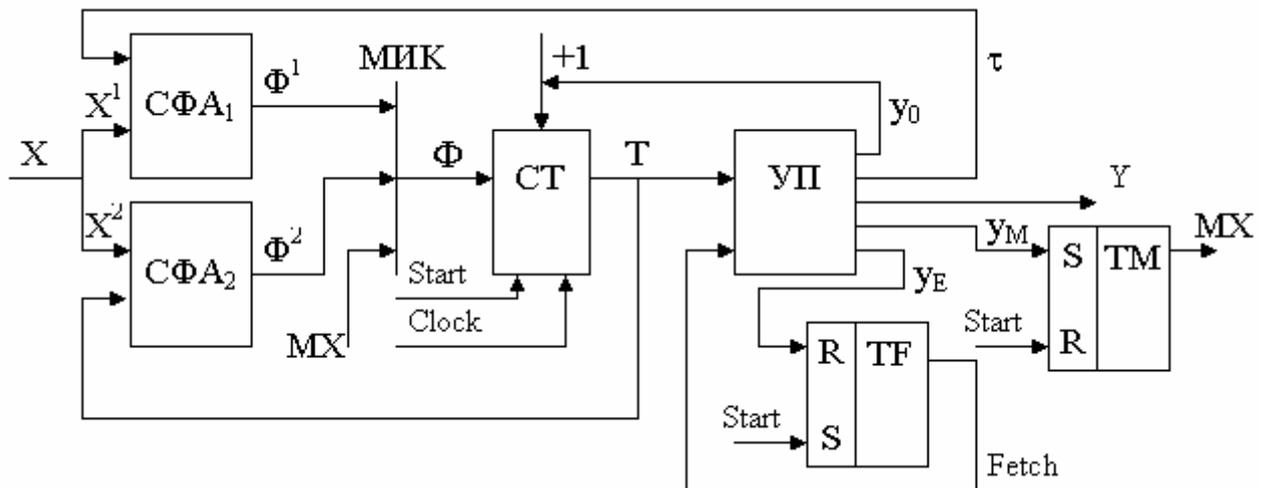


Рисунок 2 – Структурная схема КМУУ  $U_2$

В КМУУ  $U_2$  управляющая память формирует переменные  $\tau_r \in \tau$ , кодирующие классы  $V_i \in \Pi_1$ , и  $u_M$ , управляющие триггером ТМ. Если  $MX = 1$ , то в качестве функций  $\Phi$  используются выходы схемы СФА<sub>1</sub>:

$$\Phi^1 = \Phi^1(\tau, X^1). \quad (11)$$

Если  $MX = 0$ , то в качестве функций  $\Phi$  используются выходы схемы СФА<sub>2</sub>:

$$\Phi^2 = \Phi^2(T, X^2). \quad (12)$$

Таким образом, переменная  $MX$  управляет мультиплексором МИК, выбирающим один из двух источников функций  $D_r \in \Phi$ . В остальном КМУУ  $U_2$  функционирует аналогично КМУУ  $U_1$ .

Предложенный подход позволяет исключить блок ПА, а также использовать естественную избыточность блоков ЕМВ для формирования кодов классов  $V_i \in \Pi_1$ . Итак, схема имеет два источника кодов  $K(V_i)$ : счетчик СТ для  $V_i \in \Pi_2$  и память УП для  $V_i \in \Pi_1$ . В настоящей работе предлагается метод синтеза КМУУ  $U_2$ , включающий следующие этапы:

1. Формирование множества ОЛЦ С для ГСА Г.
2. Формирование разбиения  $\Pi_C$  множества С.
3. Оптимальная адресация микрокоманд.
4. Кодирование классов  $V_i \in \Pi_1$ .
5. Формирование таблицы переходов для классов  $V_i \in \Pi_1$ .
6. Формирование таблицы переходов для классов  $V_i \in \Pi_2$ .
7. Формирование таблицы содержимого управляющей памяти.
8. Реализация схемы КМУУ в заданном элементном базисе.

### ***Пример применения предложенного метода***

Пусть для некоторой ГСА Г получено множество ОЛЦ  $S = \{\alpha_1, \dots, \alpha_{12}\}$ , где  $\alpha_1 = \langle b_1, \dots, b_4 \rangle$ ,  $\alpha_2 = \langle b_5 \rangle$ ,  $\alpha_3 = \langle b_6, b_7, b_8 \rangle$ ,  $\alpha_4 = \langle b_9, b_{10} \rangle$ ,  $\alpha_5 = \langle b_{11}, b_{12}, b_{13}, b_{14} \rangle$ ,  $\alpha_6 = \langle b_{15}, b_{16} \rangle$ ,  $\alpha_7 = \langle b_{17}, \dots, b_{20} \rangle$ ,  $\alpha_8 = \langle b_{21}, \dots, b_{24} \rangle$ ,  $\alpha_9 = \langle b_{25}, b_{26} \rangle$ ,  $\alpha_{10} = \langle b_{27}, b_{28} \rangle$ ,  $\alpha_{11} = \langle b_{29}, b_{30}, b_{31} \rangle$ ,  $\alpha_{12} = \langle b_{32} \rangle$ . Пусть эти ОЛЦ могут быть разбиты на  $I = 5$  классов, где  $V_1 = \{\alpha_1, \alpha_2, \alpha_3\}$ ,  $V_2 = \{\alpha_4, \alpha_5, \alpha_6\}$ ,  $V_3 = \{\alpha_7, \alpha_8, \alpha_9\}$ ,  $V_4 = \{\alpha_{10}, \alpha_{11}\}$ ,  $V_5 = \{\alpha_{12}\}$ . Пусть выход ОЛЦ  $\alpha_{12} \in S$  связан с входом вершины  $b_E$ . Как известно, переходы из таких ОЛЦ не рассматриваются, так как их последняя вершина должна включать переменную  $u_E$  [2].

Выполним адресацию микрокоманд так, чтобы выполнялось условие (2) и максимально возможное число классов представлялось одним обобщенным интервалом  $R$ -мерного булева пространства. В рассматриваемом примере  $M = 32$ , то есть  $R = 5$  и  $T = \{T_1, \dots, T_5\}$ . Один из возможных вариан-

тов оптимальной адресации микрокоманд приведен на рис. 3. Этот рисунок содержит видоизмененную карту Карно, которая достаточна для получения обобщенных интервалов. Из рис. 3 имеем: класс  $V_1$  соответствует интервалу  $00^{***}$ , класс  $V_2$  –  $01^{***}$ , класс  $V_3$  соответствует двум интервалам:  $10^{***}$  и  $1100^*$ , класс  $V_4$  соответствует двум интервалам:  $11011$  и  $11110$ . Интервал для класса  $V_5$  не имеет значения. Символ  $U_2(\Gamma)$  означает, что КМУУ  $U_2$  реализуется по ГСА  $\Gamma$ .

		$T_1 T_2 T_3$							
		000	001	010	011	100	101	110	111
$T_4 T_5$	00	$b_1$	$b_5$	$b_9$	$b_{13}$	$b_{17}$	$b_{21}$	$b_{25}$	$b_{29}$
	01	$b_2$	$b_6$	$b_{10}$	$b_{14}$	$b_{18}$	$b_{22}$	$b_{26}$	$b_{30}$
	10	$b_3$	$b_7$	$b_{11}$	$b_{15}$	$b_{19}$	$b_{23}$	$b_{27}$	$b_{31}$
	11	$b_4$	$b_8$	$b_{12}$	$b_{16}$	$b_{20}$	$b_{24}$	$b_{28}$	$b_{32}$
		$V_1$		$V_2$		$V_3$		$V_4$	$V_5$

Рисунок 3 – Адреса микрокоманд КМУУ  $U_2(\Gamma)$

Итак,  $\Pi_1 = \{V_3, V_4\}$ ,  $I_1 = 2$ ,  $\Pi_2 = \{V_1, V_2\}$ . Для кодирования классов  $V_i \in \Pi_1$  достаточно  $R_1 = 1$  переменной, то есть  $\tau_r \in \{\tau_1\}$ . Коды классов  $V_i \in \Pi_2$  уже были найдены, так как они совпадают с соответствующими интервалами. Таким образом,  $K(V_1) = 00^{***}$ ,  $K(V_2) = 01^{***}$ . Так как для обоих кодов  $T_1 = 0$ , то классы  $V_1$  и  $V_2$  можно идентифицировать переменной  $T_2$ . Пусть  $K(V_3) = 0$ ,  $K(V_4) = 1$ .

Таблицы переходов для классов  $V_i \in \Pi_C$  строятся по системе обобщенных формул перехода [2,3]. Пусть в рассматриваемом примере эта система имеет вид:

$$\begin{aligned}
 V_1 &\rightarrow x_1 b_9 \vee \overline{x_1} x_2 b_{11} \vee \overline{x_1} \overline{x_2} x_3 b_{13} \vee \overline{x_1} x_2 \overline{x_3} b_{15}; \\
 V_2 &\rightarrow x_2 b_5 \vee \overline{x_2} x_5 b_6 \vee \overline{x_2} \overline{x_5} x_6 b_{17} \vee \overline{x_2} x_5 \overline{x_6} b_{22}; \\
 V_3 &\rightarrow x_4 x_6 b_{21} \vee \overline{x_4} \overline{x_6} b_{25} \vee \overline{x_4} x_3 b_{27} \vee \overline{x_4} x_3 b_{29}; \\
 V_4 &\rightarrow x_1 b_{32} \vee \overline{x_1} b_{23}.
 \end{aligned} \tag{13}$$

Таблицы переходов имеют следующие столбцы  $V_i$ ,  $K(V_i)$ ,  $b_q$ ,  $A(b_q)$ ,  $X_h$ ,  $\Phi_h$ ,  $h$ . Здесь  $X_h$  – набор ЛУ, который определяет связь между классом  $V_i$  и вершиной  $b_q$  в формулах перехода вида (13);  $\Phi_h$  – набор функций возбуждения СТ, равных единице для формирования адреса  $A(b_q)$ . Разница между таблицами заключается в том, что для классов  $V_i \in \Pi_1$  код представляется переменными  $\tau_r \in \tau$ , а для  $V_i \in \Pi_2$  – переменными  $T_r \in T$ . Для

КМУУ  $U_2(\Gamma)$  переходы для классов  $V_i \in \Pi_1$  представлены в табл. 1, а для классов  $V_i \in \Pi_2$  – в табл. 2. Первая таблица имеет  $N_1 = 6$  строк, а вторая –  $N_2 = 8$  строк.

Таблица 1. Таблица переходов для классов  $V_i \in \Pi_1$

$V_i$	$K(V_i)$	$b_q$	$A(b_q)$	$X_h$	$\Phi_h$	$h$
$V_3$	0	$b_{21}$	10100	$x_4 x_6$	$D_1 D_3$	1
		$b_{25}$	11000	$\overline{x_4 x_6}$	$D_1 D_2$	2
		$b_{27}$	11010	$\overline{x_4 x_3}$	$D_1 D_2 D_4$	3
$V_4$	1	$b_{32}$	11111	$x_1$	$D_1 D_2 D_3 D_4 D_5$	4
		$b_{23}$	10110	$\overline{x_1}$	$D_1 D_3 D_4$	5

Таблица 2. Таблица переходов для классов  $V_i \in \Pi_2$

$V_i$	$K(V_i)$	$b_q$	$A(b_q)$	$X_h$	$\Phi_h$	$h$
$V_1$	0	$b_9$	01000	$x_1$	$D_2$	1
		$b_{11}$	01010	$\overline{x_1 x_2}$	$D_2 D_4$	2
		$b_{13}$	01100	$\overline{x_1 x_2 x_3}$	$D_2 D_3$	3
		$b_{15}$	01110	$\overline{x_1 x_2 x_3}$	$D_2 D_3 D_4$	4
$V_2$	1	$b_5$	00100	$x_2$	$D_3$	5
		$b_6$	00101	$\overline{x_2 x_5}$	$D_3 D_5$	6
		$b_{17}$	10000	$\overline{x_2 x_5 x_6}$	$D_1$	7
		$b_{22}$	10101	$\overline{x_2 x_5 x_6}$	$D_1 D_3 D_5$	8

Для табл.1 и табл.2 адреса микрокоманд берутся из рис. 3. Из табл.1 имеем систему функций (11). Например,  $D_1 = 1$ ,  $D_2 = \overline{\tau_1 x_4 x_6} \vee \overline{\tau_1 x_4} \vee \overline{\tau_1 x_1}$  (после минимизации). Из табл. 2 имеем систему (12). Например,  $D_1 = \overline{T_1 x_2 x_5}$ ,  $D_2 = \overline{T_1}$  (после минимизации). Из этих таблиц имеем также множества  $X^1 = \{x_1, x_3, x_4, x_6\}$ ,  $X^2 = \{x_1, x_2, x_3, x_5, x_6\}$ ,  $\Phi^1 = \Phi^2 = \Phi$ .

Пусть для реализации управляющей памяти используются блоки ЕМВ, имеющие конфигурации  $256 \times 1$ ,  $128 \times 2$ ,  $64 \times 4$ ,  $32 \times 8$ . Таким образом  $O_F = \{1, 2, 4, 8\}$ , а  $V = 256$  бит при  $t = 1$ . Для рассматриваемого примера из (8) имеем  $t_F = 256 : 32 = 8$ , таким образом,  $t_F = 8$ . Пусть для ГСА  $\Gamma$  имеем  $N = 28$ , тогда  $n_F = 4$ ,  $t_0 = 32$  и условие (10) выполняется. Таким образом, модель  $U_2$  может быть использована для этих условий.

Для формирования содержимого УП предлагается следующая методика, сводящаяся к преобразованию содержимого опера торных вершин ГСА  $\Gamma$ :

1. Если вершина  $b_q \in D^g$  не является выходом ОЛЦ  $\alpha_g \in C$ , то в неё включается переменная  $y_0$ .

2. Если вершина  $b_q \in D^g$  связана с вершиной  $b_E$ , то в неё включается переменная  $y_E$ .

3. Если вершина  $b_q \in D^g$  является выходом ОЛЦ  $\alpha_g \in V_i$ , где  $V_i \in \Pi_1$ , то в неё вводится переменная  $y_M$  и переменные  $\tau_r \in \tau$ , равные единице в коде  $K(V_i)$ .

Например, в вершину  $b_2 \in D^1$  вводится  $y_0$ , в вершину  $b_{20} - y_M$ , в вершину  $b_{28} - y_M$  и  $\tau_1$ , а в  $b_{32} - y_E$ . Далее в ячейку УП с адресом  $A(b_q)$  записывается набор  $Y(b_q)$  и элементы множества  $\tau \cup \{y_0, y_M, y_E\}$ , записанные в вершине  $b_q \in E_1$  при преобразовании ГСА Г.

Реализация схемы КМУУ  $U_2$  сводится к реализации систем функций (11)-(12) на LUT элементах и УП на ЕМВ. Для решения этой задачи используются стандартные промышленные пакеты [6,7]. Этот этап выходит за пределы нашей статьи.

## **Заключение**

Предлагаемый в работе метод оптимизации схемы КМУУ основан на использовании двух источников кодов классов псевдоэквивалентных ОЛЦ, а также мультиплексора для выбора одного из источников. Такой подход позволяет исключить из КМУУ преобразователь адреса микрокоманд, а для преобразования адресов выходов ОЛЦ в коды классов используется естественная избыточность блоков ЕМВ, используемых для реализации управляющей памяти.

Проведенные авторами исследования показали, что предложенный метод позволяет на 12-14% уменьшить число LUT-элементов в КМУУ  $U_2$  по сравнению с эквивалентным КМУУ  $U_1$ . Однако применение модели  $U_2$  возможно только при выполнении условия (1). Если  $t_0 - (N + 2) \geq R_C$ , то целесообразно коды всех классов  $V_i \in \Pi_C$  формировать при помощи УП. При этом исчезает мультиплексор МХ и триггер ТМ.

Научная новизна предложенного метода заключается в использовании особенностей КМУУ (наличие классов псевдоэквивалентных ОЛЦ) для уменьшения числа LUT элементов в схеме КМУУ.

Практическая значимость метода заключается в уменьшении площади кристалла FPGA, занимаемой схемой КМУУ с преобразователем адреса микрокоманд, что позволяет получить схемы, обладающие меньшей стоимостью, чем известные из литературы аналоги.

Дальнейшие направления работы связаны с разработкой метода вы-

бора оптимальной структуры КМУУ с преобразователем адреса микрокоманд при реализации схемы в базисе FPGA.

### **Список литературы**

1. Barkalov A., Titarenko L. Logic synthesis for compositional microprogram control units. – Berlin: Springer, 2008. – 272 pp.
2. Баркалов А.А., Титаренко Л.А. Синтез микропрограммных автоматов на заказных и программируемых СБИС. – Донецк: УНИТЕХ, 2009.—336 с.
3. Barkalov A., Titarenko L. Logic synthesis for FSM-based control units. – Berlin: Springer, 2009. – 233 pp.
4. Maxfield S. The Design Warrior's Guide to FPGAs. – Amsterdam: Elsevier, 2004. – 541 pp.
5. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах с программируемой структурой – С-Пб: БХВ – Петербург, 2006. – 736 с.
6. xilinx.com.
7. altera.com.
8. Baranov S. Logic and System Design of Digital Systems. – Tallinn: TTU, 2008. – 266 pp.
9. Баркалов А.А., Вишневский Р., Ефименко К.Н. Реализация композиционного микропрограммного устройства управления на FPGA // Радиоэлектроника. Информатика. Управление. – 2005. – №2. – С.127-131.

*Надійшла до редакції 17.11.2010 р.*

*Рецензент:*

**О.О. Баркалов<sup>1</sup>, Л.О. Титаренко<sup>1</sup>, К.М. Єфіменко<sup>2</sup>, Я.М. Липинський<sup>1</sup>**  
Університет Зеленогурський (Польща)<sup>1</sup>,  
Донецький національний технічний університет<sup>2</sup>

**Оптимізація схеми КМПК із перетворювачем адреси мікрокоманд.** В роботі запропоновано метод зменшення апаратних витрат у схемі КМПК із перетворювачем адреси мікрокоманд, який орієнтовано на технологію FPGA. Метод засновано на використанні двох джерел кодів класів псевдоеквівалентних ОЛЛ та мультиплексу, який дозволяє вибрати одне з цих джерел. Такий підхід дозволить зменшити число LUT елементів у схемі адресації КМПК. Наведено приклад використання запропонованого методу.

**КМПК, ГСА, ОЛЛ, FPGA, логічна схема**

**A.A Barkalov<sup>1</sup>, L.A. Titarenko<sup>1</sup>, K.N. Efimenko<sup>2</sup>, J.M. Lipinsky<sup>1</sup>**  
University of Zelena Gora, Poland<sup>1</sup>, Donetsk National Technical University<sup>2</sup>

**Optimization of circuit for CMCU with transformer of microinstruction address.** A method for reducing the hardware amount in the circuit of CMCU with transformer of microinstruction address is proposed oriented on FPGA technology. The method is based on the use of two sources of codes classes of pseudoequivalent OLC and a multiplexer to choose one of these sources. Such an approach would reduce the number of LUT elements in the addressing circuit of CMCU. An example of the proposed method application is given.

**CMCU, GSA, OLC, FPGA, logic circuit**