

УДК 004.274

А.А. Баркалов<sup>1</sup>, д.т.н., проф.,  
Л.А. Титаренко<sup>1</sup>, д.т.н., проф.,  
К.Н. Ефименко<sup>2</sup>, к.т.н., доц.,  
И.Я. Зеленева<sup>2</sup>, к.т.н., доц.<sup>1</sup> Зеленогурский университет, г. Зеленая гора, Польша  
A.Barkalov@iie.uz.zgora.pl<sup>2</sup> Донецкий национальный технический университет, г. Донецк, Украина  
KN\_Efimenko@mail.ru, irina@cs.dgtu.donetsk.ua

## ОПТИМИЗАЦИЯ КМУУ С РАЗДЕЛЕНИЕМ КОДОВ

Предлагается метод уменьшения аппаратных затрат в схеме КМУУ с разделением кодов, 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.

**Композиционное микропрограммное устройство управления, граф-схема алгоритма, операторная линейная цепь, FPGA, логическая схема**

### Введение

Одним из путей реализации схемы устройства управления (УУ) по линейной граф-схеме алгоритма (ГСА) является использование модели композиционного микропрограммного устройства управления (КМУУ) с разделением кодов [1,2]. В настоящее время для реализации схем УУ широко используются программируемые логические интегральные схемы (ПЛИС) вида FPGA (Field-Programmable Gate Arrays) [3-6]. Основу FPGA представляют макроячейки LUT (Look-Up Table), имеющие ограниченное число входов (4-6) [7,8]. Для оптимизации схемы УУ на FPGA необходимо уменьшить количество аргументов и термов в реализуемых системах булевых функций [9]. В настоящей работе предлагается метод решения этой задачи для КМУУ с разделением кодов. Метод основан на использовании двух источников кодов классов псевдоэквивалентных операторных линейных цепей (ОЛЦ) и мультиплексора источников кодов (МИК). Предлагаемый метод является развитием идей, предложенных в работе [10].

Целью исследования является оптимизация схемы КМУУ с разделением кодов за счет мультиплексирования источников кодов псевдоэквивалентных ОЛЦ.

Задачей исследования является разработка метода синтеза КМУУ с разделением кодов, позволяющего уменьшить число LUT-элементов в комбинационной части КМУУ.

### Базовая информация о КМУУ с разделением кодов

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  $\bar{b}_g = \langle b_{g^1}, \dots, b_{g^z} \rangle$  такая, что для любой пары соседних компонент  $b_{g^i}, b_{g^{i+1}}$ , где  $i$  – номер компоненты кортежа  $\bar{b}_g$ , существует дуга  $\langle b_{g^i}, b_{g^{i+1}} \rangle \in E$ .

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

Определение 4. ОЛЦ  $\bar{b}_i, \bar{b}_j$  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  $C = \{b_1, \dots, b_G\}$ , 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

$$R = \lceil \log_2 M \rceil. \tag{1}$$

Пусть  $F_{\max} = \max(F_1, \dots, F_G)$  – максимальное число компонент в ОЛЦ. Закодируем каждую ОЛЦ  $b_g \in C$  двоичным кодом  $K(b_g)$ , имеющим  $R_1$  разрядов, где

$$R_1 = \lceil \log_2 G \rceil. \tag{2}$$

Для определения любой вершины  $b_q \in D^s$  достаточно  $R_2$  разрядов, представляющих код  $K(b_q)$ . При этом

$$R_2 = \lceil \log_2 F_{\max} \rceil. \tag{3}$$

Пусть для ГСА  $\Gamma$  выполняется следующее условие:

$$R_1 + R_2 = R. \tag{4}$$

В этом случае для реализации алгоритма  $\Gamma$  целесообразно использовать модель КМУУ с разделением кодов (рис. 1). В этой модели для кодирования ОЛЦ используются переменные  $\tau_i \in \tau$ , где  $|\tau| = R_1$ . Для кодирования компонент ОЛЦ используются переменные  $T_i \in T$ , где  $|T| = R_2$ . Коды компонент выбраны так, чтобы выполнялась естественная адресация микрокоманд [1]. Для этого код первой компоненты любой ОЛЦ равен 0, второй – 1 и так далее. Естественно, что эти десятичные числа представлены их двоичными  $R_2$ -разрядными эквивалентами.

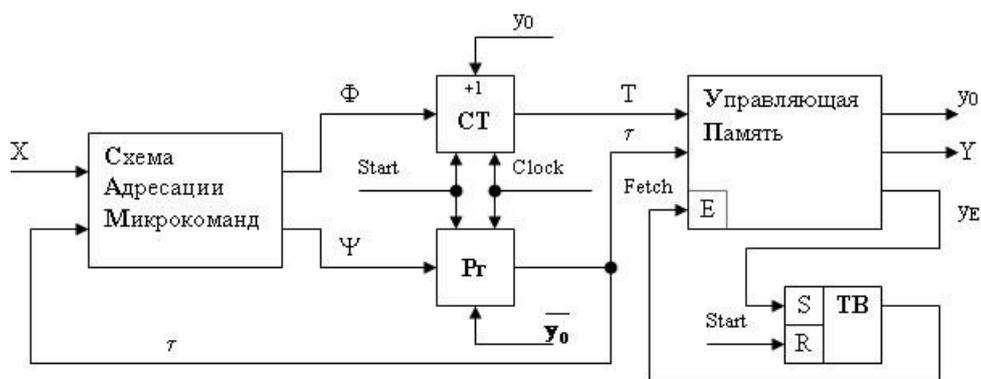


Рисунок 1 – Структурная схема КМУУ с разделением кодов

Условимся в дальнейшем обозначать КМУУ (рис. 1) символом  $U_1$ .

В КМУУ  $U_1$  схема адресации микрокоманд (САМ) реализует систему функций возбуждения счетчика СТ и триггера Pr

$$\begin{aligned} \Phi &= \Phi(\tau, X), \\ \Psi &= \Psi(\tau, X). \end{aligned} \tag{5}$$

При этом адрес микрокоманды  $MI_q$  представляется в виде

$$A(b_q) = K(\alpha_g) * K(b_q), \tag{6}$$

где вершина  $b_q$  входит в состав ОЛЦ  $b_g \in C$ , \* – знак операции конкатенации.

Композиционное МУУ  $U_1$  функционирует следующим образом. По сигналу Start в Pr и СТ заносится начальный адрес микропрограммы, а триггер выборки ТВ устанавливается в единичное состояние. При этом Fetch = 1, что разрешает выборку команд из управляющей памяти (УП). Если считанная микрокоманда не соответствует выходу ОЛЦ, то одновременно с микрооперациями  $Y(b_q)$  формируется сигнал  $y_0$ . Если  $y_0 = 1$ , то к содержимому СТ прибавляется единица и адресуется следующая компонента

текущей ОЛЦ. Если выход ОЛЦ достигнут, то  $y_0 = 0$ . При этом адрес входа следующей ОЛЦ формируется схемой САМ. При достижении окончания микропрограммы формируется сигнал  $y_E$ , триггер ТВ обнуляется, и выборка микрокоманд прекращается.

Число LUT-элементов в схеме САМ зависит от числа аргументов и термов в системе (5). В настоящей работе предлагается метод, позволяющий уменьшить сложность функций в системе (5) и, следовательно, уменьшить аппаратные затраты в схеме САМ.

### Основная идея предлагаемого метода

Пусть ОЛЦ  $b_g \in C_1$ , если  $O_g$  не связан с конечной вершиной ГСА  $\Gamma$ . Найдем разбиение  $P_C = \{B_1, \dots, B_l\}$  множества  $C_1$  на классы псевдоэквивалентных ОЛЦ (ПОЛЦ). Выполним кодирование  $b_g \in C$  так, чтобы максимально возможное число классов  $B_i \in P_C$ , где  $|P_C| = l$ , представлялось одним обобщенным интервалом  $R_1$ -мерного булева пространства. Пусть  $n_i$  – число обобщенных интервалов, представляющих класс.

Представим множество  $\Pi_C$  в виде  $\Pi_C = \Pi_A \cup \Pi_B$ . При этом множества  $\Pi_A$  и  $\Pi_B$  строятся следующим образом:

$$\begin{aligned} (n_i = 1) &\rightarrow V_i \in \Pi_A, \\ (n_i > 1) &\rightarrow V_i \in \Pi_B. \end{aligned} \tag{7}$$

Источником кодов классов  $V_i \in \Pi_A$  является регистр  $R_i$ . При этом код класса  $V_i \in \Pi_A$  определяется соответствующим интервалом  $R_1$ -мерного булева пространства.

Закодируем классы  $V_i \in \Pi_B$  двоичными кодами  $C(V_i)$  разрядности

$$R_3 = \lceil \log_2(|\Pi_B| + 1) \rceil. \tag{8}$$

Используем для кодирования классов  $V_i \in \Pi_B$  переменные из множества  $Z = \{z_1, \dots, z_{R_3}\}$ . Для формирования кодов  $C(V_i)$  необходим блок преобразователя кодов (БПК). Этот блок реализует систему функций

$$Z = Z(\tau). \tag{9}$$

Очевидно, для реализации схемы БПК необходимы некоторые ресурсы кристалла FPGA. В настоящей работе предлагается уменьшить эти ресурсы за счет использования избыточности встроенных блоков памяти EMB, входящих в

состав FPGA [7,8]. Эти блоки имеют фиксированную емкость, однако их конфигурация может меняться.

Для реализации схемы  $Y \cup \{y_0, y_E\}$  необходимо

$$n_f = \left\lceil \frac{N+2}{t_f} \right\rceil \tag{10}$$

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

$$t_f = \left\lceil \frac{V}{2^R} \right\rceil. \tag{11}$$

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

$$t_o = \lceil n_f \cdot t_f \rceil. \tag{12}$$

Если выполняется условие

$$t_o - (N+3) \geq 1, \tag{13}$$

то для реализации схемы устройства управления предлагается модель КМУУ  $U_2$  (рис. 2).

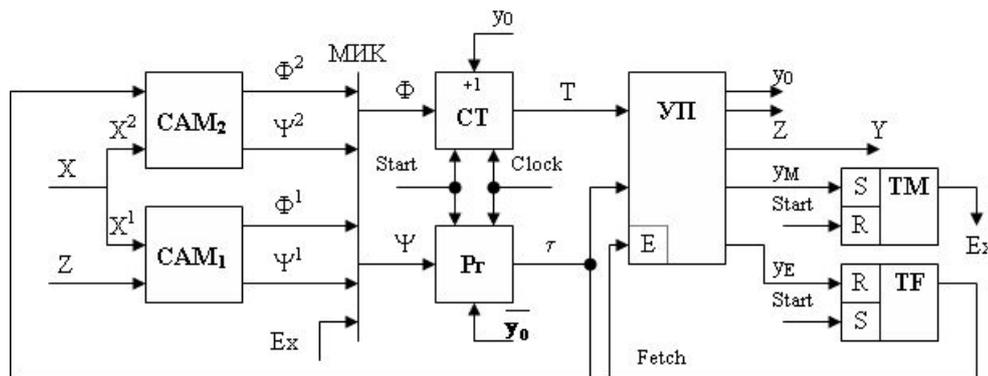


Рисунок 2 – Структурная схема КМУУ  $U_2$

Эта модель имеет ряд отличий от модели  $U_1$ . Во-первых, блок CAM разделен на два блока. Блок  $CAM_1$  реализует переходы, определяемые множеством  $\Pi_B$ , а блок  $CAM_2$  – множеством  $\Pi_A$ . Мультиплексор МИК служит для выбора источника кодов, используя переменную  $E_x$ . Значение переменной  $E_x$  определяется состоянием триггера ТМ, управляемого дополнительной переменной  $y_M$ . Блок УП является источником кодов классов для схемы  $CAM_1$ . Источниками кодов для схемы  $CAM_2$  является регистр  $R_i$ . Предлагаемое КМУУ функционирует следующим образом.

По сигналу Start в  $R_i$  и  $CT$  заносятся нулевые коды (адрес первой микрокоманды), а триггера TF и ТМ устанавливаются соответственно в "1" (Fetch = 1) и "0" ( $E_x = 0$ ). Пока адрес входа не достигнут, КМУУ  $U_2$

функционирует как  $U_1$ . При достижении адреса выхода ОЛЦ  $b_g \in V_i$  может формироваться переменная  $y_M = 1$ . В этом случае  $E_x = 1$  и адрес перехода формируется схемой  $CAM_1$ . Для этого формируется система функций:

$$\begin{aligned} \Phi^1 &= \Phi^1(Z, X^1); \\ \Psi^1 &= \Psi^1(Z, X^1). \end{aligned} \tag{14}$$

Если  $V_i \in \Pi_A$ , то переменная  $y_M$  не формируется. При этом  $E_x = 0$  и адрес перехода определяется схемой  $CAM_2$ . Для этого формируется система функций:

$$\begin{aligned} \Phi^2 &= \Phi^2(\tau, X^2); \\ \Psi^2 &= \Psi^2(\tau, X^2). \end{aligned} \tag{15}$$

Соответствующие функции передаются на выход МИК и требуемые коды  $K(b_g)$  и

$K(b_q)$  загрузаються відповідно в Рг і СТ. Функціонування продовжується обычним образом до формування змінної УЕ (досягнення закінчення алгоритма управління).

Такий підхід дозволяє зменшити число термів в системі (5) до абсолютного мінімуму. При виконанні умови

$$R_3 < R_1 \tag{16}$$

зменшується число аргументів в системі (10) по порівнянню з відповідними функціями з системи (5). Недоліком даного підходу є наявність блоку МІК, що споживає деякі ресурси кристалла. Однак схема МІК реалізується за рахунок використання тристабільних виходів макроячеек, тому доповільних LUT-елементів не потрібно. Блок УП реалізується на реконфігуруємих блоках ЕМВ, які мають строго визначене число виходів, що входять в множину {1, 2, 4, 8, 18, 36}. При цьому існує висока ймовірність наявності невикористаних виходів блоків ЕМВ, що входять в склад УП пристрою  $U_1$ .

**Особенности реализации схемы КМУУ  $U_2$**

В настоящей работе предлагается метод синтеза КМУУ  $U_2$ , включающий следующие этапы:

1. Формирование для ГСА Г множеств  $C$ ,  $C_1$ , и  $\Pi_C$ .
2. Оптимальное кодирование ОЛЦ  $b_g \in C_1$  и кодирование компонент ОЛЦ.
3. Формирование множеств  $\Pi_A$  и  $\Pi_B$ .
4. Кодирование классов  $V_i \in \Pi_B$  кодами  $C(V_i)$ .
5. Формирование таблицы переходов для классов  $V_i \in \Pi_A$ .
6. Формирование таблицы переходов для классов  $V_i \in \Pi_B$ .
7. Формирование содержимого управляющей памяти.
8. Синтез схемы КМУУ в заданном базисе.

Этапы 1-4 выполняются по известным методикам [1-3]. Этап 8 связан с разработкой VHDL-моделей КМУУ и использованием стандартных промышленных пакетов [7,8]. Эти этапы не представляют особого интереса для иллюстрации синтеза схемы КМУУ  $U_2$ . В этой связи мы не рассматриваем данные этапы в нашей статье.

Пусть для некоторой ГСА  $\Gamma_1$  получено множество ОЛЦ  $C = \{b_1, \dots, b_{16}\}$  и разбиение  $\Pi_C = \{V_1, \dots, V_7\}$ . При этом  $b_{16} \notin C_1$ , а классы  $V_i \in \Pi_C$  определяются следующим образом:  $V_1 = \{b_1\}$ ,  $V_2 = \{b_2, b_3, b_4\}$ ,  $V_3 = \{b_5, b_6\}$ ,  $V_4 = \{b_7, b_8, b_9\}$ ,  $V_5 = \{b_{10}, b_{11}\}$ ,  $V_6 = \{b_{12}\}$ ,

$V_7 = \{b_{13}, b_{14}, b_{15}\}$ . Итак,  $G = 16$ ,  $R_1 = 4$ ,  $\tau = \{\tau_1, \dots, \tau_4\}$ .

Оптимальное кодирование ОЛЦ  $b_g \in C_1$  выполняется так, чтобы максимально возможное число классов  $V_i \in \Pi_C$  представлялось одним интервалом  $R_1$ -мерного булева пространства [2]. Один из вариантов кодирования представлен на рис. 3.

		$\tau_3\tau_4$				
			00	01	11	10
$\tau_1\tau_2$	00	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	
	01	$\alpha_5$	$\alpha_6$	$\alpha_{10}$	$\alpha_{11}$	
	11	$\alpha_7$	$\alpha_8$	$\alpha_9$	$\alpha_{16}$	
	10	$\alpha_{13}$	$\alpha_{14}$	$\alpha_{15}$	$\alpha_{12}$	

Рисунок 3 – Коды ОЛЦ  $b_g \in C$

Из карты Карно (рис. 3) можно получить следующие интервалы, определяющие классы  $V_i \in \Pi_C$ . Класс  $V_1$  определяется интервалом 0000; класс  $V_2$  – интервалами 00\*1 и 001\*; класс  $V_3$  – интервалом 010\*; класс  $V_4$  – интервалами 110\* и 11\*1; класс  $V_5$  – интервалом 011\*; класс  $V_6$  – интервалом 1010; класс  $V_7$  – интервалами 100\* и 10\*1. Таким образом,  $\Pi_A = \{V_1, V_3, V_5, V_6\}$ , где  $K(V_1) = 0000$ ,  $K(V_3) = 010^*$ ,  $K(V_5) = 011^*$  и  $K(V_6) = 1010$ . Очевидно,  $\Pi_B = \{V_2, V_4, V_7\}$  и для их кодирования необходимо  $R_3 = 2$  разряда. Итак,  $Z = \{z_1, z_2\}$ . Закодируем классы  $V_i \in \Pi_B$  следующим образом:  $C(V_2) = 00$ ,  $C(V_4) = 01$ ,  $C(V_7) = 10$ .

Таблица переходов формируется на основе обобщённых формул переходов [4]. Пусть, например, из ГСА  $\Gamma_1$  можно получить следующие формулы:

$$V_1 \rightarrow x_1 b_3 \vee \overline{x_1} x_2 b_8 \vee \overline{x_1} x_2 b_{12}; \tag{17}$$

$$V_2 \rightarrow x_4 b_{12} \vee \overline{x_4} b_{17}.$$

Столбцы таблицы переходов включают следующую информацию: исходный класс (столбец  $V_i$ ); код класса (для  $\Pi_A$  это код  $K(V_i)$ , а для  $\Pi_B$  –  $C(V_i)$ ); адрес перехода ( $A(b_q)$ ); логические условия определяющие переход (столбец  $X_h$ ); функции возбуждения триггеров регистра Рг (столбец  $\Psi_h^1$  для  $\Pi_B$  и  $\Psi_h^2$  для  $\Pi_A$ ); функции возбуждения триггеров счетчика СТ (столбец  $\Phi_h^2$  для  $\Pi_A$  и  $\Phi_h^1$  для  $\Pi_B$ ); номер перехода (столбец h).

Пусть  $A(b_3) = 000100$ ,  $A(b_8) = 001000$ ,  $A(b_{12}) = 010101$ ,  $A(b_{17}) = 110010$ . Тогда фрагменты таблиц переходов для формул (17) приведены в

табл. 1 и табл. 2.

Таблица 1 – Таблица переходов для класса

$$B_1 \in \Pi_A$$

$B_i$	$K(B_i)$	$A(b_q)$	$X_h$	$\Psi_h^2$	$\Phi_h^2$	$h$
$B_1$	0000	000100	$x_1$	$D_4$	–	1
		001000	$\overline{x_1 x_2}$	$D_3$	–	2
		010101	$\overline{x_1 x_2}$	$D_2 D_4$	$D_6$	3

Таблица 2 – Таблица переходов для класса

$$B_2 \in \Pi_B$$

$B_i$	$C(B_i)$	$A(b_q)$	$X_h$	$\Psi_h^1$	$\Phi_h^1$	$h$
$B_2$	00	010101	$x_4$	$D_2 D_4$	$D_6$	1
		110010	$\overline{x_4}$	$D_1 D_2$	$D_5$	2

Система (15) может быть получена из таблицы переходов для классов  $B_i \in \Pi_B$ . Так, из табл. 2 имеем, например

$$D_1 = \overline{z_1 z_2 x_4};$$

$$D_6 = z_1 z_2 x_4.$$

Система (14) может быть получена из таблицы переходов для классов  $B_i \in \Pi_A$ . Так, из табл. 1, имеем, например,

$$D_2 = \overline{\tau_1 \tau_2 \tau_3 \tau_4 x_1 x_2};$$

$$D_6 = \tau_1 \tau_2 \tau_3 \tau_4 x_1 x_2.$$

Формирование содержимого УП выполняется тривиальным образом. В ячейки УП, соответствующие выходам ОЛЦ  $b_g \in B_i$ , где  $B_i \in \Pi_B$ , записывается код  $K(B_i)$ . Этот этап мы также не рассматриваем.

## Заключение

Предлагаемый в работе метод оптимизации КМУУ основан на мультиплексировании двух

источников кодов классов псевдоэквивалентных ОЛЦ. Такой подход позволяет гарантированно уменьшить число термов в системе функций возбуждения триггеров регистра и счетчика адресов микрокоманд до максимально возможной величины. Если КМУУ с разделением кодов рассматривать как автомат Мура, то предлагаемый подход позволяет уменьшить число термов до величины этого параметра у эквивалентного автомата Мили. Применение предложенного метода имеет смысл только при наличии свободных ресурсов встроенных блоков памяти, входящих в состав микросхемы FPGA.

Недостатком предложенного подхода является введение мультиплексора, который вносит дополнительную задержку в цикл работы КМУУ. Однако уменьшение числа термов ведёт к уменьшению числа уровней в схеме и задержка от введения МИК компенсируется. Проведенные авторами исследования показали, что предложенный метод позволяет до 36% уменьшить число LUT-элементов по отношению к исходному КМУУ. При этом время цикла КМУУ  $U_2$  всегда было меньше, чем у КМУУ  $U_1$ .

Научная новизна предложенного метода заключается в использовании особенностей КМУУ (наличие классов псевдоэквивалентных ОЛЦ) и кристалла FPGA (фиксированное число выходов блоков EMB) для уменьшения числа LUT-элементов в схеме КМУУ с разделением кодов.

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

## Список литературы

1. Палагин А.В., Баркалов А.А., Титаренко Л.А., Опанасенко В.Н. Проектирование реконфигурируемых цифровых систем. – Луганск: Издательство ВНУ, 2011. – 432 с.
2. Barkalov A., Titarenko L. Logic synthesis for compositional microprogram control units. – Berlin: Springer, 2009. 272 pp.
3. Баркалов А.А., Титаренко Л.А. Синтез микропрограммных автоматов на заказных и программируемых СБИС. – Донецк: УНИТЕХ, 2009. —336 с.
4. Barkalov A., Titarenko L. Logic synthesis for FSM-based control units. – Berlin: Springer, 2009. 233 pp.
5. Maxfield S. The Design Warrior's Guide to FPGAs. – Amsterdam: Elsevier, 2004. – 541 pp.
6. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах с программируемой структурой – С-Пб: БХВ – Петербург, 2006. – 736 с.
7. xilinx.com.
8. altera.com.
9. Baranov S. Logic and System Design of Digital Systems. – Tallinn: TTU, 2008. – 266 pp.
10. Баркалов А.А., Титаренко Л.А., Ефименко К.Н., Липински Я.М. Оптимизация схемы КМУУ с преобразователем адреса микрокоманд // Наукові праці ДонНТУ. Серія „Проблеми

моделювання та автоматизації проектування" (МАП-2011). Вип.9 (179). – Донецьк: ДонНТУ. – 2011. – с.26-35.

Надійшла до редколегії 30.03.2012

**О.О. БАРКАЛОВ<sup>1</sup>, Л.О. ТИТАРЕНКО<sup>1</sup>,  
К.М. ЄФІМЕНКО<sup>2</sup>, І.Я. ЗЕЛЕНЬОВА<sup>2</sup>**

<sup>1</sup> Зеленогурський університет, м. Зелена гора,  
Польща

<sup>2</sup> Донецький національний технічний університет

**A.A. BARKALOV<sup>1</sup>, L.A. TITARENKO<sup>1</sup>,  
K.N. EFIMENKO<sup>2</sup>, I.J. ZELENYOVA<sup>2</sup>**

<sup>1</sup> University of Zielona Gora, Poland,

<sup>2</sup> Donetsk National Technical University, Ukraine

#### **Оптимізація КМПК із розподілом кодів**

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

*Композиційний мікропрограмний пристрій керування, граф-схема алгоритму, операторний лінійний ланцюг, FPGA, логічна схема*

#### **Optimization of CMCU with code sharing**

A method for reducing the hardware amount in the circuit of CMCU with code sharing 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.

*Compositional microprogram control unit, GSA, operational linear object, FPGA, logic circuit*