

**А.А. Баркалов<sup>1</sup>, Л.А. Титаренко<sup>1</sup>,  
К.Н. Ефименко<sup>2</sup>, И.Я. Зеленёва<sup>2</sup>**

Университет Зеленогурский, г. Зеленая Гура, Польша<sup>1</sup>,

A.Barkalov@iie.uz.zgora.pl<sup>1</sup>

ГВУЗ «Донецкий национальный технический университет», г. Донецк<sup>2</sup>

## **ОПТИМИЗАЦИЯ КМУУ С ЭЛЕМЕНТАРНЫМИ ЦЕПЯМИ**

Предлагается метод уменьшения аппаратных затрат в схеме КМУУ с элементарными цепями, ориентированный на технологию FPGA. Метод основан на использовании двух источников кодов классов псевдоэквивалентных ЭОЛЦ 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, ЭОЛЦ, FPGA, логическая схема, ЕМВ**

### **Введение**

Композиционные микропрограммные устройства управления (КМУУ) являются эффективным средством реализации линейных алгоритмов управления [1,2]. Одной из моделей КМУУ является модель с разделением кодов [3], позволяющая при определенных условиях уменьшить аппаратные затраты в схеме адресации микрокоманд. В настоящее время микросхемы типа FPGA (field-programmable gate arrays) широко используются при реализации схем цифровых устройств [4,5]. Основу этих СБИС представляют макроячейки табличного типа, называемые LUT (look-up table). Как правило, LUT-элементы имеют ограниченное число входов (4-6) [6,7]. Для уменьшения числа LUT в схеме КМУУ необходимо уменьшить число аргументов и термов в системе функций адресации микрокоманд [1,8]. В настоящей работе предлагается один из подходов к решению этой задачи, основанный на мультиплексировании двух источников кодов классов псевдоэквивалентных элементарных операторных линейных цепей (ЭОЛЦ). Предлагаемый метод является развитием результатов, полученных в работе [9], и использует избыточность встроенных блоков памяти ЕМВ.

**Целью исследования** является уменьшение числа LUT-элементов в схеме КМУУ с элементарными цепями за счет мультиплексирования источников кодов классов псевдоэквивалентных ЭОЛЦ и использования избыточности встроенных блоков памяти ЕМВ.

**Задачей исследования** является разработка метода синтеза КМУУ с элементарными цепями, позволяющего оптимизировать схему адресации микрокоманд.

Алгоритм управления представлен в виде граф-схемы алгоритма (ГСА) [8]. Этот выбор определен наглядностью подобного представления и широким применением аппарата ГСА в практике инженерного проектирования.

### **Базовая информация о КМУУ с элементарными цепями**

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 (ОЛЦ) ГСА  $\Gamma$  называется конечная последовательность операторных вершин  $\bar{b}_g = \langle b_{g1}, \dots, b_{gF_g} \rangle$  такая, что для любой пары соседних компонент  $b_{gi}, b_{gi+1}$ , где  $i$  – номер компоненты кортежа  $\bar{b}_g$ , 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  $\bar{b}_g$ , называется входом ОЛЦ  $\bar{b}_g$ , если существует дуга  $\langle b_t, b_q \rangle \in E$ , где  $b_t \notin D^g$ . Элементарная ОЛЦ (ЭОЛЦ) имеет только один вход.

**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.** ЭОЛЦ  $\bar{b}_i, \bar{b}_j$  называются псевдоэквивалентными ЭОЛЦ, если их выходы связаны со входом одной и той же вершины  $b_q \in V$ .

Пусть для некоторой ГСА  $\Gamma$  сформировано множество ЭОЛЦ  $C = \{\bar{b}_1, \dots, \bar{b}_G\}$ , 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

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

Пусть  $F_{\max} = \max(F_1, \dots, F_G)$  – максимальное число компонент в ЭОЛЦ. Закодируем каждую ЭОЛЦ  $\bar{b}_g \in C$  двоичным кодом  $K(\bar{b}_g)$ , имею-

щим  $R_1$  разрядов, где

$$R_1 = \lceil \log_2 G \rceil. \quad 2)$$

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

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

Пусть для ГСА  $\Gamma$  выполняется следующее условие:

$$R_1 + R_2 = R. \quad 4)$$

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

Условимся в дальнейшем обозначать КМУУ (рис. 1) символом  $U_1$ . В КМУУ  $U_1$  схема адресации микрокоманд (САМ) реализует систему функций возбуждения триггера  $P_r$

$$\Psi = \Psi(\tau, X). \quad 5)$$

При этом адрес микрокоманды  $MI_q$  представляется в виде

$$A(b_q) = K(\alpha_g) * K(b_q), \quad 6)$$

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

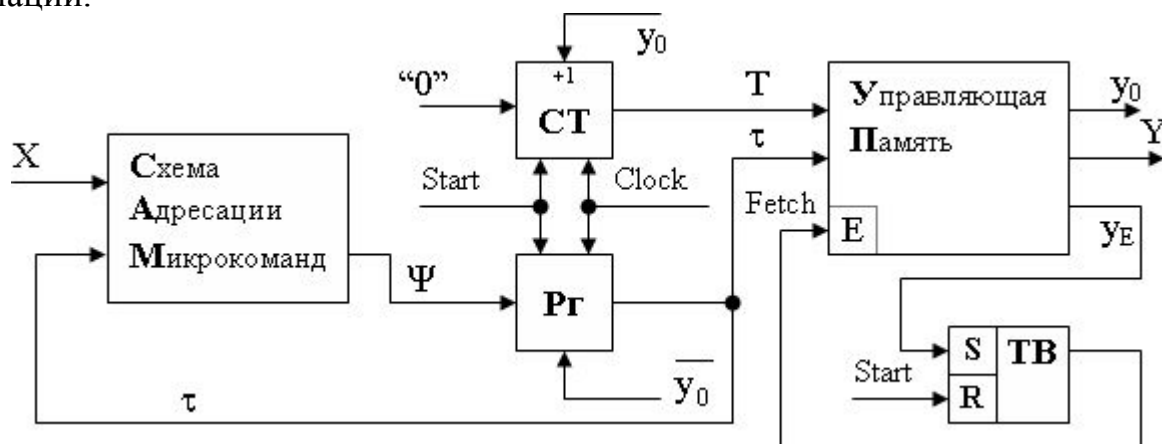


Рисунок 1 - Структурная схема КМУУ с элементарными цепями

КМУУ  $U_1$  функционирует следующим образом. По сигналу Start в  $P_r$

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 ЭОЛЦ, 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 ЭОЛЦ. Если выход ОЛЦ достигнут, то  $y_0 = 0$ . При этом адрес входа следующей ЭОЛЦ формируется схемой САМ. При достижении окончания микропрограммы формируется сигнал  $u_E$ , 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 ЭОЛЦ  $b_g \in C_1$ , если  $O_g$  не связан с конечной вершиной ГСА Г. Найдем разбиение  $\Pi_C = \{B_1, \dots, B_I\}$  множества  $C_1$  на классы псевдоэквивалентных ЭОЛЦ (ПЭОЛЦ). Выполним кодирование  $b_g \in C$  так, чтобы максимально возможное число классов  $B_i \in \Pi_C$ , где  $|\Pi_C| = I$ , представлялось одним обобщённым интервалом  $R_1$ -мерного булева пространства. Пусть  $n_i$  – число обобщённых 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:

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

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

Закодируем классы  $B_i \in \Pi_B$  двоичными кодами  $S(B_i)$  разрядности

$$R_3 = \lceil \log_2(|\Pi_B| + 1) \rceil. \quad 8)$$

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

$$Z = Z(\tau). \quad 9)$$

Очевидно, для реализации схемы БПК необходимы некоторые ре-

ресурсы кристалла FPGA. В настоящей работе предлагается уменьшить эти ресурсы за счет использования избыточности встроенных блоков памяти ЕМВ, 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

$$n_f = \left\lceil \frac{N+2}{t_F} \right\rceil \quad (10)$$

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

$$t_F = \left\lceil \frac{V}{2^R} \right\rceil. \quad (11)$$

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

$$t_O = \lceil n_f \cdot t_F \rceil. \quad (12)$$

Если выполняется условие

$$t_O - (N+3) \geq 1, \quad (13)$$

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

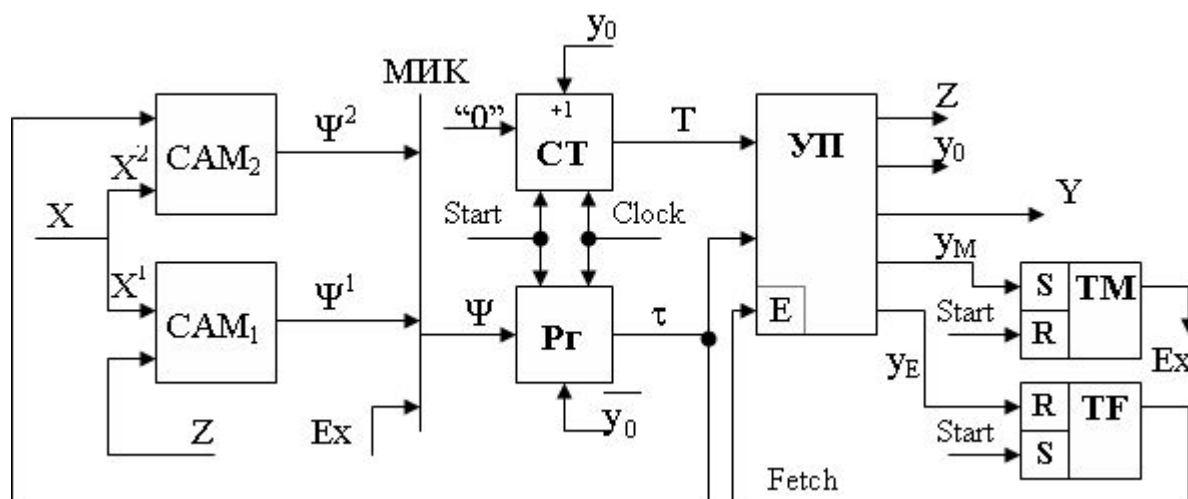


Рисунок. 2 - Структурная схема КМУУ  $U_2$

Эта модель имеет ряд отличий от модели  $U_1$ . Во-первых, блок САМ разделен на два блока. Блок САМ<sub>1</sub> реализует переходы, определяемые множеством П<sub>В</sub>, а блок САМ<sub>2</sub> – множеством П<sub>А</sub>. Мультиплексор МИК служит для выбора источника кодов, используя переменную Ех. Значение переменной Ех определяется состоянием триггера ТМ, управляемого дополнительной переменной у<sub>М</sub>. Блок УП является источником кодов клас-

сов для схемы САМ<sub>1</sub>. Источниками кодов для схемы САМ<sub>2</sub> является регистр R<sub>г</sub>. Предлагаемое КМУУ функционирует следующим образом.

По сигналу Start в R<sub>г</sub> и СТ заносятся нулевые коды (адрес первой микрокоманды), а триггера TF и TM устанавливаются соответственно в “1” (Fetch = 1) и “0” (Ex = 0). Пока адрес входа не достигнут, то U<sub>2</sub> функционирует как U<sub>1</sub>. При достижении адреса выхода ЭОЛЦ  $\bar{b}_g \in V_1$  может формироваться переменная  $y_M = 1$ . В этом случае Ex = 1 и адрес перехода формируется схемой САМ<sub>1</sub>. Для этого формируется система функций:

$$\Psi^1 = \Psi^1(Z, X^1). \quad (14)$$

Если  $V_i \in P_A$ , то переменная  $y_M$  не формируется. При этом Ex = 0 и адрес перехода определяется схемой САМ<sub>2</sub>. Для этого формируется система функций:

$$\Psi^2 = \Psi^2(\tau, X^2). \quad (15)$$

Соответствующие функции передаются на выход МИК и требуемый код  $K(\bar{b}_g)$  загружается в R<sub>г</sub>. Функционирование продолжается обычным образом до формирования переменной  $y_E$  (достижение окончания алгоритма управления).

Такой подход позволяет уменьшить число термов в системе (5) до абсолютного минимума. При выполнении условия

$$R_3 < R_1 \quad (16)$$

уменьшается число аргументов в системе (14) по сравнению с соответствующими функциями из системы (5). Недостатками данного подхода является наличие блока МИК, потребляющего некоторые ресурсы кристалла, и увеличение на  $R_3+1$  разрядности слов УП. Однако, схема МИК реализуется за счет использования тристабильных выходов макроячеек, поэтому дополнительных LUT-элементов не требуется. Блок УП реализуется на реконфигурируемых блоках памяти ЕМВ (embedded memory block) [5,6]. Эти блоки имеют строго определенное число выходов, входящих в множество {1, 2, 4, 8, 18, 36}. При этом имеется высокая вероятность наличие неиспользованных выходов блоков ЕМВ, входящих в состав УП устройства U<sub>1</sub>.

### **Особенности реализации схемы КМУУ U<sub>2</sub>**

В настоящей работе предлагается метод синтеза КМУУ U<sub>2</sub>, включающий следующие этапы:

1. Формирование для ГСА Г множеств С, С<sub>1</sub>, и П<sub>С</sub>.
2. Оптимальное кодирование ЭОЛЦ  $\bar{b}_g \in C_1$  и кодирование компонент ЭОЛЦ.
3. Формирование множеств P<sub>A</sub> и P<sub>B</sub>.

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 [6,7]. 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 ЭОЛЦ  $C = \{\alpha_1, \dots, \alpha_{16}\}$  и разбиение  $\Pi_C = \{V_1, \dots, V_7\}$ . При этом  $\alpha_{16} \notin C_1$ , а классы  $V_i \in \Pi_C$  определяются следующим образом:  $V_1 = \{\alpha_1\}$ ,  $V_2 = \{\alpha_2, \alpha_3, \alpha_4\}$ ,  $V_3 = \{\alpha_5, \alpha_6\}$ ,  $V_4 = \{\alpha_7, \alpha_8, \alpha_9\}$ ,  $V_5 = \{\alpha_{10}, \alpha_{11}\}$ ,  $V_6 = \{\alpha_{12}\}$ ,  $V_7 = \{\alpha_{13}, \alpha_{14}, \alpha_{15}\}$ . Итак,  $G = 16$ ,  $R_1 = 4$ ,  $\tau = \{\tau_1, \dots, \tau_4\}$ .

Оптимальное кодирование ЭОЛЦ  $\alpha_g \in C_1$  выполняется так, чтобы максимально возможное число классов  $V_i \in \Pi_C$  представлялось одним интервалом  $R_1$ -мерного булева пространства [1]. 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 ЭОЛЦ  $\alpha_g \in C$

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 необхо-

димо  $R_3 = 2$  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 [1]. Пусть, например, из ГСА  $\Gamma_1$  можно получить следующие формулы:

$$\begin{aligned} V_1 &\rightarrow x_1 b_3 \vee \overline{x_1 x_2} b_8 \vee \overline{\overline{x_1 x_2}} b_{12}; \\ V_2 &\rightarrow x_4 b_{12} \vee \overline{x_4} b_{17}. \end{aligned} \tag{17}$$

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  $\Pi_A$ ); 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 (13) приведены в табл. 1 и табл. 2.

Таблица 1. Таблица переходов для класса  $V_1 \in \Pi_A$

$V_i$	$K(V_i)$	$A(b_q)$	$X_h$	$\Psi_h^2$	$h$
$V_1$	0000	000100	$x_1$	$D_4$	1
		001000	$\overline{x_1 x_2}$	$D_3$	2
		010101	$\overline{x_1 x_2}$	$D_2 D_4$	3

Таблица 2. Таблица переходов для класса  $V_2 \in \Pi_B$

$V_i$	$C(V_i)$	$A(b_q)$	$X_h$	$\Psi_h^1$	$h$
$V_2$	00	010101	$x_4$	$D_2 D_4$	1
		110010	$\overline{x_4}$	$D_1 D_2$	2

Система (15) может быть получена из таблицы переходов для классов  $V_i \in \Pi_B$ . Так, из табл. 2 имеем, например  $D_1 = z_1 z_2 x_4$ . Система (14) может быть получена из таблицы переходов для классов  $V_i \in \Pi_A$ . Так, из табл. 1, имеем, например,  $D_2 = \overline{\tau_1 \tau_2 \tau_3 \tau_4} x_1 x_2$ .

Этап 7 выполняется практически по аналогии, например, с формированием содержимого УП для КМУУ с общей памятью [1]. Тем не менее, имеется одна особенность: в ячейки ЕМВ, соответствующие выходам ЭОЛЦ  $b_g \in V_i$  для блоков  $V_i \in \Pi_B$ , вводится код класса  $K(V_i)$ . Однако это дополнение производится элементарно и в настоящей работе не рассматривается.



## **Выводы**

Предлагаемый в работе метод оптимизации КМУУ с разделением кодов основан на мультиплексировании двух источников кодов классов псевдоэквивалентных ЭОЛЦ. Такой подход позволяет гарантированно уменьшить число термов в системе функций возбуждения триггеров регистра и счетчика адресов микрокоманд до максимально возможной величины. Если КМУУ с элементарными цепями рассматривать как автомат Мура, то предлагаемый подход позволяет уменьшить число термов до величины этого параметра у эквивалентного автомата Мили. Отметим, что применение метода возможно только при наличии свободных ресурсов встроенных блоков памяти 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 32% 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 ЭОЛЦ) и кристаллов FPGA (реконфигурируемость встроенных блоков памяти) для уменьшения числа LUT-элементов в схеме КМУУ с элементарными цепями.

**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. Barkalov A., Titarenko L. Logic synthesis for compositional microprogram control units. – Berlin: Springer, 2008. – 272 pp.
2. Палагин А.В., Баркалов А.А., Титаренко Л.А., Опанасенко В.Н. Проектирование реконфигурируемых цифровых систем. – Луганск: Издательство ВЛУ, 2011. – 432 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. Оптимизация схемы КМУУ с разделением кодов/ А.А. Баркалов, Л.А. Титаренко, К.Н. Ефименко // Наукові праці ДонНТУ. Серія „Інформатика, кібернетика та обчислювальна техніка” (КОТ-2011). Вип. 14 (188) – Донецьк: ДонНТУ. – 2011. – С.68-73.

*Надійшла до редакції 10.07.2012 р.*

*Рецензент:*

**О.О. Баркалов<sup>1</sup>, Л.О. Титаренко<sup>1</sup>, К.М. Єфіменко<sup>2</sup>, І.Я. Зеленьова<sup>2</sup>**  
Університет Зеленогурський (Польща)<sup>1</sup>,  
ДВНЗ «Донецький національний технічний університет» (Україна)<sup>2</sup>

**Оптимізація КМПК із елементарними ланцюгами.** В роботі запропоновано метод зменшення апаратних витрат у схемі КМПК із елементарними ланцюгами, який орієнтовано на технологію FPGA. Метод засновано на використанні двох джерел кодів класів псевдоеквівалентних ЕОЛЛ 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, GSA, ЕОЛЛ, FPGA, логічна схема, ЕМВ**

**A.A. Barkalov<sup>1</sup>, L.A. Titarenko<sup>1</sup>, K.N. Efimenko<sup>2</sup>, I.J. Zelenjova<sup>2</sup>**  
University of Zielona góra (Poland)<sup>1</sup>,  
Donetsk National Technical University (Ukraine)<sup>2</sup>

**Optimization of CMCU with elementary chains.** A method for reducing the hardware amount in the circuit of CMCU with elementary chains is proposed oriented on FPGA technology. The method is based on the use of two sources of codes classes of pseudoequivalent EOLC 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, EOLC, FPGA, logic circuit, EMB**