

УДК 004.3

А.А. Баркалов<sup>1</sup>, А.С. Лаврик<sup>2</sup>, А.Н. Мирошкин<sup>3</sup>  
<sup>1</sup>University of Zielona Gora, Zielona Gora, Poland,  
<sup>2</sup>ДонНТУ, Донецк, Украина<sup>3</sup>Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany  
<sup>1</sup>a.barkalov@iee.uz.zgora.pl, <sup>2</sup>alexandrlavrik@cs.donntu.edu.ua, <sup>3</sup>MiroshkinAN@gmail.com

## Применение кодирования совместимых микроопераций для уменьшения аппаратурных затрат в схеме устройства управления с разделением кодов

*В работе предложен метод уменьшения аппаратурных затрат, ориентированный на композиционные микропрограммные устройства управления с разделением кодов, реализованные на микросхемах типа CPLD. Метод основан на большом коэффициенте объединения по входу макроячеек ПМЛ, что позволяет использовать более одного источника для кодов классов ОЛЦ. Приведен пример реализации предложенного метода*

**Композиционное микропрограммное устройство управления, псевдоэквивалентная операторная линейная цепь, разделение кодов**

### Введение

В настоящее время для реализации цифровых схем устройств управления широко используются программируемые логические интегральные схемы (ПЛИС) типа CPLD (Complex Programmable Logic Devices) [6]. Такие ПЛИС включают в себя макроячейки с широкой входной шиной, которые могут выполнять логическую функцию от большого числа переменных (несколько десятков) [2,3]. Высокая стоимость этого базиса требует решения актуальной задачи уменьшения числа корпусов ПЛИС в схеме. При её решении следует учитывать не только специфику интерпретируемого алгоритма управления, но и особенности элементного базиса.

Если реализуемый алгоритм имеет линейный характер, для его представления целесообразно использовать модель композиционного микропрограммного устройства управления (КМУУ) с разделением кодов [4]. В таком случае, благодаря большой ширине входной шины макроячейки, коды классов псевдоэквивалентных операторных линейных цепей (ОЛЦ) могут быть представлены более чем одним источником. В данной работе предлагается метод уменьшения аппаратурных затрат в схеме КМУУ, основанный на использовании двух источников кодов, совместно с методом кодирования классов совместимых микроопераций [4].

Целью исследования является уменьшение аппаратурных затрат в комбинационной схеме КМУУ за счёт одновременного использования

нескольких источников кода классов псевдоэквивалентных ОЛЦ и кодирования классов совместимых микроопераций.

Задачей исследования является разработка метода синтеза, позволяющего уменьшить число макроячеек ПМЛ (программируемая матричная логика) и блоков постоянных запоминающих устройств (ППЗУ) в схеме КМУУ с разделением кодов.

### Особенности КМУУ с разделением кодов

Пусть ГСА  $\Gamma$  представлена множеством вершин  $V$  и дуг  $E$ . Пусть  $V = \{b_0, b_E\} \cup E_1 \cup E_2$ , где  $b_0$  – начальная вершина,  $b_E$  – конечная вершина,  $E_1$  – множество операторных вершин, где  $|E_1| = M$ ,  $E_2$  – множество условных вершин. Вершина  $b_q \in E_1$  содержит набор микроопераций (МО)  $Y(b_q) \subseteq Y$ , где  $Y = \{y_1, \dots, y_N\}$  – множество микроопераций [1]. Каждая вершина  $b_q \in E_2$  содержит элементы множества логических условий  $X = \{x_1, \dots, x_L\}$ .

Сформируем набор операторных линейных цепей  $C = \{\alpha_1, \dots, \alpha_G\}$  для ГСА  $\Gamma$ , где каждая ОЛЦ  $\alpha_g \in C$  – это последовательность операторных вершин, причём каждой паре её соседних вершин соответствует одна из дуг ГСА. Каждая ОЛЦ  $\alpha_g \in C$  имеет только один выход  $O_g$  и произвольное количество входов. Определения ОЛЦ, их входов выходов могут быть найдены в [4].

Назовём ГСА  $\Gamma$  линейной ГСА (ЛГСА),

если выполняется следующее условие:

$$\frac{M}{G} \geq 2. \tag{1}$$

Каждая вершина  $b_q \in E_1$  соответствует микрокоманде  $MI_q$ , хранимой в управляющей памяти (УП) КМУУ по адресу  $A_q$ . Для адресации микрокоманд достаточно

$$R = \lceil \log_2 M \rceil \tag{2}$$

бит. Пусть ОЛЦ  $\alpha_g \in C$  включает  $F_g$  компонент и пусть  $Q = \max(F_1, \dots, F_G)$ . Закодируем ОЛЦ  $\alpha_g \in C$  двоичными кодами  $K(\alpha_g)$  разрядности  $R_G$ , где

$$R_G = \lceil \log_2 G \rceil. \tag{3}$$

Закодируем каждый компонент ОЛЦ  $\alpha_g \in C$  двоичными кодами  $K(b_q)$  разрядности  $R_Q$ , где

$$R_Q = \lceil \log_2 Q \rceil. \tag{4}$$

Пусть компоненты закодированы таким образом, что

$$K(b_{gi+1}) = K(b_{gi}) + 1, \tag{5}$$

где  $i = 1, \dots, F_g - 1, g = 1, \dots, G$ . Пусть для кодирования компонент ОЛЦ используются переменные  $T_r \in T$ , а для кодирования ОЛЦ – переменные  $\tau_r \in \tau$ , где  $|\tau| = R_G, |T| = R_Q$ . В случае, если выполняется условие

$$R_G + R_Q = R, \tag{6}$$

ЛГСА Г может быть представлена моделью КМУУ с разделением кодов  $U_1$  (Рис. 1).

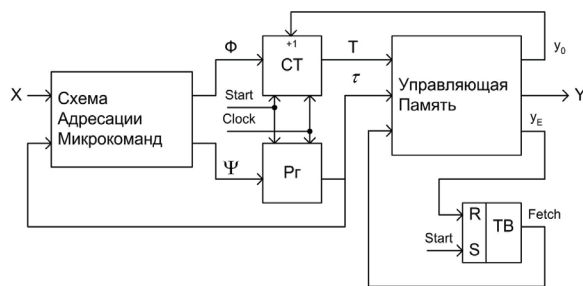


Рисунок 1 – Структурная схема КМУУ  $U_1$

В КМУУ  $U_1$ , схема адресации микрокоманд (САМ) реализует систему функций возбуждения счётчика СТ и регистра Рг:

$$\Phi = \Phi(\tau, X), \tag{7}$$

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

При этом, адрес микрокоманды  $MI_q$  представляется в виде:

$$A(b_q) = K(\alpha_g) * K(b_q), \tag{8}$$

где вершина  $b_q$  – это компонент ОЛЦ  $\alpha_g \in C$  и “\*” – знак операции конкатенации.

КМУУ  $U_1$  работает следующим образом. По сигналу  $Start = 1$  начальный адрес (все нули) заносится в Рг и СТ. В тоже время, триггер выборки ТБ устанавливается в единичное состояние. При этом  $Fetch = 1$ , что разрешает выборку микрокоманд из УП. Каждая ячейка УП содержит микрооперации  $y_n \in Y$  и специальные переменные  $y_0$  и  $y_E$ . Если  $y_0 = 1$ , то к текущему содержимому СТ прибавляется единица и адресуется следующий компонент текущей ОЛЦ. Иначе – значения СТ и Рг загружаются из САМ. Первый случай соответствует переходу из любого компонента ОЛЦ кроме выхода. Второй случай соответствует переходу из выхода ОЛЦ. Если  $y_E = 1$ , триггер выборки обнуляется, сигнал  $Fetch = 0$  и работа КМУУ прекращается. Это соответствует переходу из вершины  $b_q \in E_1$ , где  $\langle b_q, b_E \rangle \in E$ .

Отметим, что ОЛЦ  $\alpha_i, \alpha_j \in C$  являются псевдоэквивалентными ОЛЦ (ПОЛЦ) [4], если их выходы соединены с одной и той же вершиной ГСА Г. Аппаратурные затраты в логической схеме САМ могут быть уменьшены введением специального блока преобразователя кодов (ПК), преобразующего коды ОЛЦ в коды классов ПОЛЦ. В этом случае САМ реализует  $H_0$  термов, где  $H_0$  – число переходов эквивалентного автомата Мили, что является минимально возможным числом переходов [4]. Однако этот блок потребляет некоторые ресурсы ПЛИС или ППЗУ, из которых строится УП.

В данной работе предлагается метод синтеза КМУУ  $U_2$ , в котором используется два источника кодов классов ПОЛЦ, что позволяет снизить аппаратные затраты, потребляемые блоком ПК вплоть до нуля. Для уменьшения сложности блока УП предлагается использовать стратегию кодирования классов совместимых микроопераций [4].

### Основная идея предлагаемого метода

Отметим, что схемы блоков САМ, СТ, Рг и ТБ реализуются на макроячейках ПМЛ. Для реализации УП требуются внешние ППЗУ, имеющие  $t$  выходов, где  $t \in \{1, 2, 4, 8, 16\}$  [2, 3].

Пусть  $C_1 \subseteq C$  будет множеством ОЛЦ, где  $\alpha_g \in C_1$ , если её выход не соединён с конечной вершиной  $b_E$ . Пусть  $\Pi_C = \{B_1, \dots, B_l\}$  - разбиение множества  $C_1$  на классы ПОЛЦ. Закодируем ОЛЦ  $\alpha_g \in C_1$  таким образом, чтобы большинство

классов  $B_i \in \Pi_C$  было представлено одним интервалом  $R_G$ -мерного булева пространства. Для этого может быть использован широко известный алгоритм ESPRESSO [5]. Пусть  $\Pi_C = \Pi_A \cup \Pi_B$ , где  $B_i \in \Pi_A$ , если классу соответствует один интервал, иначе  $B_i \in \Pi_B$ .

Если выполняется условие

$$\Pi_B = \emptyset, \tag{9}$$

то блок ПК отсутствует и регистр  $R_g$  является источником всех кодов  $K(B_i)$  для классов  $B_i \in \Pi_C$ . В противном случае преобразованию подлежат только адреса выходов ОЛЦ, входящих в классы  $B_i \in \Pi_B$ . Для кодирования этих классов достаточно

$$R_B = \lceil \log_2(|\Pi_B| + 1) \rceil \tag{10}$$

Единица добавляется к  $|\Pi_B|$  для обозначения ситуации  $B_i \notin \Pi_B$ . Отметим, что часть кодов может быть реализована на ППЗУ.

Пусть для кодирования микроопераций используется стратегия кодирования кодов классов совместимых микроопераций, которая подробно описана в [1]. Микрооперации  $y_i, y_j \in Y$  называются совместимыми микрооперациями (СМО), если они никогда не встречаются вместе в наборах  $Y(b_q) \subseteq Y$ .

Разобьем множество  $Y$  на классы  $Y^1, \dots, Y^J$  совместимых микроопераций, что соответствует разбиению  $\Pi_Y = \{Y^1, \dots, Y^J\}$ . Пусть  $N_j = |Y^j|$ ,

закодируем каждую микрооперацию  $y_n \in Y^j$  двоичным кодом  $K(y_n)$  разрядности

$$R_j = \lceil \log_2(N_j + 1) \rceil \tag{11}$$

единица в выражение (11) добавляется для учёта ситуации, когда микрооперации  $y_n \in Y^j$  отсутствуют в наборе  $Y(b_q) \subseteq Y$ . Для реализации микроопераций  $y_n \in Y^j$  используется дешифратор  $DC_j$ . Совокупность дешифраторов  $DC_j (j=1, \dots, J)$  образует блок дешифрации микроопераций (БДМО). Суммарно этот блок имеет

$$R_Y = \sum_{j=1}^J R_j \tag{12}$$

выходов, для формирования которых используется блок шифрации микроопераций (БШМО). Этот блок реализуется на ППЗУ, имеющих  $t$  выходов.

Для реализации БШМО требуется

$$K_0 = \left\lceil \frac{R_Y + 2}{t} \right\rceil \tag{13}$$

микросхем с числом ячеек, не меньшим  $M$ . В выражении (13) учитываются две дополнительные переменные  $y_0$  и  $y_E$ , формируемые блоком БШМО.

Очевидно, что остаётся  $R_0$  свободных выходов микросхем ППЗУ, где

$$R_0 = K_0 t - R_Y - 2. \tag{14}$$

Если выполняется условие

$$R_0 \geq R_B, \tag{15}$$

то все классы  $B_i \in \Pi_B$  могут быть представлены с помощью УП и блок ПК отсутствует. Иначе может быть представлено лишь  $I_{CM}$  классов, где

$$I_{CM} = 2^{R_0}. \tag{16}$$

Оставшиеся классы  $B_i \in \Pi_B$  должны быть помещены в  $\Pi_A$  и представлены  $R_g$ .

Такой подход ведёт к КМУУ  $U_2$  (рис. 2), в котором функции  $z_r \in Z$ , используются для кодирования микроопераций, где  $|Z| = R_Y$ .

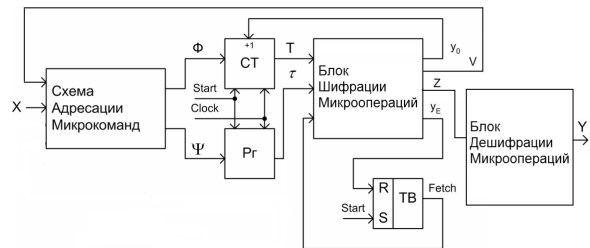


Рисунок 2 – Структурная схема КМУУ  $U_2$

В КМУУ  $U_2$ , блок САМ реализует функции:

$$\Phi = \Phi(\tau, V, X), \tag{17}$$

$$\Psi = \Psi(\tau, V, X),$$

блок БШМО реализует функции

$$\begin{aligned} V &= V(\tau, T); \\ y_0 &= y_0(\tau, T); \\ y_E &= y_E(\tau, T); \\ Z &= Z(\tau, T), \end{aligned} \tag{18}$$

а блок БДМО реализует функции

$$Y = Y(Z). \tag{19}$$

Переменные  $v_r \in V$  используются для кодирования классов  $B_i \in \Pi_B$ ,  $|V| = R_B$ . Переменные  $z_r \in Z$  используются для кодирования разрядов управления дешифраторами  $R_j$ ,  $|Z| = R_Y$ .

Принципы действия КМУУ  $U_1$  и  $U_2$  практически идентичны.

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

1. Формирование множеств  $C, C_1, \Pi_C$  для ГСА Г.
2. Присвоение макрокомандам адресов.
3. Формирование множеств  $\Pi_A, \Pi_B$ .
4. Кодирование классов  $B_i \in \Pi_B$ .
5. Кодирование микроопераций и формирование таблицы блока БДМО.
6. Формирование таблицы переходов КМУУ.
7. Формирование таблицы блока БШМО.
8. Синтез логической схемы КМУУ.

**Пример применения предложенного метода**

Пусть для некоторой ГСА  $\Gamma_1$  сформированы множества  $C = \{\alpha_1, \dots, \alpha_8\}$ ,  $C_1 = \{\alpha_1, \dots, \alpha_7\}$  и  $\Pi_C = \{B_1, \dots, B_4\}$ , где  $B_1 = \{\alpha_1\}$ ,  $B_2 = \{\alpha_2, \alpha_3, \alpha_4\}$ ,  $B_3 = \{\alpha_5, \alpha_6\}$ ,  $B_4 = \{\alpha_7\}$ , при этом  $\alpha_1 = \langle b_1, b_2, b_3 \rangle$ ,  $\alpha_2 = \langle b_4, b_5, b_6, b_7 \rangle$ ,  $\alpha_3 = \langle b_8, b_9, b_{10} \rangle$ ,  $\alpha_4 = \langle b_{11}, \dots, b_{14} \rangle$ ,  $\alpha_5 = \langle b_{15}, b_{16} \rangle$ ,  $\alpha_6 = \langle b_{17}, b_{18}, b_{19} \rangle$ ,  $\alpha_7 = \langle b_{20}, b_{21} \rangle$ ,  $\alpha_8 = \langle b_{22}, b_{23}, b_{24} \rangle$ . Итак,  $G = 8$ ,  $R_G = 3$ ,  $Q = 4$ ,  $R_Q = 2$ ,  $M = 24$ ,  $R = 5$ . Таким образом, условие (6) выполняется и метод разделения кодов может быть применён. Отметим, что условие (1) выполняется, следовательно, ГСА  $\Gamma_1$  является линейной.

Закодируем ОЛЦ  $\alpha_g \in C$  как показано на Рис. 3, используя переменные  $\tau_r \in \tau$ , где  $|\tau| = 3$ .

	$\tau_2 \tau_3$			
$\tau_1$	00	01	11	10
0	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$
1	$\alpha_5$	$\alpha_6$	$\alpha_7$	$\alpha_8$

Рисунок. 3 – Коды ОЛЦ для ГСА  $\Gamma_1$ .

Компоненты ОЛЦ  $\alpha_g \in C$  закодированы тривиально [4]: первый компонент имеет код 00, второй – 01 и так далее, для соответствия условию (5). Переменные  $T_1, T_2$  используются для кодирования компонентов.

Из Рис. 3 видно, что классы  $B_1, B_3$  и  $B_4$  представлены одним интервалом булева пространства. Эти интервалы представлены кодами  $K(B_i)$ :  $K(B_1) = 000$ ,  $K(B_3) = 10*$ ,  $K(B_4) = 111$ . Класс  $B_2 \in \Pi_C$  представлен интервалами  $00*$  and  $01*$ . Этот анализ показывает, что  $\Pi_A = \{B_1, B_3, B_4\}$  и  $\Pi_B = \{B_2\}$  ( $R_B = 1$ ). Переменная  $v_1 \in V$  используется для кодирования класса  $B_2$ . Пусть  $K(B_2) = 1$ , где  $v_1 = 0$  показывает ситуацию  $B_i \in \Pi_B$ .

Пусть для ГСА  $\Gamma_1$  найдено разбиение  $\Pi_Y = \{Y^1, \dots, Y^5\}$ , где  $Y^1 = \{y_1, y_2, y_3\}$ ,  $Y^2 = \{y_4, \dots, y_{10}\}$ ,  $Y^3 = \{y_{11}, \dots, y_{16}\}$ ,  $Y^4 = \{y_{17}, y_{18}, y_{19}\}$ ,  $Y^5 = \{y_{20}, y_{21}, y_{22}\}$ . При этом  $N_1 = N_4 = N_5 = 3$ ,  $N_2 = 7$ ,  $N_3 = 6$ ,  $R_1 = R_4 = R_5 = 2$ ,  $R_2 = R_3 = 3$ ,  $R_Y = 12$ . Пусть для реализации блока БШМО используются ППЗУ с  $t = 4$ , тогда из формул (13) и (14) можно найти  $R_0 = 2$ . Так как  $|\Pi_B| = 1$ , то  $R_B = 1$ , условие (15) выполняется и блок БПА отсутствует. Таким образом, модель КМУУ  $U_2$  может быть применена.

Для кодирования микроопераций  $y_n \in Y$  используются элементы множества  $Z = \{z_1, \dots, z_{12}\}$ . Закодируем микрооперации тривиальным образом: если в множестве  $Y^j$  индекс микрооперации  $y_n$  меньше индекса микрооперации  $y_m$ , то десятичный эквивалент кода  $K(y_n)$  меньше десятичного эквивалента кода  $K(y_m)$ . Например, для кодирования микроопераций  $y_n \in Y^2$  используются переменные  $z_3, z_4, z_5$  и  $K(y_4) = 000$ ,  $K(y_5) = 001, \dots, K(y_{10}) = 110$ . Таблица блока БДМО может рассматриваться как набор из  $J$  таблиц, где таблица  $j$  определяет  $DC_j$ . Эти таблицы строятся элементарно и в данной работе этот этап не рассматривается.

Пусть переходы для классов  $B_i \in \Pi_C$  задаются следующей системой обобщённых формул перехода (ОФП) [4]:

$$\begin{aligned}
 B_1 &\rightarrow x_1 b_4 \vee \overline{x_1} x_2 b_6 \vee \overline{x_1} x_2 x_3 b_8 \vee \overline{x_1} x_2 x_3 b_{14}; \\
 B_2 &\rightarrow x_2 b_{15} \vee x_2 x_3 b_{17} \vee x_2 x_3 x_4 b_{20} \vee x_2 x_3 x_4; \\
 B_3 &\rightarrow x_5 b_{19} \vee \overline{x_5} b_{11}; \\
 B_4 &\rightarrow x_1 b_{17} \vee \overline{x_1} b_{22};
 \end{aligned}
 \tag{20}$$

Эта система является основой для построения таблицы переходов, имеющей следующие

столбцы:  $B_i$ ,  $K_A(B_i)$ ,  $K_B(B_i)$ ,  $b_q$ ,  $A(b_q)$ ,  $X_h$ ,  $\Psi_h$ ,  $\Phi_h$ ,  $h$ . Здесь  $K_A(B_i)$  – это код  $K(B_i)$ , где  $B_i \in \Pi_A$ ;  $K_B(B_i)$  – код для  $B_i \in \Pi_B$ ;  $X_h$  – входной сигнал получаемый из ОФП;  $\Psi_h$  – набор выходных функций загружающих код  $K(\alpha_g)$  в Рг;  $\Phi_h$  – набор выходных функций загружающих код  $K(b_q)$  в СТ;  $h=1, \dots, H_0$  номер перехода. В этом случае  $\Psi = \{D_1, D_2, D_3\}$ ,  $\Phi = \{D_4, D_5\}$ . Определим несколько адресов микроопераций. Например, вершина  $b_{20}$  является первым компонентом ОЛЦ  $\alpha_7 \in C$ . Следовательно,  $A(b_{20})=11100$ . По аналогии мы можем найти  $A(b_{11})=01000$ ,  $A(b_{15})=10000$ ,  $A(b_{15})=10100$  и  $A(b_{19})=10110$ .

Пусть символ  $U_i(\Gamma_j)$  используется для обозначения интерпретации ГСА  $\Gamma_j$  с помощью модели КМУУ  $U_i(i=1,2)$ .

Часть таблицы переходов для классов  $B_2, B_3 \in \Pi_C$  показана в Табл. 1.

Таблица 1

Часть таблицы переходов для КМУУ  $U_2(\Gamma_1)$ 

$B_i$	$K_A(B_i)$	$K_B(B_i)$	$b_q$	$A(b_q)$	$X_h$	$\Psi_h$	$\Phi_h$	$h$
	$\tau_1 \tau_2 \tau_3$	$v_1$						
$B_2$	***	1	$b_{15}$	10000	$x_2$	$D_1$	-	1
			$b_{17}$	10100	$\overline{x_2 x_3}$	$D_1 D_3$	-	2
			$b_{20}$	11100	$\overline{x_2 x_3 x_4}$	$D_1 D_2 D_3$	-	3
$B_3$	10*	0	$b_{11}$	01000	$\overline{x_2 x_3 x_4}$	$D_2$	-	4
			$b_{19}$	10110	$x_5$	$D_1 D_3$	$D_4$	5
			$b_{11}$	01000	$\overline{x_5}$	$D_2$	-	6

Эта таблица используется для построения системы (17). Например, следующие выражения получены из Табл. 1:

$$D_2 = \overline{v_1 x_2 x_3} \vee \tau_1 \tau_2 v_1 x_5 \quad (\text{после минимизации});$$

$$D_4 = \tau_1 \tau_2 v_1 x_5.$$

Реализация логической схемы КМУУ  $U_2$  сводится к реализации систем (17) и (19) на макроячейках ПМЛ и системы (18) на ППЗУ. Эти задачи достаточно рассмотрены в литературе [6] и в нашей статье этот шаг не рассматривается.

### Заключение

Предложенный метод направлен на уменьшение числа макроячеек ПМЛ в схеме адресации микрокоманд, а также ёмкости используемого ППЗУ в КМУУ с разделением кодов. Это возможно благодаря таким главным особенностям как большая ширина входной шины современных макроячеек ПМЛ; избыточность выводов микросхем ППЗУ; существованию классов псевдоэквивалентных ОЛЦ, а также кодированию классов совместимых микроопераций.

Недостатком предлагаемого метода является увеличение времени такта КМУУ за счёт введения дополнительного блока дешифрации микроопераций. Таким образом, метод может быть использован, если он обеспечивает требуемое быстродействие цифровой системы. В противном случае необходимо использовать другие модели КМУУ [4]. Стоит отметить, что метод наиболее эффективен для интерпретации линейных ГСА.

Дальнейшие направления наших исследований связаны с применением предложенного подхода для реализации КМУУ на микросхемах FPGA [5,7].

### Список литературы

1. Baranov S. Logic Synthesis for Control Automata / S. Baranov. – Kluwer Academic Publishers, 1994. – 312 p.
2. Электронный ресурс. Xilinx CPLDs [http://www.xilinx.com/products/silicon\\_solutions/cplds/index.htm](http://www.xilinx.com/products/silicon_solutions/cplds/index.htm).
3. Электронный ресурс. Altera devices overview. – [http://www.altera.com/products/devices/common/devfamily\\_overview.html](http://www.altera.com/products/devices/common/devfamily_overview.html).
4. Barkalov A. Logic Synthesis for Compositional Microprogram Control Units / A. Barkalov, L. Titarenko. – Berlin: Springer, 2008. – 272 p.
5. Maxfield C. The Design Warrior's Guide to FPGAs / C. Maxfield. – Amsterdam: Elsevier, 2004. – 541 p.
6. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем / В.В. Соловьев. – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с.
7. Грушвицкий Р.И. Проектирование систем с использованием микросхем программируемой логики / Р.И. Грушвицкий, А.Х. Мурсаев, Е.П. Угрюмов. – С-Пб: БХВ. – Петербург, 2002. – 608 с.

Надійшла до редколегії 24.01.2011

**О.О. БАРКАЛОВ, О.С. ЛАВРИК,  
О.М. МИРОШКІН**

Зеленогурський університет, м. Зелена гора,  
Польща,  
Донецький національний технічний університет,  
Max Planck Institute for Dynamics of Complex  
Technical Systems, Magdeburg, Germany

**A. BARKALOV, A. LAVRIK, A. MIROSHKIN**

University of Zielona Gora, Poland,  
Donetsk National Technical University,  
Max Planck Institute for Dynamics of Complex  
Technical Systems, Magdeburg, Germany

**Застосування кодування сумісних операцій для  
зменшення апаратних витрат в схемі  
пристрою керування з розділенням кодів.**

**Application of Compatible Microoperation Subsets  
Encoding for Hardware Amount Reduction of  
Control Unit with Code Sharing**

В роботі запропоновано метод зменшення апаратних витрат, орієнтований на композиційні мікропрограмні пристрої керування з розділенням кодів, що реалізовані на мікросхемах типу CPLD. Метод базується на великому коефіцієнту об'єднання за входом макрокомірок ПМЛ, що дозволяє використовувати більше ніж одне джерело для кодів класів ОЛЛ. Наведено приклад використання запропонованого методу.

The method of hardware reduction is proposed which is oriented on compositional microprogram control units with code sharing and PAL-based CPLD chips. The method is based on a wide fan-in of PAL macrocells allowing using more than one source for codes of operational linear chains. An example of proposed method application is given.

*композиційний мікропрограмний пристрій  
керування, псевдоеквівалентний операторний  
лінійний ланцюг, розділення кодів*

*compositional microprogram control unit,  
pseudoequivalent operational linear chain, code  
sharing*