

А.А. Баркалов, Л.А. Титаренко, А.Н. Мирошкин

## Оптимизация устройства управления с разделением кодов

Предложен метод синтеза композиционных микропрограммных устройств управления с разделением кодов. Оптимизация схемы адресации микрокоманд достигается путем введения в формат микрокоманды поля с кодами классов псевдоэквивалентных операторных линейных цепей. Оптимизация емкости управляющей памяти достигается кодированием наборов микроопераций.

A method of synthesis is suggested for compositional microprogram control units with the code sharing. The optimization of the microinstruction addressing scheme is obtained by using a special field with the code of classes of pseudoequivalent operational linear chains. The optimization of the control memory is reached by the encoding of collections of microoperations.

Запропоновано метод синтезу композиційних мікропрограмних пристроїв керування з розділенням кодів. Оптимізація схеми адресації мікрокоманд досягається шляхом введення до формату мікрокоманди поля з кодами класів псевдоеквівалентних операторних лінійних ланцюгів. Оптимізація місткості керуючої пам'яті досягається кодуванням наборів мікрооперацій.

**Введение.** Одним из важных блоков цифровой системы является устройство управления (УУ) [1]. При реализации схем УУ актуальна задача уменьшения аппаратных затрат [2]. Для решения этой задачи необходимо учитывать особенности как алгоритма управления, так и используемого элементного базиса. В настоящей статье предложен метод решения этой задачи для случая интерпретации линейного алгоритма управления на программируемых логических интегральных схемах (ПЛИС). Используемые ПЛИС включают в себя макроячейки программируемой матричной логики (ПМЛ) и встроенные блоки памяти (ВБП) [3, 4]. В линейных алгоритмах управления доля операторных вершин превышает 75% от общего числа вершин [5]. Для интерпретации линейных алгоритмов целесообразно использовать композиционные микропрограммные устройства управления (КМУУ) [5]. В работах [6, 7] предложен метод оптимизации схемы КМУУ с разделением кодов, основанный на использовании классов псевдоэквивалентных операторных линейных цепей (ПОЛЦ). Однако при этом не уменьшаются аппаратные затраты в блоке формирования микроопераций. В данной статье описано развитие этого метода, основанное на кодировании наборов микроопераций [2].

Цель исследования – оптимизация комбинационной схемы КМУУ путем введения в формат

микрокоманд кодов классов ПОЛЦ и кодов наборов микроопераций. Задача исследования – разработка метода синтеза КМУУ, позволяющего уменьшить число макроячеек ПМЛ и блоков ВБП в его логической схеме. При этом алгоритм управления представляется в виде граф-схемы алгоритма [8, 9].

### Анализ композиционных устройств управления с разделением кодов

Обозначим в некоторой ГСА  $\Gamma$  множество вершин  $B = \{b_0, b_E\} \cup E_1 \cup E_2$  и множество дуг  $E$ . В этом множестве  $b_0$  – начальная вершина,  $b_E$  – конечная вершина,  $E_1$  – множество операторных вершин,  $E_2$  – множество условных вершин. Операторная вершина  $b_q \in E_1$  содержит набор микроопераций  $Y(b_q) \subseteq Y$ , где  $Y = \{y_1, \dots, y_N\}$  – множество микроопераций (выходных сигналов). Условная вершина  $b_g \in E_2$  содержит один элемент множества логических условий (входных сигналов)  $X = \{x_1, \dots, x_L\}$ .

Операторная линейная цепь (ОЛЦ) является последовательностью операторных вершин граф-схемы алгоритма. Сформируем множество ОЛЦ  $C = \{\alpha_1, \dots, \alpha_G\}$ , где  $G = |C|$  обозначает количество ОЛЦ в данной ГСА. При этом каждая пара соседних вершин ОЛЦ  $\alpha_g \in C$  соответствует некоторой дуге из множества  $E$  [5]. Каждая ОЛЦ  $\alpha_g \in C$  имеет произвольное число входов и только один выход  $O_g$ . Формальные оп-

ределения ОЛЦ, их входов и выходов можно найти в [5]. Отметим, что каждая вершина  $b_q \in E_1$  соответствует микрокоманде  $MI_q$ , хранимой в управляющей памяти (УП) по адресу  $A_q$ . Для адресации микрокоманд достаточно

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

бит, где  $M = |E_1|$ . Пусть каждая ОЛЦ  $\alpha_g \in C$  содержит  $F_g$  компонент и пусть  $Q = \max(F_1, \dots, F_G)$ . Закодируем каждую ОЛЦ  $\alpha_g \in C$  двоичным кодом  $K(\alpha_g)$  разрядности

$$R_1 = \lceil \log_2 G \rceil, \quad (2)$$

а каждую компоненту  $b_q \in E_1$  – двоичным кодом  $K(b_q)$  разрядности

$$R_2 = \lceil \log_2 Q \rceil. \quad (3)$$

Используем для кодирования ОЛЦ элементы  $\tau_r \in \tau$ , а для кодирования компонент – элементы  $T_r \in T$ , где  $|\tau| = R_1$ ,  $|T| = R_2$ . Кодирование компонент выполняется в естественном порядке, т.е.

$$K(b_{gi}) = K(b_{gi-1}) + 1, \quad (4)$$

где  $g = 1, \dots, G$ ,  $i = 1, \dots, F_g$ . Пусть для ГСА  $\Gamma$  выполняется условие

$$R_1 + R_2 = R. \quad (5)$$

В этом случае для ее интерпретации может быть использована модель КМУУ с разделением кодов (рис. 1), обозначаемая в дальнейшем символом  $U_1$ .

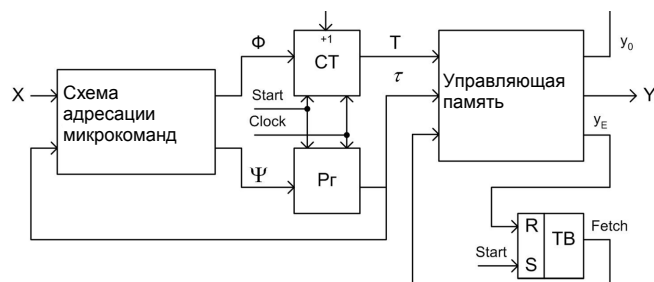


Рис. 1. Структурная схема КМУУ  $U_1$

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

$$\begin{aligned} \Phi &= \Phi(\tau, X), \\ \Psi &= \Psi(\tau, X). \end{aligned} \quad (6)$$

При этом адрес микрокоманды  $MI_q$  представляется в виде

$$A(b_q) = K(\alpha_g) * K(b_q), \quad (7)$$

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

По сигналу *Start* в Rr и СТ заносится начальный адрес микропрограммы, а триггер выборки ТВ устанавливается в единичное состояние. При этом сигнал *Fetch* = 1, что разрешает выборку микрокоманд из 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 термов в схеме САМ может быть уменьшено путем введения преобразователя кодов ОЛЦ в коды классов псевдоэквивалентных ОЛЦ [5]. Операторные линейные цепи  $\alpha_i$ ,  $\alpha_j \in C$  называются псевдоэквивалентными, если их выходы связаны со входом одной и той же вершины ГСА  $\Gamma$ . Однако реализация такого преобразователя требует дополнительных макроячеек ПМЛ.

В статье предлагается упростить преобразователь кодов, для чего используются свободные ресурсы встроенных блоков памяти. Для уменьшения числа ВБП используется максимальное кодирование наборов микроопераций.

### Основная идея предлагаемого метода

Пусть  $\alpha_g \in C_1$ , если  $\alpha_g \in C$  и ее выход не связан с входом конечной вершины ГСА  $\Gamma$ . Найдем разбиение  $\Pi_C = \{B_1, \dots, B_I\}$  множества  $C_1$  на классы ПОЛЦ. Закодируем классы  $B_i \in \Pi_C$  двоичными кодами  $K(B_i)$  разрядности

$$R_B = \lceil \log_2 I \rceil. \quad (8)$$

Используем для кодирования классов  $B_i \in \Pi_C$  переменные  $v_r \in V$ , где  $|V| = R_B$ .

При синтезе КМУУ исходная ГСА  $\Gamma$  преобразуется путем введения в ее операторные вер-

шины дополнительных переменных  $y_0$  и  $y_E$ . Таким образом, исходное множество  $Y$  преобразуется в множество  $Y_C = Y \cup \{y_0, y_E\}$ . Пусть множество  $Y_C$  включает в себя  $Q_1$  различных наборов микроопераций и дополнительных переменных. Поставим в соответствие каждому набору  $Y_q$  двоичный код  $K(Y_q)$  разрядности

$$R_Y = \lceil \log_2 Q_1 \rceil. \quad (9)$$

Используем для кодирования наборов  $Y_q \subseteq Y_C$  переменные  $Z_r \in Z$ , где  $|Z| = R_Y$ . В этом случае управляющая память состоит из двух блоков [5]: блок микропамяти (БМП), формирующий функции

$$Z = Z(T, \tau), \quad (10)$$

и блок микроопераций (БМО), формирующий переменные

$$Y_C = Y_C(Z). \quad (11)$$

Оба блока – БМО и БМП – реализуются на встроенных блоках памяти, имеющих  $t$  выходов. Примем, что емкости  $q$  одного БВП достаточно для хранения всех слов БМП и БМО:

$$q \geq \max(M, Q_1). \quad (12)$$

В статье используются микрокоманды, формат которых включает в себя поле  $K(B_i)$  и поле  $K(Y_q)$ . Разрядность  $R_I$  таких микрокоманд определяется как

$$R_I = R_B + R_Y. \quad (13)$$

Блок БМП имеет  $R_Y$  выходов и для его реализации требуется

$$n_1 = \left\lceil \frac{R_Y}{t} \right\rceil \quad (14)$$

блоков БВП. При этом  $R_3$  бит слова БМП могут быть свободными, где

$$R_3 = n_1 t - R_Y. \quad (15)$$

Эти биты могут использоваться для хранения части  $V^1$  кода  $K(B_i)$ .

При выполнении условия

$$R_3 \geq R_B \quad (16)$$

все разряды кода  $K(B_i)$  реализуются блоком микропамяти. В противном случае

$$R_4 = R_B - R_3 \quad (17)$$

разрядов кода  $K(B_i)$  формируются блоком преобразователя кодов (БПК). Эти разряды образуют часть  $V^2$  кода класса  $B_i \in \Pi_C$ . Предложенный подход ведет к КМУУ  $U_2$  (рис. 2).

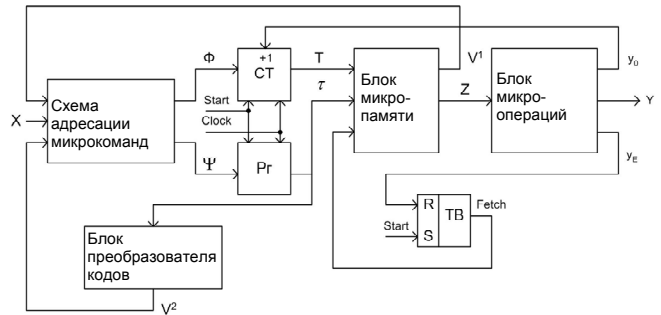


Рис. 2. Структурная схема КМУУ  $U_2$

В КМУУ  $U_2$  блок САМ реализует функции

$$\Phi = \Phi(V, X), \quad (18)$$

$$\Psi = \Psi(V, X), \quad (19)$$

а блок БПК реализует функции

$$V^2 = V^2(\tau). \quad (20)$$

При этом, естественно, выполняются условия

$$V^1 \cup V^2 = V, \quad (21)$$

$$V^1 \cap V^2 = \emptyset. \quad (22)$$

Назначение остальных блоков уже рассмотрено. Отметим, что схемы БАМ, СТ, Рг, ТВ реализуются с использованием макроячеек ПМЛ, схемы БМП и БМО – с использованием встроенных блоков памяти, а схема БПК может быть реализована в любом из этих базисов. Предлагаемый метод синтеза состоит из следующих этапов:

- Формирование множеств  $C$ ,  $C_1$  и  $\Pi_C$  для ГСА Г.
- Кодирование ОЛЦ, их компонент и классов  $B_i \in \Pi_C$ .
- Кодирование наборов микроопераций  $Y_q \subseteq Y_C$ .
- Формирование содержимого блоков управляющей памяти.
- Формирование таблицы переходов КМУУ.
- Формирование таблицы блока преобразователя кодов.
- Синтез логической схемы КМУУ.

### Пример применения предложенного метода

Пусть ГСА  $\Gamma_1$  характеризуется множествами  $C = \{\alpha_1, \dots, \alpha_8\}$ , где  $\alpha_8 \notin C_1$ , и  $\Pi_C = \{B_1, \dots, B_5\}$ , где  $B_1 = \{\alpha_1\}$ ,  $B_2 = \{\alpha_2, \alpha_3\}$ ,  $B_3 = \{\alpha_4, \alpha_5\}$ ,  $B_4 = \{\alpha_6\}$ ,  $B_5 = \{\alpha_7\}$ ,  $\alpha_1 = \langle b_1, b_2, b_3 \rangle$ ,  $\alpha_2 = \langle b_4, \dots, b_7 \rangle$ ,  $\alpha_3 = \langle b_8, b_9 \rangle$ ,  $\alpha_4 = \langle b_{10}, b_{11}, b_{12} \rangle$ ,  $\alpha_5 = \langle b_{13}, \dots, b_{16} \rangle$ ,  $\alpha_6 = \langle b_{17}, \dots, b_{19} \rangle$ ,  $\alpha_7 = \langle b_{20}, b_{21} \rangle$ ,  $\alpha_8 = \langle b_{22}, b_{23}, b_{24} \rangle$ . Следовательно, количество ОЛЦ  $G = 8$ , для их кодирования необходимо  $R_1 = 3$  переменных из множества  $\tau = \{\tau_1, \tau_2, \tau_3\}$ , максимальная длина ОЛЦ  $Q = 4$  вершины, для их кодирования достаточно  $R_2 = 2$  переменных из множества  $T = \{T_1, T_2\}$ , количество операторных вершин в ГСА  $M = 24$ , для их кодирования достаточно  $R = 5$  разрядов. Следовательно, условие (5) выполняется и применение метода разделения кодов имеет смысл. Для кодирования классов  $B_i \in \Pi_C$  достаточно  $R_B = 3$  переменные, т.е.  $V = \{v_1, v_2, v_3\}$ .

Закодируем ОЛЦ  $\alpha_g \in C$  и их классы произвольным образом:  $K(\alpha_1) = 000, \dots, K(\alpha_8) = 111$ ,  $K(B_1) = 000, \dots, K(B_5) = 100$ . Для выполнения условия (4) присвоим первой компоненте каждой ОЛЦ  $\alpha_g \in C$  код 00, второй – 01, третьей – 10, а четвертой – 11. Это позволяет определить адреса  $A(b_q)$  микрокоманд КМУУ  $U_2(\Gamma_1)$ , показанные в табл. 1. При этом столбцы соответствуют переменным  $\tau_1, \tau_2, \tau_3$ , а строки – переменным  $T_1, T_2$ .

Здесь и далее символ  $U_i(\Gamma_j)$  означает, что КМУУ  $U_i$  интерпретирует ГСА  $\Gamma_j$ .

Т а б л и ц а 1. Адреса микрокоманд КМУУ  $U_2(\Gamma_1)$

Адрес	000	001	010	011	100	101	110	111
00	$b_1$	$b_4$	$b_8$	$b_{10}$	$b_{13}$	$b_{17}$	$b_{20}$	$b_{22}$
01	$b_2$	$b_5$	$b_9$	$b_{11}$	$b_{14}$	$b_{18}$	$b_{21}$	$b_{23}$
10	$b_3$	$b_6$	*	$b_{12}$	$b_{15}$	$b_{19}$	*	$b_{24}$
11	*	$b_7$	*	*	$b_{16}$	*	*	*

Из этой таблицы получаем адреса микрокоманд. Например,  $A(b_5) = 00101$ ,  $A(b_{19}) = 10010$  и т.д. Заменяя вер-

шины наборами микроопераций, получаем содержимое УП (табл. 2).

Наборы микроопераций взяты из операторных вершин ГСА  $\Gamma_1$ , которая у нас не показана. Из табл. 2 следует, что управляющая память содержит  $Q_1 = 8$  наборов микроопераций:  $Y_1 = \{y_0, y_1, y_2\}$ ,  $Y_2 = \{y_0, y_3, y_9\}$ ,  $Y_3 = \{y_4\}$ ,  $Y_4 = \{y_0, y_3, y_5\}$ ,  $Y_5 = \{y_0, y_3, y_6\}$ ,  $Y_6 = \{y_1, y_7\}$ ,  $Y_7 = \{y_8\}$ ,  $Y_8 = \{y_1, y_2, y_E\}$ . Для их кодирования достаточно  $R_Y = 3$  переменных  $Z = \{z_1, z_2, z_3\}$ . Закодируем наборы  $Y_q \subseteq Y_C$  произвольно:  $K(Y_1) = 000, \dots, K(Y_8) = 111$ . Пусть ВБП имеет  $t = 2$  выхода, тогда количество использованных ВБП  $n_1 = 2$ . Количество неиспользованных разрядов  $R_3 = 1$ , следовательно, один разряд кода  $K(B_i)$  может быть реализован блоком БМП. Пусть  $V^1 = \{v_1\}$ .

Для задания блока БМП достаточно заменить наборы микроопераций в табл. 1 их кодами. При этом в выход каждой ОЛЦ  $\alpha_g \in B_i$  добавляется значение первого разряда кода класса  $B_i \in \Pi_C$ . В нашем примере блок БМП задан табл. 3, где  $v_1$  включен в выход ОЛЦ  $\alpha_7$ .

Для формирования содержимого блока БМО необходимо построить таблицу со столбцами  $K(Y_q), Y_q, q$ . Эта таблица строится тривиально (табл. 4).

Для формирования таблицы переходов КМУУ  $U_2$  необходимо построить систему обобщенных формул перехода [4] для классов  $B_i \in \Pi_C$ . Пусть для рассматриваемого примера эта система имеет следующий вид:

$$\begin{aligned}
 B_1 &\rightarrow x_1 b_4 \vee \overline{x_1} b_8; \\
 B_2 &\rightarrow x_3 b_{10} \vee \overline{x_3} x_4 b_{13} \vee \overline{\overline{x_3} x_4} b_{17}; \\
 B_3 &\rightarrow x_2 b_{17} \vee \overline{x_2} x_3 b_{20} \vee \overline{\overline{x_2} x_3} b_{18}; \\
 B_4 &\rightarrow b_{20}; B_5 \rightarrow x_1 b_{22} \vee \overline{x_1} b_{11}.
 \end{aligned} \tag{23}$$

Т а б л и ц а 2. Содержимое управляющей памяти КМУУ  $U_2(\Gamma_1)$

Адрес	000	001	010	011	100	101	110	111
00	$Y_0, Y_1, Y_2$	$Y_0, Y_3, Y_5$	$Y_0, Y_1, Y_2$	$Y_0, Y_3, Y_6$	$Y_0, Y_3, Y_5$	$Y_0, Y_1, Y_2$	$Y_0, Y_3, Y_6$	$Y_0, Y_3, Y_9$
01	$Y_0, Y_3, Y_9$	$Y_0, Y_3, Y_9$	$Y_1, Y_7$	$Y_0, Y_3, Y_9$	$Y_0, Y_3, Y_9$	$Y_0, Y_3, Y_5$	$Y_8$	$Y_0, Y_3, Y_6$
10	$Y_4$	$Y_0, Y_3, Y_6$	*	$Y_8$	$Y_0, Y_1, Y_2$	$Y_4$	*	$Y_1, Y_2, Y_E$
11	*	$Y_8$	*	*	$Y_1, Y_7$	*	*	*

**Таблица 3.** Содержимое блока микропамяти КМУУ  $U_2(\Gamma_1)$

Адрес	000	001	010	011	100	101	110	111
00	000	001	000	011	100	000	100	001
01	001	011	101	—	100	001	011	110 $v_1$
10	010	—	100	*	110	—	000	010
11	*	110	—	*	*	101	—	*

**Таблица 4.** Содержимое блока микроопераций КМУУ  $U_2(\Gamma_1)$

$K(Y_q)$	$Y_q$	$q$	$K(Y_q)$	$Y_q$	$q$
000	$y_0, y_1, y_2$	1	100	$y_0, y_3, y_6$	5
001	$y_0, y_3, y_9$	2	101	$y_1, y_7$	6
010	$y_4$	3	110	$y_8$	7
011	$y_0, y_3, y_5$	4	111	$y_1, y_2, y_E$	8

Система вида (23) – основа для формирования таблицы переходов КМУУ  $U_2$  со столбцами:  $B_i, K(B_i), b_q, A(b_q), X_h, \Phi_h, \Psi_h, h$ . Назначение этих столбцов видно из табл. 5, а число ее строк  $H$  определяется числом термов в системе (23).

**Таблица 5.** Фрагмент таблицы переходов КМУУ  $U_2(\Gamma_1)$

$B_i$	$K(B_i)$	$b_q$	$A(b_q)$	$X_h$	$\Phi_h$	$\Psi_h$	$h$
$B_3$	010	$b_{17}$	10100	$x_2$	—	$D_1 D_3$	6
		$b_{20}$	11000	$\bar{x}_2 x_3$	—	$D_1 D_2$	7
		$b_{18}$	10101	$\bar{x}_2 \bar{x}_3$	$D_5$	$D_1 D_3$	8

Этот фрагмент задает переходы из класса  $B_3$ , которые начинаются с термина №6 системы (23). Таблица переходов служит для формирования систем (18), (19), термы  $F_h$  которой определяются как

$$F_h = \left( \bigwedge_{r=1}^{R_B} v_r^{l_{rh}} \right) \cdot X_h \quad (h=1, \dots, H). \quad (24)$$

В системе (24)  $l_{rh} \in \{0,1\}$  – значение  $r$ -го разряда кода  $K(B_i)$  из  $h$ -й строки таблицы,  $v_r^0 = \bar{v}_r, v_r^1 = v_r$ . Например, из табл. 5 имеем  $D_1 = F_6 \vee F_7 \vee F_8 = \bar{v}_1 v_2 v_3$ ;  $D_2 = F_6 \vee F_8 = \bar{v}_1 v_2 v_3 x_2 \vee \bar{v}_1 v_2 v_3 x_2 x_3$ ;  $D_3 = F_8 = \bar{v}_1 v_2 v_3 x_2 x_3$ .

**Таблица 6.** Таблица блока преобразователя кодов (БПК) для КМУУ  $U_2(\Gamma_1)$

$\alpha_g$	$K(\alpha_g)$	$B_i$	$K(B_i)$	$V_g^2$	$g$	$\alpha_g$	$K(\alpha_g)$	$B_i$	$K(B_i)$	$V_g^2$	$g$
$\alpha_1$	000	$B_1$	000	—	1	$\alpha_5$	100	$B_3$	010	$v_2$	5
$\alpha_2$	001	$B_2$	001	$v_3$	2	$\alpha_6$	101	$B_4$	011	$v_2 v_3$	6
$\alpha_3$	010	$B_2$	001	$v_3$	3	$\alpha_7$	110	$B_5$	100	—	7
$\alpha_4$	011	$B_3$	010	$v_2$	4	$\alpha_8$	111	$B_6$	—	—	8

Таблица БПК имеет столбцы  $\alpha_g, K(\alpha_g), B_i, K(B_i), V_g^2, g$ . Для нашего примера блок преобразователя кодов задан табл. 6.

Напомним, что переменная  $v_1$  формируется блоком БМП, а блок  $B_6$  не кодируется, так как  $\alpha_8 \notin C_1$ . Очевидно, что подобная таблица задает содержимое блока БВП. Если блок БПК реализуется на макроячейках ПМЛ, то табл. 6 соответствует картам Карно для функций  $v_r \in V^2$ . Для оптимизации системы (20) в этом случае необходимо оптимально закодировать ОЛЦ  $\alpha_8 \in C_1$ , для чего может быть, например, использован алгоритм *ESPRESSO* [1]. В данной статье эта задача не рассматривается.

Реализация логической схемы КМУУ  $U_2$  сводится к реализации систем (18), (19) на ПМЛ и таблиц вида 3, 4 и 6 на блоках БВП. Для этой цели могут быть использованы стандартные пакеты [4] или известные методы [8]. Этот этап также здесь не рассматривается. Отметим, что в КМУУ  $U_1(\Gamma_1)$  управляющая память содержит  $32 \cdot 12 = 384$  бита (если  $t = 2$ ), а таблица переходов состоит из 17 строк. В КМУУ  $U_2(\Gamma_1)$  микропамять содержит  $34 \cdot 4 = 144$  бита, блок микроопераций требует  $8 \cdot 2 = 16$  бит. Таким образом, в КМУУ  $U_2(\Gamma_1)$  используется 256 бит памяти, а таблица переходов состоит из 11 строк. Следовательно, КМУУ  $U_2(\Gamma_1)$  требует в 1,5 раза меньше памяти, а его блок САМ реализует в 1,54 раза меньше термов.

**Заключение.** Предлагаемый метод включения в формат микрокоманды поля с кодом класса псевдоэквивалентных ОЛЦ позволяет уменьшить число макроячеек ПМЛ в схеме формирования адреса микрокоманд. Использование метода кодирования наборов микроопераций и дополнительных переменных позволяет уменьшить требуемую емкость управляющей памяти. Использование преобразователя кодов ОЛЦ в

коды классов ПОЛЦ гарантирует сокращение числа переходов, при этом блок БПК может отсутствовать, если выполняется условие (16). Тогда преобразование осуществляется блоком микропамяти.

Однако быстродействие КМУУ  $U_2$  снижается в сравнении с этой характеристикой базового КМУУ с разделением кодов. Отметим, что сокращение числа макроячеек в блоке САМ может привести к сокращению числа уровней в его схеме. При этом эффект увеличения времени цикла при увеличении числа уровней памяти может скомпенсироваться. Рассмотренные примеры показали, что число макроячеек ПМЛ в КМУУ  $U_2$  уменьшается до 30 процентов и требуемые ресурсы памяти – до 50 процентов в сравнении с КМУУ  $U_1$ . При этом число уровней уменьшается на два–три. Напомним, что применение этого метода целесообразно только при выполнении условия (5) для линейных ГСА.

Научная новизна предложенного метода состоит в использовании классов псевдоэквивалентных ОЛЦ и свободных ресурсов встроенных блоков памяти для уменьшения числа макроячеек ПМЛ в схеме адресации микрокоманд с одновременным уменьшением требуемых ресурсов памяти при кодировании наборов микроопераций. Практическая значимость метода состоит в сокращении числа микросхем при реализации схемы КМУУ, что позволяет получить схемы меньшей стоимости, чем известные аналоги.

Дальнейшее направление исследований связано с разработкой САПР для синтеза схем КМУУ, а также с проверкой применимости данного метода для базиса программируемых вентиляльных матриц *FPGA (field-programmable gate arrays)* [10].

1. *De Micheli G.* Synthesis and Optimization of Digital Circuits. – NY: McGraw-Hill, 1994. – 636 p.
2. *Баркалов А.А.* Синтез устройств управления на программируемых логических устройствах. – Донецк: ДНТУ, 2002. – 262 с.
3. *Macrocell Configurations in CoolRunner XPLA3 CPLDs.* – [http://www.xilinx.com/support/documentation/application\\_notes/xapp335.pdf](http://www.xilinx.com/support/documentation/application_notes/xapp335.pdf)
4. *Грушвицкий П.И., Мурсаев А.Х., Узрюмов Е.П.* Проектирование систем с использованием микросхем программируемой логики. – СПб: БХВ-Петербург, 2002. – 608 с.
5. *Barkalov A., Titarenko L.* Logic Synthesis for Compositional Microprogram Control Units – Berlin: Springer, 2008. – 272 p.
6. *Synthesis of control unit with code sharing and modified linear chains.* Машиностроение и техносфера XXI века / А.А. Barkalov, S.A. Kovalyov, J. Bieganski et al. // Сб. тр. XV междунар. науч.-техн. конф. в г. Севастополе 15–20 сент. 2008 г. В 4-х т. Т. 4. – Донецк: ДонНТУ, 2008. – С. 54–59.
7. *Баркалов А.А., Красичков А.А., Мирошкин А.Н.* Синтез устройства управления с разделением кодов и модификацией операторных линейных цепей // Наук. пр. Донецьк. нац. техн. ун-ту. Серія «Інформатика, кібернетика і обчислювальна техніка». Вип. 9 (132) – Донецьк: ДонНТУ. – 2008. – С. 183–187.
8. *Соловьев В.В.* Проектирование цифровых схем на основе программируемых логических интегральных схем. – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с.
9. *Baranov S.* Logic Synthesis for Control Automata – Boston: Kluwer Acad. Publ., 1994 – 312 p.
10. *Maxfield C.* The Design Warrior's Guide for FPGA – Amsterdam: Elsevier, 2004. – 541 p.

Поступила 31.03.2009

Тел. для справок: 8(0622) 301-0723 (Донецк)

E-mail: [A.Barkalov@iie.uz.zgora.pl](mailto:A.Barkalov@iie.uz.zgora.pl), [MiroshkinAN@gmail.com](mailto:MiroshkinAN@gmail.com)

© А.А. Баркалов, Л.А. Титаренко, А.Н. Мирошкин, 2010