

**О.О. БАРКАЛОВ**, доктор техн. наук, професор,  
Зеленогурський Університет (Польща),  
65246, Зелена Гура, вул. Підгірна, 50, Польща,  
[a.barkalov@iie.uz.zgora.pl](mailto:a.barkalov@iie.uz.zgora.pl)

**Л.О. ТИТАРЕНКО**, доктор техн. наук, професор,  
Зеленогурський Університет (Польща),  
65246, Зелена Гура, вул. Підгірна, 50,  
Польща; Харківський нац. ун-т радіоелектроніки,  
61000, м. Харків, просп. Науки, 14,  
[L.Titarenko@iie.uz.zgora.pl](mailto:L.Titarenko@iie.uz.zgora.pl)

**О.М. ГОЛОВІН**, канд. техн. наук, ст. наук. співробітник,  
Ін-т кібернетики ім. В.М. Глушкова НАН України,  
03187, м. Київ, просп. Академіка Глушкова, 40, Україна,  
[yaviz@ukr.net](mailto:yaviz@ukr.net)

**О.В. МАТВІЄНКО**, наук. співробітник,  
Ін-т кібернетики ім. В.М. Глушкова НАН України,  
03187, м. Київ, просп. Академіка Глушкова, 40, Україна,  
[yaviz@ukr.net](mailto:yaviz@ukr.net)

## ОПТИМІЗАЦІЯ КОМПОЗИЦІЙНОГО МІКРОПРОГРАМНОГО ПРИБОРУ УПРАВЛІННЯ З ЕЛЕМЕНТАРНИМИ ЛАНЦЮГАМИ

*Запропоновано метод зменшення апаратних витрат у схемі композиційного мікропрограмного пристрою управління (КМПУ), що реалізується в базисі FPGA. Метод заснований на перетворенні адрес мікрокоманд на коди виходів елементарних лінійних операторних ланцюгів (ЕЛОЛ). Для оптимізації схеми КМПУ множини ЕЛОЛ розбивається на класи. Розбиття здійснюється так, що блок адресації мікрокоманд має точно два рівня елементів табличного типу. Пам'ять управління КМПУ реалізується на вбудованих блоках пам'яті. У роботі розглянуто приклад синтезу схеми КМПУ і виконано аналіз запропонованого методу.*

**Ключові слова:** композиційний мікропрограмний пристрій управління, мікрокоманда, LUT, ЕМВ, синтез.

### Вступ

Одним із центральних блоків цифрових систем є пристрій управління [1]. Пристрій управління — це послідовна схема [2], яка координує взаємодію інших блоків системи. Як правило, при синтезі схем пристроїв управління виникає необхідність зменшення витрат апаратури [3]. Методи вирішення цього зав-

дання залежать від особливостей архітектури пристрою управління й елементного базису. У цій статті розглядається задача зменшення витрат апаратури в схемі композиційного мікропрограмного пристрою управління (КМПУ) [4]. Як елементний базис обрано мікросхеми типу *FPGA* (*field-programmable logic array*) [5, 6], широко використовувані для реалізації сучасних цифрових систем [7, 8].

КМПУ є автоматом Мура, до складу якого входить пам'ять управління (ПУ) з мікропрограмою, яка відповідає початковому алгоритму управління [4]. У КМПУ використовується природна адресація мікрокоманд [4], що веде до необхідності використання лічильника адреси мікрокоманд (ЛА). Мікрокоманди (МК) містять тільки мікрооперації (МО) та додаткові сигнали управління [4]. Система функцій збудження пам'яті формується схемою адресації мікрокоманд.

При використанні базису *FPGA* [7, 8] схема адресації МК реалізується на елементах *LUT* (*look-up table*), а ПУ — на вбудованих блоках пам'яті *EMB* (*Embedded Memory Block*). Лічильник реалізується у вигляді розподіленої схеми, що використовує тригери конфігураційних блоків *CLB* (*configurable logic block*). Елементи схеми КМПУ з'єднуються за допомогою програмованих з'єднань.

Для оптимізації характеристик *CLB* необхідно, щоб елемент *LUT* мав 5–6 входів. Це забезпечує достатню гнучкість проектування, а також уможливорює досягнення компромісу між займаною площею кристала, швидкодією та споживаною елементом *LUT* електричною енергією [9, 10]. Тому в сучасних мікросхемах *FPGA* елементи *LUT* мають  $S_L = 6$  входів. Це значення є занадто малим для реалізації схем пристрою управління, які можуть мати до 50 логічних умов (ЛУ) і до 15 адресних розрядів [7, 8]. Як наслідок, схеми пристрою управління мають багато рівнів елементів *LUT* і складну систему з'єднань.

Збільшення числа рівнів у схемі різко збільшує часову затримку та споживану потужність [7]. Тому доцільно зменшувати число рівнів і з'єднань у схемі пристрою управління. У роботі ми розглядаємо один із шляхів вирішення цієї проблеми для КМПУ з елементарними лінійними операторними ланцюгами (ЕЛОЛ) [4].

Пропонований метод засновано на ідеї дворівневого кодування станів [11] автомата Мілі. Стани є аналогами мікрокоманд. Тому в роботі розглядається метод подвійної адресації

мікрокоманд. Наші дослідження показали, що за певних умов пропонується метод дає змогу зменшити кількість рівнів у схемі адресації мікрокоманд до двох. Крім того, цей підхід дає змогу забезпечити регулярну систему з'єднань, що спрощує завдання розміщення і трасування [11]. Мова граф-схем алгоритму (ГСА) [3] використовується в нашій статті для специфікації вихідного алгоритму управління.

## Синтез КМПУ в базисі *FPGA*

Будь-яка ГСА  $\Gamma$  складається з множин вершин і дуг, що з'єднують ці вершини [3]. Розгляньмо ГСА  $\Gamma_1$  (рис.1).

У ГСА  $\Gamma_1$  можна виокремити початкову ( $b_0$ ) й кінцеву вершини ( $b_E$ ), операторні вершини  $b_q \in B = \{b_1, \dots, b_M\}$  й умовні вершини, що містять ЛУ  $x_i \in X = \{x_1, \dots, x_L\}$ . У операторній вершині  $b_{mq} \in B$  записуються набори мікрооперацій (НМО)  $Y_q \subseteq Y = \{y_1, \dots, y_N\}$ . Для розглянутого прикладу маємо  $M = 17$ ,  $L = 6$  і  $N = 9$ . Введімо низку визначень з роботи [4], необхідних для подальшого викладу матеріалу.

**Визначення 1.** Кінцева послідовність операторних вершин  $\alpha_g = \langle b_{g_1}, \dots, b_{g_{F_g}} \rangle$  називається ЛОЛ ГСА  $\Gamma$ , якщо для будь-якої пари сусідніх компонент  $\alpha_g$  існує дуга, де  $i$  — номер компоненти вектора  $\alpha_g$ .

**Визначення 2.** Входом ЛОЛ  $\alpha_g$  називається вершина, вхід якої пов'язаний з виходом початкової або умовної вершини, або з виходом операторної вершини, що не входить в ЛОЛ  $\alpha_g$ .

**Визначення 3.** Виходом ЛОЛ  $\alpha_g$  називається вершина, вихід якої пов'язаний з входом вершини, що не входить у ЛОЛ  $\alpha_g$ .

**Визначення 4.** ЛОЛ  $\alpha_g \in C$  є елементарною, якщо вона має тільки один вхід.

Операторна вершина  $b_m \in B$  відповідає мікрокоманді  $MI_m$ , що зберігається в ПУ [12]. Якщо ГСА  $\Gamma$  включає  $M$  операторних вершин, то в ПУ зберігається  $M_0$  мікрокоманд. Якщо вершина  $b_0$  з'єднана з умовною вершиною, то  $M_0 = M + 1$  (в цьому випадку МК  $MI_0$  відповідає вершині  $b_0$ ). В іншому випадку число МК збігається з числом операторних вершин ( $M_0 = M$ ).

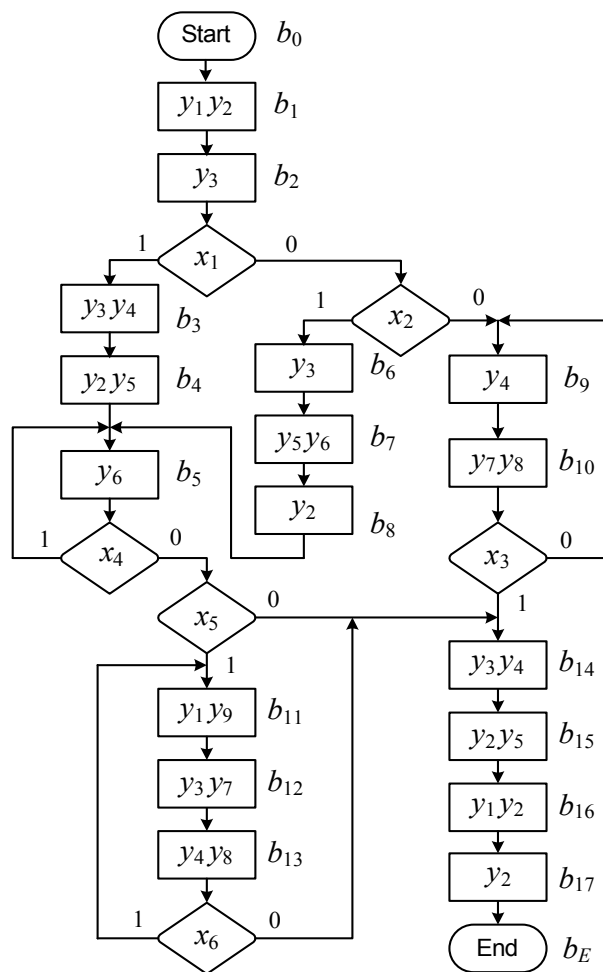


Рис. 1. Первинна ГСА  $\Gamma_1$

Мікрокоманді  $MI_m$  відповідає адреса  $A_m$ , що має  $R$  розрядів, де

$$R = \lceil \log_2 M_0 \rceil. \quad (1)$$

Для представлення адреси МК використовуються внутрішні змінні  $T_r \in T = \{T_1, \dots, T_R\}$ . Для зберігання адреси МК використовується  $R$ -розрядний лічильник ЛА з інформаційними входами типу  $D$  [4]. Для запису адреси в ЛА використовуються функції збудження пам'яті ( $\Phi$ ЗП), що утворюють множину  $\Phi = \{D_1, \dots, D_R\}$ .

Використовуючи метод [4], побудуємо множину ЕЛОЛ  $C = \{\alpha_1, \dots, \alpha_G\}$  для ГСА  $\Gamma$ . Ця множина має такі властивості:

- кожен елемент множини  $C \in$  ЕЛОЛ;
- все операторні вершини включені до деяких ЕЛОЛ;
- кожна вершина включена тільки в одну ЕЛОЛ;
- число елементів множини  $C \in$  мінімально можливим.

Виконаємо природну адресацію МК в межах кожної ЕЛОЛ. У цьому випадку умова

$$A_{gi+1} = A_g + 1 \quad (g = \overline{1, G}; i = \overline{1, F_g - 1})$$

виконується для кожної пари сусідніх компонент  $b_{gi}, b_{gi+1}$  кожної ЕЛОЛ. Тепер пристрій управління можна представити у вигляді КМПУ  $U_1$  (рис.2).

КМПУ  $U_1$  функціонує у такий спосіб. За сигналом *Start* в ЛА заноситься нульовий код, що відповідає початковій адресі прошивки. Чергова мікрокоманда вибирається з ПУ. Якщо ця МК відповідає будь-якій компоненті ЛОЛ  $\alpha_g \in C$  (крім виходу), то формується змінна  $y_0$ . У цьому випадку за сигналом синхронізації *Clock* вміст ЛА збільшується на одиницю. Якщо обрана МК відповідає виходу  $O_g$  ЛОЛ  $\alpha_g \in C$ , то  $y_0 = 0$ . У цьому випадку схема адресації МК формує функції

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

Функції (2) формують в ЛА адресу переходу. Якщо чергова МК відповідає виходу ЕЛОЛ, пов'язаному дугою з кінцевою вершиною ГСА, то формується змінна  $y_E$ . Якщо  $y_E = 1$ , то імпульси *Clock* не надходять в ЛА і функціонування КМПУ припиняється.

Для кожного моменту часу  $t$  ( $t = 0, 1, 2, \dots$ ) адреса переходу  $A^{t+1}$  визначається поточною адресою  $A^t$  і змінними  $x_i \in X, y_0$  та  $y_E$ . Ця залежність може бути представлена формулою

$$A^{t+1} = \begin{cases} A^t + 1, & \text{якщо } y_0 = 1 \text{ і } y_E = 1; \\ A^t, & \text{якщо } y_E = 1; \\ \Phi(T, X), & \text{якщо } y_0 = 0 \text{ і } y_E = 0. \end{cases} \quad (3)$$

Вміст ЛА використовується як адреса ПУ, і як вхідна інформація для схеми адресації МК. Тому КМПУ  $U_1$  називається КМПУ із загальною пам'яттю [4].

При втіленні схеми КМПУ  $U_1$  в базисі *FPGA* схема адресації і ЛА реалізуються на блоках *CLB*, що включають елементи *LUT* і тригери. Назвімо блок, що складається з елементів *LUT*, блоком *LUTer* [7]. Позначмо схему адресації МК символом *LUTerA*.

Сучасні *FPGA* містять вбудовані блоки пам'яті *EMB*. Ці блоки можна налаштовувати так, що за постійної ємності  $V_0$  змінюються кількість входів ( $S_A$ ) і виходів ( $t_F$ ) [13–15]. При цьому залежність між  $V_0$ ,  $S_A$ , і  $t_F$  представляється виразом

$$V_0 = 2^{S_A} \times t_F.$$

Отже, пара  $\langle S_A, t_F \rangle$  визначає конфігурацію *EMB*. Для сучасних *FPGA* характерні наступні пари [6]:  $\langle 15, 1 \rangle$ ,  $\langle 14, 2 \rangle$ ,  $\langle 13, 4 \rangle$ ,  $\langle 12, 8 \rangle$ ,  $\langle 11, 16 \rangle$ ,  $\langle 10, 32 \rangle$  та  $\langle 9, 64 \rangle$ . Очевидно, зменшення величини  $S_A$  на 1 подвоює число виходів *EMB*.

Блоки *EMB* доцільно використовувати для реалізації схеми ПУ. Це можливо, якщо існує пара  $\langle S_0, t_0 \rangle$ , для якої виконується така умова

$$S_0 = R.$$

Якщо кожна змінна  $y_n \in Y$  представлена одним розрядом слова ПУ, то кожна МК має  $N + 2$  розрядів. Число мікрооперацій збільшується на 2 для обліку змінних  $y_0$  і  $y_E$ . Кількість блоків *EMB* визначається виразом

$$n_{EMB} = \left\lceil \frac{N + 2}{t_0} \right\rceil.$$

Використовуємо символ *EMBer* для позначення блоку, що складається з *CLB* типу *EMB*. Для реалізації ПУ використовується блок *EMBer*, а схема адресації МК реалізується у вигляді блоку *LUTerA* (рис.3).

У КМПУ  $U_1$  (рис. 3) тригери лічильника розподілено між *CLB* блоку *LUTerA*. Цим пояснюється наявність сигналів *Clock*, *Start*,  $y_0$  і  $y_E$  на входах блоку *LUTerA*. За виконання певних умов схема блоку *LUTerA* має кілька рівнів логіки. Пропонований у цій статті метод орієнтовано на оптимізацію числа елементів *LUT* і їхніх рівнів у схемі блоку *LUTerA*.

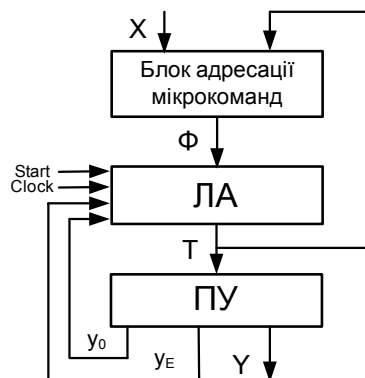


Рис.2. Структурна схема КМПУ  $U_1$

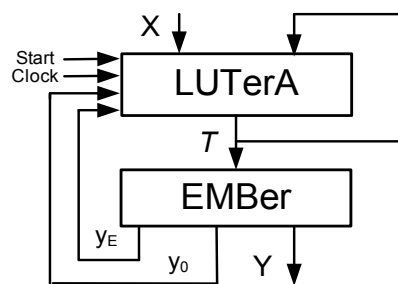


Рис.3. Реалізація КМПУ  $U_1$  в базисі *FPGA*

### Основна ідея пропонуваного методу

Як впливає з рис.3, блок *LUTerA* реалізує систему (2). Кожна функція  $D_r \in \Phi$  залежить від  $NA(D_r)$  змінних. Якщо виконується умова

$$NA(D_r) > SL ((r=1, R)),$$

то для реалізації функції  $D_r \in \Phi$  необхідна багаторівнева схема. У цьому випадку характеристики блоку *LUTerA* значно погіршуються [16]: збільшується споживана потужність і зменшується швидкодія. Отже, для поліпшення характеристик пристрою управління необхідно зменшити кількість елементів *LUT* і їхніх рівнів у схемі КМПУ  $U_1$ .

У цій роботі ми адаптуємо ідею подвійного кодування станів [11] для оптимізації схеми КМПУ  $U_1$ . За виконання певних умов пропо-

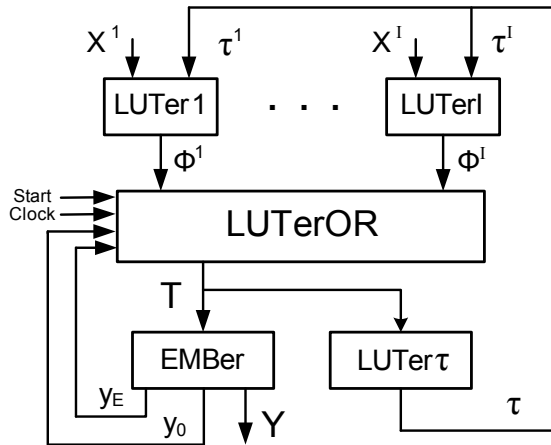


Рис.4. Структурна схема КМПУ  $U_2$

нований метод дає змогу зменшити кількість рівнів логіки в схемі блоку  $LUTerA$  до двох.

Нехай для ГСА  $\Gamma$  знайдено множину ЕЛОЛ, що складається з  $G$  елементів. Кожна ЕЛОЛ  $\alpha_g \in C$  має точно один вихід  $O_g$ . Ці виходи утворюють множину  $O(\Gamma) = \{O_1, \dots, O_G\}$ . Функції (2) залежать від адрес виходів ЕЛОЛ і змінних  $x_1 \in X$ . Отже, виходи  $O_g \in O(\Gamma)$  є аналогами станів автомата Мура, а їхні адреси — аналогами кодів станів.

Адреси  $A(O_g)$  виходів ЕЛОЛ  $\alpha_g \in C$  формуються в процесі природної адресації МК. У цьому сенсі вони є функціями адрес входів і кількості компонент ЕЛОЛ. Однак адреси входів ЕЛОЛ можна призначати довільно.

Знайдімо розбиття  $\Pi_0 = \{O^1, \dots, O^I\}$  множини  $O(\Gamma)$  таке, що: число класів  $I$  є мінімально можливим і для кожного класу  $O^i \in \Pi_0$  виконується умова

$$R_i + L_i \leq SL, (i \in \{1, \dots, I\}).$$

У виразі (4) символ  $R_i$  означає число додаткових змінних, необхідних для кодування виходів  $O_g \in O^i$  кодами  $C(O_g)$ . Символ  $L_i$  означає число ЛУ  $x_1 \in X$ , що визначають переходи з ЕЛОЛ для блоку  $i (i \in \{1, \dots, I\})$ .

При кодуванні виходів  $O_g \in O^i$  необхідно врахувати співвідношення  $O_g \notin O^i$ . Таким чином, число кодованих об'єктів на 1 перевищує

число  $M_i = |O^i|$  елементів в класі  $O^i \in \Pi_0$ . Число змінних  $R_i$  визначається виразом

$$R_i = \lceil \log_2(M_i + 1) \rceil, (i = \overline{1, I}).$$

Адреси  $A(O_g)$  виходів ЕЛОЛ представляються змінними  $T_r \in T$ . Ці адреси зберігаються в ЛА. Використовуємо для кодування виходів ЕЛОЛ змінні з множини  $\tau = \{\tau_1, \dots, \tau_{R_0}\}$ . Потужність множини  $\tau$  визначається виразом

$$R_0 = \sum_{i=1}^I R_i.$$

Клас  $O^i \in \Pi_0$  однозначно визначає три множини. Множина  $X^i \subseteq X$  включає ЛУ  $x_1 \in X$ , що визначають переходи з виходів  $O_g \in O^i$ . Множина  $\Phi^i \subseteq \Phi$  включає змінні  $D_r \in \Phi$ , рівні одиниці для формування адрес переходів з виходів  $O_g \in O^i$ . Множина  $\tau^i \subseteq \tau$  включає змінні  $\tau_r \in \tau$ , використовувани для формування кодів  $C(O_g)$  виходів  $O_g \in O^i$ .

Кожному класу  $O^i \in \Pi_0$  відповідає СБФ

$$\Phi^i = \Phi(\tau^i, X^i). \quad (5)$$

Для реалізації системи (5) використовуються елементи  $LUT$ , що утворюють блок  $LUTeri$ .

Функції збудження пам'яті, реалізовані блоком  $LUTeri$ , є частковими. Позначимо їх символом  $D_r^i$ . Для формування повних ФВП  $D_r \in \Phi$  необхідно знайти СБФ

$$D_r = \bigvee_{i=1}^I D_r^i, (r = \overline{1, R}). \quad (6)$$

Функції (6) реалізуються блоком  $LUTerT$ . У найкращому випадку блок  $LUTerT$  складається з  $R$  елементів  $LUT$ . Цей випадок визначається виразом

$$I \leq S_L.$$

Змінні  $T_r \in T$  зберігаються в ЛА, а для формування змінних  $\tau_r \in \tau$  необхідно реалізувати СБФ

$$\tau = \tau(T). \quad (7)$$

Система (7) складається з  $R_0$  рівнянь, для реалізації яких потрібен блок  $LUTert$ .

Грунтуючись на цій попередній інформації, ми пропонуємо структуру КМПУ  $U_2$ . Цей пристрій управління складається з блоків  $LUTer1, \dots, LUTerI, LUTerOR, LUTert$  і  $EMBer$  (рис.4).

Блок  $EMBer$  складається з  $n_{EMB}$  блоків  $EMB$ , що мають в сумі  $t_E = t_0 \times n_{EMB}$  виходів. У цій статті ми розглядаємо випадок, коли виконується умова

$$t_E = N + 2.$$

У цьому випадку блок  $EMBer$  використовується тільки для формування функцій  $y_n \in Y$  і змінних  $y_0$  і  $y_E$ .

КМПУ  $U_2$  функціонує так. За сигналом  $Start$  в ЛА записується нульова адреса, і відповідна МК вибирається з блоку  $EMBer$ . Якщо з  $EMBer$  обрано МК, яка не відповідає виходу  $O_g \in O(\Gamma)$ , то разом із МО  $y_n \in Y$  формується змінна  $y_0 = 1$ . Вміст ЛА збільшується на 1, що відповідає переходу між сусідніми компонентами ЕЛОЛ  $\alpha_g \in C$ . Якщо обрана МК відповідає виходу  $O_g \in O(\Gamma)$ , то  $y_0 = 0$  і адреса переходу формується блоком  $LUTer_i$  таким, що  $O_g \in O^i$ . Для цього на виході  $LUTer_t$  формується код  $C(O_g)$ . Якщо обрана МК відповідає вершині  $b_m \in B$  такої, що існує дуга  $\langle b_m, b_E \rangle$ , то формується  $y_E = 1$ . При цьому надходження імпульсів синхронізації  $Clock$  блокується і функціонування  $U_2$  припиняється.

У цій статті пропонується метод синтезу КМПУ  $U_2$ , який включає наступні етапи:

1. Формування множини ЕЛОЛ з мінімальною кількістю елементів по ГСА  $\Gamma$ .
2. Природна адресація МК, що оптимізує системи (5).
3. Формування множини виходів ЕЛОЛ  $O(\Gamma)$ .
4. Формування розбиття  $\Pi_0$ , що задовольняє (4).
5. Кодування виходів ЕЛОЛ кодами  $C(O_g)$ .
6. Формування таблиць блоків  $LUTer_1 - LUTer_l$ .
7. Формування систем функцій (5)–(6).
8. Формування таблиці блоку  $LUTer_t$  і системи (7).
9. Формування таблиці блоку  $EMBer$ .
10. Реалізація схеми КМПУ в заданому базисі.

Нехай символ  $U_i(\Gamma_j)$  означає, що КМПУ  $U_i$  реалізується за ГСА  $\Gamma_j$ . Розгляньмо приклад синтезу КМПУ  $U_2(\Gamma_1)$  за використання елементів  $LUT$  з числом входів  $S_L = 5$ .

		$T_1 T_2 T_3$							
$T_4 T_5$		000	001	010	011	100	101	110	111
00		1	3	5	6	9	11	14	*
01		2	4	*	7	10	12	15	*
10		*	*	*	8	*	14	16	*
11		*	*	*	*	*	*	17	*

Рис.5. Адреса мікрокоманд КМПУ  $U_2(\Gamma_1)$

### Приклад синтезу КМПК $U_2$

Використовуємо метод [4] для формування множини ЕЛОЛ  $C$ . Для нашого прикладу метод [4] дає змогу отримати множини  $C = \{\alpha_1, \dots, \alpha_7\}$ , де  $\alpha_1 = \langle b_1, b_2 \rangle$ ,  $\alpha_2 = \langle b_3, b_4 \rangle$ ,  $\alpha_3 = \langle b_5 \rangle$ ,  $\alpha_4 = \langle b_6, b_7, b_8 \rangle$ ,  $\alpha_5 = \langle b_9, b_{10} \rangle$ ,  $\alpha_6 = \langle b_{11}, b_{12}, b_{13} \rangle$ ,  $\alpha_7 = \langle b_{14}, \dots, b_{17} \rangle$ . Кожен ЛОЛ  $\alpha_g \in C$  має точно один вхід, тобто кожен елемент множини  $C$  є ЕЛОЛ.

Для оптимізації системи (5) ми пропонуємо призначати входам ЕЛОЛ адреси з максимальною кількістю нулів. В цьому випадку мікрокоманда  $MI_q$ , яка відповідає вершині  $b_q$ , такої, що існує дуга  $\langle b_0, b_q \rangle$ , повинна мати нульову адресу.

Для ГСА  $\Gamma_1$  маємо  $M_0 = 17$ . Використовуючи (1), отримуємо  $R = 5$ , що дає множини  $T = \{T_1, \dots, T_5\}$  і  $\Phi = \{D_1, \dots, D_5\}$ . Один із варіантів адресації МК для нашого прикладу наведено на рис.5. Тут показано номери мікрокоманд  $MI_m$ .

Як впливає з рис.5, для входів ЕЛОЛ виконується рівність  $T_4 = T_5 = 0$ . Із цього випливає, що  $LUTer_t = D_5 = 0$ . Отже, блоки  $LUTer_i$  не формують функції  $D_4^i$  і  $D_5^i$ , а блок  $LUTer_t$  — функції  $D_4$  і  $D_5$ . Крім того на входи блоку  $LUTer_t$  потрібно подавати тільки змінні  $T_1 - T_3$ .

Множина виходів ЕЛОЛ формується у тривіальний спосіб. Множина  $O(\Gamma)$  включає останні компоненти кожного ЕЛОЛ. У розглянутому прикладі маємо множину  $O(\Gamma_1) = \{b_2, b_4, b_5, b_8, b_{10}, b_{13}, b_{17}\}$ .

Для знаходження розбиття  $\Pi_0$  можна використовувати алгоритм з роботи [11]. Пошук розбиття виконується на множині  $O'(\Gamma) \subseteq O(\Gamma)$ . Множина  $O'(\Gamma)$  не включає ЕЛОЛ, виходи яких мають безпосередній зв'язок із входом кінцевої

вершини ГСА Г. У розглянутому прикладі є дуга  $\langle b_{17}, b_E \rangle$ , тому  $\alpha_7 \notin O'(\Gamma_1) = \{O_1, \dots, O_6\}$ .

Для нашого прикладу  $S_L = 5$ . Тому необхідно знайти розбиття множини  $O(\Gamma_1)$ , для кожного класу якого виконується умова  $R_i + L_i \leq 5$ . Цій умові відповідає розбиття  $\Pi_0 = \{O^1, O^2\}$ , де  $O^1 = \{O_1, O_2, O_5\}$  і  $O^2 = \{O_3, O_4, O_6\}$ .

Для класу  $O^1 \in \Pi_0$  маємо множину  $X^1 = \{x_1, I_2, x_3\}$  з  $L_1 = 3$ . Так як  $M_1 = 3$ , то  $R_1 = 2$ . Отже,  $L_1 + R_1 = 5$ . При цьому маємо множину  $\tau^1 = \{\tau_1, \tau_2\}$ . Для класу  $O^2 \in \Pi_0$  маємо  $X^2 = \{x_4, x_5, x_6\}$  з  $L_2 = 3$ . Оскільки  $M_2 = 3$ , то  $R_2 = 2$  і  $\tau^2 = \{\tau_3, \tau_4\}$ . Як і для  $O^1 \in \Pi_0$  умова (4) виконується для класу  $O^2 \in \Pi_0$ . Очевидно, що  $\tau^1 \cup \tau^2 = \tau = \{\tau_1, \dots, \tau_4\}$ .

Очевидно, коди  $C(O_g)$  не впливають на число елементів  $LUT$  першого рівня логіки. Тому закодуємо виходи у тривіальний спосіб:  $C(O_1) = C(O_3) = 01$ ,  $C(O_2) = C(O_4) = 10$  і  $C(O_5) = C(O_6) = 11$ . Код 00 залишимо для індикації відношення  $O_g \notin O'$ .

Формування таблиць блоків  $LUTer1-LUTer2$  починається з формування систем формул переходу для виходів  $O_g \in O'$ . Формули переходу задають зв'язок між виходами та входами ЕЛОЛ [4].

Блок  $LUTer1$  задається системою формул переходу (СФП), яка формується за ГСА  $\Gamma_1$  і має такий вигляд:

$$\begin{aligned} O_1 &\rightarrow x_1 b_3 \vee \overline{x_1 x_2} b_6 \vee \overline{x_1 x_2} b_9 \\ O_2 &\rightarrow b_5, O_5 \rightarrow x_3 b_{14} \vee \overline{x_3 x_2} b_6 \vee \overline{x_3 x_2} b_9 \end{aligned} \quad (8)$$

Таблиця 1. Таблиця блоку  $LUTer1$

$O_g$	$C(O_g)$	$b_q$	$A_q$	$X_h$	$\Phi_h$	$h$
$O_1$	01	$b_3$	00100	$x_1$	$D_3^1$	1
		$b_6$	01100	$\overline{x_1 x_2}$	$D_2^1 D_3^1$	2
		$b_9$	10000	$\overline{x_1 x_2}$	$D_1^1$	3
$O_3$	10	$b_5$	01000	1	$D_2^1$	4
$O_4$	11	$b_{14}$	11000	$x_3$	$D_1^1 D_2^1$	5
		$b_6$	01100	$x_2 \overline{x_3}$	$D_2^1 D_3^1$	6
		$b_9$	10000	$\overline{x_2 x_3}$	$D_1^1$	7

Блок  $LUTer2$  задається наступною СФП:

$$\begin{aligned} O_3 &\rightarrow x_4 b_5 \vee \overline{x_4 x_5} b_{11} \vee \overline{x_4 x_5} b_{14} \\ O_4 &\rightarrow b_5, O_6 \rightarrow x_6 b_{11} \vee \overline{x_6} b_{14} \end{aligned} \quad (9)$$

Таблиці блоків  $LUTeri$  мають такі стовпці:  $O_g$ ;  $C(O_g)$   $b_q$  — вершина, до якої є перехід з виходу  $O_g$ ;  $A_q$  — адреса МК, яка відповідає вершині  $b_q \in B$ ;  $X_h$  — кон'юнкція ЛУ, що визначає перехід з  $O_g$  в  $b_q$ ;  $\Phi_h$  — набір ФВП, рівних одиниці для формування в лічильнику ЛА адреси  $A_q$ ;  $h$  — номер переходу.

Із СФП (8) маємо таблицю блоку  $LUTer1$  (табл.1), із СФП (9) — таблицю блоку  $LUTer2$  (табл.2).

Верхній індекс у функціях  $D_r^i \in \Phi^i$  показує, що дані функції генеруються блоком  $LUTeri$ . Адреси МК беруться з рис.5.

З табл.1 і 2 формуються СБФ (5). Ці СБФ доцільно мінімізувати. Однак мінімізація має сенс тільки за умови, що зменшується число однойменних аргументів у всіх термах системи. З табл.1 маємо:

$$\begin{aligned} D_1^1 &= \overline{\tau_1} \tau_2 \overline{x_1 x_2} \vee \tau_1 \tau_2 x_3 \vee \tau_1 \tau_2 \overline{x_2} \\ D_2^1 &= \overline{\tau_1} \tau_2 \overline{x_1 x_2} \vee \tau_1 \overline{\tau_2} \vee \tau_1 \tau_2 x_3 \vee \tau_1 \tau_2 x_2 \\ D_3^1 &= \overline{\tau_1} \tau_2 x_1 \vee \overline{\tau_1} \tau_2 x_2 \vee \tau_1 \tau_2 \overline{x_1 x_2} \end{aligned}$$

З табл.2 отримуємо СБФ для блоку  $LUTer2$ :

$$\begin{aligned} D_1^2 &= \overline{\tau_3} \tau_4 \overline{x_4} \vee \tau_1 \tau_2 \\ D_2^2 &= \overline{\tau_3} \tau_4 x_4 \vee \overline{\tau_3} \tau_4 \overline{x_5} \vee \tau_3 \overline{\tau_4} \vee \tau_1 \tau_2 \overline{x_6} \\ D_3^2 &= \overline{\tau_3} \tau_4 x_4 x_5 \vee \tau_3 \overline{\tau_4} \vee \tau_3 \tau_4 x_6 \end{aligned}$$

Таблиця 2. Таблиця блоку  $LUTer2$

$O_g$	$C(O_g)$	$b_q$	$A_q$	$X_h$	$\Phi_h$	$h$
$O_3$	01	$b_5$	01000	$x_4$	$D_2^2$	1
		$b_{11}$	10100	$\overline{x_4 x_5}$	$D_1^2 D_3^2$	2
		$b_{14}$	11000	$\overline{x_4 x_5}$	$D_1^2 D_2^2$	3
$O_4$	10	$b_6$	01000	1	$D_2^2$	4
$O_6$	11	$b_{15}$	10100	$x_6$	$D_1^2 D_3^2$	5
		$b_{17}$	11000	$\overline{x_6}$	$D_1^2 D_2^2$	6

Система (6) формується у тривіальний спосіб: функції з однойменними нижніми індексами об'єднуються знаком диз'юнкції. Для нашого прикладу маємо:

$$D_1 = D_1^1 \vee D_1^2; D_2 = D_2^1 \vee D_2^2; D_3 = D_3^1 \vee D_3^2.$$

Як наслідок, кожен із блоків  $LUTer1$ ,  $LUTer2$  і  $LUTerT$  має по 3 елементи  $LUT$  з  $S_L=5$ . Отже, схема адресації МК має 9 елементів  $LUT$  і два рівня логіки.

Як випливає з рис.5, змінні  $T_1 = T_2 = T_3$  для всіх мікрокоманд, відповідних ЕЛОЛ  $\alpha_g \in C$ . Тому функції  $\tau = \tau(T)$  можна формувати на основі цих внутрішніх змінних. При цьому код  $C(O_g)$  формуватиметься для всіх мікрокоманд ЕЛОЛ  $\alpha_g \in C$ . Однак запис адрес переходу в лічильник здійснюватиметься тільки при  $y_0 = 0$ .

Таблиця блоку  $LUTert$  має стовпці:  $O_g$  — вихід ЕЛОЛ  $\alpha_g \in C$ ;  $A_g$  — кон'юнкція змінних, які мають постійне значення для всіх елементів ЕЛОЛ  $\alpha_g \in C$ ;  $C(O_g)$  — код виходу  $O_g \in O^i$ ;  $\alpha_g$  — змінні  $\tau_r \in \tau$ , рівні одиниці в коді  $C(O_g)$ ;  $g$  — номер ЕЛОЛ.

Для КМПУ  $U_2(\Gamma_1)$  блок  $LUTert$  представлено табл.3. Таблиця має 6 рядків.

Із цієї таблиці маємо СБФ

$$\tau_1 = \overline{T_1} \overline{T_2} T_3 \vee T_1 \overline{T_2} T_3$$

$$\tau_2 = \overline{T_1} T_2 \overline{T_3} \vee T_1 T_2 \overline{T_3}$$

$$\tau_3 = T_1 \overline{T_3}, \tau_4 = T_2 T_3$$

Очевидно, блок  $LUTert$  складається з  $R_0 = 4$  елементів  $LUT$  і має один рівень логіки. У

Таблиця 3. Таблиця блоку  $LUTert$

$O_g$	$A_g$	$C(O_g)$	$\tau_g$	$g$
$O_1$	$\overline{T_1} \overline{T_2} T_3$	0100	$\tau_2$	1
$O_2$	$\overline{T_1} T_2 \overline{T_3}$	1000	$\tau_1$	2
$O_3$	$\overline{T_1} T_2 T_3$	0001	$\tau_4$	3
$O_4$	$T_1 \overline{T_2} \overline{T_3}$	0010	$\tau_3$	4
$O_5$	$T_1 T_2 \overline{T_3}$	1100	$\tau_1 \tau_2$	5
$O_6$	$T_1 T_2 T_3$	0011	$\tau_3 \tau_4$	6

загальному випадку число літералів у системі  $\tau(T)$  визначається як  $R \times R_0$ , що дорівнює 20. Це число збігається з числом з'єднань між блоками  $LUTerT$  і  $LUTert$ .

Завдяки прийнятому методу адресації, СБФ (10) має 10 літералів. Отже, число з'єднань між блоками  $LUTerT$  і  $LUTert$  зменшується вдвічі у порівнянні з максимальним значенням.

Блок  $EMBer$  представлено таблицею з наступними стовпцями:  $MI_m, A_m, y_0, y_E, y_1, \dots, y_N, m$ . Ця таблиця заповнюється у такий спосіб:

1. У стовпцях  $y_1 - y_N$  рядки  $m$  записуються одиниці для МО  $y_N \in Y$  з операторної вершини  $b_m$  ( $m = 1, \dots, M$ ).

2. Якщо вершина  $b_m$  не є виходом ЕЛОЛ, то для МК  $MI_m$  маємо  $y_0 = 1$ .

3. Якщо вершина  $b_m$  пов'язана дугою  $\langle b_m, b_E \rangle$  з кінцевою вершиною ГСА  $\Gamma$ , то для рядка  $m$  необхідно записати  $y_E = 1$ .

Для нашого прикладу блок  $EMBer$  представлено табл.4. Ця таблиця має  $M = 17$  рядків. Адреси мікрокоманд взято з рис.5. Рядки табл.4 заповнюються на основі ГСА  $\Gamma_1$  і множини ЕЛОЛ  $C$ .

Останній етап запропонованого методу пов'язано з розв'язанням складних оптимізаційних задач. Для розв'язання цих завдань необхідно застосовувати стандартні САПР фірм-виробників  $FPGA$ . Наприклад, для  $FPGA$  фірми  $Xilinx$  використовується САПР  $Vivado$  [17]. У нашому прикладі цей крок не розглядається.

## Висновок

До складу КМПУ входить блок адресації мікрокоманд, що формує адреси переходів на основі поточної адреси та значень логічних умов. Для складних алгоритмів управління цей блок може мати багаторівневу структуру. Якщо цифрова система реалізується в базисі  $FPGA$ , то блок адресації мікрокоманд будується на елементах  $LUT$ . Наслідком малого числа входів цих елементів є різке збільшення кількості їх рівнів у схемі КМПУ.

У цій роботі пропонується метод оптимізації схеми блоку адресації МК, заснований на розбитті



Таблиця 4. Таблиця блоку *EMVer*

$MI_m$	$A_m$	$y_0$	$y_F$	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$	$y_7$	$y_8$	$y_9$	$m$
$MI_1$	00001	1	0	1	0	0	0	0	0	0	0	0	1
$MI_2$	00100	0	0	0	0	1	0	0	0	0	0	0	2
$MI_3$	00101	1	0	0	0	1	1	0	0	0	0	0	3
$MI_4$	00101	0	1	0	1	0	0	1	0	0	0	0	4
$MI_5$	01000	0	0	0	0	0	0	0	1	0	0	0	5
$MI_6$	01100	1	0	0	0	1	0	0	0	0	0	0	6
$MI_7$	01101	1	0	0	0	0	0	1	1	0	0	0	7
$MI_8$	01110	0	0	0	1	0	0	0	0	0	0	0	8
$MI_9$	10000	1	0	0	0	0	1	0	0	0	0	0	9
$MI_{10}$	10001	0	0	0	0	0	0	0	0	1	1		10
$MI_{11}$	10100	1	0	1	0	0	0	0	0	0	0	1	11
$MI_{12}$	10101	1	0	0	0	1	0	0	0	1	0	0	12
$MI_{13}$	10110	0	0	0	0	0	1	0	0	0	1		13
$MI_{14}$	11000	1	0	0	0	1	1	0	0	0	0	0	14
$MI_{15}$	11001	1	0	0	1	0	0	1	0	0	0	0	15
$MI_{16}$	11010	1	0	1	1	0	0	0	0	0	0	0	16
$MI_{17}$	11011	0	1	0	1	0	0	0	0	0	0	0	17

множини виходів ЕЛОЛ. Цей метод базується на ідеї подвійного кодування станів [11].

Дослідження ефективності запропонованого методу проводилося з використанням бібліотеки стандартних автоматів [18] і *FPGA Virtex-7* фірми *Xilinx* [19]. Аналіз бібліотеки [18] показав, що для 68 відсотків прикладів виконується умова (3) і блок адресації має кілька рівнів елементів *LUT*.

Дослідження [11] показали, що подвійне кодування станів дає змогу збільшити швидкодію, зменшити апаратні витрати (число *LUT* і їхніх з'єднань) і споживану потужність у схе-

мах автоматів Мілі. Грунтуючись на цих результатах, можна очікувати, що за виконання умови (3) запропонований підхід уможливіє поліпшення характеристики КМПУ в порівнянні з еквівалентним КМПУ  $U_1$ .

Наші подальші дослідження спрямовуються на застосування запропонованого підходу для оптимізації схем КМПУ [20, 21], що мають структури, відмінні від КМПУ  $U_1$  (рис.2). Крім того, ми плануємо дослідити можливість застосування методів оптимізації суміщених автоматів [22–23] для поліпшення характеристик КМПУ.

#### ЛІТЕРАТУРА

1. Соловьев В. В. Проектирование цифровых схем на основе программируемых логических интегральных схем. Москва : Горячая линия – ТЕЛЕКОМ, 2001. 636 с.
2. DeMicheli G. Synthesis and optimization of digital circuits. New York : McGraw-Hill, 1994. 576 p.
3. Baranov S. Logic synthesis for control automata. Dordrecht : Kluwer Academic Publishers, 1994. 312 p.
4. Баркалов А. А., Титаренко Л. А. Синтез композиционных микропрограммных устройств управления. Харьков : Коллегиум, 2007. 304 с.
5. Maxfield C. The design warrior's guide to FPGAs. Orlando : Academic Press, 2004. 542 p.

6. White paper FPGA architecture. URL: [www.altera.com](http://www.altera.com).
7. Sklyarov V., Sklyarova I., Barkalov A., Titarenko L. Synthesis and optimization of FPGA-based systems. Berlin : Springer, 2014. 432 p.
8. Grout I. Digital systems design with FPGAs and CPLDs. Amsterdam : Elsevier, 2008. 784 p.
9. Garcia-Vargas I., Senhadji-Navarro R., Jimenez-Moreno G., Civit-Balcells A., Guerra-Gutierrez P. ROM-based finite state machines implementation in low-cost FPGAs. ISIE'07 : IEEE Intern. Simp. on Industrial Electronics (Vigo, 2007). 2007. P. 2342–2347.
10. Sklyarova I., Sklyarov V., Sudnitson A. Design of FPGA-based circuits using hierarchical finite state machines. Tallinn : TUT Press, 2012. 240 p.
11. Barkalov A., Titarenko L., Mielcetek K. Hardware reduction for LUT-based Mealy FSMs. International Journal of Applied Mathematics and Computer Science. 2018, 28 (3). P. 595–607. DOI: <https://doi.org/10.2478/amcs-2018-0046>.
12. Barkalov A., Titarenko L. Logic synthesis for FSM-based control units. Berlin : Springer, 2009. 233 p.
13. Грушницький Р. И., Мурсаев А. Х., Угрюмов Е. П. Проектирование систем с использованием микросхем программируемой логики. СПб : БХВ-Петербург, 2002. 608 с.
14. Tiwari A., Tomko K. Saving power by mapping finite state machines into embedded memory blocks in FPGAs. DATE 2004 : Proc. Design, Automation and Test in Europe Conference and Exhibition (Paris, France, 6–20 Feb. 2004). 2004. 2. P. 916–921.
15. Garcia-Vargas L., Senhadji-Navarro R. Finite state machines with input multiplexing: A performance study. IEEE Transactions on CAD of Integrated Circuits and Systems. 2015. 34 (5). P. 867–871.
16. Rawski M., Selvaraj H., Luba T. An application of functional decomposition in ROM-based FSM implementation in FPGA devices. Journal of System Architecture. 2005. 51 (6–7). P. 424–434.
17. Vivado Design Suite. Xilinx. URL: <https://www.xilinx.com/products/design-tools/vivado.html>.
18. Yang S. Logic synthesis and optimization benchmarks user guide. Version 3.0. Techn. Rep. Microelectronics Center of North Carolina, 1991. 43 p.
19. Virtex-7 FPGAs. Xilinx. URL: <https://www.xilinx.com/products/silicon-devices/fpga/virtex-7.html>.
20. Баркалов А. А., Титаренко Л. А., Ефименко К. Н. Оптимизация схем композиционных микропрограммных устройств управления. Кибернетика и системный анализ. 2011. 1. С. 179–188. URL: <http://dspace.nbu.gov.ua/bitstream/handle/123456789/72212/15-Barkalov.pdf?sequence=1>.
21. Баркалов А. А., Титаренко Л. А. Преобразование кодов в композиционных микропрограммных устройствах управления. Кибернетика и системный анализ. 2011. 5. С. 107–118. URL: <http://dspace.nbu.gov.ua/bitstream/handle/123456789/84238/09-Barkalov.pdf?sequence=1>.
22. Баркалов А. А., Титаренко Л. А., Визор Я. Е., Матвиенко А. В. Оптимальное кодирование состояний в совмещенном автомате. Управляющие системы и машины. 2016. 6. С. 34–39. DOI: <https://doi.org/10.15407/usim.2016.06.034>.
23. Баркалов А. А., Титаренко Л. А., Визор Я. Е., Матвиенко А. В. Уменьшение числа LUT элементов в схеме совмещенного автомата. Управляющие системы и машины. 2016. 3. С. 16–22. DOI: <https://doi.org/10.15407/usim.2016.03.016>.
24. Баркалов А. А., Титаренко Л. А., Визор Я. Е., Матвиенко А. В. Уменьшение аппаратурных затрат в совмещенных автоматах. Управляющие системы и машины. 2017. 4. С. 43–50. DOI: <https://doi.org/10.15407/usim.2017.04.043>

Надійшла 15.02.2021

## REFERENCES

1. Soloviev V. V., 2001. *Proyektirovaniye tsifrovyykh skhem na osnove programmiruyemykh logicheskikh integralnykh skhem* [Design of digital circuits based on programmable logic integrated circuits], Hotline – TELECOM, Moscow, 636 p. (In Russian).
2. DeMicheli G., 1994. *Synthesis and optimization of digital circuits*, McGraw-Hill, New York, 576 p.
3. Baranov S., 1994. *Logic synthesis for control automata*, Kluwer Academic Publishers, Dordrecht, 312 p.
4. Barkalov A. A., Titarenko L. A., 2007. *Sintez kompozitsionnykh mikroprogrammnykh ustroystv upravleniya* [Synthesis of composite microprogram control devices], Collegium, Kharkov, 304 p. (In Russian).
5. Maxfield C., 2004. *The design warrior's guide to FPGAs*. Academic Press, Orlando, 542 p.
6. "White paper FPGA architecture". [online] Available at: [www.altera.com](http://www.altera.com).

7. Sklyarov V., Skliarova I., Barkalov A., Titarenko L., 2014. Synthesis and optimization of FPGA-based systems. Springer, Berlin, 432 p.
8. Grout I., 2008. Digital systems design with FPGAs and CPLDs, Elsevier, Amsterdam, 784 p.
9. Garcia-Vargas I., Senhadji-Navarro R., Jimenez-Moreno G., Civit-Balcells A., Guerra-Gutierrez P., 2007. “ROM-based finite state machines implementation in low cost FPGAs”, IEEE Intern. Simp. on Industrial Electronics (ISIE'07), Vigo, 2007, pp. 2342–2347.
10. Skliarova I., Sklyarov V., Sudnitson A., 2012. Design of FPGA-based circuits using hierarchical finite state machines, TUT Press, Tallinn, 240 p.
11. Barkalov A., Titarenko L., Mielcetek K., 2018. “Hardware reduction for LUT-based Mealy FSMs”, International Journal of Applied Mathematics and Computer Science, 28 (3), pp. 595–607.
12. Barkalov A., Titarenko L., 2009. Logic synthesis for FSM-based control units. Springer, Berlin, 233 p.
13. Grushnitskiy R. I., Mursaev A. Kh., Ugryumov E. P., 2002. *Proyektirovaniye sistem s ispolzovaniyem mikroskhem programmiruyemoy logiki* [Designing systems using programmable logic chips], BHV-Petersburg, St. Petersburg, 608 p. (In Russian).
14. Tiwari A., Tomko K., 2004. “Saving power by mapping finite state machines into embedded memory blocks in FPGAs”, Proc. Design, Automation and Test in Europe Conference and Exhibition, Paris, France, 6–20 Feb. 2004, 2, pp. 916–921.
15. Garcia-Vargas L., Senhaji-Navarro R., 2015. “Finite state machines with input multiplexing: A performance study”, IEEE Transactions on CAD of Integrated Circuits and Systems, 34 (5), pp. 867–871.
16. Rawski M., Selvaraj H., Luba T., 2005. “An application of functional decomposition in ROM-based FSM implementation in FPGA devices”, Journal of System Architecture, 51 (6–7), pp. 424–434.
17. “Vivado Design Suite”, Xilinx. [online] Available at: <<https://www.xilinx.com/products/design-tools/vivado.html>>.
18. Yang S., 1991. Logic synthesis and optimization benchmarks user guide. Version 3.0. Techn. Rep. Microelectronics Center of North Carolina, 43 p.
19. “Virtex-7 FPGAs”, Xilinx. [online] Available at: <<https://www.xilinx.com/products/silicon-devices/fpga/virtex-7.html>>.
20. Barkalov A. A., Titarenko L. A., Efimenko K. N., 2011. “Optimizatsiya skhem kompozitsionnykh mikroprogrammnykh ustroystv upravleniya” [“Optimization of the circuits of composite microprogram control devices”], Cybernetics and Systems Analysis, 1, pp. 179–188. [online] Available at: <<http://dSPACE.nbuv.gov.ua/bitstream/handle/123456789/72212/15-Barkalov.pdf?sequence=1>> (In Russian).
21. Barkalov A. A., Titarenko L. A., 2011. “Preobrazovaniye kodov v kompozitsionnykh mikroprogrammnykh ustroystvakh upravleniya” [“Converting codes in composite microprogram control devices”], Cybernetics and Systems Analysis, 5, pp. 107–118. [online] Available at: <<http://dSPACE.nbuv.gov.ua/bitstream/handle/123456789/84238/09-Barkalov.pdf?sequence=1>> (In Russian).
22. Barkalov A. A., Titarenko L. A., Vizor Ya. E., Matvienko A. V., 2016. “Optimalnoye kodirovaniye sostoyaniy v sovmeshchennom avtomate” [“Optimal coding of states in a combined automaton”], Control systems and machines, 6, pp. 34–39. DOI: 10.15407/usim.2016.06.034 (In Russian).
23. Barkalov A. A., Titarenko L. A., Vizor Ya. E., Matvienko A. V., 2016. “Umensheniye chisla LUT elementov v skheme sovmeshchennogo avtomata” [“Reducing the number of LUT elements in the combined machine circuit”], Control systems and machines, 3, pp. 16–22. DOI: 10.15407/usim.2016.03.016 (In Russian).
24. Barkalov A. A., Titarenko L. A., Vizor Ya. E., Matvienko A. V., 2017. “Umensheniye apparaturnykh zatrat v sovmeshchennykh avtomatakh” [“Reduction of hardware costs in combined machines”], Upravlyayushchiye sistemy i mashiny, 4, pp. 43–50. DOI: 15407/usim.2017.04.043 (In Russian)

Received 15.02.2021

*O.O. Barkalov*, Doctor of Technical Sciences, Professor,  
Computer Science and Telecommunications University of Zielona Gora (Poland),  
Podgorna str., 50, Zielona Gora, 65246, Poland,  
A.Barkalov@iie.uz.zgora.pl

*L.O. Titarenko*, Doctor of Technical Sciences, Professor,  
Computer Science and Telecommunications University of Zielona Gora (Poland),  
Podgorna str., 50, Zielona Gora, 65246, Poland,  
L.Titarenko@iie.uz.zgora.pl

*O.M. Golovin*, Ph.D. Eng. Sciences, Senior Research Associate,  
V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine,  
03187, Kiev, Glushkov Avenue, 40, Ukraine,  
yaviz@ukr.net

*O.V. Matvienko*, Researcher Associate,  
V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine,  
03187, Kiev, Glushkov Avenue, 40, Ukraine,  
matv@online.ua

#### OPTIMIZATION OF A COMPOSITION MICROPROGRAM CONTROL UNIT WITH ELEMENTARY CIRCUITS

**Introduction.** The control unit coordinating an interaction of all other blocks of a digital system is one of the central blocks and is a sequential circuit. As a rule, when synthesizing control unit circuits, the problem arises of reducing hardware costs. The methods for solving this problem depend on the features of both the architecture of the control unit and the elemental basis.

**Purpose.** The main goal of this work is to reduce hardware costs and power consumption of control units of digital systems taking into account the features of the element base of the control unit and rational organization of addressing microinstructions. FPGA (field-programmable logic array) microcircuits, widely used for the implementation of modern digital systems, were chosen as an elementary basis.

**Methods.** Methods of set theory, synthesis of automata and software modeling as well as the library of standard automata and FPGA Virtex-7 from Xilinx were used for assessment the effectiveness of solving the problem.

**Results.** The paper proposes a method for optimizing the circuit of the microinstruction addressing unit based on splitting the set of outputs of elementary linear operator circuits, which is based on the idea of double coding of states. The proposed method, under certain conditions, makes it possible to reduce the number of levels in microinstruction addressing circuit to two.

**Conclusion.** Studies have shown that double coding of states can increase performance, reduce hardware costs (the number of LUTs and their interconnections) and power consumption in Mealy's circuitry. Based on these results, it can be expected that, with the number of conditions exceeding the number of LUT inputs, the proposed approach will improve characteristics of the composition microprogram control unit in comparison with the equivalent control unit  $U_1$ .

**Keywords:** *composition microprogram control device, microinstruction, LUT, EMB, synthesis.*